ASSIGNMENT 2

GUIDE.ME

*Nguyễn Thanh Hoàng*

*Nguyễn Trí Nhân*

*Đặng Văn Hùng*

*Phạm Gia Bảo*

CONNECT MENTOR USE CASE DETAIL

GUIDE.ME

# User Story Description

* As a mentee, I want to connect with another user as a mentor so that he/she can help me to make a to-do list for my issue.

# Business Objectives

* Users can easily find an appropriate mentor(search by issue or search direct mentor name) and connect with the mentor.
* Provide the chat feature for the mentee and mentor when they are connected.

# Acceptance Criteria

## Business rules

* Users must put the keyword to the search component in the navigation bar.
* The keyword can be related to the problem the user is having or mentor’s name.
* System will show a list of mentors available to users in a list.
* If no results are returned, a message will be displayed to the user
* Each item of the list contains all the public information of the mentor including the feedback’s score and feedback’s message.
* After connecting successfully with mentor, user can find the mentor in connection page(provide chat function when user click into mentor item)

## Scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Given** | **When** | **Then** |
| 1 | Search bar | User input keyword to search bar and click search | System will show a list of mentors available to users in a list |
| 2 | Lists of mentors | User choose a mentor AND click connect | A connect request should send to mentor |

NOTIFICATION USE CASE DETAIL

GUIDE.ME

# User Story Description

* As a mentee, I want to receive notifications so that I can keep track of my to-do list

# Business Objectives

* Users receive notification when the task is nearing deadline.

# Acceptance Criteria

## Business rules

* Notification will be based on the due date of todo-list.
* Notification will be show two time
  + The first is before the due date a amount of time(will be calculated by the system)
  + the second will be at the due date of todo-list
* Notification will contains todo-list title, a message(send by notification system)
* When the user will show on the device’s notification panel.
* Click into notification will navigated user’s screen to todo-list page

## Scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Given** | **When** | **Then** |
| 1 | Mentor dashboard | Mentee click connect with mentor | A notification is pushed to mentor device |
| 2 | A todo list | Is time out within 10 minutes | A notification is pushed to mentee’s device |

FEEDBACK USE CASE DETAIL

GUIDE.ME

# User Story Description

* As a user, I want to feedback to another user so that everyone can read the feedback and help them find the best user

# Business Objectives

* Users can feedback the score and message(optional), after the user connected with the mentor and completed the todo-list.

# Acceptance Criteria

## Business rules

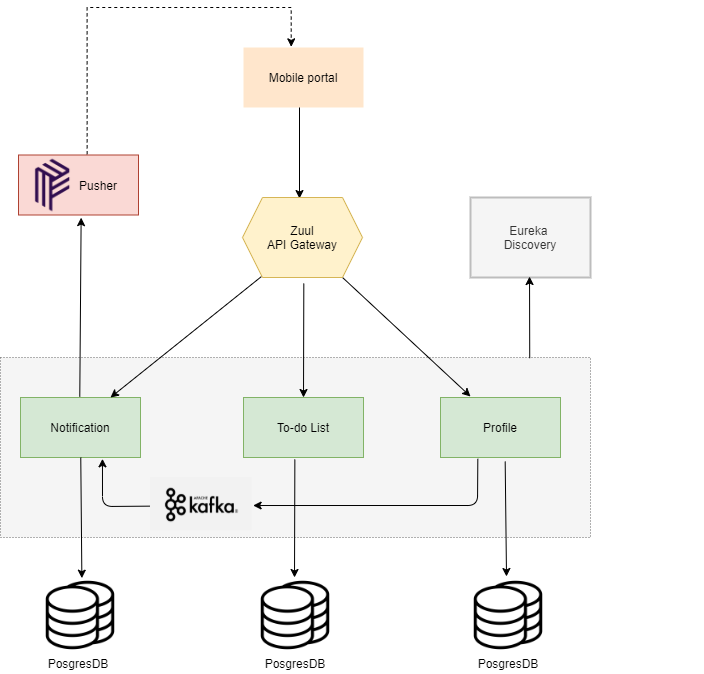
* Users can give feedback to a mentor by clicking the "feedback" button on the connection page.
* Users can see the feedback button only when users have successfully connected with that mentor and the mentor completes todo-list creation.
* Feedback score will be scale from 0 to 5.
* The feedback score button will be a strip of 5 stars.
* When the user clicks to the star, it will correspond to the appropriate score.
* Users can also provide feedback messages(optional).
* Users click the “submit” button in the right bottom corner to submit feedback.
* Mentor should receive feedback notification

## Scenarios

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Given** | **When** | **Then** |
| 1 | A mentee connected with a mentor AND mentor finished making todo-list | mentee go to connection page | feedback button in mentor item must available for mentee |
| 2 | Mentee in feedback page | Mentee click submit button | A notification send to Mentor |

ARCHITECTURE DETAIL

GUIDE.ME

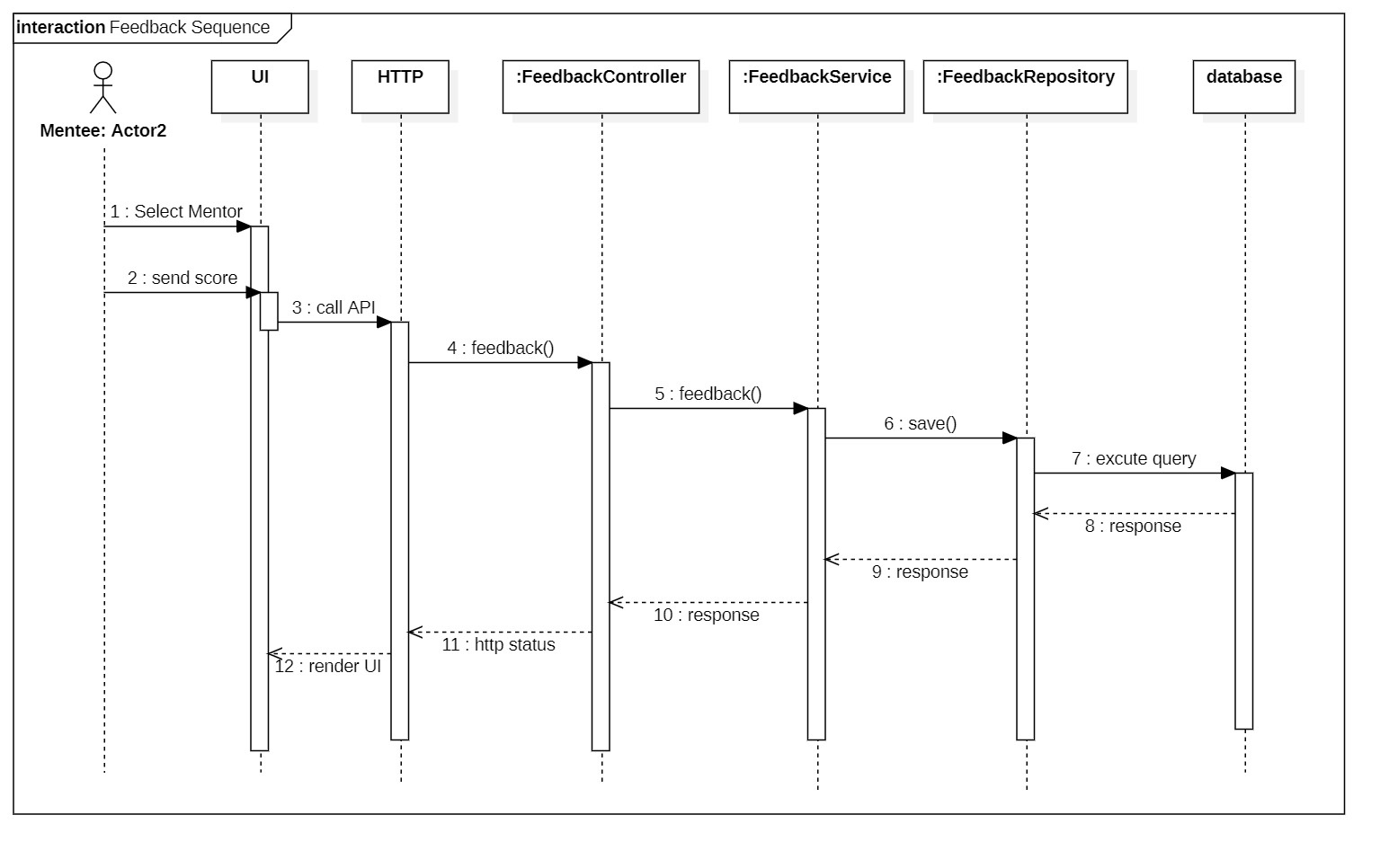


* **Goal**
  + The microservice architecture is chosen as main design approach by following purpose:
    - Because microservices let you independently scale services up or down, the ease—and cost—of scaling is dramatically less than in a monolithic system. Adding new capabilities usually means adding discrete new microservices, not redoing the entire application, which increases both development speed and application stability.
    - Independent development teams can collaborate more easily and speed up time-to-market.
    - If one microservice fails, all the others will likely continue to work.

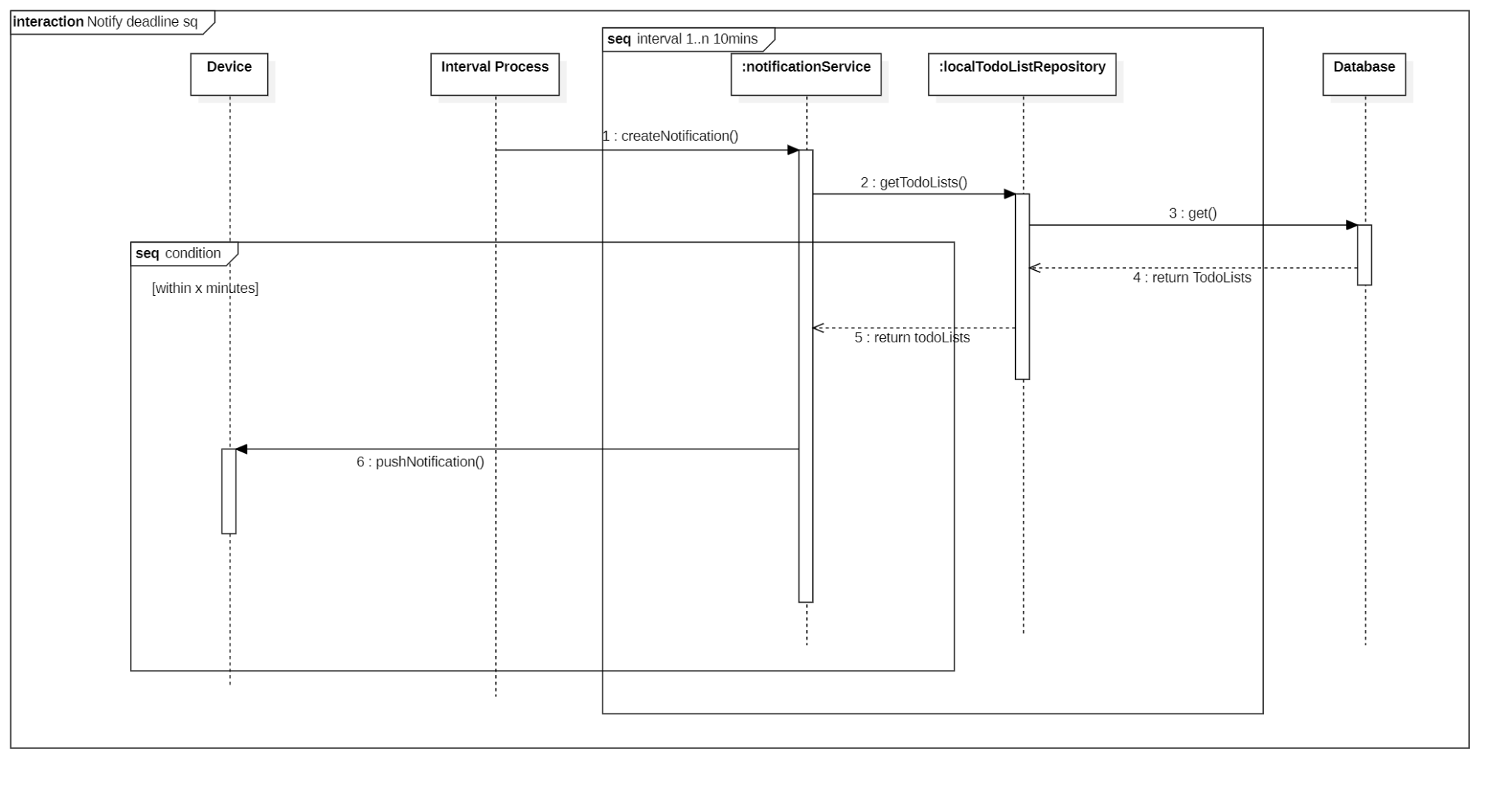
UML MODEL

GUIDE.ME

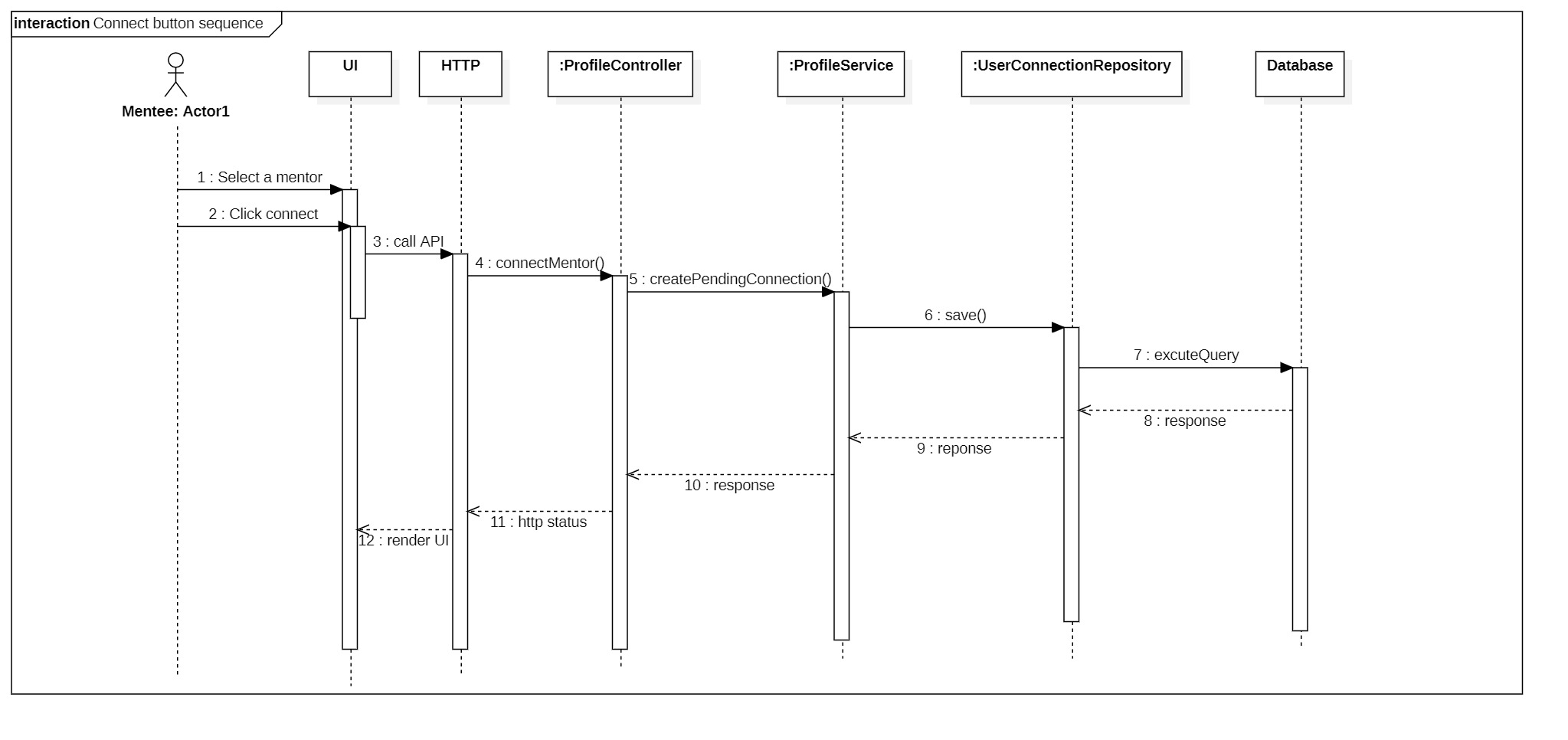
* Feedback sequence diagram



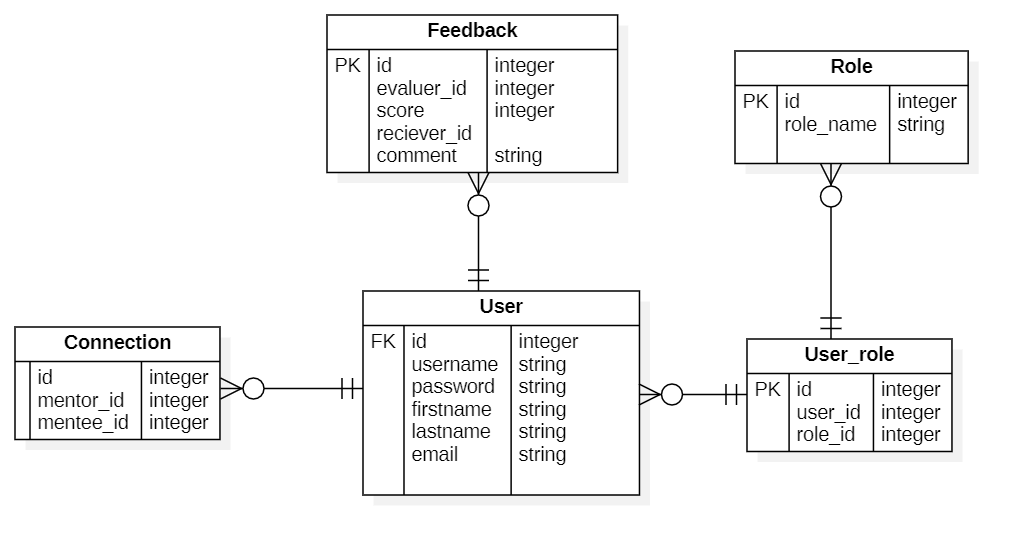
* Notification sequence diagram



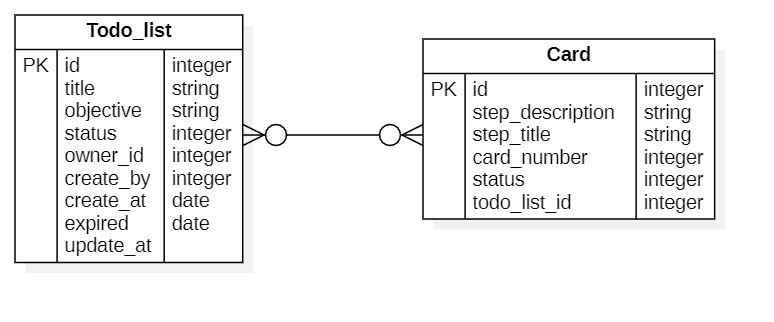
* Connect mentor sequence diagram



* Profile service ERD



* Todo-list service ERD



* Notification service ERD

