**TEAMDUTY Web-application**

Software Requirement Specification

**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

## Objective

The Objective of this (SRS) Software requirement specification is establish requirement for TEAMDUTY Web application and show all of description and specification of TEAMDUTY Web application. And this Software requirement specification is guidelines for develop software correctly and match with requirement.

## Intended Audience and Reading Suggestions

### 1.2.1 Intended Audience

***Development team***

* Software Requirement specification help Development team understand the scope of the project and detail of software before implementation.
* Software Requirement specification help development team develop program efficiently.
* Software Requirement specification help development team can check software product that match with requirement of program or not.

***User***

* Software requirement specification will help the user to know detail and description of software product.
* User can check that the requirements in software requirement specification match with their requirement or not.
* Software requirement specificationhelp user and development understand in program corresponding

### 1.2.2 Reading Suggestions

This software requirement specification will classify requirements by user into 3 type including **service request customer**, **freelance supply** and **Administrator**. And for each progress we will separated to send in each progress as followed;

|  |  |
| --- | --- |
| **Submit on-** | **User Requirement specification of-** |
| May | Feature 1 # Account Management System  Feature 2 # Service request customer management system. |
| July | Feature 3 # Search system  Feature 4 # Software-freelance supply management system  Feature 5 # Communication system |
| September | Feature 6 # Verification system  Feature 7 # Admistrator Management system |

## 1.3 Project Scope

The scope of this software requirement speciation is strategies to establish the system that:

* The Software System is Web-application.
* The Software System provide only software side.
* The Software System divide to two main part that is software human resource, and software job auction project.
* Show requirement and detail of Software system

TEAMDUTY – web application is a website application. This will provide a system to manage about software freelance finding job software human resource.

# Chapter Two | Overall Description

## 2.1 Product Perspective

TEAMDUTY – Web application to contract both of Software freelancer and Software employer. The application provides feature help Software freelancer to find software job. And help Software employer to find Software human resource in software project.

## 2.2 Product Features (need to add more)

TEAMDUTY – Web application will provide feature is different feature in difference type of user. Both of Software Freelancer and Software Employer must be register before use Webapllication put their E-mail and password to the registration system. They will got verify e-mail on their e-mail that they put on previously.

TEAMDUTY – Web application has feature difference if type of user is difference but every feature in this system will be related in this system.

## 2.3 User Classes and Characteristics

There are 3 type of user, which interact to TEAMDUTY-web application system. That is **service request customer, freelance supply** and **Administrator**

URS1 - The users can login, log out after register and confirm register themselves as both as *service request customer* and *freelance supply*.

URS2 -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*.

URS3 -The ***freelance supply*** can manage to Add or Edit Picture Name, Skill, Address, Introduce yourself, and Telephone number in Profile detail as *freelance supply*.

URS4 -The ***service request customer*** can post job and Detail as Project Name, Budget, Information and skill required to system that show to both of user as *service request customer* and *freelance supply*.

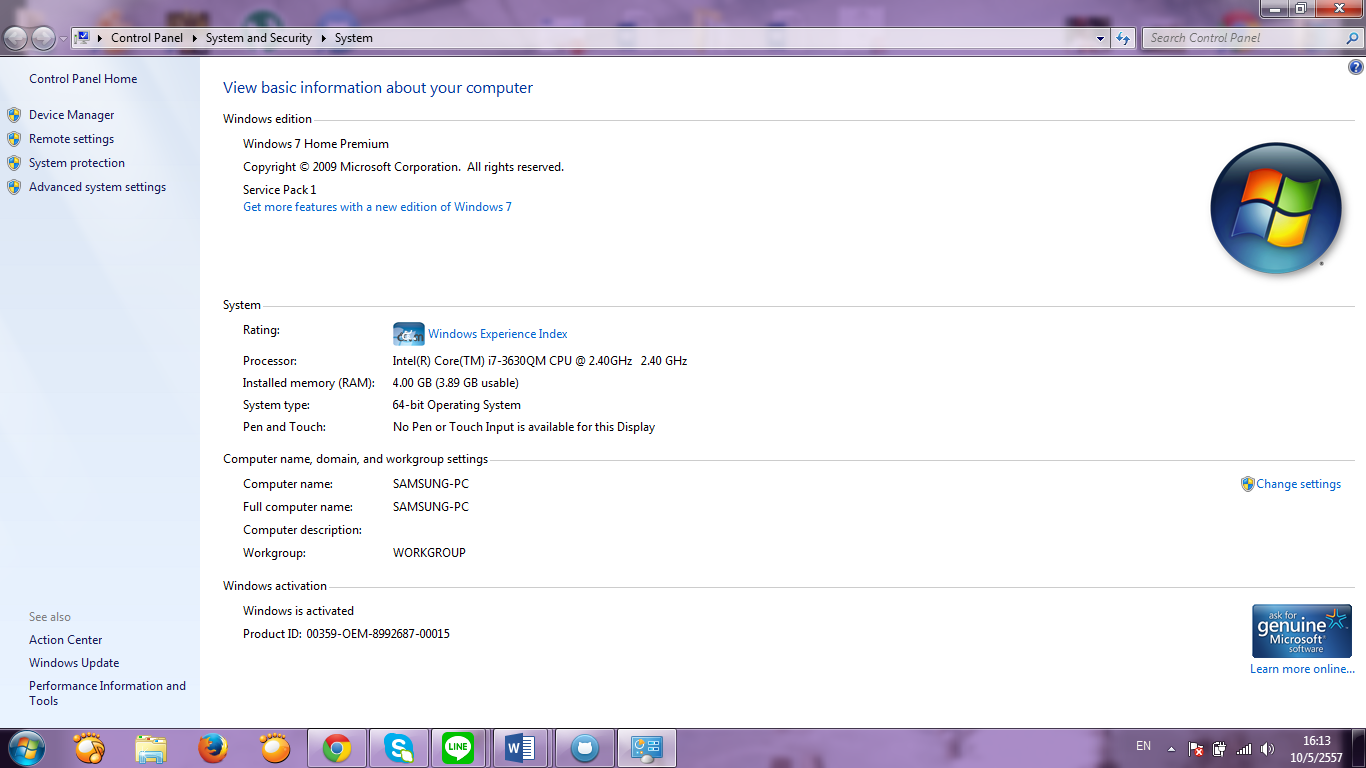
URS5 -The ***service request customer*** can send request to freelancer form ‘freelancer requested bid message’ on the project that posted.

URS6 -The ***service request customer*** can cancel request form ‘freelancer requested bid message’ on the project that posted.

URS7 -The ***service request customer*** can view Picture Name, Skill, and Address, Introduce yourself, and Telephone number in Profile detail of freelancer.

## 2.4 Operation Environment

**Notebook or PC**



**Browser**

* Google Chrome
* Firefox

## 2.5 Design and Implementation Constraints

Internet Connection is required of use TEAMDUTY-Web application

* 1 email can be only used to register for 1 account.

# Chapter Three | User Requirement

User is a person using a generic system. There are 3 type of user, which interact to Team Duty Web application system is **service request customer**, **freelance supply** and **Administrator**. We will be separated type

**Feature #1 - Account management system.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URS.NO** | **Description** | **Users related** | | |
| **service request customer** | **freelance supply** | **Administrator** |
| URS-01 | The users can login, log out after register and confirm register themselves as both as *service request customer* and *freelance supply*. |  |  |  |
| URS-02 | 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*. |  |  |  |
| URS-03 | The ***freelance supply*** can manage to Add or Edit Picture Name, Skill, Address, Introduce yourself, and Telephone number in Profile detail as *freelance supply*. |  |  |  |

**Feature #2 - Service request customer management system.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URS.NO** | **Description** | **Users related** | | |
| **service request customer** | **freelance supply** | **Administrator** |
| URS-04 | The ***service request customer*** can post job and Detail as Project Name, Budget, Information and skill required to system that show to both of user as *service request customer* and *freelance supply*. |  |  |  |
| URS-05 | The ***service request customer*** can send request to freelancer form ‘freelancer requested bid message’ on the project that posted. |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **URS.NO** | **Description** | **Users related** | | |
| **service request customer** | **freelance supply** | **Administrator** |
| URS-06 | The ***service request customer*** can cancel request form ‘freelancer requested bid message’ on the project that posted. |  |  |  |
| URS-07 | The ***service request customer*** can view Picture Name, Skill, and Address, Introduce yourself, and Telephone number in Profile detail of freelancer. |  |  |  |

* = User associated with that #URS

# Chapter Four | System Feature

## 4.1 Software Requirement Specification

**Feature #1 account management system.**

**URS1** - The users can login, log out after register and confirm register themselves as both as *service request customer* and *freelance supply*.

**URS2** - 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*.

**URS3** - The ***freelance supply*** can manage to Add or Edit Picture Name, Skill, Address, Introduce yourself, and Telephone number. Profile detail as *freelance supply*.

**URS1: The users can login, log out after register themselves as both as *service request customer* and *freelance supply*.**SRS 01- The system provides the UI which receive email and register-code that random by system   
SRS 02 - The system shall check the email format.

SRS 03 - The system can check the password format. The password format must be 6-20 characters.

SRS 04 - The system can check the matching of password and confirm password.

SRS 05 - The system shall send link to email to confirm registration.

SRS 06 - The system shall add new user information to database using email, password and confirm password.

SRS 07 - The system shall display error message “The email must not empty”

SRS 08 - The system shall display error message “The password must not empty”

SRS 09 - The system shall display error message “The confirm password must not empty”

SRS 10 - The system shall display error message “The email is not valid”

SRS 11 - The system shall display the error message “The password length must be 6-20”

SRS 12 - The system shall display the error message “The password and confirm password is not match”

SRS 13 - The system shall display the error message “The email is already existed”.

SRS 14 - The system shall connect to database

**URS2** - 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*

SRS 15 - The system shall provide login interface which receive username (email) and password.

SRS 16 - The system shall validate username and password which matched with database.

SRS 17 - The system shall display the error message. “Error, the username and password is not correct”.

SRS 18 - The system can check the password format. The password format must be 6-20 characters.

SRS 07- The system shall display error message “The email must not empty”

SRS 08 - The system shall display error message “The password must not empty”

SRS 14 - The system shall connect to database

**URS3** - The ***freelance supply*** can manage to Add or Edit Picture Name, Skill, Address, Introduce yourself, and Telephone number. Profile detail as *freelance supply*.

SRS 17- The system shall provide logout interface for logged out the system.

SRS 18- The system shall log username (email) and password out of the system.

SRS 19- The system shall provide user interface for receive name, address, company name, Address, Introduce yourself, and Telephone number

SRS 20- The system shall provide button to browse the logo/picture.

SRS 21- The system shall allow user edit the profile.

SRS 14- The system shall connect to database

SRS 22- The system shall record profile information to database.

SRS 23- The system shall display the error message “some information is blank”.  
  
  
**//URS5-The user can manage their profile as freelancer.**

SRS 19 - The system shall provide user interface for receive name, Address, Introduce yourself, and Telephone number.

SRS 20 - The system shall provide button to browse the logo/picture.

SRS 21 - The system shall allow user edit the profile.

SRS 22 - The system shall record information to database.

SRS 23 - The system shall display the error message “some information is blank”.

SRS 24 - The system shall provide list of skill to receive skill from freelancer.

SRS 25 - The system shall display the error message “you must choose at least one skill”.

SRS 14- The system shall connect to database

**Feature #2 enterprise management system.**

**URS4 -** The ***service request customer*** can post job and Detail as Project Name, Budget, Information and skill required to system that show to both of user as *service request customer* and *freelance supply*.

**URS5 -** The ***service request customer*** can send request to freelancer form ‘freelancer requested bid message’ on the project that posted.  
**URS6 -** The ***service request customer*** can cancel request form ‘freelancer requested bid message’ on the project that posted.

**URS7 -** The ***service request customer*** can view Picture Name, Skill, and Address, Introduce yourself, and Telephone number in Profile detail of freelancer.   
  
**URS4 -** The ***service request customer*** can post job and Detail as Project Name, Budget, Information and skill required to system that show to both of user as *service request customer* and *freelance supply*.

SRS 26 - The system shall provide user interface for receive Project Name, Budget,

Project Detail and skill required to system

SRS 27 - The system shall record job’s information to database

SRS 14 - The system shall connect to database

SRS 28 - The system shall display information of job to user

SRS 29 - The system shall display the message “Success, you are already posted job.”

SRS 30 - The system shall display the error message “Some information must be require.”

**URS5 -** The ***service request customer*** can send request to freelancer form ‘freelancer

requested bid message’ on the project that posted.

SRS 31 - The system shall provide list of bidding of freelancer.

SRS 32 - The system shall provide button to request freelancer.

SRS 33 - The system shall send request to freelancer.

SRS 14 - The system shall connect to database.

SRS 34 - The system shall display the message “Success, You’ve already sent your request!”

**URS6 -** The ***service request customer*** can cancel request form ‘freelancer requested bid

message’ on the project that posted.

SRS 35 - The system shall provide list of requested freelancer.

SRS 36 - The system shall provide button to cancel requested freelancer.

SRS 37 - The system shall send message to freelancer.

SRS 14 - The system shall connect to database.

SRS 38 - The system shall display the message “Success, You’ve already cancel the request!”

**URS7 -** The ***service request customer*** can view Picture Name, Skill, and Address, Introduce

Yourself, and Telephone number in Profile detail of freelancer..

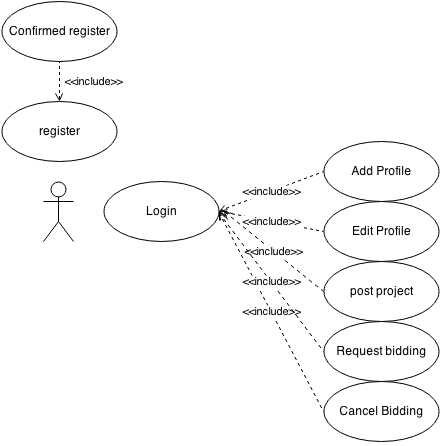
SRS 39 - The system shall display information of freelancer.

SRS 14 - The system shall connect to database.

## 4.2 Use case Description & Activity Diagram

### 4.2.1 Use case Description and Activity Diagram

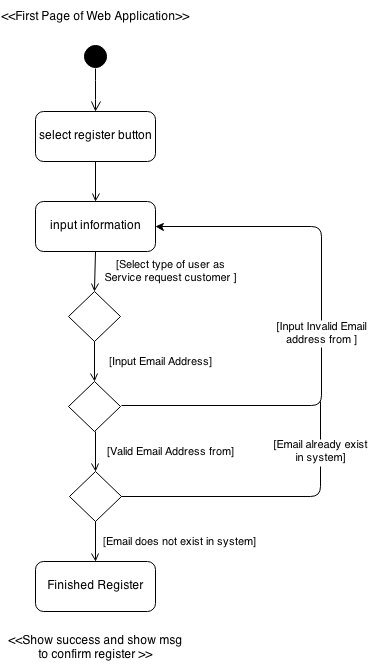
Use case description will be create follow the submitted user requirement in each progress. Some use cases will not have their use case description. That means the user requirements, which related to those use cases, were not submitted in any progress yet. Both use case and activity diagram will be grouped follow users of system (**service request customer**, **freelance supply** and **Administrator**) same as grouping of requirements.

***Use case of service request customer for progress 1***

***Figure 4.2.1.1 Use Case Description of service request customer for progress 1***

EUC – 01: Register for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-01 | |
| Use Case Name: | Register for **service request customer** | |
| Short Description: | **service request customer** get account for using web-application | |
| Actor: | **service request customer** | |
| Pre-condition: | **service request customer** does not have account | |
| Post-condition | **service request customer** get link form E-mail that they using for register | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **service request customer** selected register button in first page of Web-application |  |
|  | 2.system provide UI to input Email, select type of user and confirm registration code |
| 3. **Service request customer** Selected type of user as service request customer, input their email address and registration code. |  |
|  | 4.System will send link to confirm register to email that user input on previously |
| 5. **Service request customer** click link on Email to open their profile. |  |
| Alternative Flow: | 3. Service request customer input invalid type of email or email already existed in system   * 3.1 system display error message * 3.2 **Service request customer** input email address again. | |
| Exceptional: | - | |
| Priority: | High | |

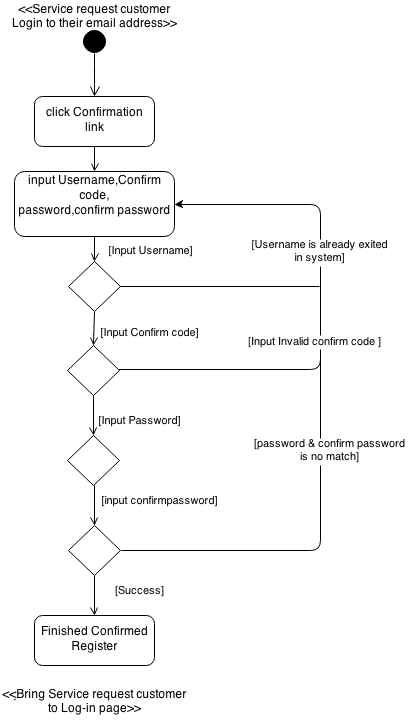
EAC – 01: Register for **service request customer**

- Information Include type of User that system provided as *service request customer* and *freelance supply*, Email Address

- In Registration page they will send confirmation code to email that *service request customer* input on previously.

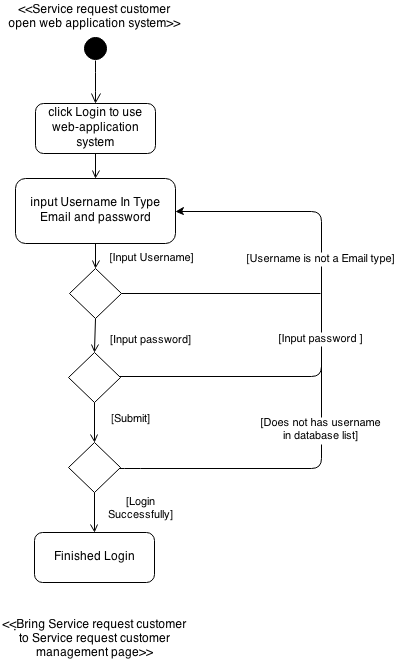
EUC – 02: Confirm register for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-02 | |
| Use Case Name: | Confirm register for **service request customer** | |
| Short Description: | **service request customer** confirm register in their email address | |
| Actor: | **service request customer** | |
| Pre-condition: | **service request customer** Register as **service request customer** on web-application | |
| Post-condition | **Service request customer** get account to using web-application. | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **service request customer** login to their email address |  |
| 2. **service request customer** click link in Email address |  |
|  | 3.system provide Confirmation Page to Input Username, Confirm-code, password and re-password in confirmation page |
| 4. **service request customer** input Username ,Confirm-code, password and re-password in confirmation page |  |
|  | 5.system display message to show success confirmation Register |
| Alternative Flow: | 4. Service request customer input Username that already exited ,wrong confirmed code or password & confirmed password does not match   * 4.1 system display error message * 4.2 **Service request customer** input all of Username confirmed code, password & confirmed-password again. | |
| Exceptional: | - | |
| Priority: | High | |

******EAC – 02: Confirm register for **service request customer**

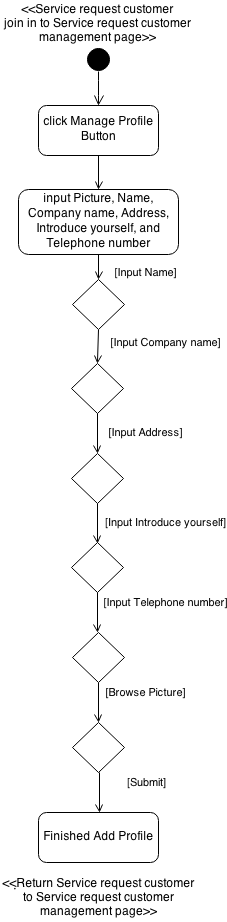
EUC – 3: Login for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-03 | |
| Use Case Name: | Login for **service request customer** | |
| Short Description: | **service request customer** Login to using web-application system | |
| Actor: | **service request customer** | |
| Pre-condition: | **service request customer** Confirm register as **service request customer** on web-application | |
| Post-condition | **Service request customer** Login to use Web-application system successfully. | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **Service request customer** join to web-application |  |
|  | 2. Systemprovide UI to input as Username in type E-mail and password. |
| 3. **Service request customer** input Username in type E-mail and password. |  |
|  | 4.System find Username in database list |
|  | 5.System bring **Service request customer** to main page of **Service request customer** management |
| Alternative Flow: | 3. **Service request customer** input wrong type Username(not Email type) to Login or wrong password that not match with username   * 3.1 system display error message * 3.2 **Service request customer** input all of Username and password to login again. | |
| Exceptional: | 4. **Service request customer** account **does not exist in the database**   * 4.1 System validation finds that the user record does not exist in the database. * 4.2 Systems alerts the user that their record does not exist. * 4.3 Use case ends | |
| Priority: | High | |

EAC – 03: Login for **service request customer**

EUC – 04: Add Profile for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-04 | |
| Use Case Name: | Add Profile for **service request customer** | |
| Short Description: | **service request customer** Add profile to introduce about themselves to show in web application | |
| Actor: | **service request customer** | |
| Pre-condition: | **Service request customer** Login to use Web-application system successfully. | |
| Post-condition | **Service request customer** show information about themselves to other user in web application | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **Service request customer** come to **Service request customer** main page |  |
| 2. **Service request customer** click Profile manage |  |
|  | 3. Systemprovide UI to input Picture, Name, Company name, Address, Introduce yourself, and Telephone number. |
| 4. **Service request customer** input Picture, Name, Company name, Address, Introduce yourself, and Telephone number in to UI of system |  |
|  | 5.System retrieve Picture, Name, Company name, Address, Introduce yourself, and Telephone number information save in to database |
|  | 6.System bring **Service request customer** to main page of **Service request customer** management |
| Alternative Flow: | - | |
| Exceptional: | - | |
| Priority: | Normal | |

EAC – 04: Add Profile for **service request customer**

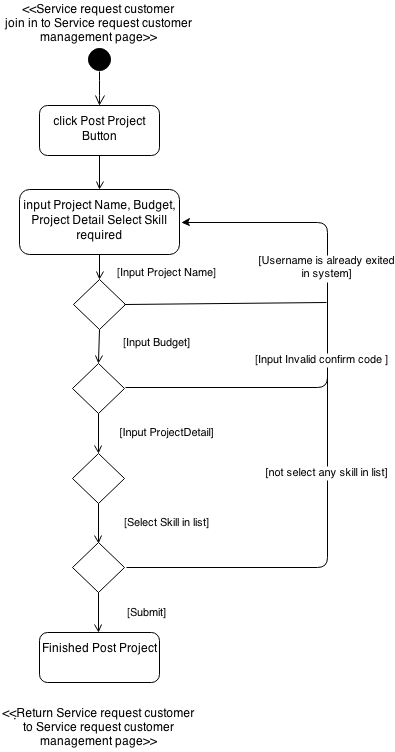
EUC – 05: Edit Profile for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-05 | |
| Use Case Name: | Edit Profile for **service request customer** | |
| Short Description: | **service request customer** Edit profile of themselves to show in web application | |
| Actor: | **service request customer** | |
| Pre-condition: | **Service request customer** Login to use Web-application system successfully. | |
| Post-condition | **Service request customer** show information about themselves to other user in web application | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **service request customer** |  |
|  |  |
|  |  |
|  |  |
|  |  |
| Alternative Flow: |  | |
| Exceptional: |  | |
| Priority: |  | |

EAC – 04: Edit Profile for **service request customer**

EUC – 05: Post Project for **service request customer**

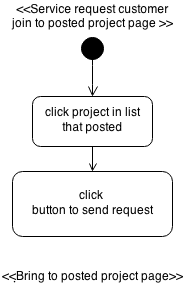
|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-05 | |
| Use Case Name: | Post Project for **service request customer** | |
| Short Description: | **service request customer** Post and show Project to show other user in web-application system | |
| Actor: | **service request customer** | |
| Pre-condition: | **Service request customer** Login to use Web-application system successfully. | |
| Post-condition | **Service request customer** show project information into web application | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **Service request customer** come to **Service request customer** main page |  |
| 2. **Service request customer** click post project button |  |
|  | 3. Systemprovide UI to input Project Name, Budget, Information and skill required |
| 4. **Service request customer** input Project Name, Budget, Project Detail and skill required in to UI of system |  |
|  | 5.System retrieve Project Name, Budget, Project Detail and Skill required save in to database |
|  | 6.System bring **Service request customer** to main page of **Service request customer** management |
| Alternative Flow: | 3. **Service request customer** does not input project Name, Budget or not select any skill in list   * 3.1 system display error message * 3.2 **Service request customer** input all of project Name, Budget and Select Skill required in list again | |
| Exceptional: | - | |
| Priority: | High | |

EAC – 05: Post Project for **service request customer**

EUC – 06: **Service request customer** send request to Freelancer

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-06 | |
| Use Case Name: | **Service request customer** send request to Freelancer | |
| Short Description: | **service request customer** send request to other Freelancer that come to bid project that posted | |
| Actor: | **service request customer** | |
| Pre-condition: | **Service request customer** post project in to system | |
| Post-condition | Message will send to **Freelancer** that **Service request customer** select to send request from bided **Freelancer** list | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **Service request customer** come to **Service request customer** Project posted page |  |
| 2. **Service request customer** click project that showed in list |  |
|  | 3.Systemshow Project Name, Budget, Project Detail and Skill required below of Project Information system show **Freelancer** Name that bided , bided detail , and button to send request. |
| 4. **Service request customer** click button to send request |  |
|  | 5. System will send message to Freelancer that bided project. |
| Alternative Flow: | - | |
| Exceptional: | - | |
| Priority: | Normal | |

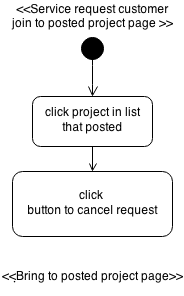
EAC – 06: **Service request customer** send request to Freelancer



EUC – 07: **Service request customer** cancel request to Freelancer

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-07 | |
| Use Case Name: | **Service request customer** cancel request to Freelancer | |
| Short Description: | **service request customer** cancel request to other Freelancer that **service request customer** send request in previously | |
| Actor: | **service request customer** | |
| Pre-condition: | **Service request customer** send request to Freelancer | |
| Post-condition | Message will send to **Freelancer** that **Service request customer** and name of Freelancer will be remove from list | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **Service request customer** come to **Service request customer** Project posted page |  |
| 2. **Service request customer** click project that showed in list |  |
|  | 3.Systemshow Project Name, Budget, Project Detail and Skill required below of Project Information system show **Freelancer** Name that bided , bided detail , and button to send request. |
|  | 4. System will show list of Freelancer that send request on previously. |
| 4. **Service request customer** click button to cancel request |  |
|  | 5. System will send message to **Freelancer** that bided project.and remove list of **freelancer** from project |
| Alternative Flow: | - | |
| Exceptional: | 4. If **Service request customer** didn’t sent request to **Freelancer** cancel request will be not show in project detail page. | |
| Priority: | Normal | |

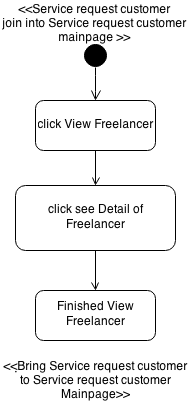
EUC – 07: **Service request customer** cancel request to Freelancer



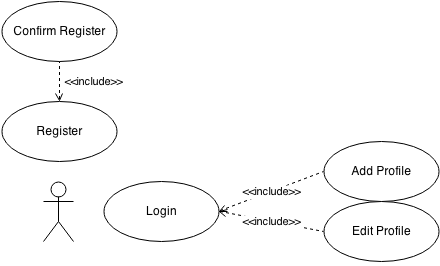
EUC – 08: **Service request customer** can view profile of Freelancer

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-08 | |
| Use Case Name: | **Service request customer** can view profile of Freelancer | |
| Short Description: | **service request customer** can view profile information of freelancer | |
| Actor: | **service request customer** | |
| Pre-condition: | **Service request customer** Login to using web application successfully | |
| Post-condition |  | |
| Flow of Event: | **service request customer** | **system** |
|  | 1. **Service request customer** come to **Service request customer** main page |  |
| 2. **Service request customer** click View freelancer |  |
|  | 3. Systemprovide list of freelancer that available in database. |
| 4. **Service request customer** click detail of freelancer in list. |  |
|  | 5.System will show Picture Name, Skill, Address, Introduce yourself, and Telephone number in Profile detail of **Freelancer** |
| Alternative Flow: | - | |
| Exceptional: | - | |
| Priority: | Normal | |

EAC – 08: **Service request customer** can view profile of Freelancer



***Use case of freelance supply for progress 1***

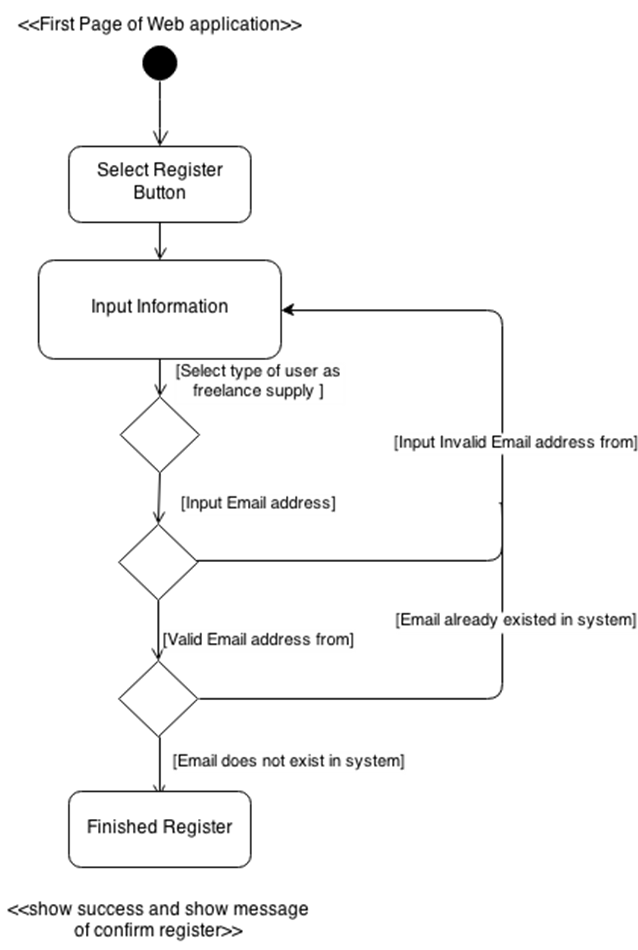
******

***Figure 4.2.1.2 Use Case Description of freelance supply for progress 1***

FUC – 01: Register for **freelance supply**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-01 | |
| Use Case Name: | Register for **freelance supply** | |
| Short Description: | **freelance supply** get account for using web-application | |
| Actor: | **freelance supply** | |
| Pre-condition: | **freelance supply** does not have account | |
| Post-condition | **freelance supply** get link form E-mail that they using for register | |
| Flow of Event: | **service request customer** | **system** |
|  | 1.**freelance supply** selected register button in first page of Web-application |  |
|  | 2.system provide UI to input Email, select type of user and confirm registration code |
| 3. **freelance supply** Selected type of user as service request customer, input their email address and registration code. |  |
|  | 4.System will send link to confirm register to email that user input on previously |
| 5. **freelance supply** click link on Email to open their profile. |  |
| Alternative Flow: | 3. Service request customer input invalid type of email or email already existed in system   * 3.1 system display error message * 3.2 **freelance supply** input email address again. | |
| Exceptional: | - | |
| Priority: | High | |

FAC – 01: Register for **freelance supply**

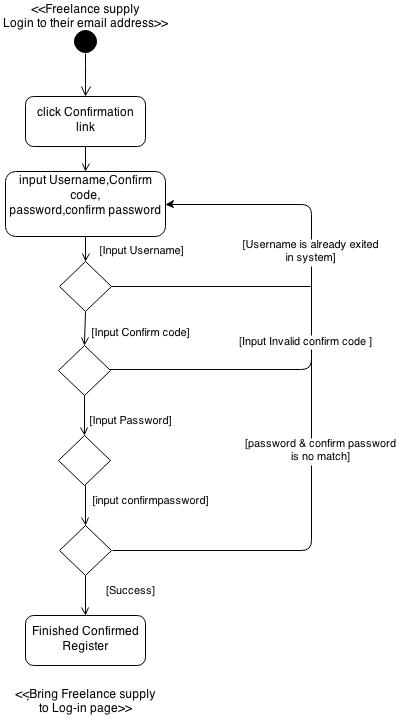


- Information Include type of User that system provided as *service request customer* and *freelance supply*, Email Address

- In Registration page they will send confirmation code to email that *service request customer* input on previously.

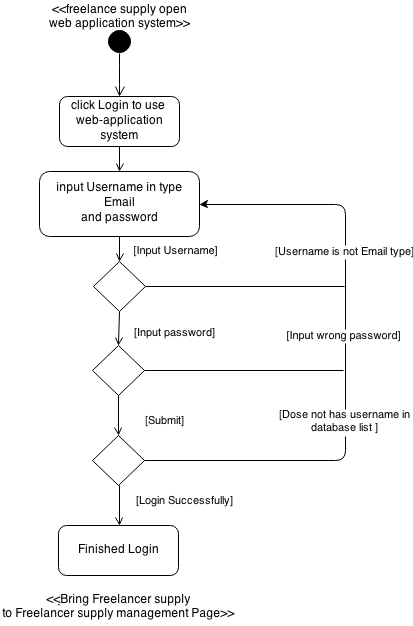
FUC – 02: Confirm register for **freelance supply**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-02 | |
| Use Case Name: | Confirm register for **freelance supply** | |
| Short Description: | **freelance supply** confirm register in their email address | |
| Actor: | **freelance supply** | |
| Pre-condition: | **freelance supply** Register as **freelance supply** on web-application | |
| Post-condition | **freelance supply** get account to using web-application. | |
| Flow of Event: | **freelance supply** | **system** |
|  | 1. **freelance supply** login to their email address |  |
| 2. **freelance supply** click link in Email address |  |
|  | 3.system provide Confirmation Page to Input Username, Confirm-code, password and re-password in confirmation page |
| 4. **freelance supply** input Username ,Confirm-code, password and re-password in confirmation page |  |
|  | 5.system display message to show success confirmation Register |
| Alternative Flow: | 4. Service request customer input Username that already exited ,wrong confirmed code or password & confirmed password does not match   * 4.1 system display error message * 4.2 **freelance supply** input all of Username confirmed code, password & confirmed-password again. | |
| Exceptional: | - | |
| Priority: | High | |

FUC – 02: Confirm register for **service request customer**

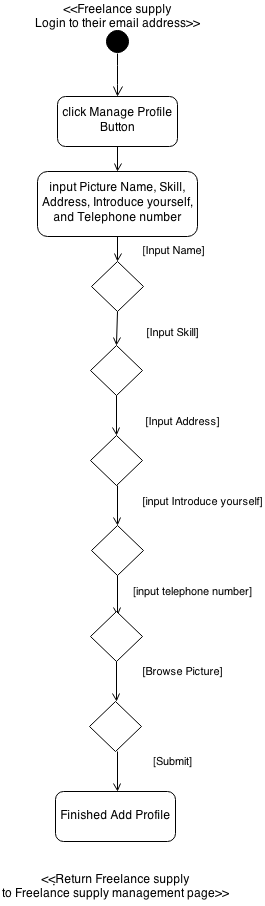
FUC – 3: Login for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-03 | |
| Use Case Name: | Login for **freelance supply** | |
| Short Description: | **freelance supply** Login to using web-application system | |
| Actor: | **service request customer** | |
| Pre-condition: | **freelance supply** Confirm register as **freelance supply** on web-application | |
| Post-condition | **freelance supply** Login to use Web-application system successfully. | |
| Flow of Event: | **freelance supply** | **system** |
|  | 1. **freelance supply** join to web-application |  |
|  | 2. Systemprovide UI to input as Username in type E-mail and password. |
| 3. **freelance supply** input Username in type E-mail and password. |  |
|  | 4.System find Username in database list |
|  | 5.System bring **freelance supply** to main page of **freelance supply** management |
| Alternative Flow: | 3. **freelance supply** input wrong type Username(not Email type) to Login or wrong password that not match with username   * 3.1 system display error message * 3.2 **freelance supply** input all of Username and password to login again. | |
| Exceptional: | 4. **freelance supply** account **does not exist in the database**   * 4.1 System validation finds that the user record does not exist in the database. * 4.2 Systems alerts the user that their record does not exist. * 4.3 Use case ends | |
| Priority: | High | |

EAC – 03: Login for **freelance supply**

FUC – 04: Add Profile for **service request customer**

|  |  |  |
| --- | --- | --- |
| Use Case ID: | EUC-04 | |
| Use Case Name: | Add Profile for **freelance supply** | |
| Short Description: | **freelance supply** Add profile to introduce about themselves to show in web application | |
| Actor: | **freelance supply** | |
| Pre-condition: | **freelance supply** Login to use Web-application system successfully. | |
| Post-condition | **freelance supply** show information about themselves to other user in web application | |
| Flow of Event: | **freelance supply** | **system** |
|  | 1. **freelance supply** come to **Service request customer** main page |  |
| 2. **freelance supply** click Profile manage |  |
|  | 3. Systemprovide UI to input Picture, Name, Company name, Address, Introduce yourself, and Telephone number. |
| 4. **freelance supply** input Picture Name, Skill, Address, Introduce yourself, and Telephone number in to UI of system |  |
|  | 5.System retrieve Picture Name, Skill, Address, Introduce yourself, and Telephone number information save in to database |
|  | 6.System bring **freelance supply** to main page of **freelance supply** management |
| Alternative Flow: |  | |
| Exceptional: |  | |
| Priority: | Normal | |

FAC – 04: Add Profile for **service request customer**

## 4.3 Quality Requirement

**Product operation factors**

* **Correctness** - The software product should able to provide more than 90% correctness of data from user traditional request.
* **Reliability** - The software should able to handle more than 90% of traditional activity with less than 10% of software’s failure.
* **Integrity** - The software should able identify between customer and shop owner. - The software should able to limit a group of person who can modify the data.
* **Usability** - The people who use software product as his first time should able to use all features within 20 minutes.

**Product revision factors**

* **Maintainability** - The software should have 20-30% of comment comparing with the whole LOC to support the future maintenance activity.
* **Testability** - The software should able to be tested 100% of it defined routine and functionality.

**Product transition factors**

* **Reusability** - More than 20% part of finished software product should able to be reused in future development.

# Chapter Five | Traceability Record