

**Happy Programming**

**Software Design Document**

– HCMC,June 2022 –

**Table of Contents**

[I. Overview 7](#_Toc106650336)

[1. Code Packages/Namespaces 7](#_Toc106650337)

[2. Database Schema 9](#_Toc106650338)

[II. Code Designs 11](#_Toc106650339)

[1. Change Password 11](#_Toc106650340)

[a. Class Diagram 11](#_Toc106650341)

[b. Class Specifications 11](#_Toc106650342)

[c. Sequence Diagram(s) 12](#_Toc106650343)

[d. Database queries 12](#_Toc106650344)

[2. Reset Password 13](#_Toc106650345)

[a. Class Diagram 13](#_Toc106650346)

[b. Class Specifications 13](#_Toc106650347)

[c. Sequence Diagram(s) 14](#_Toc106650348)

[d. Database queries 14](#_Toc106650349)

[3. Update Mentee profile 15](#_Toc106650350)

[a. Class Diagram 15](#_Toc106650351)

[b. Class Specifications 15](#_Toc106650352)

[c. Sequence Diagram(s) 16](#_Toc106650353)

[d. Database queries 16](#_Toc106650354)

[4. Create request 17](#_Toc106650355)

[a. Class Diagram 17](#_Toc106650356)

[b. Class Specifications 17](#_Toc106650357)

[c. Sequence Diagram(s) 18](#_Toc106650358)

[d. Database queries 18](#_Toc106650359)

[5. Update request 19](#_Toc106650360)

[a. Class Diagram 19](#_Toc106650361)

[b. Class Specifications 19](#_Toc106650362)

[c. Sequence Diagram(s) 20](#_Toc106650363)

[d. Database queries 20](#_Toc106650364)

[6. Rate & comment mentor 21](#_Toc106650365)

[a. Class Diagram 21](#_Toc106650366)

[b. Class Specifications 21](#_Toc106650367)

[c. Sequence Diagram(s) 23](#_Toc106650368)

[d. Database queries 24](#_Toc106650369)

[7. User Authorization 25](#_Toc106650370)

[a. Class Diagram 25](#_Toc106650371)

[b. Class Specifications 25](#_Toc106650372)

[c. Sequence Diagram(s) 26](#_Toc106650373)

[d. Database queries 26](#_Toc106650374)

[8. Sign Up 27](#_Toc106650375)

[a. Class Diagram 27](#_Toc106650376)

[b. Class Specifications 27](#_Toc106650377)

[c. Sequence Diagram(s) 28](#_Toc106650378)

[d. Database queries 29](#_Toc106650379)

[9. Sign In 30](#_Toc106650380)

[a. Class Diagram 30](#_Toc106650381)

[b. Class Specifications 30](#_Toc106650382)

[c. Sequence Diagram(s) 31](#_Toc106650383)

[d. Database queries 32](#_Toc106650384)

[10. View profile-cv of mentor 32](#_Toc106650385)

[a. Class Diagram 32](#_Toc106650386)

[b. Class Specifications 32](#_Toc106650387)

[c. Sequence Diagram(s) 33](#_Toc106650388)

[d. Database queries 34](#_Toc106650389)

[11. View all skills 35](#_Toc106650390)

[a. Class Diagram 35](#_Toc106650391)

[b. Class Specifications 35](#_Toc106650392)

[c. Sequence Diagram(s) 36](#_Toc106650393)

[d. Database queries 36](#_Toc106650394)

[12. Create CV of mentor 37](#_Toc106650395)

[a. Class Diagram 37](#_Toc106650396)

[b. Class Specifications 37](#_Toc106650397)

[c. Sequence Diagram(s) 38](#_Toc106650398)

[d. Database queries 39](#_Toc106650399)

[13. Update CV of mentor 40](#_Toc106650400)

[a. Class Diagram 40](#_Toc106650401)

[b. Class Specifications 40](#_Toc106650402)

[c. Sequence Diagram(s) 42](#_Toc106650403)

[d. Database queries 42](#_Toc106650404)

[14. View list following Request 43](#_Toc106650405)

[a. Class Diagram 43](#_Toc106650406)

[b. Class Specifications 43](#_Toc106650407)

[c. Sequence Diagram(s) 44](#_Toc106650408)

[d. Database queries 45](#_Toc106650409)

[15. View statistic request 46](#_Toc106650410)

[a. Class Diagram 46](#_Toc106650411)

[b. Class Specifications 46](#_Toc106650412)

[c. Sequence Diagram(s) 48](#_Toc106650413)

[d. Database queries 48](#_Toc106650414)

[16. View list all mentor 49](#_Toc106650415)

[a. Class Diagram 49](#_Toc106650416)

[b. Class Specifications 49](#_Toc106650417)

[c. Sequence Diagram(s) 50](#_Toc106650418)

[d. Database queries 51](#_Toc106650419)

[17. Statistic of all mentee 52](#_Toc106650420)

[b. Class Specifications 52](#_Toc106650421)

[c. Sequence Diagram(s) 53](#_Toc106650422)

[d. Database queries 54](#_Toc106650423)

[18. Create skill 55](#_Toc106650424)

[a. Class Diagram 55](#_Toc106650425)

[b. Class Specifications 55](#_Toc106650426)

[c. Sequence Diagram(s) 56](#_Toc106650427)

[d. Database queries 56](#_Toc106650428)

[19. Update skill 57](#_Toc106650429)

[a. Class Diagram 57](#_Toc106650430)

[b. Class Specifications 57](#_Toc106650431)

[c. Sequence Diagram(s) 58](#_Toc106650432)

[d. Database queries 58](#_Toc106650433)

[20. View list of skills 59](#_Toc106650434)

[a. Class Diagram 59](#_Toc106650435)

[b. Class Specifications 59](#_Toc106650436)

[c. Sequence Diagram(s) 60](#_Toc106650437)

[d. Database queries 60](#_Toc106650438)

[21. View list all request 61](#_Toc106650439)

[a. Class Diagram 61](#_Toc106650440)

[b. Class Specifications 61](#_Toc106650441)

[c. Sequence Diagram(s) 63](#_Toc106650442)

[d. Database queries 63](#_Toc106650443)

[22. List request by me 64](#_Toc106650444)

[a. Class Diagram 64](#_Toc106650445)

[b. Class Specifications 64](#_Toc106650446)

[c. Sequence Diagram(s) 66](#_Toc106650447)

[d. Database queries 66](#_Toc106650448)

[23. Statistic request by me 67](#_Toc106650449)

[a. Class Diagram 67](#_Toc106650450)

[b. Class Specifications 67](#_Toc106650451)

[c. Sequence Diagram(s) 69](#_Toc106650452)

[d. Database queries 70](#_Toc106650453)

[24. List Mentor suggestion (for Mentee) 70](#_Toc106650454)

[a. Class Diagram 70](#_Toc106650455)

[b. Class Specifications 70](#_Toc106650456)

[c. Sequence Diagram(s) 72](#_Toc106650457)

[d. Database queries 73](#_Toc106650458)

[III. Database Tables 73](#_Toc106650459)

[1. Account 73](#_Toc106650460)

[2. Invite 74](#_Toc106650461)

[3. Mentee 74](#_Toc106650462)

[4. Mentor 74](#_Toc106650463)

[5. MentorSkill 74](#_Toc106650464)

[6. Rating 75](#_Toc106650465)

[7. Request 75](#_Toc106650466)

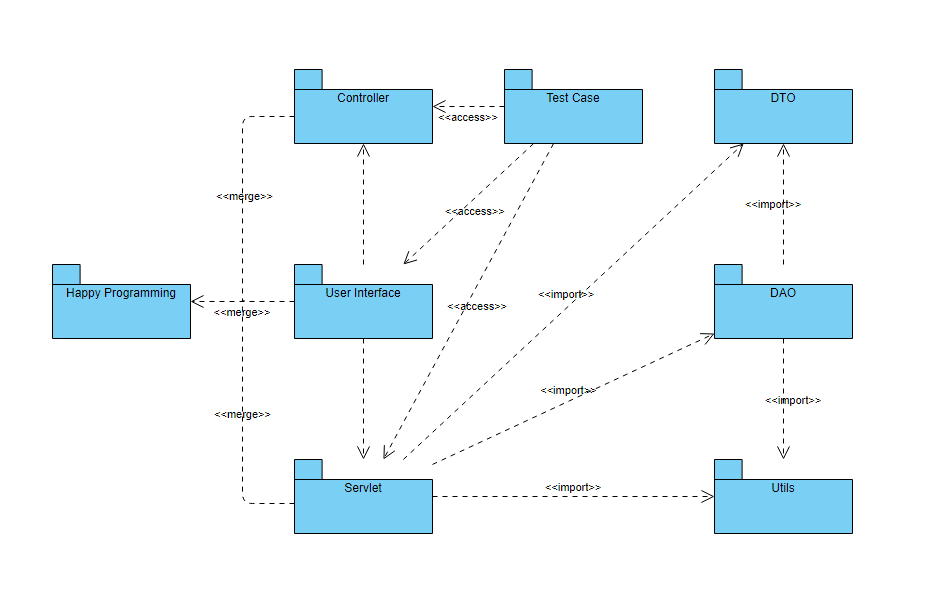
[8. RequestSkill 75](#_Toc106650467)

[9. Roles 75](#_Toc106650468)

[10. Skill 76](#_Toc106650469)

# I. Overview

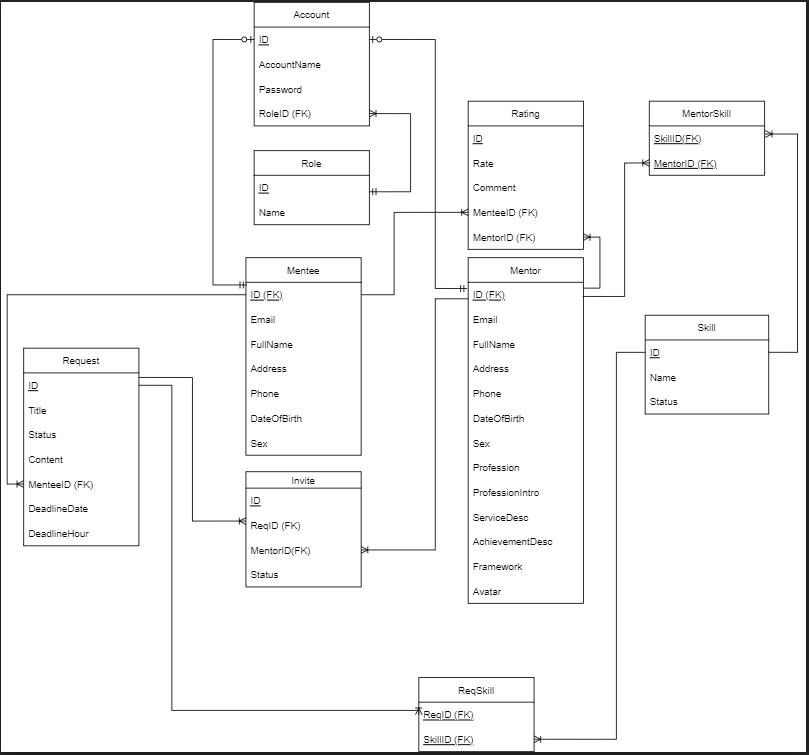
## 1. Code Packages/Namespaces



***Package descriptions & package class naming conventions***

|  |  |  |
| --- | --- | --- |
| **No** | **Package** | **Description** |
| *01* | *User Interface* | *This package contains everything that will be used in creating UI including pages, images, css, bootstrap and web configuration*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole word to avoid acronyms and abbreviations* |
| *02* | *Servlet* | *In this package are the servlets that are responsible for getting data and display that data to the pages in user interface*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |
| *03* | *Controller* | *This package contains controller that will redirect the data flow to the correct servlet*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |
| *04* | *Test Case* | *In this packet contain all of the test case use to test whether the method and connection fit the requirements*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |
| *05* | *DTO* | *This package contains DTOs (or Data Transfer Object) to reduce the number of methods calls*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |
| *06* | *DAO* | *In this package is DAOs (or Data Access Object), this package is used to improve the readability for the dev team*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |
| *07* | *Utils* | *Utils package contains frameworks, connections, … to simplify the development process*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |
| *08* | *Happy Programming* | *This packet contains every package mention above to create a complete web app*  *Class names should be nouns, in mixed case with the first letter of each internal word capitalized, keep class names simple and descriptive. Use whole words-avoid acronyms and abbreviations* |

## 2. Database Schema



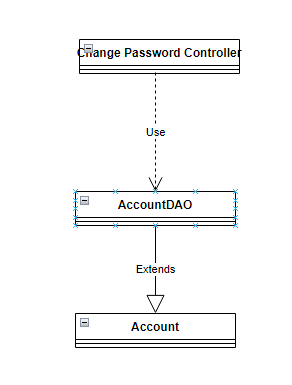
***Table descriptions & package class naming conventions are as below***

|  |  |  |
| --- | --- | --- |
| **No** | **Table** | **Description** |
| *01* | *Account* | *This table is uses to store information about user’s account including ID, AccountName, Password, Role ID*  *- Primary keys: ID*  *- Foreign keys: RoleID* |
| *02* | *Roles* | *This table contain user ID and their role*  *- Primary keys: ID* |
| *03* | *Mentee* | *All of mentees’ information will be store in this table including ID, Email, FullName, Address, Phone,DateOfBirth, Sex, Avatar*  *- Foreign keys: ID* |
| *04* | *Mentor* | *Information about mentor going to be store in this table including ID, Email, FullName, Address, Phone,DateOfBirth, Sex, Profession, ProfessionIntro, ServiceDesc, AchivementDesc, FrameWorrk, Avatar*  *- Foreign keys: ID* |
| *05* | *Request* | *This table will be the place to store request’s information including ID, Title, Status, Content, MenteeID, Deadline, DeadlineDate, DeadlineHour*  *- Primary keys: ID*  *- Foreign keys: MenteeID* |
| *06* | *Invite* | *Invitation will be store in this table including ID, ReqID, MentorID, Status*  *- Primary keys: ID*  *- Foreign keys: ReqID*  *MentorID* |
| *07* | *Rating* | *Mentor’s rating and comment will be store in this table including ID, Rate, Comment, MenteeID, MentorID*  *- Primary keys: ID*  *- Foreign keys: MenteeID*  *MentorID* |
| *08* | *Skill* | *All skill available in the system will be store in this table including ID, Name, Status*  *- Primary keys: ID* |
| *09* | *MentorSkill* | *Skills of a specific mentor will be in this table including SkillID, MentorID*  *- Primary keys: (SkillID, MentorID)*  *- Foreign keys: SkillID*  *MentorID* |
| *10* | *RequestSkill* | *Skills of a specific request will be in this table including SkillID, RequestID*  *- Primary keys: (SkillID, RequestID)*  *- Foreign keys: SkillID*  *RequestID* |

# II. Code Designs

## 1. Change Password

### a. Class Diagram



### b. Class Specifications

#### AccountDAO Class

***Class Methods***

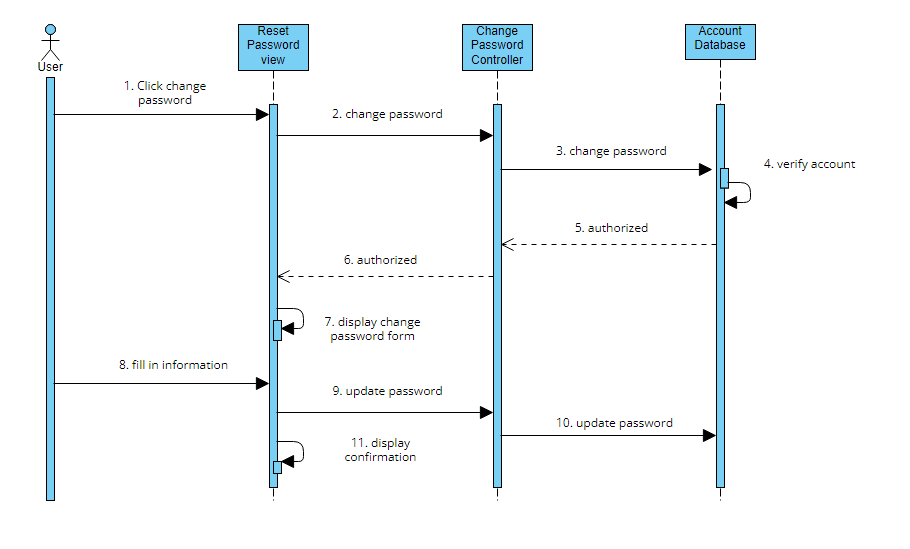
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Change password* | *This method is uses to change an account’s password, the input is the new password and the output will be Boolean to indicate the status of the method. The new password will be entered and check if it is a new password or not then the change will be made* |

#### Account Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new account, the input is an account and the output is void. The new account will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update an account, the input is an account’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete an account, the input is an account and the output is void. All information about this account will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific account, the input is an account id and the output is an account. All information about this account will be display using Sql query* |

### c. Sequence Diagram(s)



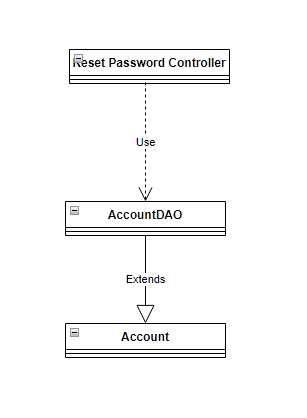
### d. Database queries

*Select \* from Account where ID = ? to make sure the account exist*

*Update Account set password = ? where ID = ? to change password*

## 2. Reset Password

### a. Class Diagram

**

### b. Class Specifications

#### AccountDAO Class

***Class Methods***

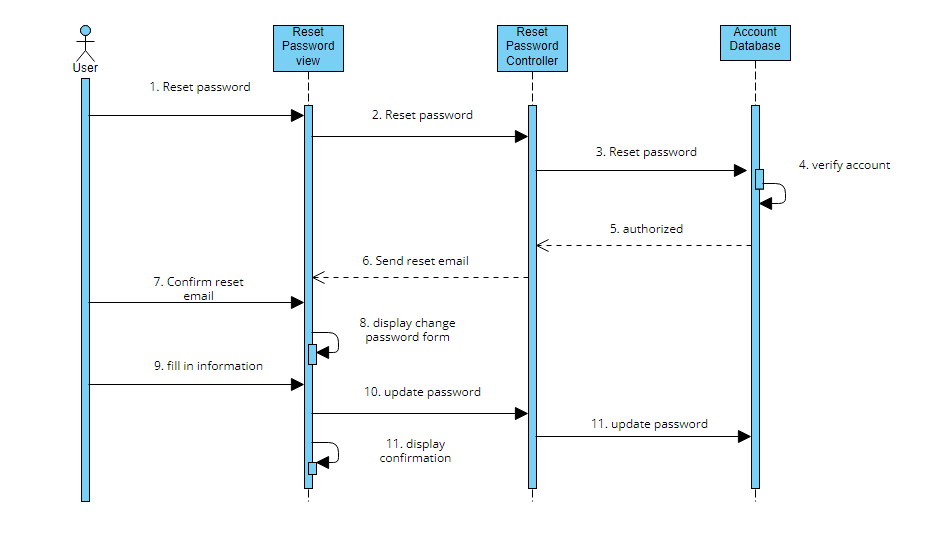
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Reset password* | *This method is uses to reset an account’s password when the user forgot, the input is the new password and the output will be Boolean to indicate the status of the method. A link will be sent to user’s email and it will lead to the reset page, the new password will be entered and check if it is a new password or not then the change will be made* |
| *02* | *Send reset email* | *This method will be use to seen an email to the user, the input is the user email and the output will be Boolean to indicate the status of the method. After confirm email the user will be able to reset their password* |

#### Account Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new account, the input is an account and the output is void. The new account will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update an account, the input is an account’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete an account, the input is an account and the output is void. All information about this account will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific account, the input is an account id and the output is an account. All information about this account will be display using Sql query* |

### c. Sequence Diagram(s)



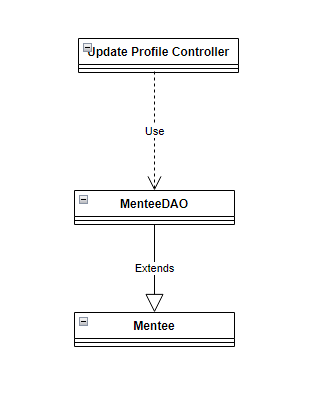
### d. Database queries

*Select \* from Account where ID = ? to make sure the account exist*

*Update Account set password = ? where ID = ? to change password*

## 3. Update Mentee profile

### a. Class Diagram

**

### b. Class Specifications

#### MenteeDAO Class

***Class Methods***

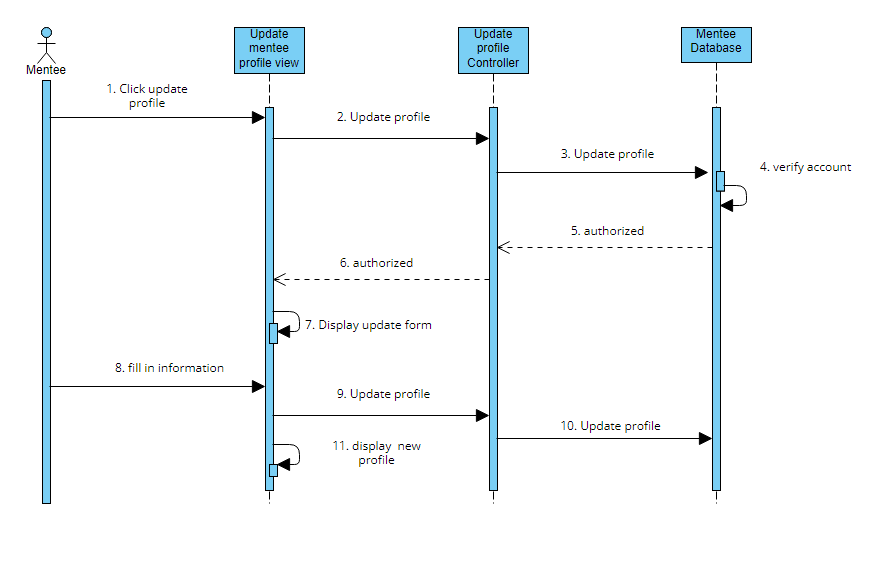
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Profile* | *This method is uses to Update a mentee’s profile, the input will be mentee’s Email, Fullname, Address, Phone, DateOfBirth, Sex , Avatar the out put will be Boolean to let the user know the result, the information will be enter and check then store in the database using Sql query* |

#### Mentee Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentee, the input is an mentee and the output is void. The new mentee will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentee, the input is a mentee’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentee, the input is a mentee and the output is void. All information about this mentee will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentee, the input is a mentee id and the output is a mentee. All information about this mentee will be display using Sql query* |

### c. Sequence Diagram(s)



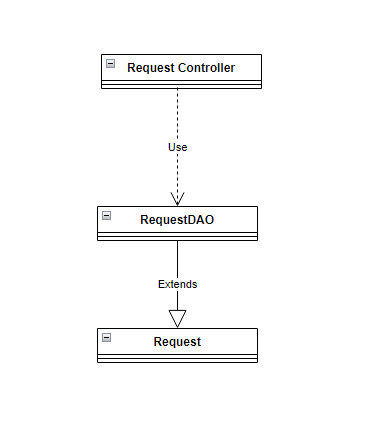
### d. Database queries

*Select \* from Mentee where ID = ? to make sure the account exist*

*Update Mentee set Email = ?, Fullname = ?, Address = ? , Phone = ? , DateOfBirth = ? , Sex = ? , Avatar = ? where ID = ? to update profile*

## 4. Create request

### a. Class Diagram

**

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

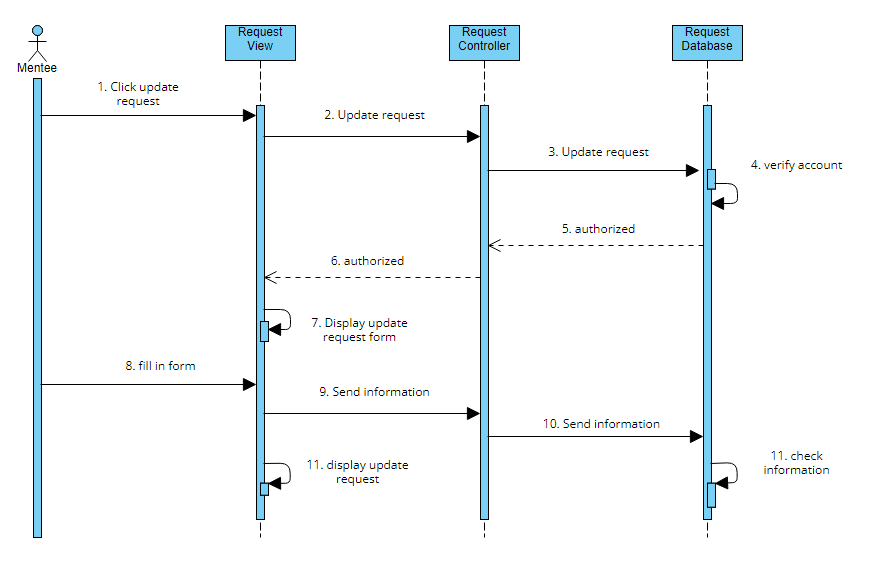
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create Request* | *This method allow mentee to create a new request, the input is the information about the request and the output will be Boolean to indicate the method successes or fail. The information will be entered by mentee after that it will be check and store in database* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

### c. Sequence Diagram(s)



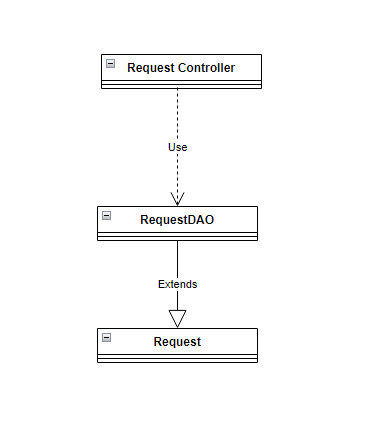
### d. Database queries

*INSERT into Request (Title, Status, Content, MenteeID, DeadlineDate, DeadlineHour)*

*Values (?, ?, ?, ?, ?, ?)*

## 5. Update request

### a. Class Diagram

**

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

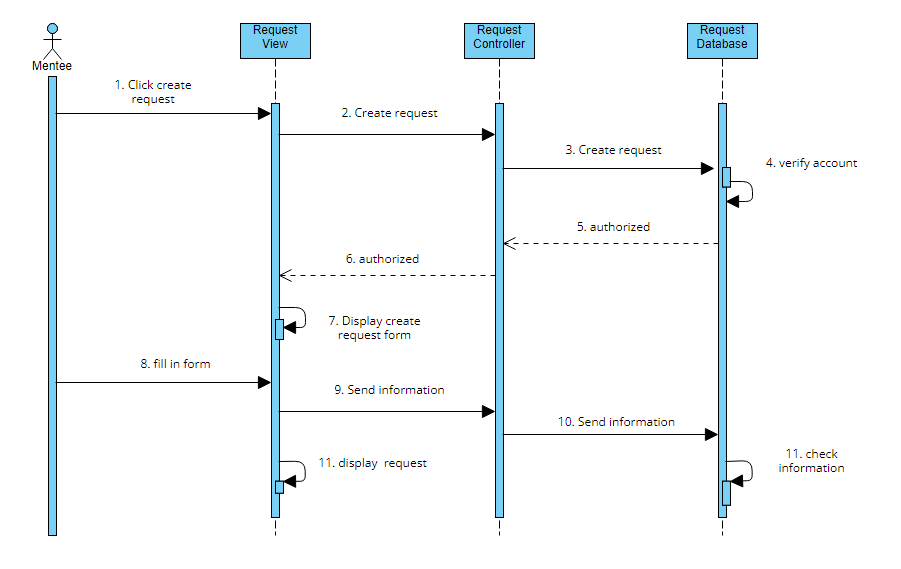
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Request* | *This method allow mentee to update an existing request, the input is the change about the request and the output will be Boolean to indicate the method successes or fail. The information will be entered by mentee when they need to change a request and after that it will be check and store in database* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

### c. Sequence Diagram(s)

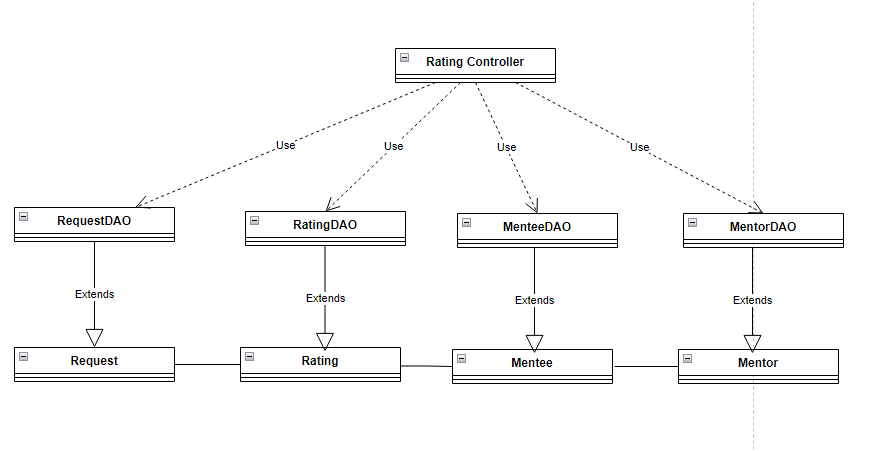


### d. Database queries

*Update Request set Title = ?, Status = ?, Content = ? , MenteeID = ? , DeadlineDate = ? , DeadlineHour = ? , where ID = ? to update request*

## 6. Rate & comment mentor

## a. Class Diagram

**

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Verify* | *This method will be call to check if the mentee have work with this mentor before or not by search in the request table and return the request with this mentor id and mentee id* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### MenteeDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Profile* | *This method is uses to Update a mentee’s profile, the input will be mentee’s Email, Fullname, Address, Phone, DateOfBirth, Sex , Avatar the out put will be Boolean to let the user know the result, the information will be enter and check then store in the database using Sql query* |

#### Mentee Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentee, the input is an mentee and the output is void. The new mentee will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentee, the input is a mentee’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentee, the input is a mentee and the output is void. All information about this mentee will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentee, the input is a mentee id and the output is a mentee. All information about this mentee will be display using Sql query* |

#### MentorDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Profile* | *This method is uses to Update a mentee’s profile, the input will be mentee’s Email, Fullname, Address, Phone, DateOfBirth, Sex , Avatar the out put will be Boolean to let the user know the result, the information will be enter and check then store in the database using Sql query* |

#### Mentor Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

#### RatingDAO Class

***Class Methods***

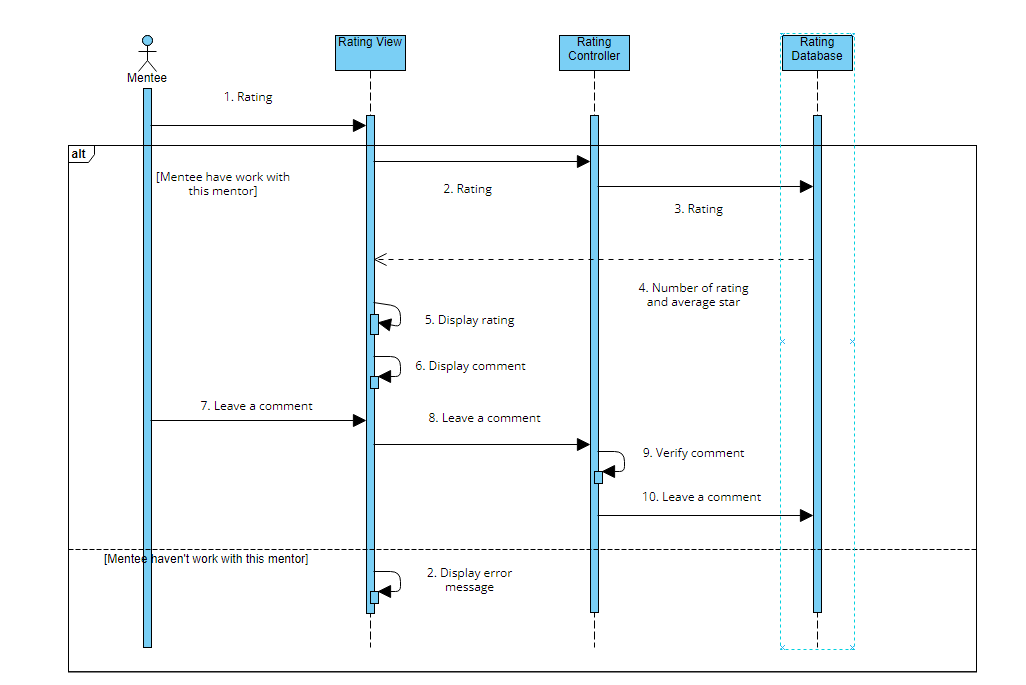
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Rating* | *This method allow mentee to give a mentor rate base on their skill. The input will be number of star and the output will be void. When the mentee rating the information will go to database and store there* |
| *02* | *Comment* | *This method allow mentee to leave a comment about a mentor. The input will be content of the comment and the output will be void. After mentee rate they will be able to leave comment if they want and this will also be store in the rating database* |

#### Rating Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

### c. Sequence Diagram(s)



### d. Database queries

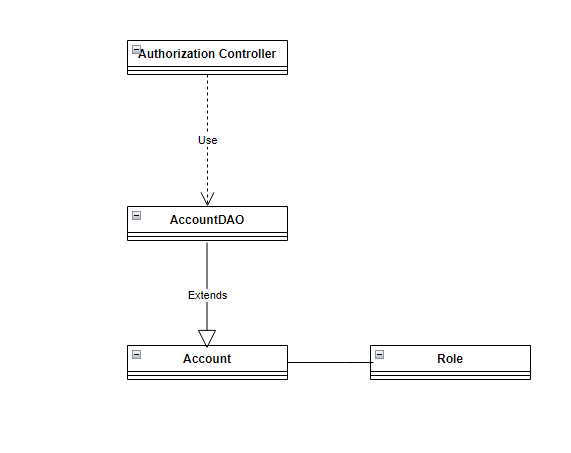
*INSERT into Rating (ID, Rate, Comment, MenteeID, MentorID)*

*Values (?, ?, ?, ?, ?) to create rating*

*Select \* from Request where MentorId = ? and mentorId = ? to verify if they have work before*

## 7. User Authorization

## a. Class Diagram

**

### b. Class Specifications

#### AccountDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Authorization* | *This method will be use to redirect user to their page and prevent enter the URL directly the input will be AccountRole and the output will be void. When user enter their account the account role will be take from the database and redirect them to the correct page* |

#### Account Class

***Class Methods***

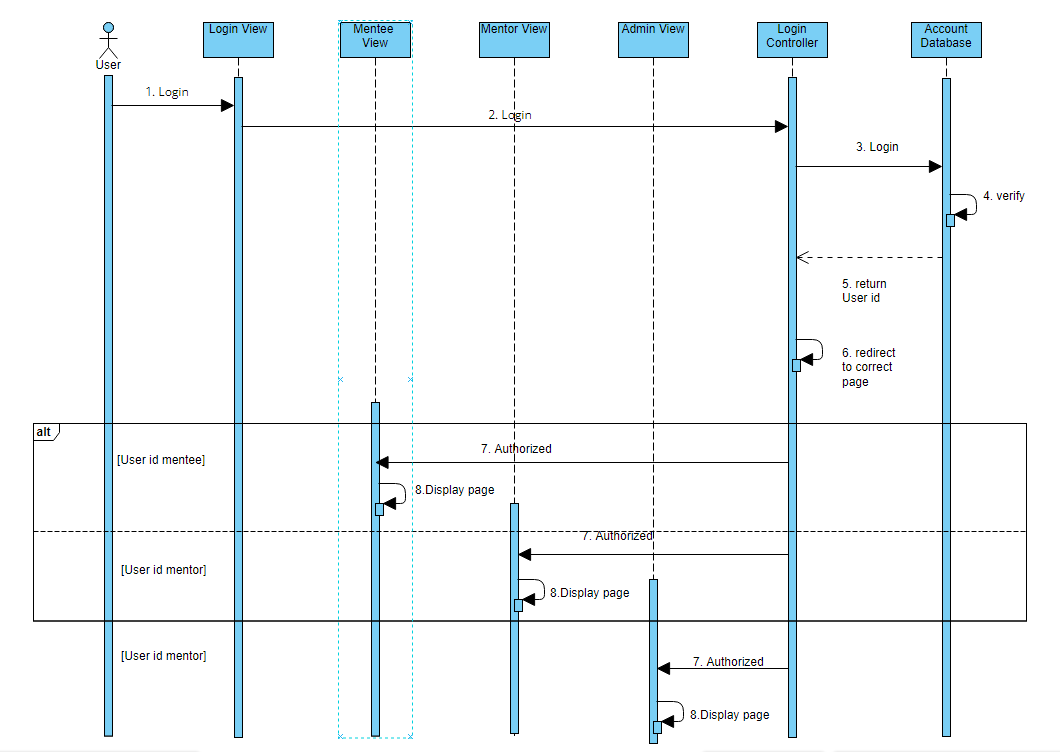
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new account, the input is an account and the output is void. The new account will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update an account, the input is an account’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete an account, the input is an account and the output is void. All information about this account will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific account, the input is an account id and the output is an account. All information about this account will be display using Sql query* |

#### Role Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Get* | *This method uses to find a specific Role, the input is a role id and the output is a role. All information about this role will be display using Sql query* |

### c. Sequence Diagram(s)

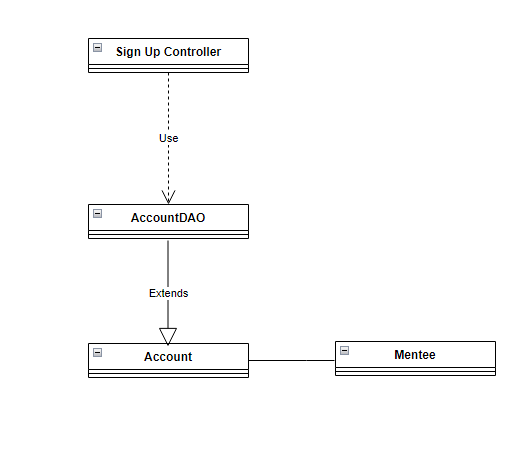


### d. Database queries

*Select \* from Account join Roles on Account.RoleID = Roles.ID*

## 8. Sign Up

## a. Class Diagram

**

### b. Class Specifications

#### AccountDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Verify filed* | *Using this method will let the user know if the will is missing or in incorrect format or not, if it isn’t a alert will be raise* |
| *02* | *Verify existing* | *This method will show the user if their account existed or not, if the account existed a error message will be alert and if it is not the account will sign up successfully* |

#### Account Class

***Class Methods***

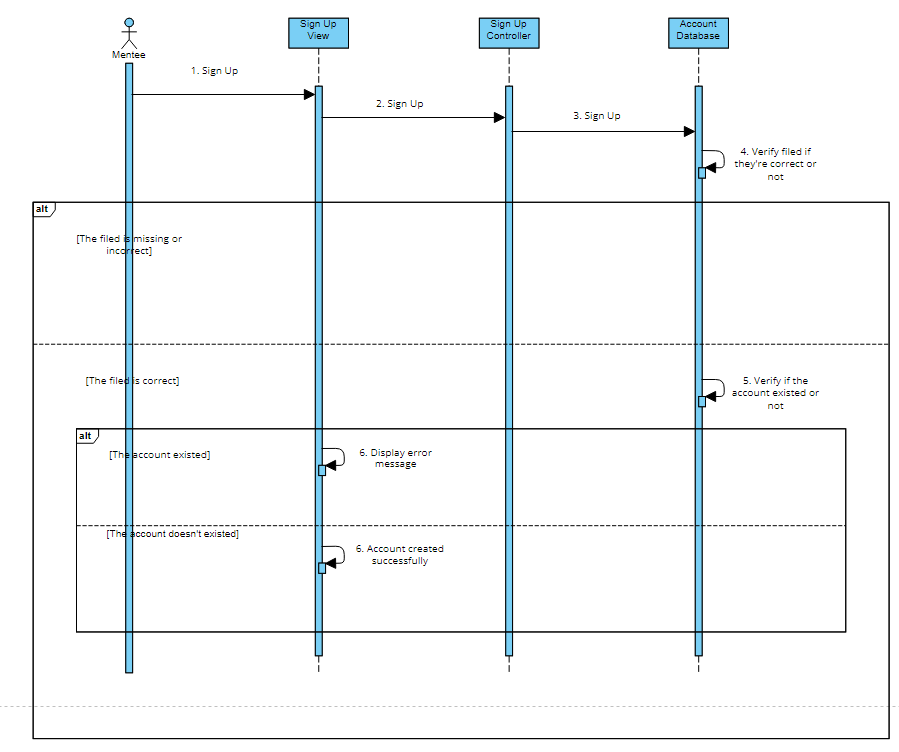
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new account, the input is an account and the output is void. The new account will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update an account, the input is an account’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete an account, the input is an account and the output is void. All information about this account will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific account, the input is an account id and the output is an account. All information about this account will be display using Sql query* |

#### Mentee Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentee, the input is an mentee and the output is void. The new mentee will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentee, the input is a mentee’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentee, the input is a mentee and the output is void. All information about this mentee will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentee, the input is a mentee id and the output is a mentee. All information about this mentee will be display using Sql query* |

### c. Sequence Diagram(s)



### d. Database queries

*Select \* from Account join Mentee on Account.ID = Mentee.ID where AccountName = ? to check if the account existed or not*

*Insert into Account(AccountName, Password, RoleID )*

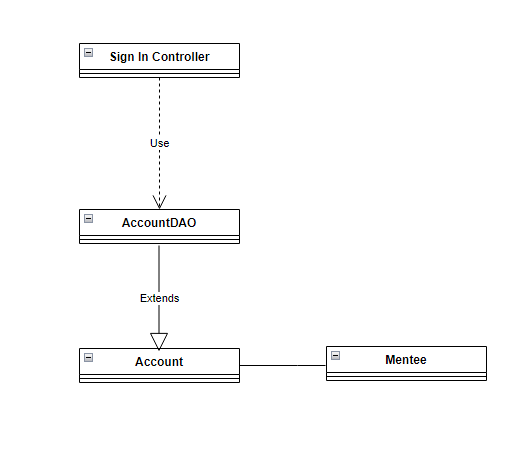
*Values(?, ?, ?)*

*Insert into Mentee(Email, FullName, Address, Phone, DateOfBirth, Sex, Avatar )*

*Values(?, ?, ?, ?, ?, ?, ?)*

## 9. Sign In

## a. Class Diagram

**

### b. Class Specifications

#### AccountDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Verify filed* | *Using this method will let the user know if the will is missing or in incorrect format or not, if it isn’t a alert will be raise* |
| *02* | *Verify existing* | *This method will show the user if their account existed or not, if the account existed a error message will be alert and if it is not the account will sign up successfully* |

#### Account Class

***Class Methods***

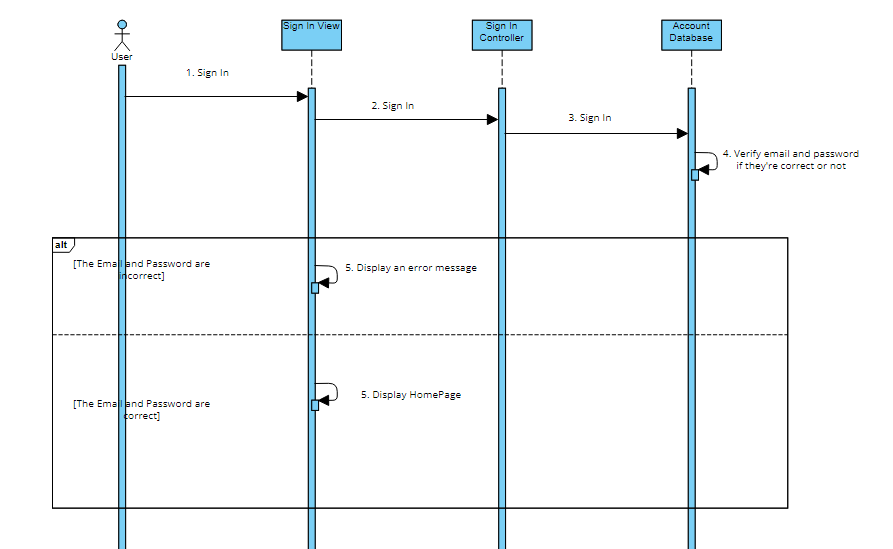
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new account, the input is an account and the output is void. The new account will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update an account, the input is an account’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete an account, the input is an account and the output is void. All information about this account will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific account, the input is an account id and the output is an account. All information about this account will be display using Sql query* |

#### Mentee Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentee, the input is an mentee and the output is void. The new mentee will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentee, the input is a mentee’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentee, the input is a mentee and the output is void. All information about this mentee will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentee, the input is a mentee id and the output is a mentee. All information about this mentee will be display using Sql query* |

### c. Sequence Diagram(s)

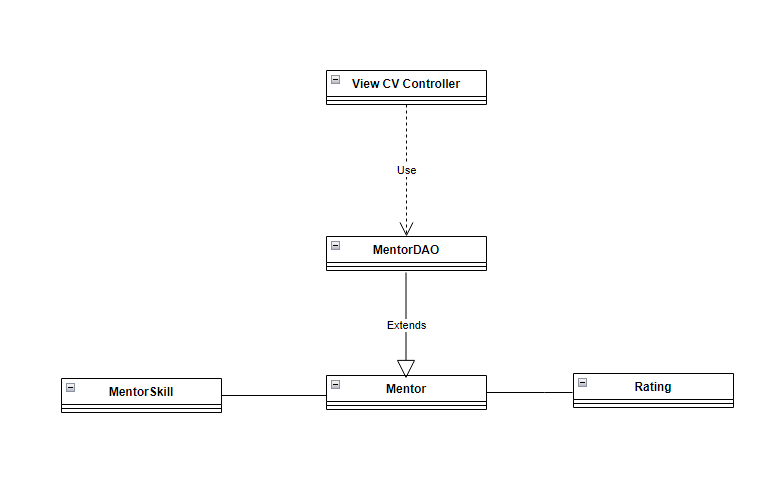


### d. Database queries

*Select \* from Account join Mentee on Account.ID = Mentee.ID where Email = ? and Password = ? to check if the email and password is correct*

## 10. View profile-cv of mentor

## a. Class Diagram

**

### b. Class Specifications

#### MentorDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *View Profile* | *This method will display summary information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get summary about this mentor using Sql query* |
| *02* | *View CV* | *This method will display information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get every things about this mentor using Sql query* |

#### Mentor Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

#### Rating Class

***Class Methods***

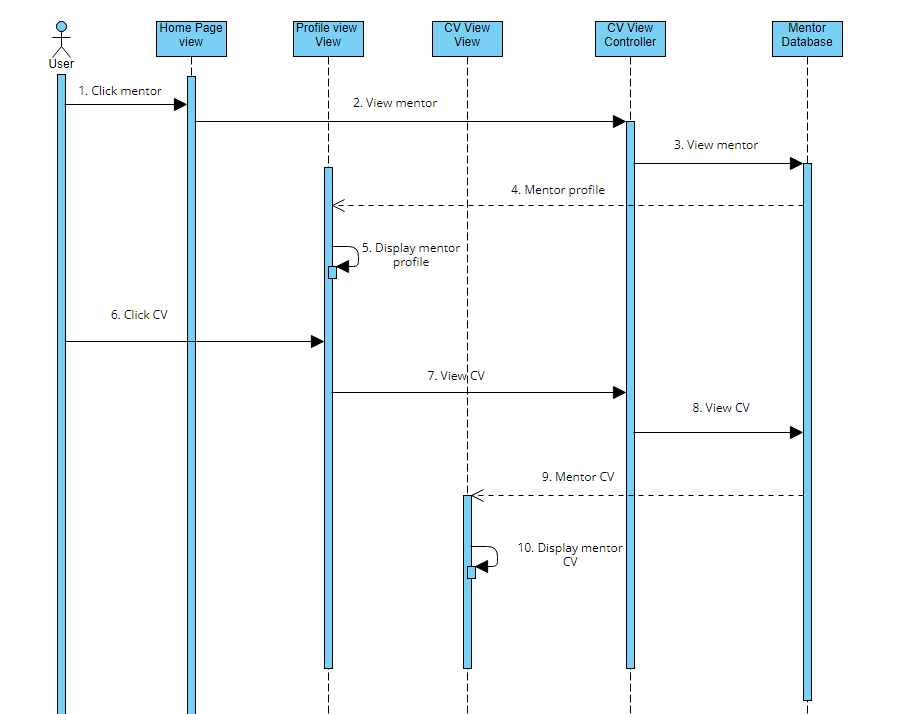
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

#### MentorSkill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a mentor’s skill, the input is a mentor’s skill and the output is void. When a new mentor create his/ her CV the skill will also be created and store in MentorSkill database* |
| *02* | *Update* | *This method uses to update a mentor’s skill, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor’s skill, the input is a mentor’s skill and the output is void. All information about this mentor’s skill will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor’s skill, the input is a mentor id and the output is a mentor’s skill. All information about this mentor’s skill will be display using Sql query* |

### c. Sequence Diagram(s)

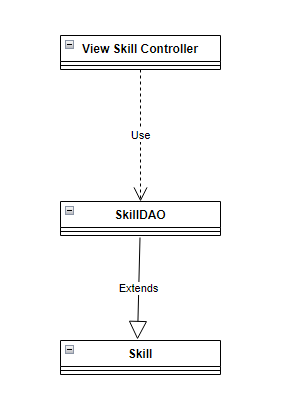


### d. Database queries

*Select \* from (Mentor join MentorSkill on Mentor.ID = MetorSkill.MentorID) join Rating on Mentor.ID = Rating.MentorID to display all information about a mentor*

## 11. View all skills

## a. Class Diagram

**

### b. Class Specifications

#### SkillDAO Class

***Class Methods***

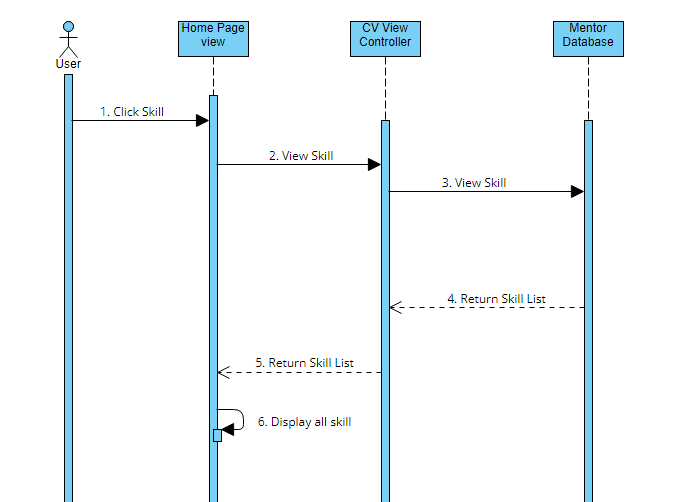
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Get List of Skills* | *This method will display the list of skills available on the website, the input will be void and output is a list of skill* |

#### Skill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

### c. Sequence Diagram(s)

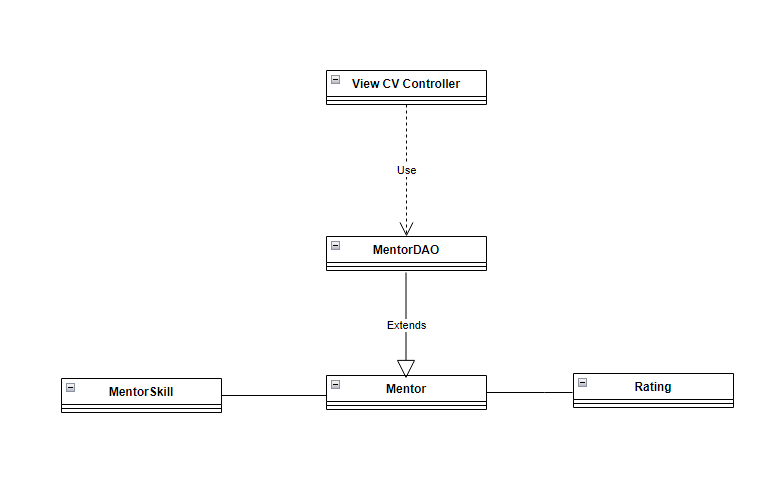


### d. Database queries

*Select \* from Skill*

## 12. Create CV of mentor

## a. Class Diagram

**

### b. Class Specifications

#### MentorDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *View Profile* | *This method will display summary information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get summary about this mentor using Sql query* |
| *02* | *View CV* | *This method will display information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get every things about this mentor using Sql query* |
| *03* | *Create CV* | *This method will be use to create a new CV for mentor, the input will be information about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |

#### Mentor Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

#### Rating Class

***Class Methods***

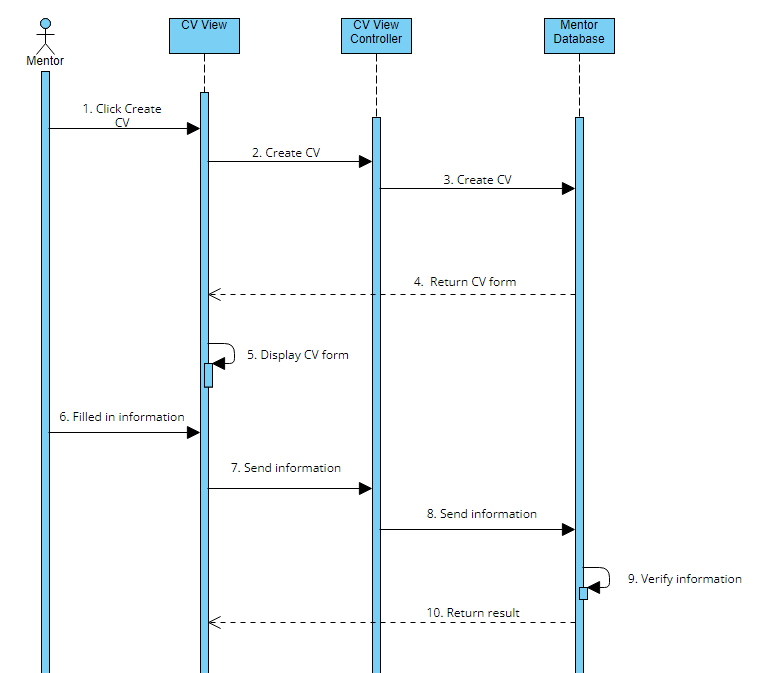
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

#### MentorSkill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a mentor’s skill, the input is a mentor’s skill and the output is void. When a new mentor create his/ her CV the skill will also be created and store in MentorSkill database* |
| *02* | *Update* | *This method uses to update a mentor’s skill, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor’s skill, the input is a mentor’s skill and the output is void. All information about this mentor’s skill will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor’s skill, the input is a mentor id and the output is a mentor’s skill. All information about this mentor’s skill will be display using Sql query* |

### c. Sequence Diagram(s)



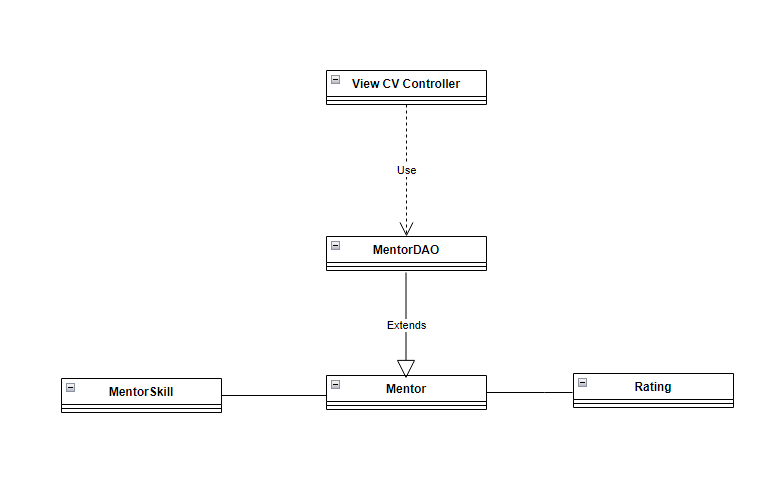
### d. Database queries

*Insert into Mentor(Email, FullName, Address, Phone, DateOfBirth, Sex, Profession, ProfessionIntro, ServiceDesc, FrameWork, Avatar)*

*Values(?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)*

## 13. Update CV of mentor

## a. Class Diagram

**

### b. Class Specifications

#### MentorDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *View Profile* | *This method will display summary information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get summary about this mentor using Sql query* |
| *02* | *View CV* | *This method will display information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call, it will go to database and get every things about this mentor using Sql query* |
| *03* | *Create CV* | *This method will be used to create a new CV for mentor, the input will be information about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |
| *04* | *Update CV* | *This method will be used to update a CV for mentor, the input will be changes about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |

#### Mentor Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

#### Rating Class

***Class Methods***

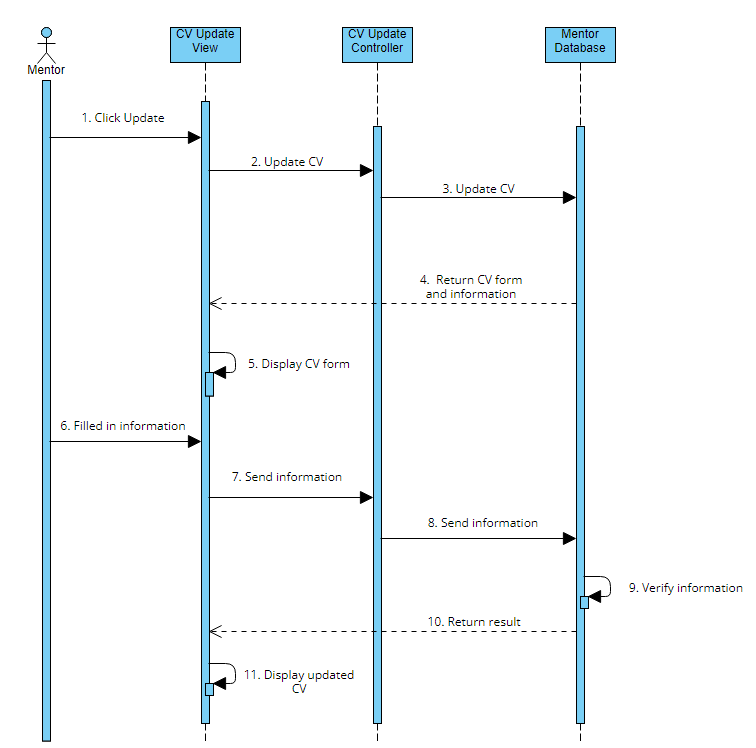
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

#### MentorSkill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a mentor’s skill, the input is a mentor’s skill and the output is void. When a new mentor create his/ her CV the skill will also be created and store in MentorSkill database* |
| *02* | *Update* | *This method uses to update a mentor’s skill, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor’s skill, the input is a mentor’s skill and the output is void. All information about this mentor’s skill will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor’s skill, the input is a mentor id and the output is a mentor’s skill. All information about this mentor’s skill will be display using Sql query* |

### c. Sequence Diagram(s)

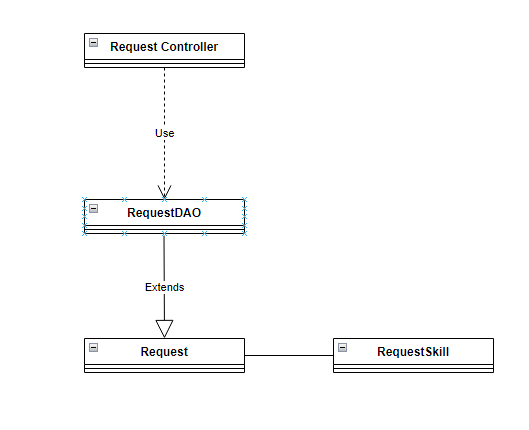


### d. Database queries

*Update Mentor set Email = ?, FullName = ?, Address = ? , Phone = ? , DateOfBirth = ? , Sex = ? , Profession =?, ProfessionIntro = ?, ServiceDesc = ?, FrameWork = ?, Avatar = ? where ID = ? to update mentor’s CV*

## 14. View list following Request

## a. Class Diagram

******

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Verify* | *This method will be call to check if the mentee have work with this mentor before or not by search in the request table and return the request with this mentor id and mentee id* |
| *02* | *Get List Request* | *This method will return all request belonging to a mentor by going to the database and select every request where MentorID = id* |

#### Request Class

***Class Methods***

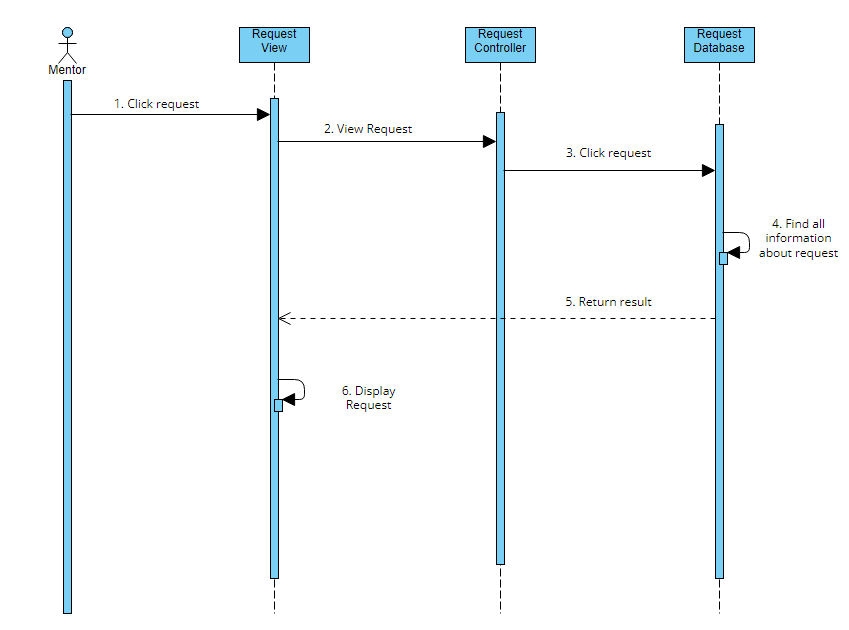
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### RequestSkill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

### c. Sequence Diagram(s)

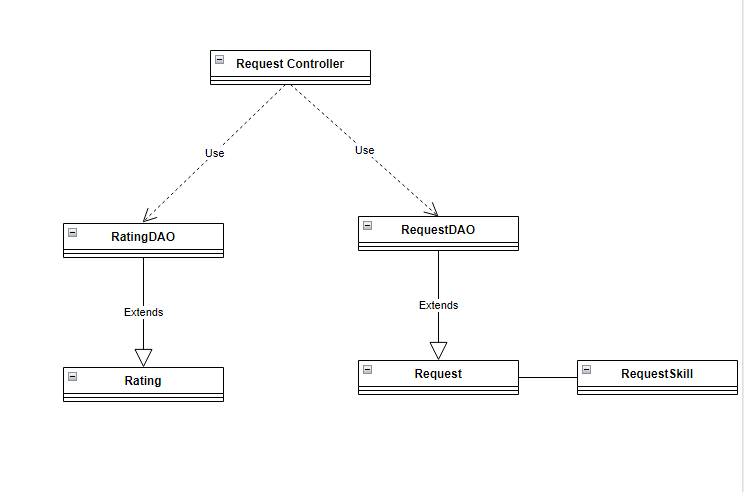


### d. Database queries

*Select \* from Request join RequestSkill on Request.ID = RequestSkil.RequestID to get all request*

## 15. View statistic request

## a. Class Diagram

******

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Verify* | *This method will be call to check if the mentee have work with this mentor before or not by search in the request table and return the request with this mentor id and mentee id* |
| *02* | *Get List Request* | *This method will return all request belonging to a mentor by going to the database and select every request where MentorID = id* |
| *03* | *Summary of request* | *This method will show summary of all request belong to a mentor. The input is MentorID and the output will be request summary, when call it will return the number of accepted request, inviting request and cancel request* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### RequestSkill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

#### RatingDAO Class

***Class Methods***

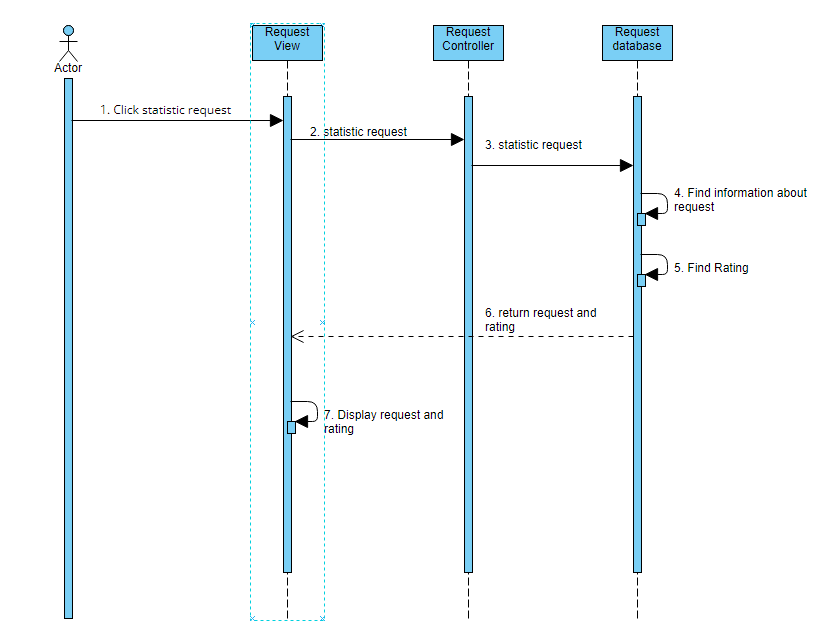
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Rating* | *This method allow mentee to give a mentor rate base on their skill. The input will be number of star and the output will be void. When the mentee rating the information will go to database and store there* |
| *02* | *Comment* | *This method allow mentee to leave a comment about a mentor. The input will be content of the comment and the output will be void. After mentee rate they will be able to leave comment if they want and this will also be store in the rating database* |
| *03* | *Get Rating* | *This method will allow the user to see the rating of a specific mentor. The input will be mentored and the out put will be rating, when this method is call the database will return all rating information belong to that mentor* |

#### Rating Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

### c. Sequence Diagram(s)



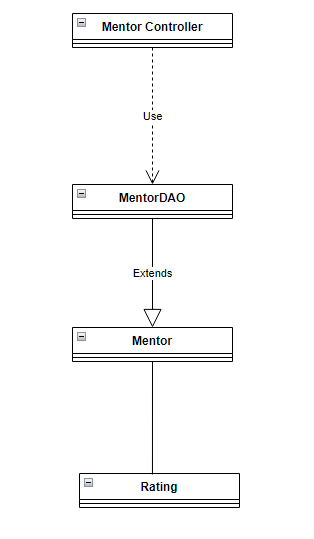
### d. Database queries

*Count(Select \* from Request join RequestSkill on Request.ID = RequestSkil.RequestID where Request.Status = ?)to calculate the total request*

*Select Rating from Rating where MentorID = ? to get mentor’s rating*

## 16. View list all mentor

## a. Class Diagram

******

### b. Class Specifications

#### MentorDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *View Profile* | *This method will display summary information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get summary about this mentor using Sql query* |
| *02* | *View CV* | *This method will display information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call, it will go to database and get every things about this mentor using Sql query* |
| *03* | *Create CV* | *This method will be used to create a new CV for mentor, the input will be information about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |
| *04* | *Update CV* | *This method will be used to update a CV for mentor, the input will be changes about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |
| *05* | *Get all mentor* | *When call this method will return the statistic of all mentor, rating and status of mentor by using sql query to select from the database* |

#### Mentor Class

***Class Methods***

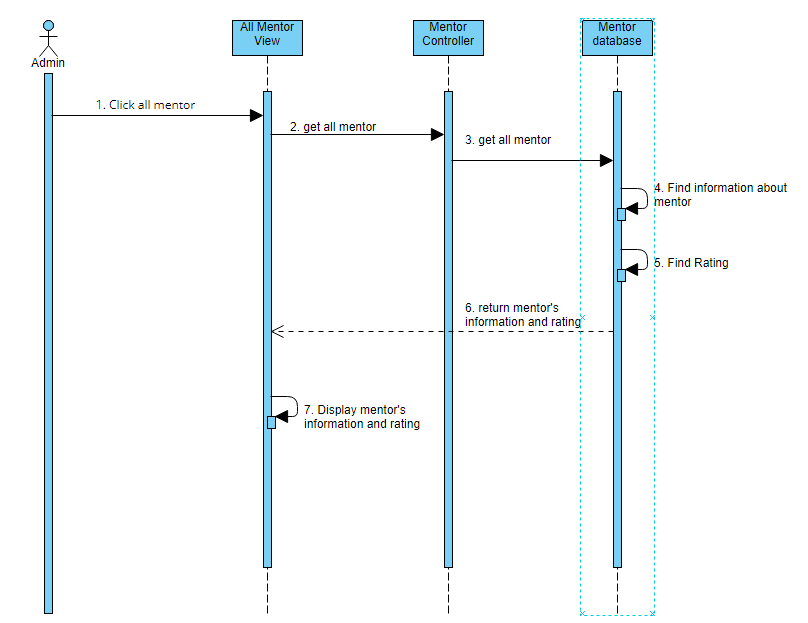
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

#### Rating Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

### c. Sequence Diagram(s)

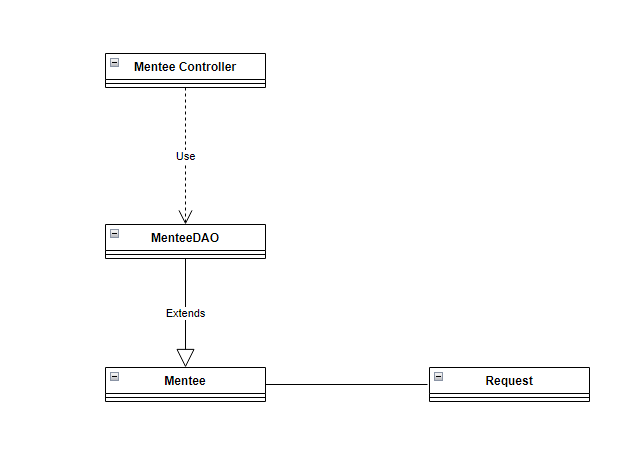


### d. Database queries

*Select \* from Mentor to get all mentor available in the system*

*Select Rating from Rating where MentorID = ? to get mentor’s rating*

## 17. Statistic of all mentee

******

### b. Class Specifications

#### MenteeDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Profile* | *This method is uses to Update a mentee’s profile, the input will be mentee’s Email, Fullname, Address, Phone, DateOfBirth, Sex , Avatar the out put will be Boolean to let the user know the result, the information will be enter and check then store in the database using Sql query* |
| *02* | *Get all mentee* | *When call this method will access database and get information about all mentee, the input is void and the output will be list of all mentee including the request summary* |

#### Mentee Class

***Class Methods***

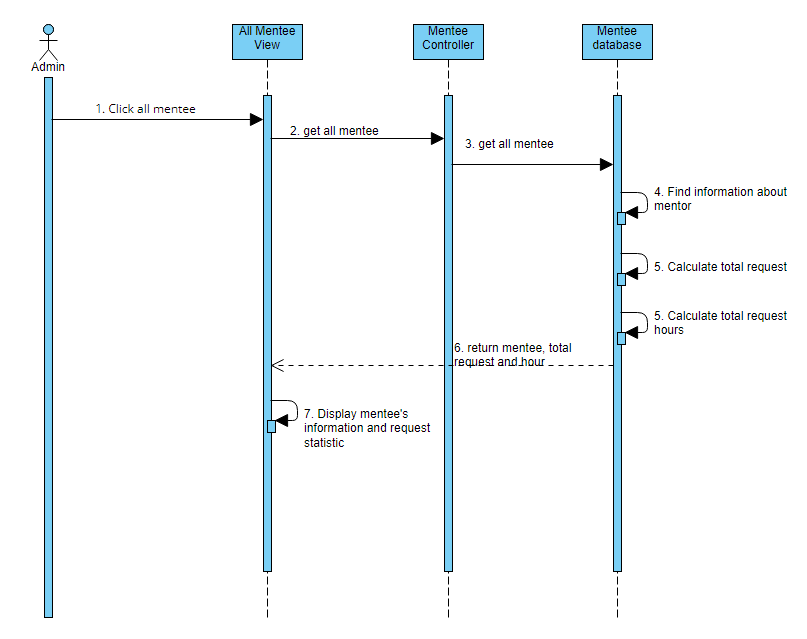
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentee, the input is an mentee and the output is void. The new mentee will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentee, the input is a mentee’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentee, the input is a mentee and the output is void. All information about this mentee will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentee, the input is a mentee id and the output is a mentee. All information about this mentee will be display using Sql query* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

### c. Sequence Diagram(s)



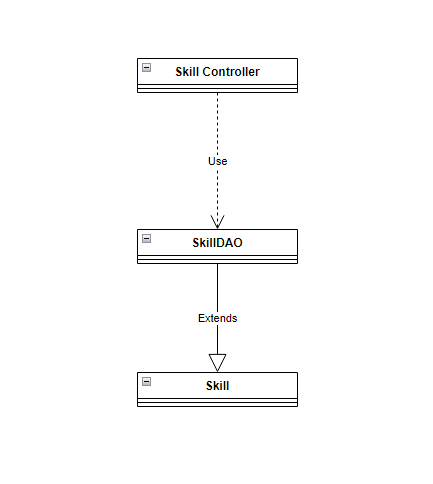
### d. Database queries

*Select \* from Mentee to get all mente available in the system*

*Count(Select \* from Mentee where MenteeID =?) To calculate the number of total requqest*

## 18. Create skill

## a. Class Diagram

******

### b. Class Specifications

#### SkillDAO Class

***Class Methods***

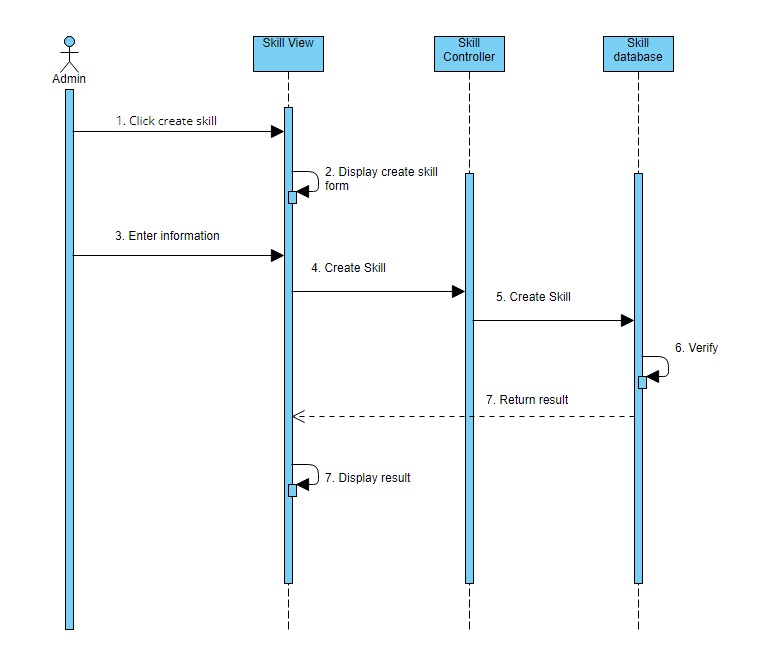
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Get List of Skills* | *This method will display the list of skills available on the website, the input will be void and output is a list of skill* |
| *02* | *Create Skill* | *When call this method will allow admin to input name and status of a new skill after being check it will be store in the database* |

#### Skill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

### c. Sequence Diagram(s)



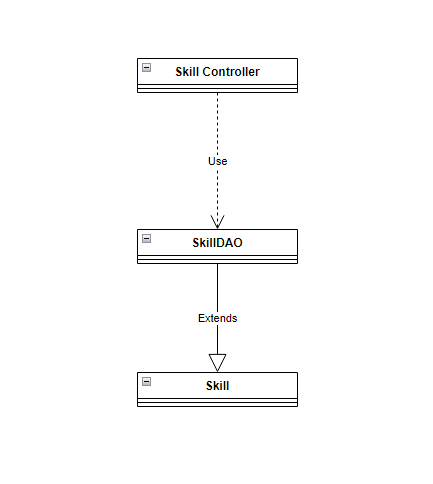
### d. Database queries

*Insert into Skill(Name, Status)*

*Values(?, ?)*

## 19. Update skill

## a. Class Diagram

******

### b. Class Specifications

#### SkillDAO Class

***Class Methods***

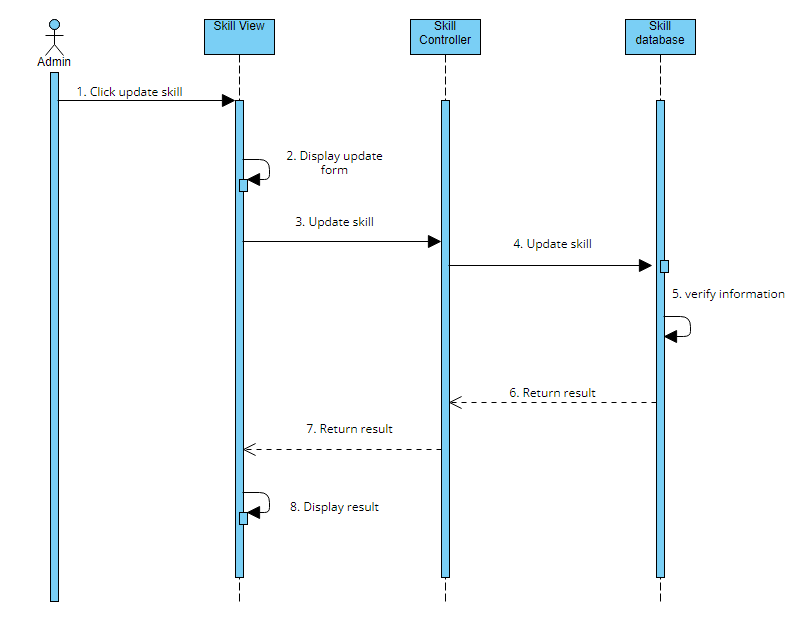
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Get List of Skills* | *This method will display the list of skills available on the website, the input will be void and output is a list of skill* |
| *02* | *Create Skill* | *When call this method will allow admin to input name and status of a new skill after being check it will be store in the database* |
| *03* | *Update Skill* | *This method will allow admin to update a skill that already existed in the database, when call the input is the changes and the out put will be Boolean after check it will be updata into the table* |

#### Skill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

### c. Sequence Diagram(s)

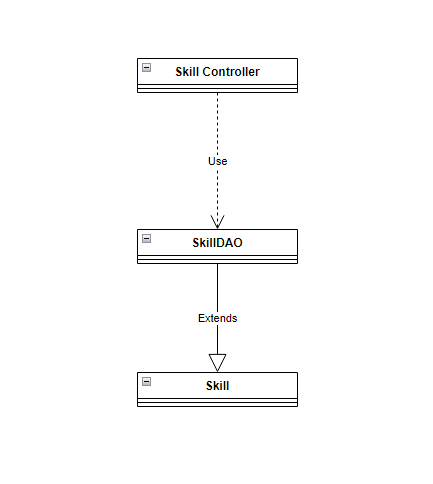


### d. Database queries

*Update Skill set (Name = ?, Status = ?) where ID =? To update skill base on ID*

## 20. View list of skills

## a. Class Diagram

******

### b. Class Specifications

#### SkillDAO Class

***Class Methods***

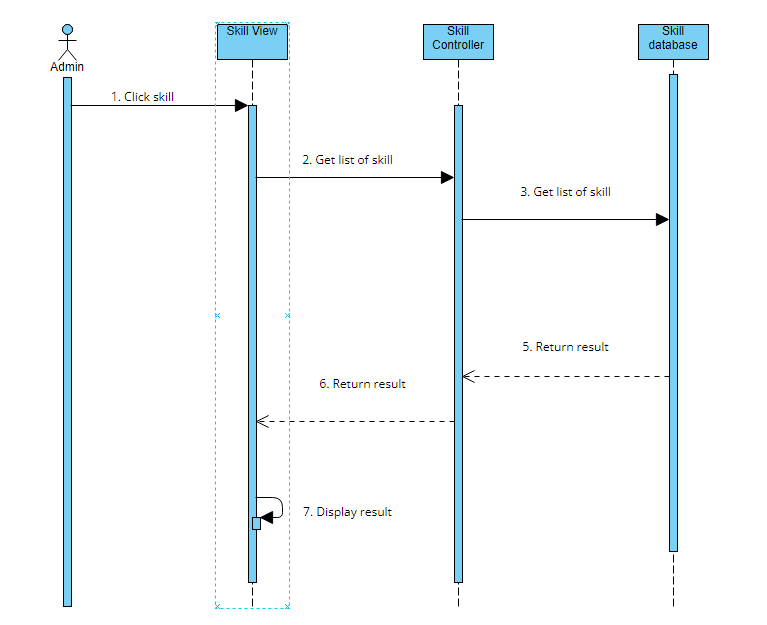
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Get List of Skills* | *This method will display the list of skills available on the website, the input will be void and output is a list of skill* |
| *02* | *Create Skill* | *When call this method will allow admin to input name and status of a new skill after being check it will be store in the database* |
| *03* | *Update Skill* | *This method will allow admin to update a skill that already existed in the database, when call the input is the changes and the out put will be Boolean after check it will be updata into the table* |

#### Skill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

### c. Sequence Diagram(s)

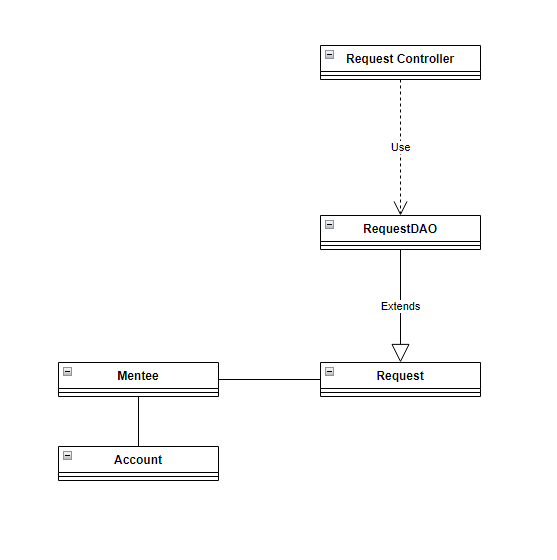


### d. Database queries

*Select \* from Skill to get all skill*

## 21. View list all request

## a. Class Diagram

******

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Request* | *This method allow mentee to update an existing request, the input is the change about the request and the output will be Boolean to indicate the method successes or fail. The information will be entered by mentee when they need to change a request and after that it will be check and store in database* |
| *02* | *Filter* | *This method allows admin to enter deadlinedate or status to search for requests that have the same status or end from date1 to date2. When call this method will get all request fit the criteria from the database using sql query* |
| *03* | *Get all request* | *This method will get every available requests in the database and display it using sql query* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### Mentee Class

***Class Methods***

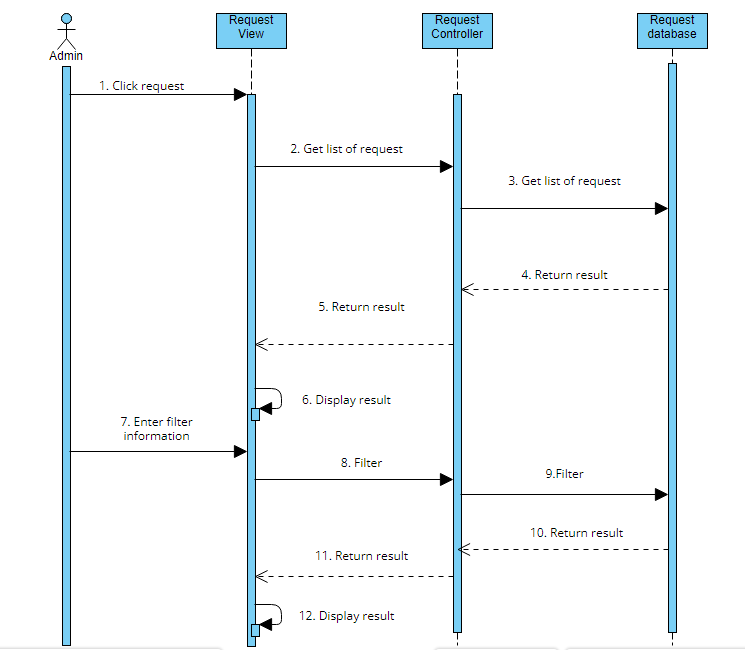
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentee, the input is an mentee and the output is void. The new mentee will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentee, the input is a mentee’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentee, the input is a mentee and the output is void. All information about this mentee will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentee, the input is a mentee id and the output is a mentee. All information about this mentee will be display using Sql query* |

#### Account Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new account, the input is an account and the output is void. The new account will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update an account, the input is an account’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete an account, the input is an account and the output is void. All information about this account will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific account, the input is an account id and the output is an account. All information about this account will be display using Sql query* |

### c. Sequence Diagram(s)



### d. Database queries

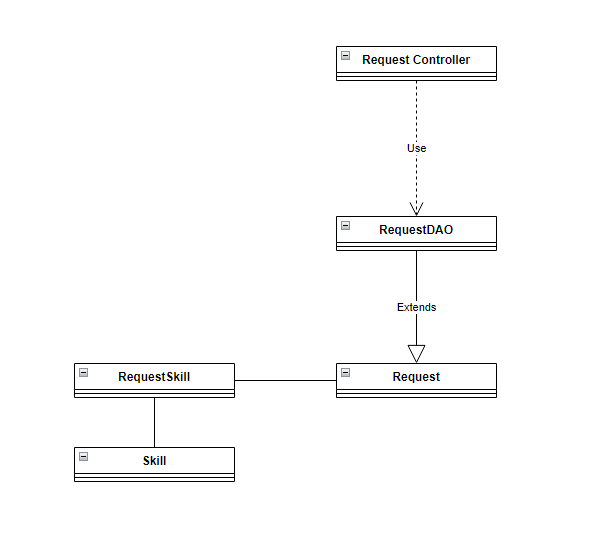
*Select \* from Request to get all request*

*Select \* from skill where DealineDate between ? and ? / Select \* request where status = ?*

*To filter request using deadline date and/or status*

## 22. List request by me

## a. Class Diagram

******

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Request* | *This method allow mentee to update an existing request, the input is the change about the request and the output will be Boolean to indicate the method successes or fail. The information will be entered by mentee when they need to change a request and after that it will be check and store in database* |
| *02* | *Filter* | *This method allows admin to enter deadlinedate or status to search for requests that have the same status or end from date1 to date2. When call this method will get all request fit the criteria from the database using sql query* |
| *03* | *Get all request mentee* | *This method will get every request belong to a mentee in the database and display it using sql query the input will be mentee id and the output will be a list of request* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### RequestSkill Class

***Class Methods***

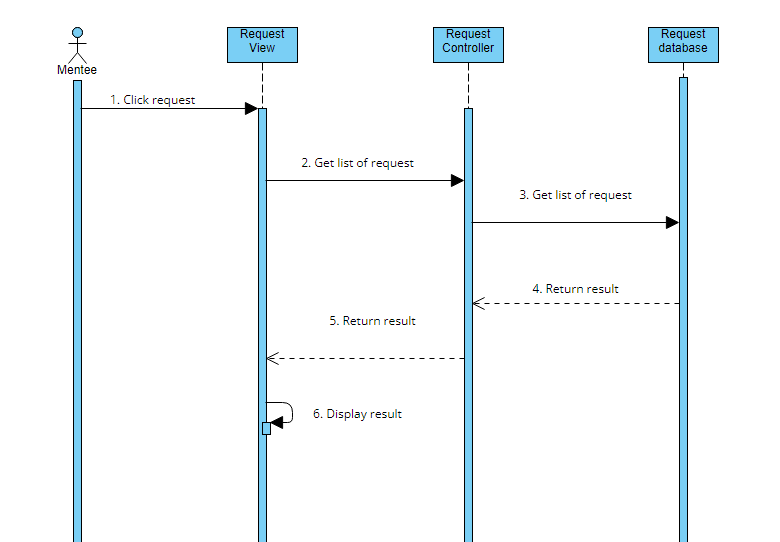
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

#### Skill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

### c. Sequence Diagram(s)

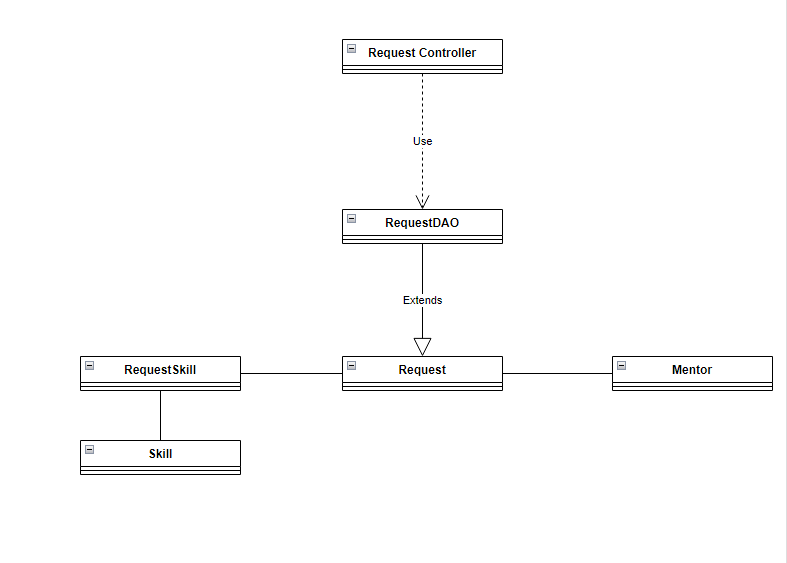


### d. Database queries

*Select \* from Request where MenteeID = ? to get all request belong to a specific mentor*

## 23. Statistic request by me

## a. Class Diagram

**

### b. Class Specifications

#### RequestDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Update Request* | *This method allow mentee to update an existing request, the input is the change about the request and the output will be Boolean to indicate the method successes or fail. The information will be entered by mentee when they need to change a request and after that it will be check and store in database* |
| *02* | *Filter* | *This method allows admin to enter deadlinedate or status to search for requests that have the same status or end from date1 to date2. When call this method will get all request fit the criteria from the database using sql query* |
| *03* | *Get all request mentee* | *This method will get every request belong to a mentee in the database and display it using sql query the input will be mentee id and the output will be a list of request* |
| *04* | *Summary of request* | *When call this method will be able to calculate the total requests, the total hours and the number of mentor invole in request the input will be menteeID and the out put will be summary of all mentee’s request* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### RequestSkill Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

#### Skill Class

***Class Methods***

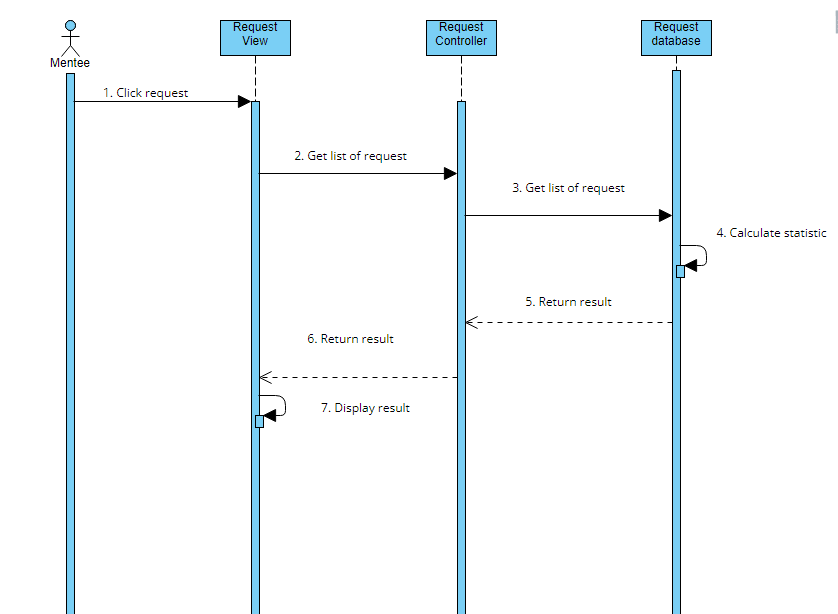
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new skill, the input is a skill and the output is void. The new skill will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a skill, the input is a skill’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a skill, the input is a skill and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific skill, the input is a skill id and the output is a skill. All information about this skill will be display using Sql query* |

#### Mentor Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

### c. Sequence Diagram(s)



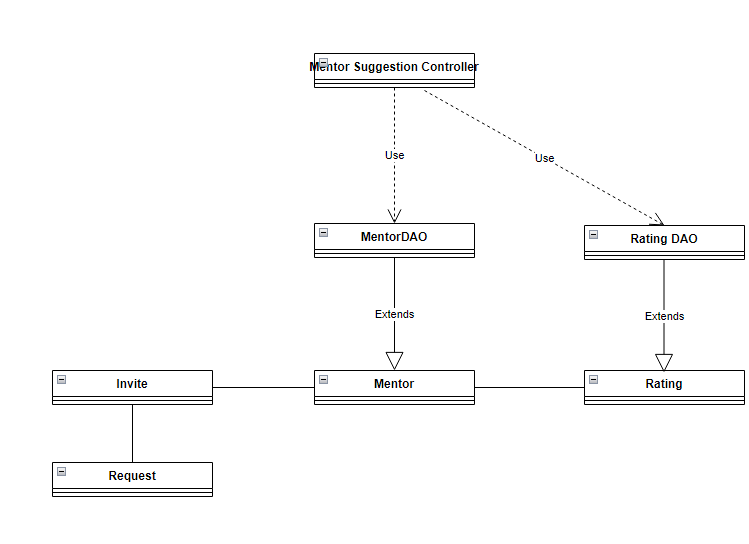
### d. Database queries

*Select \* from Request where MenteeID = ? to get all request belong to a specific mentor*

*Count(Select \* from Request where MenteeID = ?) to calculate the total requests*

## 24. List Mentor suggestion (for Mentee)

## a. Class Diagram

******

### b. Class Specifications

#### MentorDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *View Profile* | *This method will display summary information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call it will go to database and get summary about this mentor using Sql query* |
| *02* | *View CV* | *This method will display information about a mentor, the input will be MentorID and the output will be information about that mentor, when the method is call, it will go to database and get every things about this mentor using Sql query* |
| *03* | *Create CV* | *This method will be used to create a new CV for mentor, the input will be information about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |
| *04* | *Update CV* | *This method will be used to update a CV for mentor, the input will be changes about mentor and the output is void, when the mentor put in information about themselves after that it will be check and store in database* |
| *05* | *Get all mentor* | *When call this method will return the statistic of all mentor, rating and status of mentor by using sql query to select from the database* |

#### RatingDAO Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Rating* | *This method allow mentee to give a mentor rate base on their skill. The input will be number of star and the output will be void. When the mentee rating the information will go to database and store there* |
| *02* | *Comment* | *This method allow mentee to leave a comment about a mentor. The input will be content of the comment and the output will be void. After mentee rate they will be able to leave comment if they want and this will also be store in the rating database* |
| *03* | *Get Rating* | *This method will allow the user to see the rating of a specific mentor. The input will be mentored and the output will be rating, when this method is call the database will return all rating information belong to that mentor* |

#### Request Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new request, the input is a request and the output is void. The new request will be check and it will be store in the database* |
| *02* | *Update* | *This method uses to update a request, the input is a request’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a request, the input is a request and the output is void. All information about this request will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific request, the input is a request id and the output is a request. All information about this mentee will be display using Sql query* |

#### Mentor Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new mentor, the input is a mentor and the output is void. The new mentor will be check if it doesn’t exist it will be store in the database* |
| *02* | *Update* | *This method uses to update a mentor, the input is a mentor’s change and the output is void. The changes will be store in the database* |
| *03* | *Delete* | *This method uses to delete a mentor, the input is a mentor and the output is void. All information about this mentor will be delete from the database* |
| *04* | *Get* | *This method uses to find a specific mentor, the input is a mentor id and the output is a mentor. All information about this mentor will be display using Sql query* |

#### Rating Class

***Class Methods***

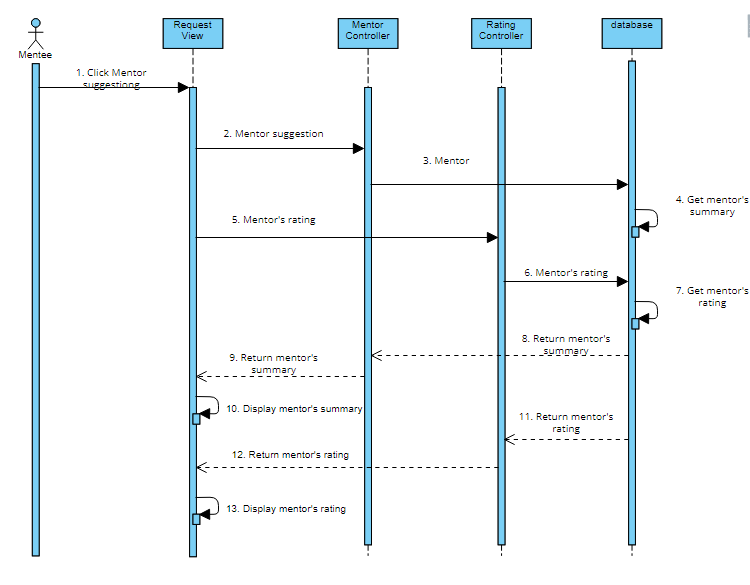
|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new rating for a mentor, the input is rating and comment. the output is void. The mentee will be check if they have work with the mentor or not if not the rating and comment will be store in database* |
| *02* | *Update* | *This method allow mentee to update their rating on a mentor including rating and comment. The input will be changes and the output will be void. When mentee want to change their rating a mentor, they will reenter the rating and comment then it will be store in database* |
| *03* | *Get* | *This method uses to find a specific rating on a mentor, the input is a rating id and the output is a rating and comment. All information about this rating will be display using Sql query* |

#### Invite Class

***Class Methods***

|  |  |  |
| --- | --- | --- |
| **No** | **Method** | **Description** |
| *01* | *Create* | *This method uses to create a new Invitation, the input is an invite and the output is void. The new invitation will be check if the mentor accepted it or not, if accepted the invitation will be store in database* |
| *0* | *Get* | *This method uses to find a specific invitation, the input is a invite id and the output is a invite. All information about this mentor will be display using Sql query* |

### c. Sequence Diagram(s)



### d. Database queries

*Select \* from Mentor where MentorID = ? to get mentor suggestion for request*

*Select Rating from Rating Where MentorID = ? to get rating for a specific mentor*

*Insert into Invite(ReqID, MentorID, Status)*

*Values(?, ?, ?) to create a new invitation*

# III. Database Tables

## 1. Account

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Primary key |
| 2 | AccountName | varchar | 50 |  |  |  |  |
| 3 | Password | varchar | 60 |  |  |  |  |
| 4 | RoleID | int |  |  |  | X | Foreign key references Roles(ID) |

## 2. Invite

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Primary key |
| 2 | ReqID | int |  |  |  | X | Foreign key references Request(ID) |
| 3 | MentorID | int |  |  |  | X | Foreign key references Mentor(ID) |
| 4 | Status | nvarchar | 250 |  |  |  |  |

## 3. Mentee

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Foreign key references Acount(ID), Primary key |
| 2 | Email | nvarchar | 250 |  |  |  |  |
| 3 | FullName | nvarchar | 250 |  |  |  |  |
| 4 | Address | nvarchar | 250 |  |  |  |  |
| 5 | Phone | nvarchar | 250 |  |  |  |  |
| 6 | DateOfBirth | date |  |  |  |  |  |
| 7 | Sex | nvarchar | 250 |  |  |  |  |
| 8 | Avatar | nvarchar | 250 |  |  |  |  |

## 4. Mentor

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Foreign key references Acount(ID), Primary key |
| 2 | Email | nvarchar | 250 |  |  |  |  |
| 3 | FullName | nvarchar | 250 |  |  |  |  |
| 4 | Address | nvarchar | 250 |  |  |  |  |
| 5 | Phone | nvarchar | 250 |  |  |  |  |
| 6 | DateOfBirth | date |  |  |  |  |  |
| 7 | Sex | nvarchar |  |  |  |  |  |
| 8 | Profession | nvarchar | 250 |  |  |  |  |
| 9 | ProfessionIntro | nvarchar | 250 |  |  |  |  |
| 10 | ServiceDesc | nvarchar | 250 |  |  |  |  |
| 11 | AchievementDesc | nvarchar | 250 |  |  |  |  |
| 12 | Framework | nvarchar | 250 |  |  |  |  |
| 13 | Avatar | nvarchar | 250 |  |  |  |  |

## 5. MentorSkill

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | SkillID | int |  |  | X | X | Foreign key references Skill(ID) |
| 2 | MentorID | int |  |  | X | X | Foreign key references Mentor(ID)  Primary key(SkillID, MentorID) |

## 6. Rating

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Primary key |
| 2 | Rate | int |  |  |  |  |  |
| 3 | Comment | nvarchar | 250 |  |  |  |  |
| 4 | MenteeID | int |  |  | X | X | Foreign key references Mentee(ID) |
| 5 | MentorID | int |  |  | X | X | Foreign key references Mentor(ID) |

## 7. Request

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Primary key |
| 2 | Title | nvarchar | 250 |  |  |  |  |
| 3 | Status | nvarchar | 250 |  |  |  |  |
| 4 | Content | nvarchar | 250 |  |  |  |  |
| 5 | MenteeID | int |  |  | X | X | Foreign key references Mentee(ID) |
| 6 | DeadlineDate | date |  |  |  |  |  |
| 7 | DeadlineHour | int |  |  |  |  |  |

## 8. RequestSkill

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | SkillID | int |  |  | X | X | Foreign key references Skill(ID) |
| 2 | RequestID | int |  |  | X | X | Foreign key references Request(ID)  Primary Key(SkillID, MentorID) |

## 9. Roles

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Primary key |
| 2 | Name | nvarchar | 50 |  |  |  |  |

## 10. Skill

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **#** | **Field name** | **Type** | **Size** | **Unique** | **Not Null** | **PK/FK** | **Notes** |
| 1 | ID | int |  | X | X | X | Primary key |
| 2 | Name | nvarchar | 250 |  |  |  |  |
| 3 | Status | nvarchar | 250 |  |  |  |  |