Group Assignment Topic

Semester 2, Session 2021/2022

Topic C: Confess Time!

We play many different roles and wear many different hats in life. For example, John can be a dad, a lecturer, and a researcher. Having all these dueling identities can be a stressful ordeal and finding a place to voice out bottled inner thoughts is definitely a good way to release all the stress. But therapy is expensive. Fortunately, students of today have access to a budget crowd-sourced version called confession pages. Here, you can employ the help of your fellow stressed out students to solve your problems. For free!

The confession page is also a good place to voice out what needs to be said. People ask or talk just about anything on confession page, whether are life related questions, homework, or even career suggestion. And even sometimes, people even confess to their crush.

A confession was thus intended as a community thing where people come together to speak and admit. As we can no longer come together safely due to the pandemic, we must now do so through virtual means via the power of the internet.

1.0 Problem Statement

Now, after reading the introduction, you as a software developer, decided to create a confession page as a contribution to the world. We have prepared the requirements for you to start off:

- Create a confession page simulator application in any form (CLI, GUI desktop, web, or mobile apps).
- You are required to implement some data structures & algorithms to achieve requirements stated below.
- Example images shown below are for references only, you may design your own interface.

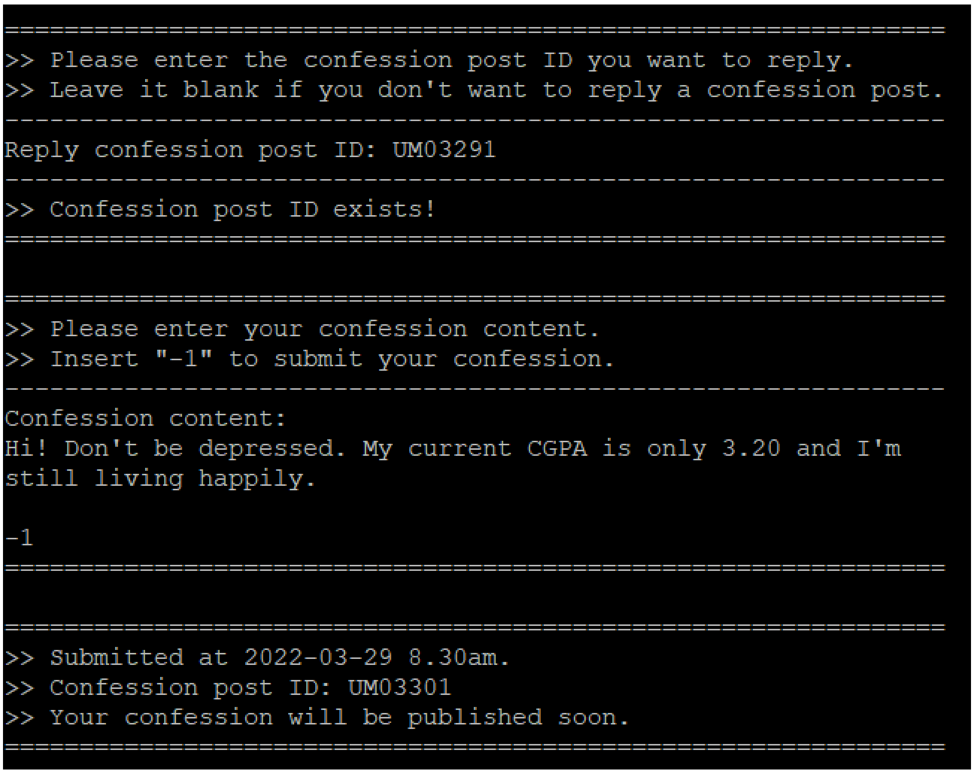
2.0 Basic Requirements

You are required to complete all the basic requirements stated below and build a confession application that allows users to submit confession post. The program must be interactable and supports user operations such as navigating into different parts of the program and prompts for user input. More details about each requirement are stated below.

2.1 Main Application Features

2.1.1 Submit Confession Post

Create an input interface which allows users to submit their confessions. The sample output for confession post submission is as shown in Figure 1.



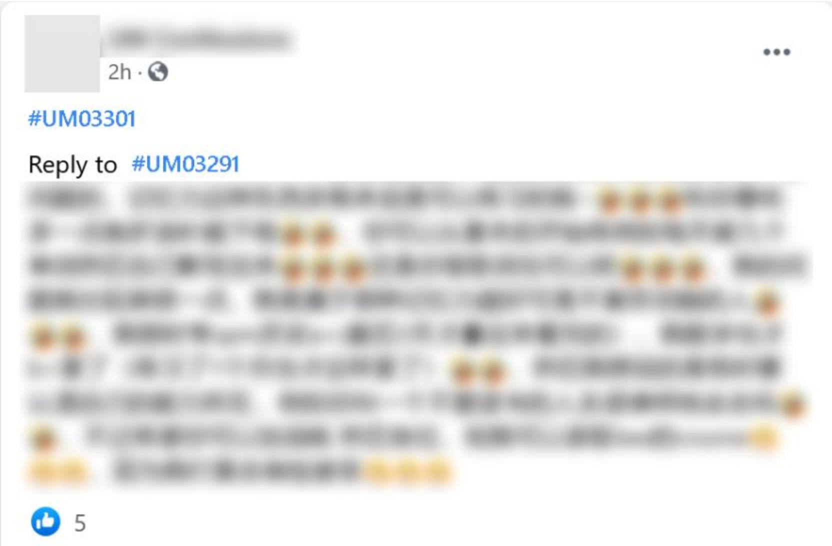
```
===============================================================
>> Please enter the confession post ID you want to reply.
>> Leave it blank if you don't want to reply a confession post.
---------------------------------------------------------------
Reply confession post ID: UM03291
---------------------------------------------------------------
>> Confession post ID exists!
===============================================================


===============================================================
>> Please enter your confession content.
>> Insert "-1" to submit your confession.
---------------------------------------------------------------
Confession content:
Hi! Don't be depressed. My current CGPA is only 3.20 and I'm
still living happily.

-1
===============================================================


===============================================================
>> Submitted at 2022-03-29 8.30am.
>> Confession post ID: UM03301
>> Your confession will be published soon.
===============================================================
```

Figure 1: Sample output for post submission.

There will be two main input fields for the user:

a) Reply confession post ID

One confession post only can reply to one published confession post. However, one published confession post can be replied by many confession posts. If the user leaves the reply confession post ID input field blank, the confession he/she submitted will not be replying to any published confession post. Figure 2 shows the sample of a real-world confession page to help you understand the concept of replying.



2h ·

#UM03301

Reply to #UM03291

5

Figure 2: Example of replying to a confession post.

This input field is optional, users either can leave it blank or enter another existing post ID. You can also implement a search function for the user to search for a published confession post ID so that they can reply to an existing confession post with a new confession post easily.

b) Confession content

This field is mandatory and must be filled in by the user upon creating a post. The user can fill in one or many sentences in different lines and the program should stop capturing when the line starts with "-1" (refer Figure 1).

Once the users have submitted their confession, a message should be displayed to show that their confession will be published soon. The message should also include the confession post ID and the submission date and time as well.

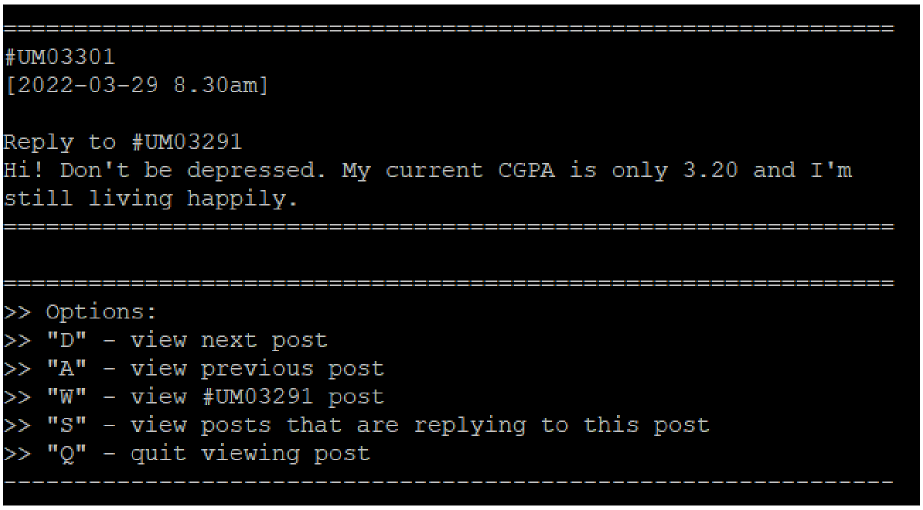
2.1.2 View Published Confession Posts

Create a display interface so that the users can view the published confession posts. The display interface should show the confession posts starting from the latest. The contents of a published confession post must include:

a) Its confession post ID
b) Published date & time
c) Reply confession post ID (if exist)
d) Confession content

**Order of the content element are not important.*

Figure 3 shows the sample output for viewing confession post, where the output should contain all the above-mentioned contents.



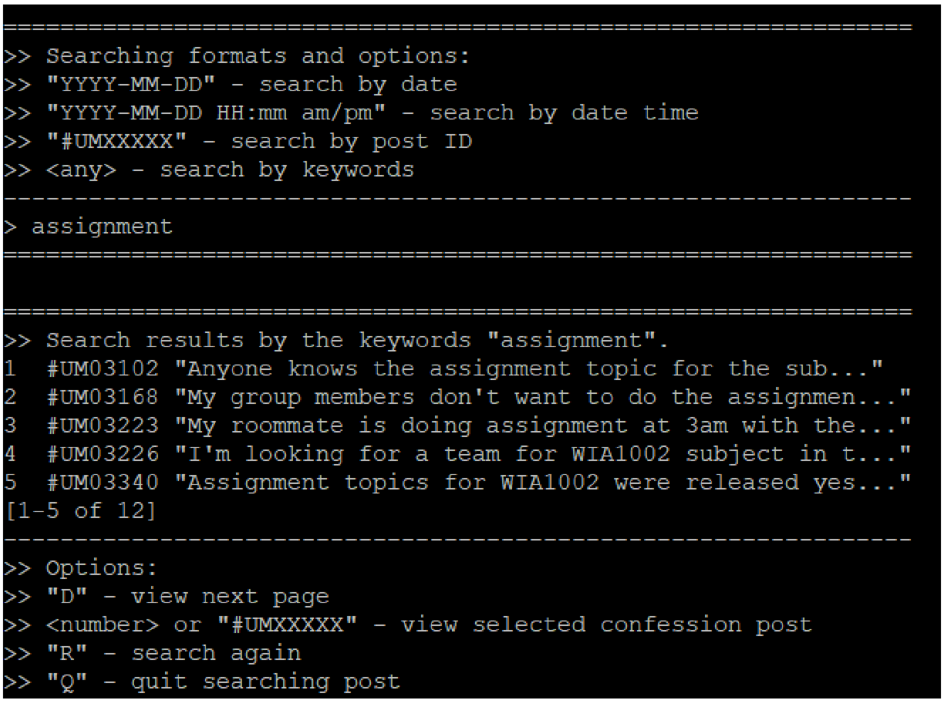
```
===============================================================
#UM03301
[2022-03-29 8.30am]

Reply to #UM03291
Hi! Don't be depressed. My current CGPA is only 3.20 and I'm
still living happily.
===============================================================

===============================================================
>> Options:
>> "D" - view next post
>> "A" - view previous post
>> "W" - view #UM03291 post
>> "S" - view posts that are replying to this post
>> "Q" - quit viewing post
---------------------------------------------------------------
```

Figure 3: Sample output for viewing post.

Besides, if a published confession post has a reply-to confession post or has other confession posts replying to this published confession post, there should be the options for the user to view the reply-to confession post or the replying confession posts.

2.1.3 Search Published Confession Posts

Create another interface/input field where the user can search for the published confession posts based on keywords, confession post ID, published date or publish date and time. The returned search result may contain multiple confession posts. The following figure shows the sample output when the user searches for the keyword "assignment".



```
================================================================
>> Searching formats and options:
>> "YYYY-MM-DD" - search by date
>> "YYYY-MM-DD HH:mm am/pm" - search by date time
>> "#UMXXXXX" - search by post ID
>> <any> - search by keywords
----------------------------------------------------------------
> assignment
================================================================

================================================================
>> Search results by the keywords "assignment".
1  #UM03102 "Anyone knows the assignment topic for the sub..."
2  #UM03168 "My group members don't want to do the assignmen..."
3  #UM03223 "My roommate is doing assignment at 3am with the..."
4  #UM03226 "I'm looking for a team for WIA1002 subject in t..."
5  #UM03340 "Assignment topics for WIA1002 were released yes..."
[1-5 of 12]
----------------------------------------------------------------
>> Options:
>> "D" - view next page
>> <number> or "#UMXXXXX" - view selected confession post
>> "R" - search again
>> "Q" - quit searching post
```

Figure 4: Sample output for searching post.

2.2 Admin Panels and System

2.2.1 Waiting / Queuing list

If you have submitted a confession on a Facebook confession page or other social media platform before, you might realize that your confession post took time to be published. There are 2 main reasons behind this:

i. Confession page admins need time to review or check your confession content.
ii. Admin does not have time to post all the submitted confessions.

In your case, you are required to simulate and automate this process with some predefined rules:

From submission to publishing flows

1. User submits a confession through the input interface.

2. Program check for content spamming (more details in next section).
3. The confession data is pushed into a data structure.
4. Waiting time elapsed.
5. The confession data is popped from that data structure.
6. The popped confession data is stored in the program permanently (meaning that the confession post has been published).

Rules & Requirements:

1. You need to create a data structure to store the pending confession post.
2. Make sure that this data structure supports First In, First Out (FIFO) operations.
3. You need to implement a scheduling process:
    a. If the number of elements in the data structure is less than or equals to 5, pop the data every 15 minutes.
    b. If the number of elements in the data structure is less than or equals to 10, pop the data every 10 minutes.
    c. If the number of elements in the data structure is more than 10, pop the data every 5 minutes.

2.2.2 Spam Checking

Some mischievous users will try to spam their confession, either repeatedly submitting similar confession content or repeatedly submitting meaningless content. To avoid this kind of misbehavior, you are required to use at least one algorithm with appropriate data structures to implement your solution to block their confession submission.

2.2.3 Batch Removal

Sometimes, there exist some posts that violate the terms and conditions in the platform. For example, a post that is too controversial and lead to unnecessary argument in the community. The admin will need to remove all the related posts as soon as possible. Design your program to allow batch removal, where given a confession ID, delete all the posts that are replying to this post either directly or indirectly (refer image below), and remove itself as well.

For instance, with referring to Figure 5, if the post ID #UM3002 is post that needs to be removed, you will need to delete the entire blue section.

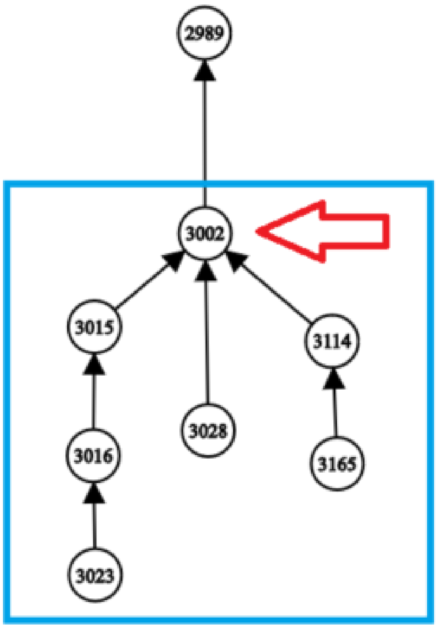


Figure 5: Example of batch removal to remove the post.

2.2.4 Save and Load Files

A good confession page should hold the posts as long as possible. Hence, a permanent storage solution is a must in your program. You're required to save all the confession posts data locally (or cloud-based if you decided to do so) before exiting the program, so that you can load the data back when you restart your application. Make sure to take care of the pending posts and their respective orders to be published.

3.0 Additional Challenges / Extra Features

I. GUI

CLI programs are useful when dealing with simple and straight forward purpose. However, application like confession page that allows constant user interaction should be developed with Graphical User Interface (GUI) instead.

II. Vacation Mode

Vacation Mode. The moderators want to rest sometimes, vacation mode will automatically approve / disapprove a submitted post based on the sentiment analysis. For example, a post with a neutral or good tone would be automatically approved, meanwhile a post with a bad tone (rude words / harassment / blackmail/ racist and many more) will be automatically rejected. You may also ban the user permanently to prevent further submission.

III. Facebook API Integration

Operate the application on Facebook. You can use Facebook API to help manage your new page on Facebook.

IV. More than Text

Contents other than text. A good confession page should allow the users to upload not only text contents, but media files too. Images and videos are the most common things that you can find on a confession page. If you are to implement this feature without coding your program in GUI, you can think of a way to launch the media files externally through your program.

V. Always Think Out of the Box

Anything that you think is interesting and worth trying.

4.0 Others

4.1 Marking

Kindly refer to the assignment instruction and marking rubrics for more information.

4.2 Tips

Here are some tips to help you in completing this assignment:

- Always plan ahead in terms of work distribution among team members. As you have already known, this is a group assignment, hence you're expected to work in a team and present the outcome together.
- Use code tracking tools such as Git or SVN to better align the codes among your team members.
- Give meaningful comments in your code. This will help the readers (your team members and lecturer) to understand your logic easier and faster, reducing the amount of time needed to figure out the whole picture.

4.3 Enquiries

If you have any questions regarding the topic, please check on this FAQ page (which will be constantly updated). You may also write an email to helpdeskdatastructure@gmail.com with the title (Confess Time!).

```
### end of assignment topic description ###
       ## good luck and have fun! ##
```