TDT4145 Project delivery part 2

24.03.2021

Textual description that documents the application including diagrams

Group Number: 136

Word Count: 549

Name	Email	Phonenumber
Philip Emiliam Ditlevsen	philiped@stud.ntnu.no	+47 902 81 229
Natalie Sørensen Forshaw	natalisf@stud.ntnu.no	+47 467 93 624
Yngve Tryggestad Larsen	yngvetl@stud.ntnu.no	+47 919 17 004

Class description

User.java:
Constructs a user with an automatically generated userID, email and password as a string, the role string either has its value as “instructor” or “student” and numCreated and numRead as integers. The class also includes a getter for each of the attributes, and setters for all attributes except userID.

DBConn.java:
This is our database connector, it connects to our mysql database.
This will not work unless there is a MySQL database running locally on the computer, with the configuration (name, password, databasename, port, ip) given in this class.

forumController.java:
We put all our operations in the controller. This class contains the operations to log in, register, make a post or a reply, search by keyword and get statistics. In addition we included getters for courses and folders, since we found out we didn’t need classes for them, so that posts could be structured inside them. In the beginning we also define a new attribute currentUser that represents the user that is currently logged in.

Main.java
Our main class showcases our 5 use cases. To illustrate the use cases we first had to register a couple of users, student 1, student 2 and instructor, and then we constructed a course.

Use case 1:
Student 1 logs in with email and password.

Use case 2:
Student 1 makes 2 posts in, where one of them is placed in the folder “exam” with the tag “questions”. One of the posts contains the keyword “wal”. For more data into use case 4 we made the other student do exactly the same.

Use case 3:
The instructor replies to 2 of the students' posts, where one of the replies goes to the post made by student 1 in the “exam” folder.

Use case 4:
Student 1 logs in again to search for the keyword “wal”, post 2 and 4 contains the keyword and are returned in the operation.

Use case 5:
The instructor logs in again to check the statistics of the users. The return is an arraylist including the 3 users ordered by read posts in descending order.

Diagrams and their explanation

Figure 1

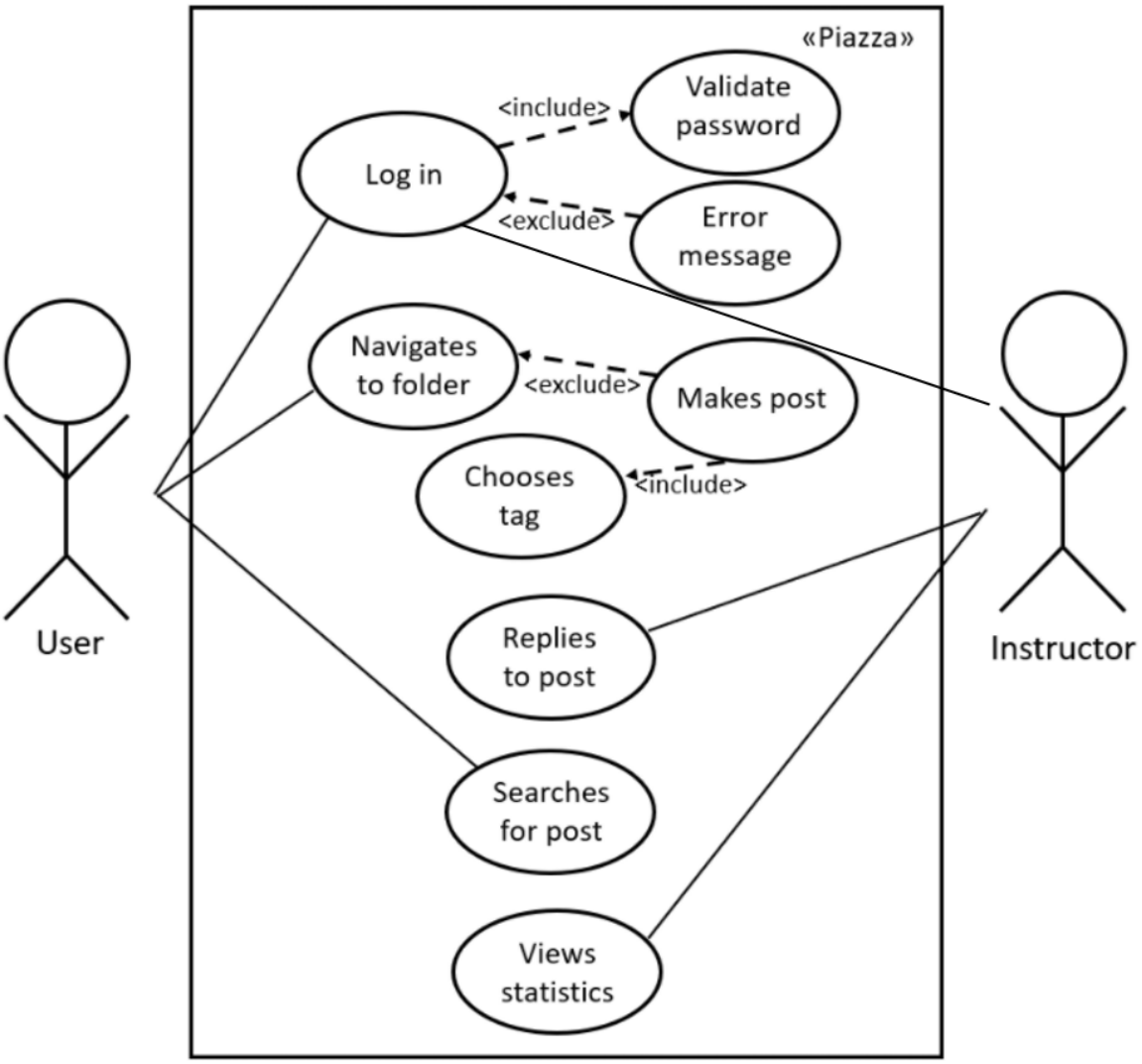

Figure 1: A use case model that shows what the users can do and shows how they solve the specific use cases we have been tasked with. It is also implicitly within the program that an instructor can do the same thing as a regular user, but the user can't do everything that an instructor can do (views statistics). This use case model is also strictly used for our specific use cases hence why a user "can't" reply, because there isn't a use case where the user replies to a post.

Figure 2

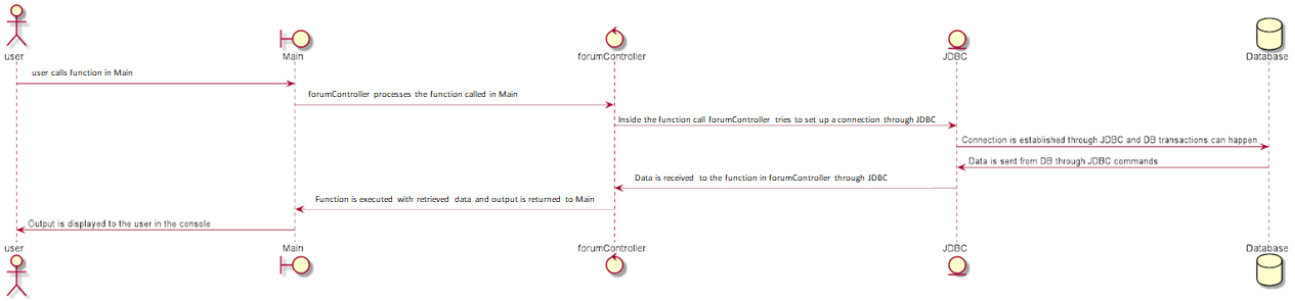

Figure 2: Sequence diagram that shows a general realization of a use case, it shows the different levels of code it runs through. Function call, input and output varies from use case to use case, but this shows the generalized version.