**TEAMDUTY Web-application**

Software Design Document

**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 (SDD) Software Design Document is establish design for TEAMDUTY Web application and show all of program design and data of TEAMDUTY Web application. And this Software Design Document is guidelines for develop software correctly and match with requirement.

## Intended Audience and Reading Suggestions

## Project Scope

# 2. Chapter Two | Overall Description

## 2.1 Product Perspective

## 2.2 Product Features

## 2.3 User Classes and Characteristics

## 2.4 Operation Environment

## 2.5 Design and Implementation Constraints

# 3. Chapter Three | System Architecture

# 4. Chapter Four | Detail Design

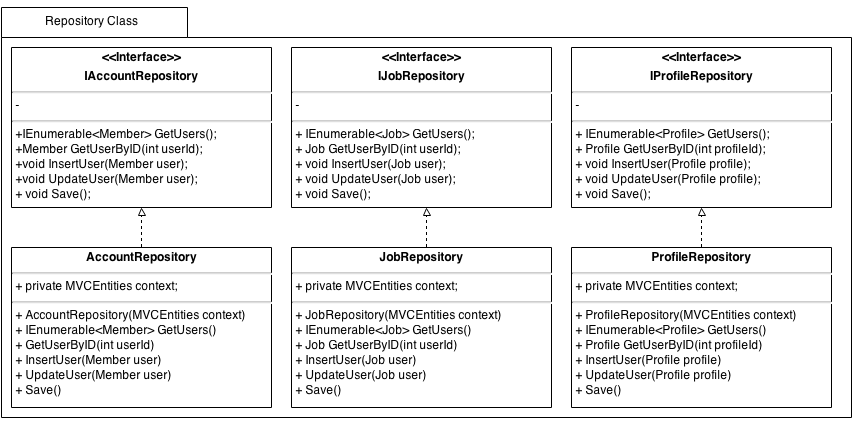
## 4.1 Class Diagram

### 4.1.1 Controller Package Class

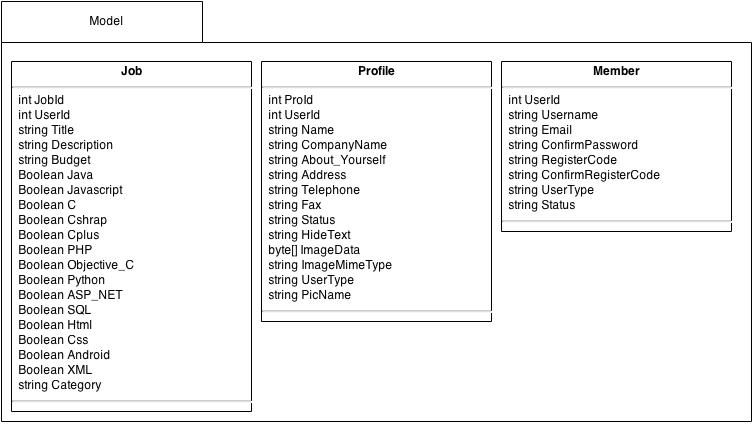
This class diagram will be show Method and attribute in each class only progress 1



### 4.1.2 Repository Package Class



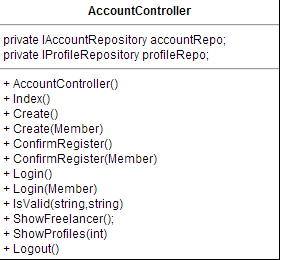
### 4.1.3 Model Class

****

## 4.2 Class Diagram Description

This class diagram Description will be show Method and attribute in each class only progress1

### 4.2.1 Controller Class Description

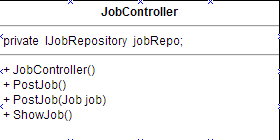


**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | accountRepo | Variable that build to call IAccountRepository Object in this class | Type: IAccountRepository object |
| 2 | profileRepo | Variable that build to call IProfileRepository Object in this class | Type: IProfileRepository object |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | Index() | The system call this method to show text list of User | returnType:View(user)  Parameter:- |
| 2 | AccountController() | The System use this method to call object from Repository | returnType:  Parameter: |
| 3 | Create(Member member) | The System use this method to insert member into database,identify type of user and send link to Email | returnType: RedirectToAction("Index", "Home");  Parameter: member |
| 4 | ConfirmRegister(Member member) | The System use this method to ConfirmRegister again  and insert other user information to register. | returnType: RedirectToAction("Index", "Home");  Parameter:member |
| 5 | Login() | The system use this method identify user to using web application | returnType: RedirectToAction("Index", "Home");  Parameter:member |
| 6 | IsValid(String username,String password) | The system use this method check username that log-in  found in database list or not. | returnType:bool  Parameter: String username, String password |
| 7 | ShowFreelancer() | The system call this method to show text list of Freelancer in system | returnType: View(users);  Parameter: |
| 8 | ShowProfiles(int) | The system use this method o show profile of user | returnType: RedirectToAction("ShowProfiles", "Profile", new { pid = id });  Parameter:- |
| 9 | Logout() | The system use this method to logout | returnType: RedirectToAction("index", "Home");  Parameter:- |

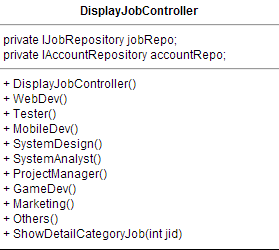


**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | jobRepo | Variable that build to call IJobRepository Object in this class | Type: IJobRepository object |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | JobController() | The System use this method to call object from Repository | returnType: -  Parameter: - |
| 2 | PostJob(Job job) | The system use method to insert job information  To database | returnType: View();  Parameter: Job Object |
| 4 | ShowJob() | The system use this method to Show list of job in  database | returnType: View(jobs);  Parameter:- |

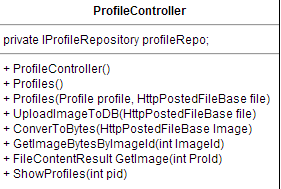


**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | jobRepo | Variable that build to call IJobRepository Object in this class | Type: IJobRepository object |
| 2 | accountRepo | Variable that build to call IAccountRepository Object in this class | Type: IAccountRepository object |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | DisplayJobController() | The System use this method to call object from Repository | returnType: View("CategoryJob", jobs);  Parameter: |
| 2 | WebDev() | The system use this method to display job of WebDeveloper Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 3 | Tester() | The system use this method to display job of  Tester Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 4 | MobileDev() | The system use this method to display job of MobileDeveloper Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 5 | SystemDesign() | The system use this method to display job of SystemDesign Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 6 | SystemAnalyst() | The system use this method to display job of SystemAnalyst Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 7 | ProjectManager() | The system use this method to display job of  Project manager Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 8 | GameDev() | The system use this method to display job of GameDeveloper Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 9 | Marketing() | The system use this method to display job of Software MarketingPage | returnType: View("CategoryJob", jobs);  Parameter: |
| 10 | Other() | The system use this method to display job of other software role job Page | returnType: View("CategoryJob", jobs);  Parameter: |
| 11 | ShowDetailCategoryJob(int jid) | The system use this method to display job of every software role job Page | returnType: PartialView("ShowDetailCategoryJob",JobModel);  Parameter:int jid |



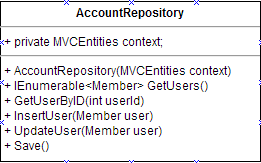
**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | profileRepo | Variable that build to call IJobRepository Object in this class | Type: IJobRepository object |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | ProfileController() | The System use this method to call object from Repository | returnType:  Parameter: - |
| 2 | Profiles(Profile profile, HttpPostedFileBase file) | The System use this method to retrieve profile of user | returnType: RedirectToAction("Index", "Home");  Parameter:profile object, HttpPostedFileBase object |
| 3 | UploadImageToDB(HttpPostedFileBase file) | The System use this method to upload image to database | returnType:  Parameter: HttpPostedFileBase object |
| 4 | ConverToBytes(HttpPostedFileBase file) | The System use this method convert type of picture to byte | returnType: imageBytes  Parameter: HttpPostedFileBase Image |
| 5 | GetImageBytesByImageId(int ImageId) | The System use this method to find image in database list | returnType:  Parameter: int |
| 6 | FileContentResultGetImage(int Proid) | The System use this method to get image to profile id | returnType: new FileContentResult(byteArray, "image/jpeg");  Parameter: int |
| 7 | ShowProfiles(int pid) | The System use this method to show profile | returnType: View(usersProfile)  Parameter: int |

### 4.2.2 Repository Class Description

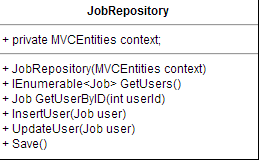


**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | context | Variable that build to call MVCEntities Object in this class | Type: MVCEntities object |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | AccountRepository(MVCEntities context) | The System use this method to call object from Database context | returnType:  Parameter: |
| 2 | GetUsers() | The System use this method to get data from database | returnType: context.Members.ToList();  Parameter: |
| 3 | InsertUser(Member user) | The System use this method to find data from database | returnType: context.Members.Add(user);  Parameter: Member user |
| 4 | UpdateUser(Member user) | The System use this method to update data to database | returnType: context.Entry(user).State = EntityState.Modified;  Parameter: Member user |
| 5 | Save() | The System use this method to save data to database | returnType:  context.SaveChanges();  Parameter: |
| 6 | GetUserByID(int userId) | The System use this method to find userId in database | returnType:  context.Members.Find(userId);  Parameter: int userId |

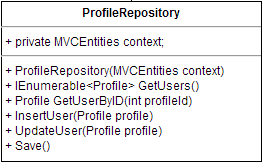
****

**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | context | Variable that build to call MVCEntities Object in this class | Type: MVCEntities object |

**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | JobRepository(MVCEntities context) | The System use this method to call object from Database context | returnType:  Parameter: |
| 2 | GetUsers() | The System use this method to get data from database | returnType: context. Jobs.ToList();  Parameter: |
| 3 | InsertUser(Job user) | The System use this method to find data from database | returnType: context. Jobs.Add(user);  Parameter: Job user |
| 4 | UpdateUser(Job user) | The System use this method to update data to database | returnType: context.Entry(user).State = EntityState.Modified;  Parameter: Job user |
| 5 | Save() | The System use this method to save data to database | returnType:  context.SaveChanges();  Parameter: |
| 6 | GetUserByID(int userId) | The System use this method to find userId in database | returnType:  context.Jobs.Find(userId);  Parameter: int userId |

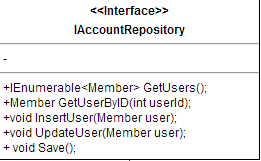


**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | context | Variable that build to call MVCEntities Object in this class | Type: MVCEntities object |

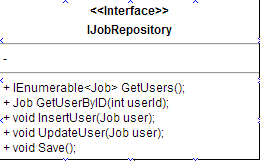
**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | ProfileRepository(MVCEntities context) | The System use this method to call object from Database context | returnType:  Parameter: |
| 2 | GetUsers() | The System use this method to get data from database | returnType: context. Profiles.ToList();  Parameter: |
| 3 | InsertUser(Profile profile) | The System use this method to find data from database | returnType: context. Profiles.Add(profileId);  Parameter: Job user |
| 4 | UpdateUser(Profile profile) | The System use this method to update data to database | returnType: context.Entry(profile).State = EntityState.Modified;  Parameter: Job user |
| 5 | Save() | The System use this method to save data to database | returnType:  context.SaveChanges();  Parameter: |
| 6 | GetUserByID(int profileId) | The System use this method to find profileId in database | returnType:  context. Profiles.Find(profileId);  Parameter: int profileId |



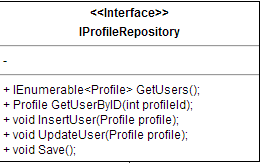
**Methods**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | GetUsers() | The System use this method from Repository | returnType:  Parameter: |
| 2 | InsertUser(Member user) | The System use this method from Repository | returnType:  Parameter: Member user |
| 3 | UpdateUser(Member user) | The System use this method from Repository | returnType:  Parameter: Member user |
| 4 | Save() | The System use this method from Repository | returnType:  Parameter: |
| 5 | GetUserByID(int userId) | The System use this method from Repository | returnType:  Parameter: int userId |



**Methods**

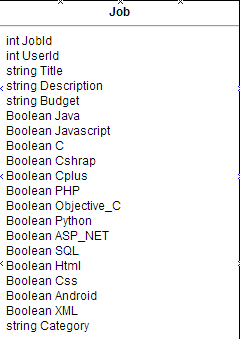
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | GetUsers() | The System use this method from Repository | returnType:  Parameter: |
| 2 | InsertUser(Job user) | The System use this method from Repository | returnType:  Parameter: Job user |
| 3 | UpdateUser(Job user) | The System use this method from Repository | returnType:  Parameter: Job user |
| 4 | Save() | The System use this method from Repository | returnType:  Parameter: |
| 5 | GetUserByID(int userId) | The System use this method from Repository | returnType:  Parameter: int userId |



**Methods**

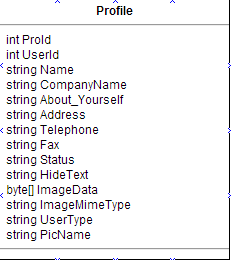
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | GetUsers() | The System use this method from Repository | returnType:  Parameter: |
| 2 | InsertUser(Profile profile) | The System use this method from Repository | returnType:  Parameter: Profile profile |
| 3 | UpdateUser(Profile profile) | The System use this method from Repository | returnType:  Parameter: Profile profile |
| 4 | Save() | The System use this method from Repository | returnType:  Parameter: |
| 5 | GetUserByID(int profileId) | The System use this method from Repository | returnType:  Parameter: int profileId |

### 4.2.3 Model Class Description



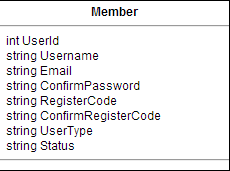
**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | JobId | Automate Value to set id of job | Type: int |
| 2 | UserId | Automate Value to set id of user | Type: int |
| 3 | Title | Value set for retrieve job Title | Type: string |
| 4 | Description | Value set for retrieve job Description | Type: string |
| 5 | Budget | Value set for retrieve budget of job | Type: Budget |
| 6 | Java | Value set for retrieve software skill job in type Java | Type: Boolean |
| 7 | Javascript | Value set for retrieve software skill job in type Javascript | Type: Boolean |
| 8 | C | Value set for retrieve software skill job in type C | Type: Boolean |
| 9 | Csharp | Value set for retrieve software skill job in type Csharp | Type: Boolean |
| 10 | Cplus | Value set for retrieve software skill job in type Cplus | Type: Boolean |
| 11 | PHP | Value set for retrieve software skill job in type PHP | Type: Boolean |
| 12 | Objective\_C | Value set for retrieve software skill job in type Objective\_C | Type: Boolean |
| 13 | Python | Value set for retrieve software skill job in type Python | Type: Boolean |
| 14 | ASP\_NET | Value set for retrieve software skill job in type ASP\_NET | Type: Boolean |
| 15 | SQL | Value set for retrieve software skill job in type SQL | Type: Boolean |
| 16 | Html | Value set for retrieve software skill job in type Html | Type: Boolean |
| 17 | CSS | Value set for retrieve software skill job in type CSS | Type: Boolean |
| 18 | Android | Value set for retrieve software skill job in type Android | Type: Boolean |
| 19 | XML | Value set for retrieve software skill job in type XML | Type: Boolean |
| 20 | Category | Value set for retrieve software role type | Type: string |



**Property**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | ProId | Automate Value to set id of Profile | Type: int |
| 2 | UserId | Automate Value to set id of UserId | Type: int |
| 3 | Name | Value set for retrieve Name of User | Type: string |
| 4 | CompanyName | Value set for retrieve CompanyName of User | Type: string |
| 5 | About\_Yourself | Value set for retrieve About\_Yourself of User | Type: string |
| 6 | Address | Value set for retrieve Address of User | Type: string |
| 7 | Telephone | Value set for retrieve Telephone of User | Type: string |
| 8 | Fax | Value set for retrieve Fax of User | Type: string |
| 9 | Status | Value set for retrieve Status of User | Type: string |
| 10 | HideText | Value set for HideText from User | Type: string |
| 11 | ImageData | Value set for retrieve ImageData of User | Type: byte[] |
| 12 | ImageMimeType | Value set for retrieve ImageMimeType of User | Type: string |
| 13 | UserType | Value set for retrieve UserType of User | Type: string |
| 14 | PicName | Value set for retrieve PicName of User | Type: string |



**Property**

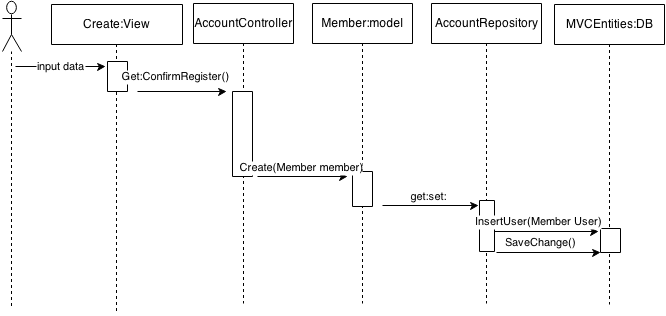
|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Name** | **Description** | **Remark** |
| 1 | UserId | Automate Value to set id of User | Type: int |
| 2 | Username | Value to retrive and set Username from User | Type: string |
| 3 | Email | Value to retrive and set Email from User | Type: string |
| 4 | Password | Value to retrive and set Password from User | Type: string |
| 5 | ConfirmPassword | Value to retrive ConfirmPassword from User | Type: string |
| 6 | RegisterCode | Value set for RegisterCode to User | Type: string |
| 7 | ConfirmRegisterCode | Value retrieve ConfirmRegisterCode from User | Type: string |
| 8 | UserType | Value set Type of User | Type: string |
| 9 | Status | Value set status of User | Type: string |

## 4.3 Sequence Diagram

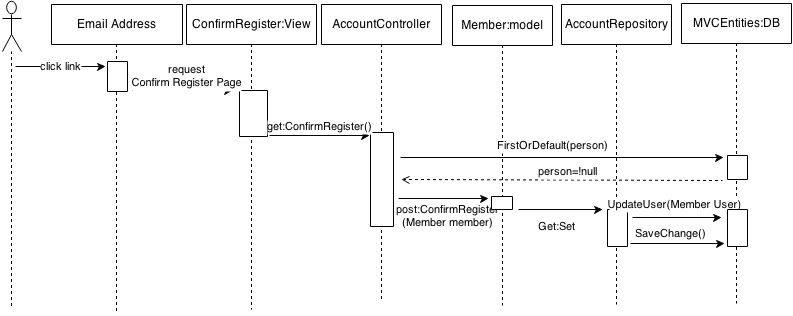
Sequence diagram separated into two user in 2 user that is **service request customer**, **freelancer supply**

***service request customer***

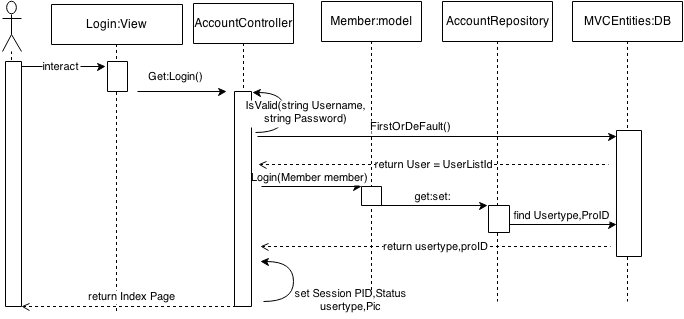
E-SQD-01: Register for **service request customer**



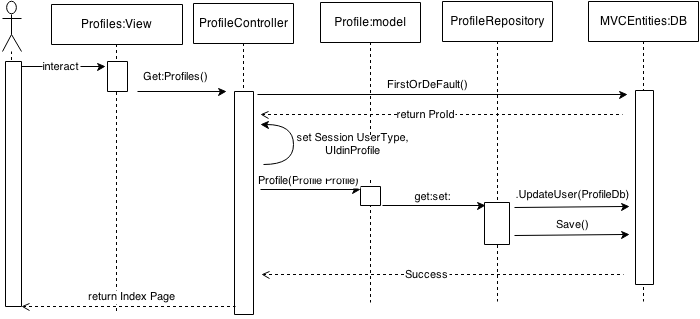
E-SQD-02: Confirm Register for **service request customer**

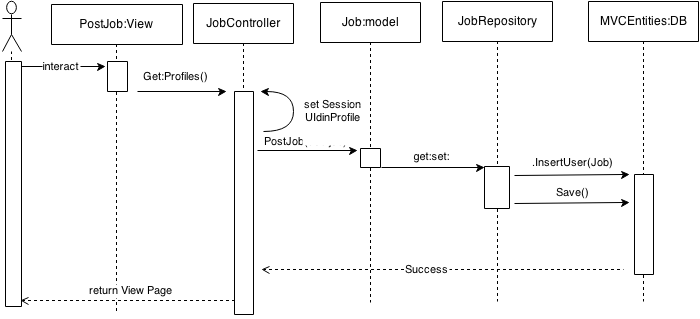


E-SQD-03: Login for **service request customer**

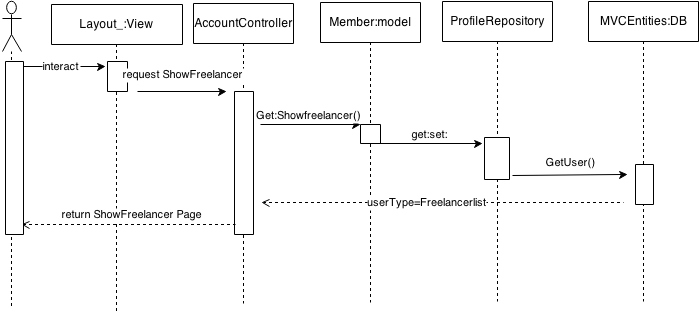


E-SQD-04: Add&Edit Profile for **service request customer**

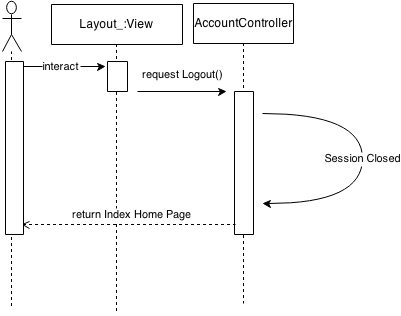


E-SQD-05: Post Project for **service request customer**

E-SQD-06: See freelancer for **service request customer**

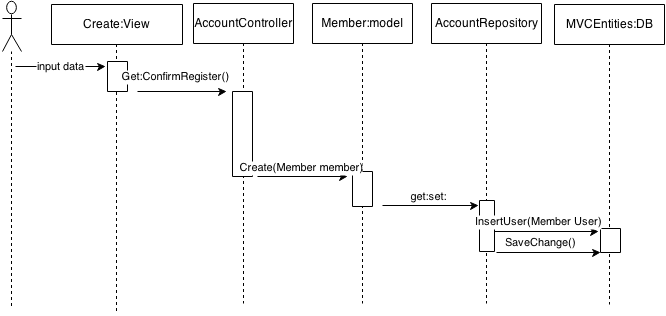


E-SQD-07: Logout for **service request customer**

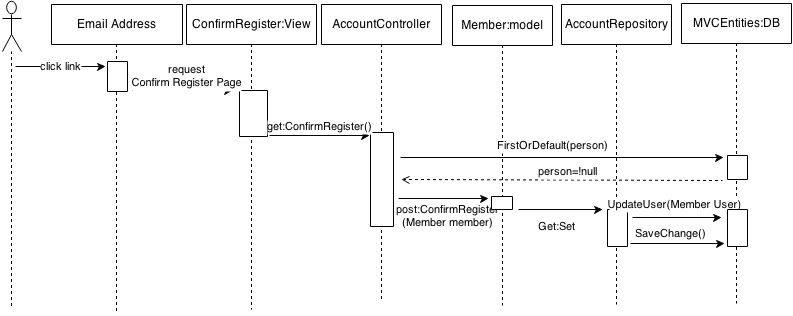


***freelancer supply***

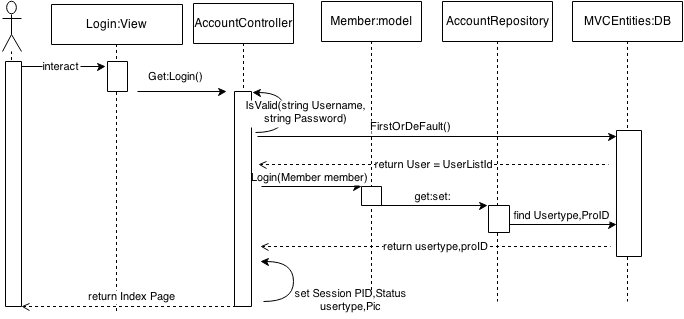
E-SQD-01: Register for **freelance supply**



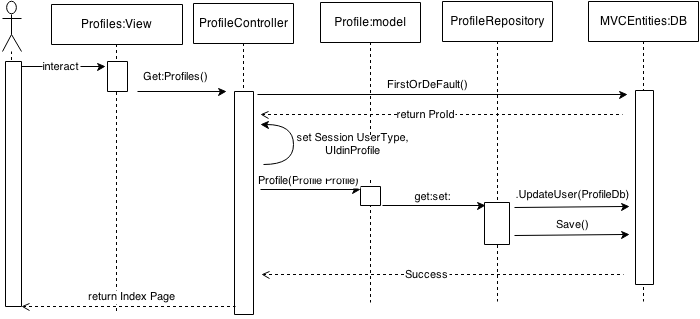
E-SQD-02: Confirm Register for **freelance supply**



E-SQD-03: Login for **freelance supply**



E-SQD-04: Add&Edit Profile for **freelance supply**



# 5. Chapter Five | Data Architecture

## 4.1 ER Diagram

## 4.2 DB Design

# 6. Chapter Six | User Interface Design

# 7. Chapter Seven | Traceability Record