

TDT4145 Project 2021 DB2

Simen Omholt-Jensen, Anna Zhang, Vebjørn Steinsholt

March 25, 2021

Contents

1	Installation	1
1.1	Requirements	1
2	Usage	1
3	Demo	1
4	Modules	2
5	Classes	3
6	Use Cases	4
7	Appendix - Runtime description	5

1 Installation

```
git clone https://github.com/simenojensen/TDT4145_Project.git
```

1.1 Requirements

- MySQL (≥ 8.0)

See `environment.yml` for python packages. For installation with `conda`:

```
conda env create -f environment.yml
```

2 Usage

```
cd TDT4145_Project/piazza
python main.py
```

Example login as student :

useremail: frumford6@ted.com

userpassword: XpdsDP085Un

Example login as instructor :

useremail: stretters@mashable.com

userpassword: AQqzBO2mTEkB

3 Demo

For a video demonstration of some of the implemented use cases, please go to https://github.com/simenojensen/TDT4145_Project/.

The repository was made public after the deadline.

4 Modules

- `main.py`
 - This module prompts the user for their MySQL login information. A database called `TDT4145ProjectGroup131` is created and filled with data from the `.csv` files in the `data` folder. Program is then run by prompting the user for either (1) log in as student, (2) log in as instructor, or (3) quit.
- `utils.py`
 - This module contains code that creates the `TDT4145ProjectGroup131` database, creates the tables of the `TDT4145ProjectGroup131` database, and fills the database with data found in the `.csv` files in the `data` folder.
 - **Functions:**
 - * `create_database()`
 - Helper function that executes the MySQL query for database creation.
 - * `setup_database()`
 - Drops the `TDT4145ProjectGroup131` database if already exists, then creates the database and executes the table initialization statements.
 - * `insert_data()`
 - Reads the `.csv` files from the `data` folder and inserts the data into the existing `TDT4145ProjectGroup131` database.
- `tables.py`
 - This module contains code which defines the MySQL tables, their fields and their constraints used to setup the `TDT4145ProjectGroup131` database. The database name is stored in the string `DB_NAME` variable. The tables are stored in a dictionary named `TABLES`.
- `piazza_user.py`
 - This module contains code for the classes implementing the functionality for the required use cases. The `PiazzaUser` class is a super class to the `Instructor` and `Student` classes. The `PiazzaUser` contains functionality shared between the `Instructor` and `Student` classes such as logging in, creating posts such as threads or replies, and keyword search. The `Student` class includes a unique action menu. The `Instructor` class includes a unique action menu and functionality for viewing statistics.

5 Classes

- **PiazzaUser** (super class):
Handles functionality shared among the **Student** and **Instructor** sub classes.
 - **Methods:**
 - * **login()**
 - Handles both **Student** and **Instructor** login functionality by prompting the user for an **useremail** and a **userpassword**. The method then verifies the input by checking if they exist in the **Login** MySQL table.
 - On successful verification, the method then stores information about the user (**UserID**) and the course (**CourseID**) which are used for later use cases.
 - * **create_post()**
 - Provides a text-based menu interface prompting the user for which type of post to create. User is prompted for (1) create a thread, (2) create a reply, or (3) go back to the action menu.
 - * **create_thread()**
 - Creates a thread post by prompting the user for post content, folder, and tag. The input is then inserted in the MySQL tables **Post**, **Thread**, **Tags**, and **ThreadInFolder** as explained in the DB1 deliverable.
 - * **create_reply()**
 - Creates a reply post by prompting the user for a post id and post content. A post is then inserted into the MySQL table **Post**, and the MySQL table **Thread** is updated to reflect that the thread post has received a reply.
 - * **search_keyword()**
 - Prompts the user for a keyword to search. The program then searches for matches in (1) the post content of posts, (2) thread tags, (3) folder names, and (4) user names. The method then prints the **PostID** for posts related to the keyword search matches.
 - * **close()**
 - Closes the MySQL connection.
- **Student** (sub class):
Provides a text-based menu interface for allowable **Student** actions.
 - **Methods:**
 - * **action_menu()**
 - User is prompted for the following options (1) create a post, (2) search for a keyword, and (3) log out.
- **Instructor** (sub class):
Provides a text-based menu interface for allowable **Instructor** actions. Also includes functionality to view user statistics.
 - **Methods:**
 - * **action_menu()**
 - User is prompted for the following options (1) create a post, (2) search for a keyword, (3) view statistics, and (4) log out.
 - * **view_statistics()**
 - Methods prints a table consisting of user names, the number of threads read by those users, and the number of posts created by those users. The table is sorted on highest read posting numbers.

6 Use Cases

All use cases were implemented within a fixed course.

1. A student or an instructor can log in by checking that their user email and user password exists in the MySQL `Login` table. This is implemented by either instantiating a `Student` or an `Instructor` sub class. The `__init__` function of the super class `PiazzaUser` is then run, which calls the `PiazzaUser.login()` method. This method implements the login functionality.
2. A student or an instructor can create an initial post, which is then considered a thread. The thread includes post content, a folder (e.g. "Exam"), and a tag (e.g. "Question"). The function for creating a thread is either called from the `Student.action_menu()` method, or the `Instructor.action_menu()` method depending on whether the user is logged in as a student or instructor. The method that implements the functionality for creating a thread is implemented in the `PiazzaUser.create_thread()` method.
3. A student or an instructor can reply to a post. The reply contains a post id to the post to reply to and some post content. The function for creating a reply is either called from the `Student.action_menu()` method, or the `Instructor.action_menu()` method depending on whether the user is logged in as a student or instructor. The method that implements the functionality for creating a reply is implemented in the `PiazzaUser.create_reply()` method.
4. A student or an instructor can search for a keyword and receive a list of post ids of posts that are related to the keyword. A related post id is a post related to a match in (1) the post content of posts, (2) the thread tags, (3) folder names, and/or (4) user names. The function for keyword search is either called from the `Student.action_menu()` method, or the `Instructor.action_menu()` method depending on whether the user is logged in as a student or instructor. The method that implements the functionality for keyword search is implemented in the `PiazzaUser.search_keyword()` method.
5. An instructor can view user statistics. This statistics view includes user names for a course, their number of threads read, and the number of posts created. This functionality is unique for an `Instructor`, and is thus implemented in the `Instructor.view_statistics()` method.

7 Appendix - Runtime description

```
(py3) + piazza git:(main) python main.py
Enter MySQL user: root
Enter MySQL password:
Database TDT4145ProjectGroup131 created successfully.
Creating table User: OK
Creating table Login: OK
Creating table PostCreator: OK
Creating table Student: OK
Creating table Instructor: OK
Creating table Post: OK
Creating table UserLikesPost: OK
Creating table Thread: OK
Creating table DiscussionPost: OK
Creating table Tags: OK
Creating table CourseForum: OK
Creating table Folder: OK
Creating table ThreadInFolder: OK
Creating table UserViewsThread: OK
Creating table UserInCourse: OK
Inserting into User : Success
Inserting into Login : Success
Inserting into PostCreator : Success
Inserting into Student : Success
Inserting into Instructor : Success
Inserting into CourseForum : Success
Inserting into Folder : Success
Inserting into UserInCourse : Success
Inserting into Post : Success
Inserting into UserLikesPost : Success
Inserting into Thread : Success
Inserting into UserViewsThread : Success
Inserting into Tags : Success
Inserting into ThreadInFolder : Success

You have three options:
- Login as Student [1]
- Login as Instructor [2]
- Quit [q]
Please enter your option: 1
```

Figure 1: A user enters their MySQL login information, the TDT4145ProjectGroup131 is created, filled with data, and the user is prompted for login options.

```
You have three options:
- Login as Student [1]
- Login as Instructor [2]
- Quit [q]
Please enter your option: 1

Please enter email: frumford6@ted.com
Please enter password:

You have three options:
- Make a post [1]
- Search for a keyword [2]
- Log out [q]
Please enter your option: 1
```

Figure 2: A logged in student is met with the pictured action menu.

```
You have three options:
- Login as Student [1]
- Login as Instructor [2]
- Quit [q]
Please enter your option: 2

Please enter email: stretters@mashable.com
Please enter password:

You have four options:
- Make a post [1]
- Search for a keyword [2]
- View Statistics [3]
- Log out [q]
Please enter your option: 1
```

Figure 3: A logged in instructor is met with the pictured action menu.

```
You have three options:
- Create a thread [1]
- Create a reply [2]
- Go back [q]
Please enter your option: 1
```

Figure 4: Both students and instructors are met with the following post creation menu if user chooses to create a post.

```
You have three options:
- Make a post      [1]
- Search for a keyword [2]
- Log out          [q]

Please enter your option: 1

You have three options:
- Create a thread  [1]
- Create a reply   [2]
- Go back         [q]

Please enter your option: 1
Please enter your post content: WAL is my favorite search keyword
Please enter your post folder: Exam
Please enter your post tag: Question

PostID of created post: 5e1ebdee-71a5-4fc1-8a8a-25d7b7133ae2

You have three options:
- Create a thread  [1]
- Create a reply   [2]
- Go back         [q]

Please enter your option: 2
Please enter id of post to reply to: 5e1ebdee-71a5-4fc1-8a8a-25d7b7133ae2
Please enter your post content: In my replies I like to include the WAL search keyword

You have three options:
- Create a thread  [1]
- Create a reply   [2]
- Go back         [q]

Please enter your option: q

You have three options:
- Make a post      [1]
- Search for a keyword [2]
- Log out          [q]

Please enter your option: 1
```

✎ *vterm* 195:26 Bot UTF-8 VTerm

Figure 5: A student or an instructor can create a thread and a reply to the created thread.

```
You have three options:
- Make a post      [1]
- Search for a keyword [2]
- Log out          [q]

Please enter your option: 2
Please enter keyword: WAL
Here are your search results:
+-----+
| PostID |
+-----+
| c7e4f3b9-6d5d-48a3-a859-4a2164f9577a |
| 8614abe5-e792-4699-9d7b-95cb162ca4ff |
| 5e1ebdee-71a5-4fc1-8a8a-25d7b7133ae2 |
| 064d8ab1-4537-4f66-930d-9b1ae67aab70 |
+-----+

You have three options:
- Make a post      [1]
- Search for a keyword [2]
- Log out          [q]

Please enter your option: 1
```

✎ *vterm* 213:26 Bot UTF-8 VTerm

Figure 6: A student or an instructor can search for a keyword.

```

You have three options:
- Login as Student [1]
- Login as Instructor [2]
- Quit [q]
Please enter your option: 2

Please enter email: stretters@ mashable.com
Please enter password:

You have four options:
- Make a post [1]
- Search for a keyword [2]
- View Statistics [3]
- Log out [q]

Please enter your option: 3

+-----+-----+-----+
| UserName | NumberOfThreadsRead | NumberOfPostsCreated |
+-----+-----+-----+
| Karry Glenister | 10 | 1 |
| Margaux Belli | 9 | 1 |
| Susanna Gutcher | 8 | 1 |
| Debi Frean | 8 | 1 |
| Fern Nutty | 7 | 1 |
| Hercule Belden | 7 | 0 |
| Jesse O'Neal | 7 | 0 |
| Frasco Loomis | 6 | 1 |
| West Hallut | 6 | 0 |
| Nan Deek | 6 | 0 |
| Tanny Ganniclyfft | 6 | 0 |
| Bastian Portman | 6 | 1 |
| Arlee Sempill | 6 | 0 |
| Louiseette Harrington | 6 | 0 |
| Edita Lamberto | 6 | 0 |
| Oriana Hailston | 6 | 0 |
| Jarrod Tanuzzi | 6 | 0 |
| Pamela Dadd | 5 | 0 |
| Allie Moylan | 5 | 1 |
| Minnaminnie Towse | 5 | 0 |
| Kerrie Freegard | 5 | 1 |
| Evvy Denyagin | 5 | 0 |
| Meaghan Maypes | 5 | 0 |
| Izak Artharg | 5 | 0 |
| Demetria Entwistle | 5 | 0 |
| Phillie Fidelus | 5 | 1 |
| Locke Dewire | 5 | 0 |
| Mil Lovering | 5 | 1 |
| Philly Ralton | 5 | 0 |
| Jacklyn Beviss | 4 | 0 |
| Bret Dimblebee | 4 | 0 |
| Ferrell Hibbart | 4 | 0 |
| Amara Collomosse | 4 | 0 |
| Pierrette Suddards | 4 | 0 |
| Agna Towson | 4 | 0 |
| Carolyne Pidgeon | 4 | 0 |
| Rik Murray | 4 | 0 |
| Orton Sibborn | 4 | 0 |
| Bail Jenton | 4 | 2 |
| Suzi Petchey | 4 | 0 |
| Ulric Bothe | 4 | 0 |
| Kerri Vicker | 4 | 0 |
| Dolli Woodall | 4 | 0 |
| Keene Jillis | 4 | 0 |
| Laurene Cordy | 4 | 0 |
| Goldie Maccrow | 4 | 0 |
| Lelia Iscowitz | 4 | 0 |
| Suzanna Fairbairn | 4 | 1 |
| Tonnie Vasin | 4 | 0 |
| Whitaker Payn | 4 | 0 |
| Aleda Riordan | 3 | 1 |
| Hillel Bedbury | 3 | 0 |
| Prent Osbidston | 3 | 0 |
| Eloise Gansbuhler | 3 | 0 |
| Gerard Baszkiewicz | 3 | 0 |
| Lorna Meatyard | 3 | 0 |
| Helli Tukely | 3 | 0 |
| Jedediah Filpo | 3 | 0 |
| Kile Stirtle | 3 | 1 |
| Jake Gother | 3 | 1 |
| Giraldo Burleigh | 3 | 0 |
| Farra Sawden | 3 | 0 |
| Lucio Matkin | 3 | 0 |
| Terese Esposi | 3 | 0 |
| Neville Arrighi | 3 | 1 |
| Toby Leech | 3 | 0 |
| Alsece Von Salzburg | 3 | 1 |
| Virgilio Stickings | 3 | 0 |
| Gabriel Brittan | 2 | 0 |
| Flinn Diggar | 2 | 0 |
| Gaylord Freathy | 2 | 1 |
| Odille Herrievan | 2 | 0 |
| Brunhilde Tregaskis | 2 | 0 |
| Gloria Casarini | 2 | 0 |
| Christoforo Strephan | 2 | 0 |
| Thoma Hedingham | 2 | 0 |
| Mordecai Margett | 2 | 0 |
| Elisa Tefft | 2 | 0 |
| Pattie Hosier | 2 | 0 |
| Roma Covell | 2 | 0 |
| Claus Ziebart | 2 | 0 |
| Elmore Wadeling | 2 | 1 |
| Chrotoem O' Hanvey | 2 | 1 |
| Adria Vousden | 1 | 0 |
| Gerianna Beaumont | 1 | 1 |
| Tedi Gwinn | 1 | 0 |
| Faustine Brenley | 1 | 0 |
| Augustina O'Cullinane | 1 | 0 |
| Merna Mounter | 1 | 0 |
| Penrod Chesher | 1 | 0 |
| Sayre Tretter | 1 | 0 |
| Benita Tierney | 1 | 0 |
| Henrik Durdle | 1 | 1 |
| Tommie Christol | 1 | 0 |
| Ruy Doran | 1 | 0 |
| Stormy Boyat | 1 | 0 |
| Munroe Litzmann | 1 | 1 |
| Marylynne Letterese | 0 | 0 |
| Renie Reeks | 0 | 0 |
| Flore Rumford | 0 | 2 |
+-----+-----+-----+

You have four options:
- Make a post [1]
- Search for a keyword [2]
- View Statistics [3]
- Log out [q]

Please enter your option: q

You have three options:
- Login as Student [1]
- Login as Instructor [2]
- Quit [q]
Please enter your option: q

Bye!
(py3) + piazza git:(main)

```

Figure 7: An instructor can view user statistics.