**TEAMDUTY Web-application**

Test Plan

**By**

Mr. Manuchet Manoi 542115049

Mr. Saknarong Noipaksa 542115061

Department of Software Engineering

College of Arts, Media and Technology

Chiang Mai University

**Project Advisor**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Aj. Jayakrit Hirisajja**

**Document History**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Document Name** | **Version** | **Status** | **Date** | **Viewable** | **Reviewer** | | **Responsible** |
| **Documents** | | | | | | | |
| TEAMDUTY - Web application - SRS v 0.1.docx | **Add Chapter 1**  - Objective  - Intended Audience and Reading Suggestions  - Project Scope | Draft | 01 May 2014 | MM,  SN,JH | | MM,  SN | MM,  SN |
| TEAMDUTY - Web application - SRS v 0.1.docx | **Add Chapter 2**  **-** Product Perspective  - Product Features  - User Classes and Characteristics  - Operation Environment  - Design and Implementation Constraints  **Add Chapter 3**  **-**User requirement  **Add Chapter 5**  -Traceability Record | Draft | 01 May 2014 | MM,  SN,JH | | MM,  SN | MM,  SN |
| TEAMDUTY - Web application - SRS v 0.2.docx | **Adjust Chapter 3**  -Approve URS & SRS | Draft | 06 May 2014 | MM,  SN,JH | | MM,  SN | MM,  SN |
| TEAMDUTY - Web application - SRS v 0.3.docx | **Add Chapter 4**  -Software Requirement Specification  -Use case Description & Activity Diagram  -Quality Requirement |  |  |  | |  |  |

**\*MM = Manuchet Manoi**

**\*SN = Saknarong Noipaksa**

**\*JH = Jayakrit Hirisajja**

**Table of Contents**

# Chapter One | Introduction

## Purpose

The objectives of the test plan of TEAMDUTY – Webapplication system is to establish test plan of the unit testing and system testing for make sure that the bugs or the defects are fixed. Unit testing covers all of implemented methods in the TEAMDUTY – Webapplication system. The system testing cover the User requirements.

## 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.

## Objective

The objective of testing TEAMDUTY – Webapplication system 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.

## Dependency Document

* Project Proposal
* Project Plan
* Software Requirement Specification
* Software Design Document
* Testing Document
* Test Plan  
  - Unit Test Document  
  - System Test Document
* Test Report  
  - Unit Test Document  
  - System Test Document
* Traceability Record
* Progress Status Report
* Change Report

## Test Duration

|  |  |
| --- | --- |
| **Progress** | **Date and Duration** |
| 1 | **Perform date:** 16th May 2014 - 18th May 2014(Unit test, System test) **Duration:** 3 days |

# Chapter Two | Unit Test Case

## Controller Package

### HomeController

**Controller Unit Test Case 1**# 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.

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ Home/Index and display category of job list to web application | ".../Home/Index" | Pass |
| 2 | This method can't redirect to Home/Index by using a wrong URL: .../ Home/Index1 | ".../Home/Index1 " | Fail |

### AccountController

**Controller Unit Test Case 2**# Create()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ /Account/Create and display category of job list to web application | ".../Account/Create " | Pass |
| 2 | This method can't redirect to /Account/Create by using a wrong URL: .../Account/Creates | ".../Account/Creates" | Fail |

**Controller Unit Test Case 3**# ConfirmRegister()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ /Account/ConfirmRegister and display category of job list to web application | ".../Account/ConfirmRegister " | Pass |
| 2 | This method can't redirect to .../Account/ConfirmRegister by using a wrong URL: .../Account/ConfirmRegisters | ".../Account/ConfirmRegisters" | Fail |

**Controller Unit Test Case 4**# Login()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ /Account/Login and display category of job list to web application | ".../Account/Login " | Pass |
| 2 | This method can't redirect to .../Account/Login by using a wrong URL: .../Account/Logins | ".../Account/Logins " | Fail |

**Controller Unit Test Case 5**# ShowFreelancer()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ /Account/ShowFreelancer and display category of job list to web application | ".../Account/ShowFreelancer" | Pass |
| 2 | This method can't redirect to .../Account/Login by using a wrong URL: .../Account/ ShowFreelancers | ".../Account/ShowFreelancers " | Fail |

**Controller Unit Test Case 6**# ShowProfile(int pid)

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 33, URL: .../ Profile/ShowProfiles?pid=33 | "?pid=33" | 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: .../ Profile/ShowProfiles?pid=0 | "?pid=0" | Fail |

**Controller Unit Test Case 7**# LogOut()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can logout and redirect to login page. | ".../Home/Index" | Pass |

### ProfileController

**Controller Unit Test Case 8**# Profile()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ Profile/Profiles and display category of job list to web application | ".../Profile/Profiles" | Pass |
| 2 | This method can't redirect to .../ Profile/Profiles by using a wrong URL: .../Profile/Profile | ".../Profile/Profile" | Fail |

**Controller Unit Test Case 9**# UploadImageToDB()

**Controller Unit Test Case 10**# GetImageBytesByImageId(int ImageId)

**Controller Unit Test Case 11**# GetImage(int ProId)

**Controller Unit Test Case 12**# ShowProfiles(int pid)

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 33, URL: .../ Profile/ShowProfiles?pid=33 | "?pid=33" | 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: .../ Profile/ShowProfiles?pid=0 | "?pid=0" | Fail |

### BidController

**Controller Unit Test Case 13**# Bidding()

*Description*

Test the Bidding() method to see that it can redirect to Patail view name = ShowDetailCategoryJob in page of recipe detail and it can't redirect if using wrong url.

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ DisplayJob/ ShowDetailCategoryJob and display category of job list to web application | ".../DisplayJob/ ShowDetailCategoryJob" | Pass |
| 2 | This method can't redirect to .../ DisplayJob/ ShowDetailCategoryJob by using a wrong URL: .../ DisplayJob/ ShowDetailCategoryJobs | ".../DisplayJob/ ShowDetailCategoryJobs" | Fail |

**Controller Unit Test Case 14**# showBiddingList()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 1, | "?bid=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 pid=0 | "?bid=0" | Fail |

**Controller Unit Test Case 15**# ShowProfileFromBid(int id)

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 33, URL: .../ Profile/ShowProfiles?pid=33 | "?pid=33" | 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: .../ Profile/ShowProfiles?pid=0 | "?pid=0" | Fail |

### JobController

**Controller Unit Test Case 16**# Postjob()

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 33, URL: .../ Job/ShowProfiles?pid=33 | "?uid=33" | 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: .../Job /ShowProfiles?pid=0 | "?uid=0" | Fail |

**Controller Unit Test Case 17**# ShowJob()

*Description*

Test the ShowJob() method to see that it can redirect to show view ShowDetailCategoryJob in page of recipe detail and it can't redirect if using wrong url.

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | This method can redirect to .../ Job/ShowJob and display category of job list to web application | ".../Job/ShowJob" | Pass |
| 2 | This method can't redirect to .../ Job/ShowJob by using a wrong URL: .../ Job/ShowJobs | ".../Job/ShowJobs" | Fail |

### DisplayJobController

**Controller Unit Test Case 18**# ShowDetailCategoryJob(int jid)

*Description*

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

*TestCase*

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | ExpextedResult |
| 1 | The method can redirect to view page of recipe detail show the information of recipe detail by using recipe id = 1, URL: .../ DisplayJob/ ShowDetailCategoryJob?pid=1 | "?jid=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: .../DisplayJob/ ShowDetailCategoryJob?pid=0 | "?jid=0" | Fail |

## Repository Package

Repository is package that we use to connect with database. We would check about in method can be work correctly or not. We will test each method in repository by build mock object in test class (it will show in test record document)

### AccountRepository

In this AccountRepository we will test method that use for Get user data in database, Insert User data and Save data after adjust database.

**Repository Unit Test Case 1**#

public IEnumerable<Member> GetUsers()

*Description*

This Unittest Case is evaluating about method in AccountRepository that use for GetMember data in database to display in system.

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | Email | Password | ConfirmPassword |
| 1 | Member1 | xxx@xxx.xxx | qwerty | qwerty |
| 2 | Member2 | oppo@go.com | 123456 | 123456 |

*ExpectResult*

Both of 2 object in Unittest = Isnotnull and pass the test

**Repository Unit Test Case 2**#

public void InsertUser(Member user)

public Member GetUserByID(int userId)

*Description*

This Unittest Case is evaluating about method in AccountRepository that use for Insert Member data to database by use id

*Input Value*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | |
| ID | userID | Email | Password | ConfirmPassword |
| 1 | Member1 | 1 | xxx@xxx.xxx | qwerty | qwerty |

*ExpectResult*

Both of 2 object in Unittest = Isnotnull , Member1&Member2 Should be Insert & pass test.

**Repository Unit Test Case 3**#

public void InsertUser(Member user)

public Member GetUserByEmail(string email)

*Description*

This Unittest Case is evaluating about method in AccountRepository that use for Insert Member data to database by use email

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | Email | Password | ConfirmPassword |
| 1 | Member1 | xxx@xxx.xxx | qwerty | qwerty |
| 2 | Member2 | oppo@go.com | 123456 | 123456 |

*ExpectResult*

Both of 2 object in Unittest = Isnotnull , Member1&Member2 Should be Insert & pass test.

**Repository Unit Test Case 4**#

public void Save()

*Description*

This Unittest Case is evaluating about method in AccountRepository that use for save after adjust data member in database

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | Email | Password | ConfirmPassword |
| 1 | Member1 | xxx@xxx.xxx | qwerty | qwerty |

*ExpectResult*

object in Unittest = Isnotnull , Member1 Should be save & pass test.

### ProfileRepository

In this ProfileRepository we will test method that use for Get profile data in database, Insert profile data, Update profile data and Save data after adjust database

**Repository Unit Test Case 5**#

public IEnumerable<Profile> GetProfile()

*Description*

This Unittest Case is evaluating about method in ProfileRepository that use for GetProfile data in database to display in system

*Input Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |
| ID | ProId | Name | CompanyName | About\_Yourself | Telephone |
| 1 | Profile1 | 6 | BigM | Daren.software | cpx.aa chaing mai thailand | 097655434 |

*ExpectResult*

object in Unittest = Isnotnull , & pass test.

**Repository Unit Test Case 6**#

public void InsertProfile(Profile profile)

public Profile GetProfileByID(int profileId)

*Description*

This Unittest Case is evaluating about method in ProfileRepository that use for Insert Profile data to database by use id

*Input Value*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |  |
| ID | ProId | Name | CompanyName | About\_Yourself | Telephone | Picture |
| 1 | Profile1 | 6 | BigM | Daren.software | cpx.aa chaing mai thailand | 097655434 | ddd.jpg |

*ExpectResult*

object in Unittest = Isnotnull and Insert & pass test.

**Repository Unit Test Case 7**#

public void UpdateProfile(Profile profile)

public Profile GetProfileByID(int profileId)

*Description*

This Unittest Case is evaluating about method in ProfileRepository that use for Update Profile data to database by use id

*Input Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |
| ID | ProId | Name | CompanyName | About\_Yourself | Telephone |
| 1 | Profile1 | 6 | BigM | Daren.software | cpx.aa chaing mai thailand | 097655434 |

*ExpectResult*

object in Unittest = Isnotnull ,Update data object & pass test.

**Repository Unit Test Case 8**#

public void DeleteProfile(Profile profile);

*Description*

This Unittest Case is evaluating about method in ProfileRepository that use for Delete Profile data to database by use id

*Input Value*

-

*ExpectResult*

object will Delete & pass test.

**Repository Unit Test Case 9**#

public void Save()

*Description*

This Unittest Case is evaluating about method in ProfileRepository that use for save after adjust data Profile in database

*Input Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |
| ID | ProId | Name | CompanyName | About\_Yourself | Telephone |
| 1 | Profile1 | 6 | BigM | Daren.software | cpx.aa chaing mai thailand | 097655434 |

*ExpectResult*

object in Unittest = Isnotnull , Profile1 Should be save & pass test.

### JobRepository

In this JobRepository we will test method that use for Get job data in database, Insert job data and save job data after adjust database

**Repository Unit Test Case 1**0#

public IEnumerable<Job> GetJobs()

*Description*

This Unittest Case is evaluating about method in JobRepository that use for get job data to display in system

*Input Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |
| ID | JobId | Title | Description | Budget | Skill |
| 1 | jobpost1 | 5 | Need people codeing | Descliption here | 20.20 | Html = True |

*ExpectResult*

object in Unittest = Isnotnull , Profile1 Should be save & pass test.

**Repository Unit Test Case 11#**

public void InsertJob(Job job)

public Job GetJobByID(int jobId)

*Description*

This Unittest Case is evaluating about method in JobRepository that use for Insert Job data to database by use id

*Input Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |
| ID | JobId | Title | Description | Budget | Skill |
| 1 | jobpost1 | 5 | Need people codeing | Descliption here | 20.20 | Html = True |

*ExpectResult*

object in Unittest = Isnotnull and Insert & pass test.

**Repository Unit Test Case 12#**

public void Save()

*Description*

This Unittest Case is evaluating about method in JobRepository that use for save after adjust data Job in database

*Input Value*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | | | |
| ID | JobId | Title | Description | Budget | Skill |
| 1 | jobpost1 | 5 | Need people codeing | Descliption here | 20.20 | Html = True |

*ExpectResult*

Object is delete data object & pass test.

### BidRepository

In this BidRepository we will test method that use for Get Bid data in database, Insert Bid data and save job data after adjust database

**Repository Unit Test Case 13#**

public IEnumerable<Bid> GetBids()

*Description*

This Unittest Case is evaluating about method in BidRepository that use for get job data to display in system

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | BidId | Detail | Budget |
| 1 | Bid1 | 2 | can finished in 2 day | 20.20 |
| 2 | Bid2 | 3 | can finished in 3 day | 20.00 |

*ExpectResult*

Both of 2 object in Unittest = Isnotnull , Bid1 and Bid2 Should be save & pass test.

**Repository Unit Test Case 14#**

public void InsertBid(Bid bid)

public Bid GetBidById(int bidId)

*Description*

This Unittest Case is evaluating about method in BidRepository that use for Insert Job data to database by use id

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | BidId | Detail | Budget |
| 1 | Bid1 | 2 | can finished in 2 day | 20.20 |
| 2 | Bid2 | 3 | can finished in 3 day | 20.00 |

*ExpectResult*

Both of 2 object in Unittest = Isnotnull and Insert & pass test.

**Repository Unit Test Case 15#**

public void Save()

*Description*

This Unittest Case is evaluating about method in BidRepository that use for save after adjust data Job in database

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | BidId | Detail | Budget |
| 1 | Bid1 | 2 | can finished in 2 day | 20.20 |
| 2 | Bid2 | 3 | can finished in 3 day | 20.00 |

*ExpectResult*

Both of 2 Object is save data object & pass test.

## Service Package

Service is package that we use to connect with database. We would check about in method can be work correctly or not. We will test each method in repository by build mock object in test class (it will show in test record document)

**Service Unit Test Case 1#**

public bool IsValid(string username, string password)

*Description*

This Unittest Case is evaluating about method in IsValid in Login Method that use for validate data in database that Username and Password

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | BidId | Detail | Budget |
| 1 | Bid1 | 2 | can finished in 2 day | 20.20 |
| 2 | Bid2 | 3 | can finished in 3 day | 20.00 |

*ExpectResult*

Both of 2 Object is save data object & pass test.

**Service Unit Test Case 2#**

public bool IsValid(string username, string password)

*Description*

This Unittest Case is evaluating about method in IsValid in Login Method that use for validate data in database that Username and Password

*Input Value*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | InputObjectName | Input Value | | |
| ID | BidId | Detail | Budget |
| 1 | Bid1 | 2 | can finished in 2 day | 20.20 |
| 2 | Bid2 | 3 | can finished in 3 day | 20.00 |

*ExpectResult*

Both of 2 Object is save data object & pass test.

**Service Unit Test Case 3#**

Public bool SendEmail(MailMessage message)

*Description*

This Unittest Case is evaluating about method that can send email to user or not

*ExpectResult*

## Can send email to user correctly

# Chapter Three | System Test Case

### Employer System Test case 01 (ESTC-01):

*service request customer* can be Register and select type of user as *service request customer*

**Description:**

This system test case testing for E-URS-01. *service request customer* can be Register and select type of user as *service request customer*

**Test Script:**

1**.** Click “Register” button in first page of web-application

2. System Provide UI system provide UI to input Email, select type of user, password, confirm password

3. Select type of user = Employer, input Email Address input password and Comfirm password and click Create

4. The System save data type of user, Email Address, input password and Confirm password to database

5. The System bring *service request customer* to index mainpage

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Select Type of user = Employer,  Input Email ,password and Confirm password password & Confirm is match | Select Type of user = Employer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System bring *service request customer* to web-application mainpage |
| 2 | Select Type of user = Employer  Input wrong type Email ,  Password & Confirm is match | Select Type of user = Employer  Input Email = “pureman.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *service request customer* need to reinput again |
| 3 | Select Type of user = Employer  Input Email is already exist in system, Password & Confirm is match | Select Type of user = Employer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *service request customer* need to reinput again |
| 4 | Select Type of user = Employer  Input Email is already exist in system, Password & Confirm is match | Select Type of user = Employer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *service request customer* need to reinput again |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 5 | Select Type of user = Employer  Input Email in system, Password & Confirm is not match | Select Type of user = Employer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure08354” | System display error message near Confirm password input box and *service request customer* need to reinput again |
| 6 | Select Type of user = Employer  Not Input Email in system, Password & Confirm Password is not match | Select Type of user = Employer  Input Email = “null”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *service request customer* need to reinput again |
| 7 | Select Type of user = Employer Input Email in system, Password & Confirm Password is more than less than 6 | Select Type of user = Employer  Input Email = “pureman9@gmail.com”  Input password = “pure”  Input Confirm password = “pure” | System display error message near Email input box and *service request customer* need to reinput again |

### Employer System Test case 02 (ESTC-02):

*service request customer* can be ConfirmRegister via email address after Register

**Description:**

This system test case testing for E-URS – 02:*service request customer* can be ConfirmRegister in their Email that they input on previously

**Prerequisite:**

*service request customer* need to Register first

**Test Script:**

1**.** Login to Email address that register in previously (in this case use pureman9@gmail.com)

2. System will send link of ConfirmRegister to Email Address that registered on previously

3. Click link in Email address

4. The System bring *service request customer* to input Username & ConfirmRegisterCode

5. The System bring *service request customer* to index mainpage

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input Username &ConfirmRegisterCode right type | Input Username = “pureqwerty”  Input ConfirmRegisterCode = “54117316” | System bring *service request customer* to web-application mainpage |
| 2 | Not Input Username but InputConfirmRegisterCode right type | Input Username = “null”  Input ConfirmRegisterCode = “54117316” | System display error message near Username input box and *service request customer* need to reinput again |
| 3 | Input Username is already exist in database but input ConfirmRegisterCode right type | Input Username = “pureman9@gmail.com”  Input ConfirmRegisterCode = “54117316” | System display error message near Username input box and *service request customer* need to reinput again |
| 4 | Input Username but wrong ConfirmRegisterCode | Input Username = “pureman9@gmail.com”  Input ConfirmRegisterCode = “11111111” | System display error message near ConfirmRegisterCode input box and *service request customer* need to reinput again |

### Employer System Test case 03 (ESTC-03):

*service request customer* can be Login to use Web-Application

**Description:**

This system test case testing for E-URS – 03:*service request customer* can be Login to use Web Application after get user account.

**Prerequisite:**

*service request customer* need to ConfirmRegister first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *service request customer* to *service request customer* mainpage

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input correct username and password. | username = pureqwerty  password = pure0835 | System bring *service request customer* to *service request customer* mainpage |
| 2 | Input some fields. | username = null  password = pure0835 | System display error message and *service request customer* need to reinput again |
| 3 | Not enter both username and password | username = null  password = null | System display error message and *service request customer* need to reinput again |
| 4 | Input invalid username and password. | username = abcd  password = 123456 | System display error message and *service request customer* need to reinput again |
| 5 | Input invalid username and password less than 6. | username = pureqwerty  password = pure | System display error message and *service request customer* need to reinput again |

### Employer System Test case 04 (ESTC-04):

*service request customer* can manage to add or edit Picture, Name, Company name, Address, Introduce yourself, and Telephone number in profile detail as *service request customer*.

**Description:**

This system test case testing for E-URS – 04:The *service request customer* can manage to add or edit Picture, Name, Company name, Address, Introduce yourself, and Telephone number in profile detail as *service request customer*.

**Prerequisite:**

*service request customer* need to Login to using system first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *service request customer* to *service request customer* mainpage

5. Click “Profile Here” to add or edit Profile

6. Input Name, Company name, Address, Introduce yourself, and Telephone number in to profile page and Browse picture to profile

7. System Update the recipe detail to database

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input all required field in add recipe detail form with all correct information. | Name : = Manuchet Manoi  Company name : =Perfect.vc  Address : =56 m.4 muang chiangmai 50133  Introduce yourself : =Employment HR at  Telephone number : =0855678909  Browse Pic : man.jpg | System bring *service request customer* to *service request customer* mainpage |
| 2 | Input some field in profile detail form. | Name : =Manuchet manoi  Company name : =Perfect.vc  Address : 56 m.4 muang chiangmai 50133  Introduce yourself :  Telephone number := 0855678909  Browse Pic : =man.jpg | System bring *service request customer* to *service request customer* mainpage |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 3 | Input Telephone number field in wrong type (not equal number) | Name : =Manuchet manoi  Company name : =Perfect.vc  Address : =56 m.4 muang chiangmai 50133  Introduce yourself : =Employment HR at  Telephone number := 08556aaaa  Browse Pic : =man.jpg | System display error message and *service request customer* need to reinput again |
| 4 | Input invalid wrong image type. | Name : =Manuchet manoi  Company name : =Perfect.vc  Address : =56 m.4 muang chiangmai 50133  Introduce yourself : = Employment HR  Telephone number : = 0855678909  Browse Pic : =man.gif | System display error message and *service request customer* need to reinput again |

### Employer System Test case 05 (ESTC-05):

*service request customer* can be Logout from web application

**Description:**

This system test case testing for E-URS – 05:*service request customer* can be Logout from web application

**Prerequisite:**

*service request customer* need to Login to using system first

**Test Script:**

1. *service request customer* clicks logout.

2. *service request customer* loged out of system.

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Click logout. | logout | The *service request customer* logging out from system and redirect first main page. |

### Employer System Test case 07 (ESTC-07):

***service request customer*** can post job and Detail as Project Name, Budget, Information, skill required and Software role categories to system that show to both of user as *service request customer* and *freelance supply*.

**Description:**

This system test case testing for E-URS – 07:The ***service request customer*** can post job and Detail as Project Name, Budget, Information, skill required and Software role categories to system that show to both of user as *service request customer* and *freelance supply*.

**Prerequisite:**

*service request customer* need to Login to using system first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *service request customer* to *service request customer* mainpage

5. Click “Project Here” to add Project

6. Detail as Project Name, Budget, Information, skill required and and Software role categories to system

7. System Update the recipe detail to database

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input all required field in add recipe detail form with all correct information. | Title : = Example Project  Budget : = 200.20  Information : = Example Information  Skill required : = HTML  Software role categories : = Web develop | System bring *service request customer* to *service request customer* mainpage |
| 2 | Not Selected any Skill required | Title : = Example Project  Budget : = 200.20  Information : = Example Information  Skill required : =  Software role categories : = Web develop | System bring *service request customer* to *service request customer* mainpage |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 3 | Input wrong type Budget | Title : = Example Project  Budget : = aaaaa  Information : = Example Information  Skill required : = HTML  Software role categories : = Web develop | System display error message and *service request customer* need to reinput again |
| 4 | Blank value in budget | Title : = Example Project  Budget : =  Information : = Example Information  Skill required : = HTML  Software role categories : = Web develop | System display error message and *service request customer* need to reinput again |

### Employer System Test case 08 (ESTC-08):

The ***service request customer*** can view Picture Name, Skill, and Address, Introduce yourself, and Telephone number in Profile detail of freelancer.

**Description:**

This system test case testing for E-URS – 08:The ***service request customer*** can view Picture Name, Skill, and Address, Introduce yourself, Skill required and Telephone number in Profile detail of freelancer.

**Prerequisite:**

*service request customer* need to Login to using system first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *service request customer* to *service request customer* mainpage

5. Click “find Freelancer Here” to see list of Freelancer

6. Click “See freelancer”

7. *service request customer* see freelancer Detail as Picture Name, Skill, and Address, Introduce yourself, Skill required and Telephone number to system

8. System Update the recipe detail to database

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | *service request customer* click See Freelancer in Freelancer list Name = pure | See Freelancer Button in Freelancer Name = pure | Display Freelancer  Name : = Manuchet Manoi  Company name : =Perfect.vc  Address : =56 m.4 muang chiangmai 50133  Skill Required : = HTML  Introduce yourself : =Dev  Telephone number : =0855678909  Browse Pic : man.jpg |

### Employer System Test case 09 (ESTC-09):

The ***service request customer*** can view posted job list in system.

**Description:**

This system test case testing for E-URS – 08:The ***service request customer*** can view posted job list in system.

**Prerequisite:**

*service request customer* need to Login to using system first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *service request customer* to *service request customer* mainpage

5. Click “total job” to see list of job in system

6. Click “See more”

7. *service request customer* see job Detail as Project Name, Budget, Information, skill required and Software role categories

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | *service request customer* click this job in job list Titlename = Example job | This job botton in job list titlename = Example job | Display Job  Name : Manuchet Manoi  Address : 56 m.4 muang chiangmai 50133  Introduce yourself : Dev kub pom  Telephone number : 0855678922  Browse Pic : man.jpg |

### Employer System Test case 10 (ESTC-10):

E-URS – 10: The ***service request customer*** can view list of bidder in each project include budget & bidding detail

**Description:**

This system test case testing for E-URS – 10:The ***service request customer*** can view list of bidder in each project include budget & bidding detail

**Prerequisite:**

*service request customer* need to Login to using system first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *service request customer* to *service request customer* mainpage

5. Click “total job” to see list of job in system

6. Click “This job”

7. *service request customer* see Bid Detail as Budget , Software role Require and project owner

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | *service request customer* click this job in job list Titlename = Example job | This job botton in job list titlename = Example job | Display Bid Below jobdetail as Bid Detail as Budget , Software role Require and project owner |

### Freelancer System Test case 01 (FSTC-01):

*Freelancer Supply* can be Register and select type of user as *Freelancer Supply*

**Description:**

This system test case testing for E-URS-01. *Freelancer Supply* can be Register and select type of user as *Freelancer Supply*

**Test Script:**

1**.** Click “Register” button in first page of web-application

2. System Provide UI system provide UI to input Email, select type of user, password, confirm password

3. Select type of user = Freelancer, input Email Address input password and Comfirm password and click Create

4. The System save data type of user, Email Address, input password and Confirm password to database

5. The System bring *Freelancer Supply* to index mainpage

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Select Type of user = Freelancer,  Input Email ,password and Confirm password password & Confirm is match | Select Type of user = Freelancer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System bring *Freelancer Supply* to web-application mainpage |
| 2 | Select Type of user = Freelancer  Input wrong type Email ,  Password & Confirm is match | Select Type of user = Freelancer  Input Email = “pureman.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *Freelancer Supply* need to reinput again |
| 3 | Select Type of user = Freelancer  Input Email is already exist in system, Password & Confirm is match | Select Type of user = Freelancer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *Freelancer Supply* need to reinput again |
| 4 | Select Type of user = Freelancer  Input Email is already exist in system, Password & Confirm is match | Select Type of user = Freelancer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *Freelancer Supply* need to reinput again |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 5 | Select Type of user = Freelancer  Input Email in system, Password & Confirm is not match | Select Type of user = Freelancer  Input Email = “pureman9@gmail.com”  Input password = “pure0835”  Input Confirm password = “pure08354” | System display error message near Confirm password input box and *Freelancer Supply* need to reinput again |
| 6 | Select Type of user = Freelancer  Not Input Email in system, Password & Confirm Password is not match | Select Type of user = Freelancer  Input Email = “null”  Input password = “pure0835”  Input Confirm password = “pure0835” | System display error message near Email input box and *Freelancer Supply* need to reinput again |
| 7 | Select Type of user = Freelancer Input Email in system, Password & Confirm Password is more than less than 6 | Select Type of user = Freelancer  Input Email = “pureman9@gmail.com”  Input password = “pure”  Input Confirm password = “pure” | System display error message near Email input box and *Freelancer Supply* need to reinput again |

### Freelancer System Test case 02 (FSTC-02):

*Freelancer Supply* can be Register and select type of user as *Freelancer Supply*

**Description:**

This system test case testing for E-URS – 02:*Freelancer Supply* can be ConfirmRegister in their Email that they input on previously

**Prerequisite:**

*Freelancer Supply* need to Register first

**Test Script:**

1**.** Login to Email address that register in previously (in this case use pureman9@gmail.com)

2. System will send link of ConfirmRegister to Email Address that registered on previously

3. Click link in Email address

4. The System bring *Freelancer Supply* to input Username & ConfirmRegisterCode

5. The System bring *Freelancer Supply* to index mainpage

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input Username &ConfirmRegisterCode right type | Input Username = “pureqwerty”  Input ConfirmRegisterCode = “54117316” | System bring *Freelancer Supply* to web-application mainpage |
| 2 | Not Input Username but InputConfirmRegisterCode right type | Input Username = “null”  Input ConfirmRegisterCode = “54117316” | System display error message near Username input box and *Freelancer Supply* need to reinput again |
| 3 | Input Username is already exist in database but input ConfirmRegisterCode right type | Input Username = “pureman9@gmail.com”  Input ConfirmRegisterCode = “54117316” | System display error message near Username input box and *Freelancer Supply* need to reinput again |
| 4 | Input Username but wrong ConfirmRegisterCode | Input Username = “pureman9@gmail.com”  Input ConfirmRegisterCode = “11111111” | System display error message near ConfirmRegisterCode input box and *Freelancer Supply* need to reinput again |

### Freelancer System Test case 03 (FSTC-03):

*service request customer* can be Register and select type of user as *service request customer*

**Description:**

This system test case testing for E-URS – 03:*service request customer* can be Login to use Web Application after get user account.

**Prerequisite:**

*service request customer* need to ConfirmRegister first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *Freelancer Supply* to *Freelancer Supply* mainpage

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input correct username and password. | username = pureqwerty  password = pure0835 | System bring *Freelancer Supply* to *Freelancer Supply* mainpage |
| 2 | Input some fields. | username = null  password = pure0835 | System display error message and *Freelancer Supply* need to reinput again |
| 3 | Not enter both username and password | username = null  password = null | System display error message and *Freelancer Supply* need to reinput again |
| 4 | Input invalid username and password. | username = abcd  password = 123456 | System display error message and *Freelancer Supply* need to reinput again |
| 5 | Input invalid username and password less than 6. | username = pureqwerty  password = pure | System display error message and *Freelancer Supply* need to reinput again |

### Freelancer System Test case 04 (FSTC-04):

*Freelancer Supply* can manage to add or edit Picture, Name, Address, Introduce yourself, and Telephone number in profile detail as *Freelancer Supply*.

**Description:**

This system test case testing for E-URS – 04:The *Freelancer Supply* can manage to add or edit Picture, Name, Company name, Address, and Telephone number in profile detail as *Freelancer Supply*.

**Prerequisite:**

*Freelancer Supply* need to Login to using system first

**Test Script:**

1**.** Click “Login” button in first page of web-application

2. System Provide UI system provide UI to input Username & Password

3. Select input Username & password and click confirm.

4. The System bring *Freelancer Supply* to *Freelancer Supply* mainpage

5. Click “Profile Here” to add or edit Profile

6. Input Name, Company name, Address, Introduce yourself, and Telephone number in to profile page and Browse picture to profile

7. System Update the recipe detail to database

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Input all required field in add recipe detail form with all correct information. | Name : Manuchet Manoi  Address : 56 m.4 muang chiangmai 50133  Introduce yourself : Employment HR at  Telephone number : 0855678909  Browse Pic : man.jpg | System bring *Freelancer Supply* to *Freelancer Supply* mainpage |
| 2 | Input some field in profile detail form. | Name : Manuchet manoi  Address : 56 m.4 muang chiangmai 50133  Introduce yourself :  Telephone number : 0855678909  Browse Pic : man.jpg | System bring *Freelancer Supply* to *Freelancer Supply* mainpage |

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 3 | Input Telephone number field in wrong type (not equal number) | Name : Manuchet manoi  Address : 56 m.4 muang chiangmai 50133  Introduce yourself : Employment HR at  Telephone number : 08556aaaa  Browse Pic : man.jpg | System display error message and  *Freelancer Supply* need to reinput again |
| 4 | Input invalid wrong image type. | Name : Manuchet manoi  Address : 56 m.4 muang chiangmai 50133  Introduce yourself : Employment HR at  Telephone number : 0855678909  Browse Pic : man.gif | System display error message and  *Freelancer Supply* need to reinput again |

### Freelancer System Test case 05 (FSTC-05):

*Freelancer Supply* can be Logout from web application

**Description:**

This system test case testing for E-URS – 05:*Freelancer Supply* can be Logout from web application

**Prerequisite:**

*Freelancer Supply* need to Login to using system first

**Test Script:**

1. *Freelancer Supply* clicks logout.

2. *Freelancer Supply* logging out of system.

**Test Case:**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Description | Input | Expect output |
| 1 | Click logout. | logout | The  *Freelancer Supply* logging out from system and redirect first main page. |