WIA1002/WIB1002 DATA STRUCTURE

TOPIC C: CONFESS TIME

LECTURER'S NAME: DR. MOHD HAIRUL NIZAM BIN MD NASIR

GROUP OCCURRENCE: 5

GROUP NAME: PSVM

Name	**No Matric**
Muhammad Danial Haikal Bin Mastirana	U2001426
Wan Amir Imran Bin Wan Azezi	U2000911
Megat Faris Putra Bin Megat Mehamat Kaharrudin	U2001086
Leonardo Laiherman	S2105894

Table of Content

1. Introduction

People with many identities can be stressed because it is tough for them to voice bottled inner thoughts since they need to keep professional on social media. To solve this problem, we need to create or design a confession application for the user to confess their feelings. All their identities will be kept secret as we need to care about their privacy so that the person can confess things anonymously. Confession is an interface for people to come together to speak and admit. For example, people use confession apps or pages to release stress, ask for advice, and birthday wishes, and confess to their crush and others. To ensure our confession project is done perfectly, we implement some data structures & algorithms that we learned from the lectures to create this confession app systematically.

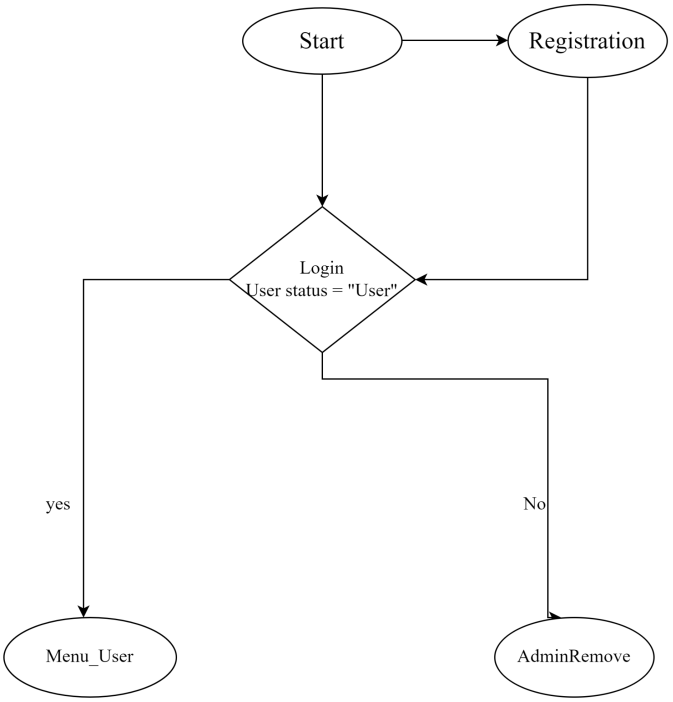
This confession app is very user-friendly so that every person of every age can use this app in the easiest way to confess what's in their mind. We make it user-friendly by implementing fewer buttons on every page, so the users don't have to explore the functionality of a button, as the buttons have different functions. Also, we put some moving pictures as a background for every page to make it more interesting.

2. Member and Responsibility

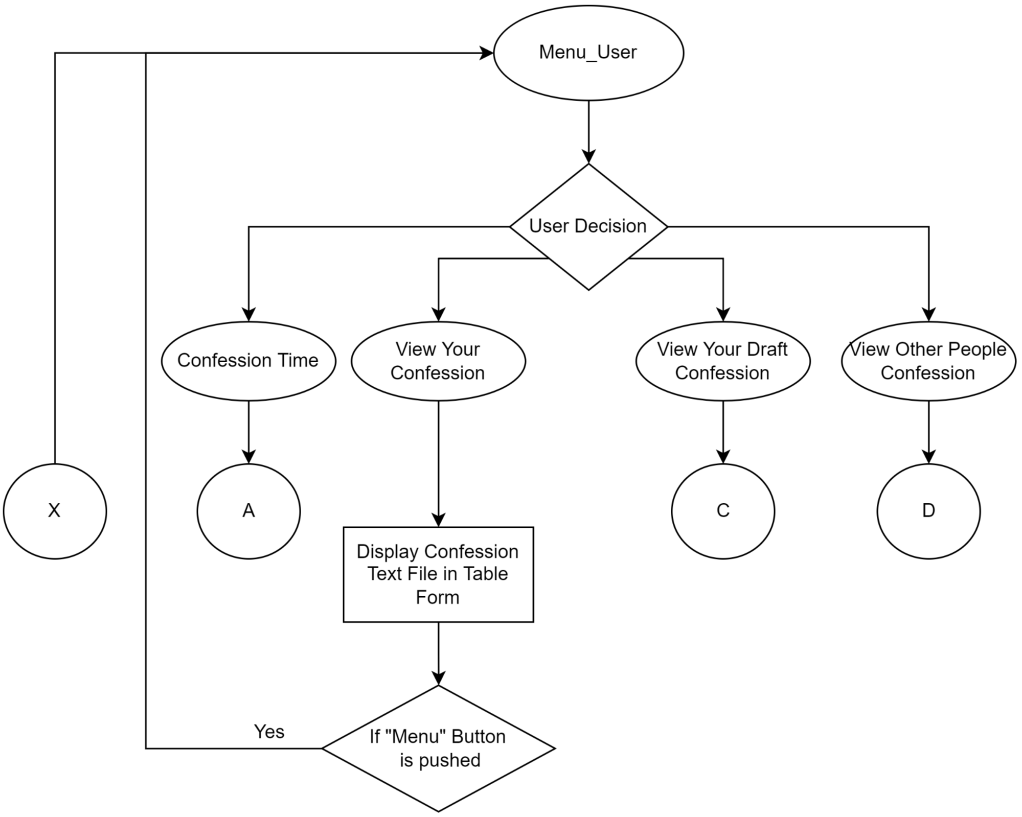
Name	No Matric	Responsibility
Muhammad Danial Haikal Bin Mastirana	U2001426	Project Leader//Backend Programmer//Main Debugger
Wan Amir Imran Bin Wan Azezi	U2000911	Backend Programmer// Debugger//Report Writer
Megat Faris Putra Bin Megat Mehamat Kaharrudin	U2001086	Frontend Programmer//Debugger//Report Manager
Leonardo Laiherman	S2105894	Programmer // Report Writer

3. Project Flow Chart

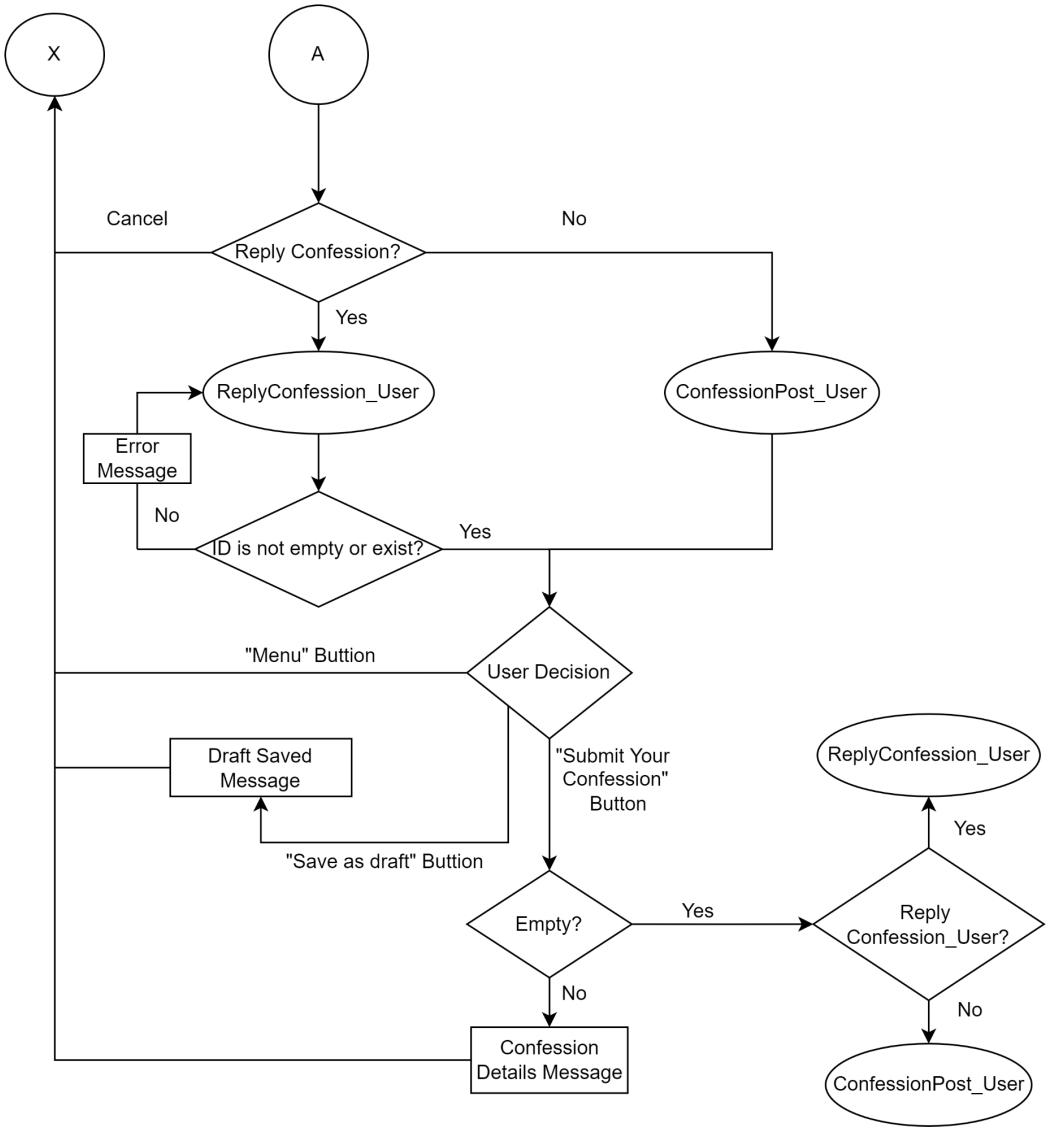
- Login Algorithm Flow Chart



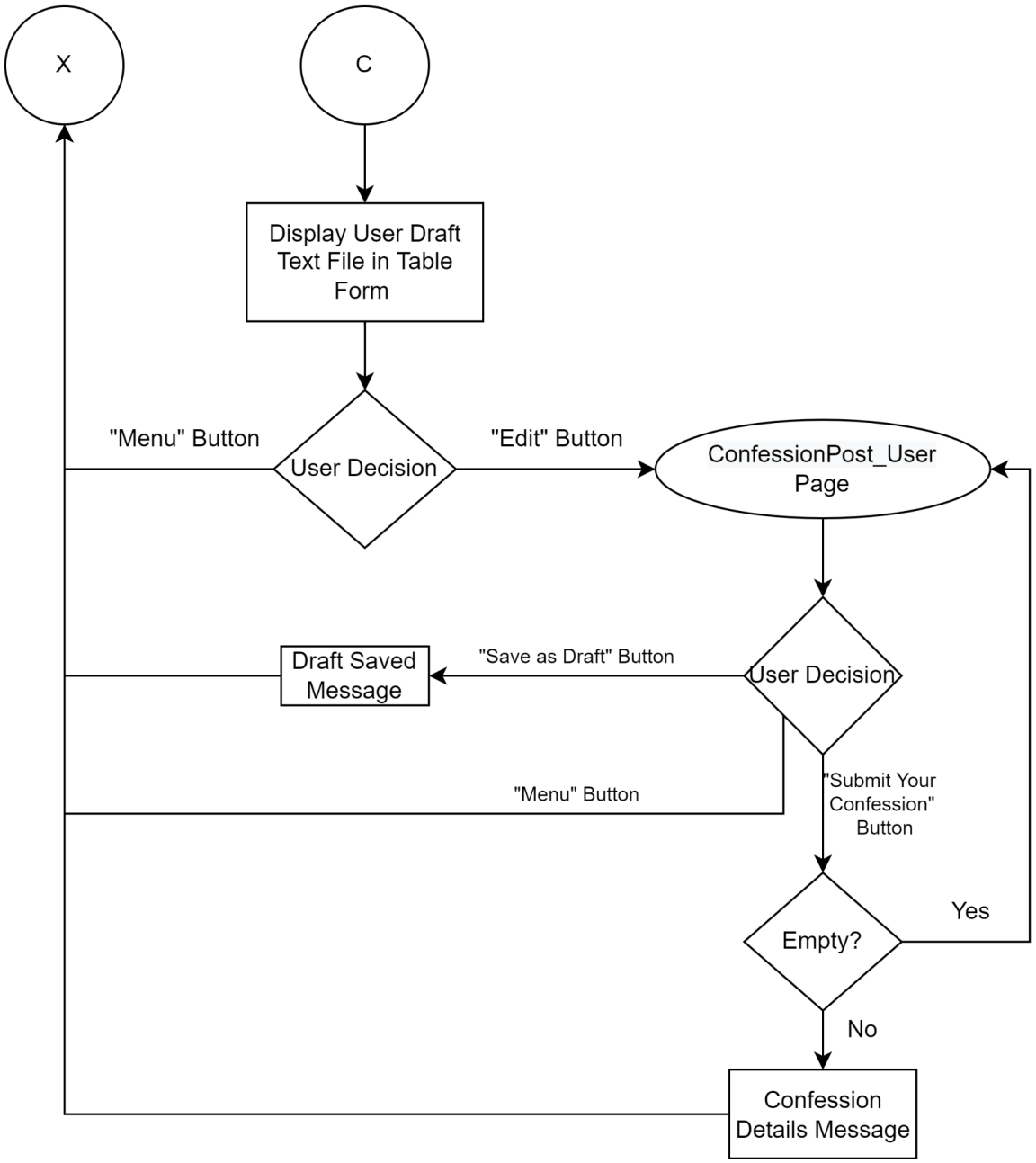
- User Main Menu Algorithm Flow Chart



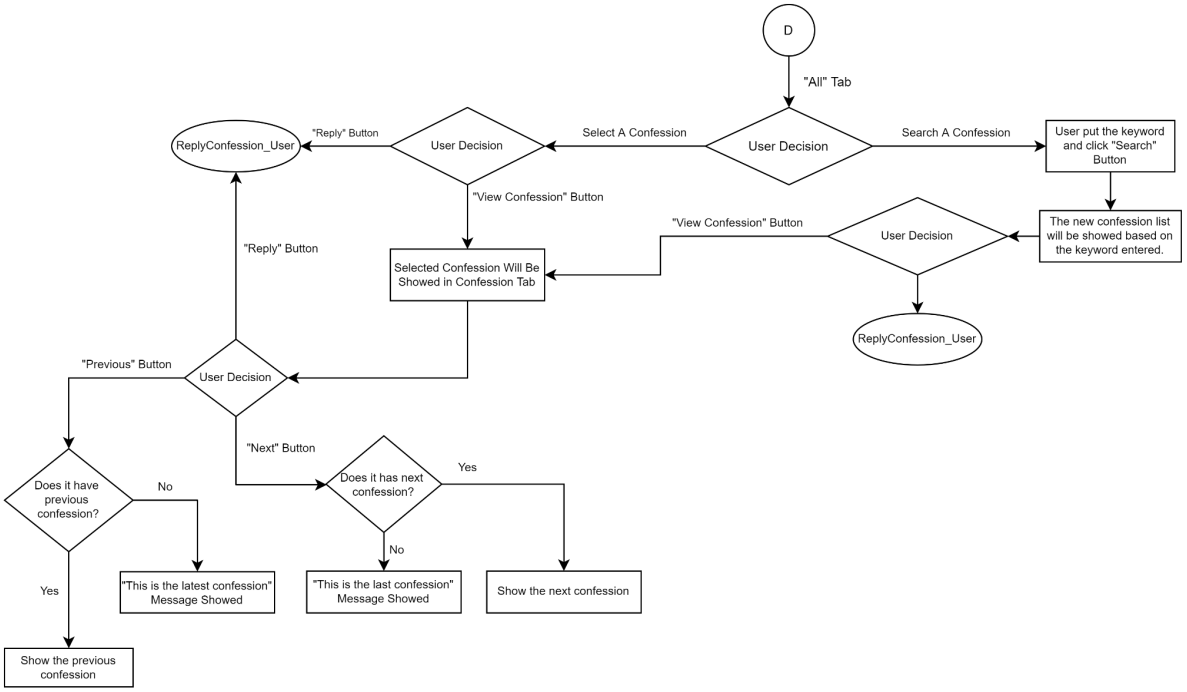
- Confession Time Algorithm Flow Chart



- View Confession Draft Algorithm Flow Chart



- View Other People Confession Algorithm Flow Chart



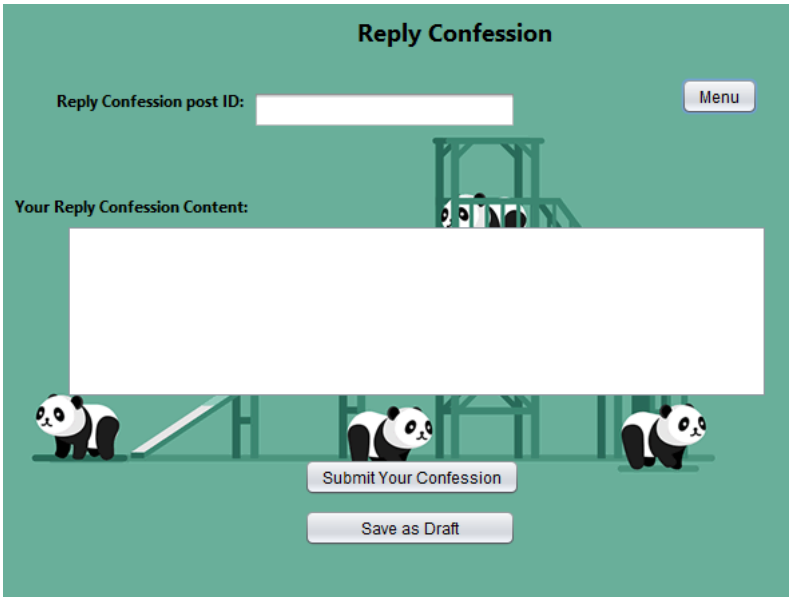
4. Basic Requirements (explain and screenshot the solution)

- Submit Confession Post

  For this part, a window pops up asking whether we want to reply to another confession post, create a new confession post or cancel it. For creating a new confession post, there will be a JTextField where the user can write anything they want to confess. Then, the user can either submit it as a draft or post it as a confession. Before the confession is submitted, it must pass the spam checking first. If the confession is detected as spam, the user must enter the message again before submitting it. If successful, it will show us the time when we submit the post and the post ID

.

For creating a confession post

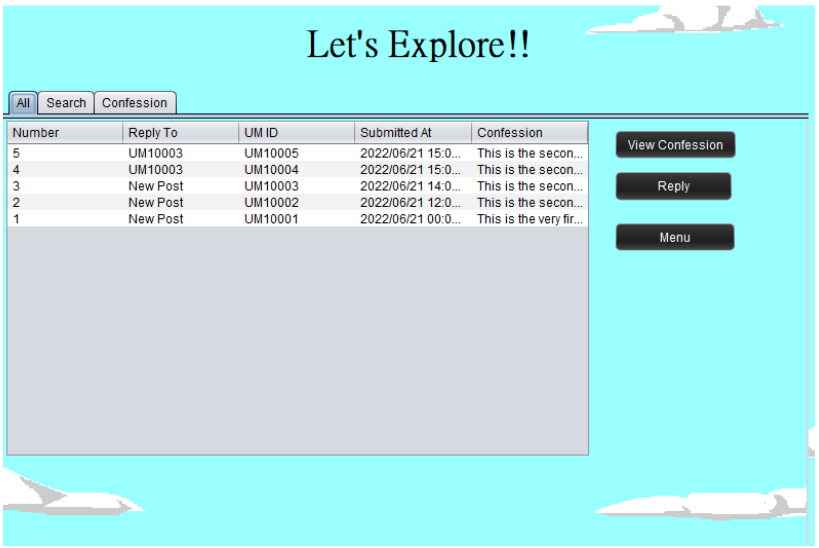


For replying a confession post

- View Published Content

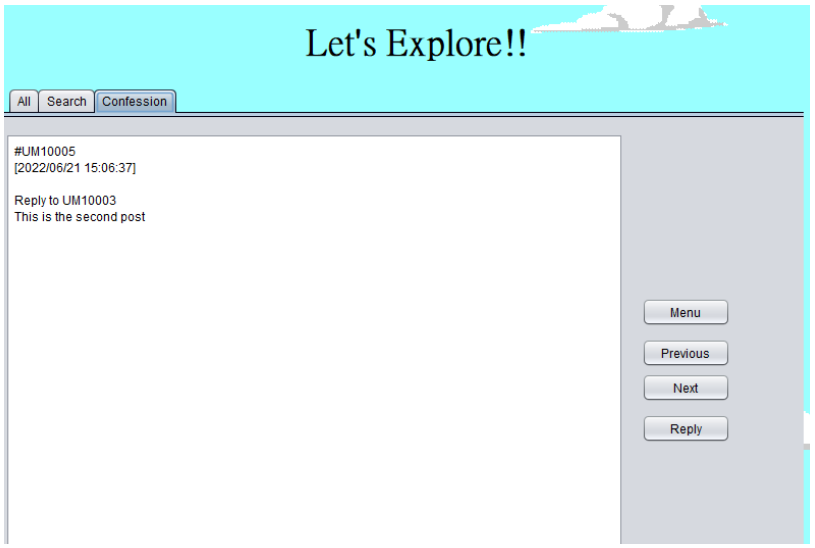
  Here, we can see three panels: All, Search, and Confession. There will be three buttons for the All and Search tabs. The first one is for viewing the confession. By clicking on one of the rows and clicking the view confession, it will direct us to the last tab, and the confession post will be shown to us. The second one is to reply to the post. The program will later direct us to the Replying Confession page, where the user Post ID will automatically be on the Reply Confession Post ID. And the last one is for the user to go back to the user menu, where we can choose the other app function.

  For the All tab, we can see all of the confessions that have been published. The list on the table will be inverted as we want to show the latest post to the user. To achieve this, we used Stack to store the information from the Confession.txt file and then push it into SortedConfession.txt to make the data be inverted. In this tab, the user can see the specific information of the confession post, like which post ID is the confession post replying to, the confession post ID when it was submitted, and also the content of that specific confession.



For viewing all of the confession that has been posted

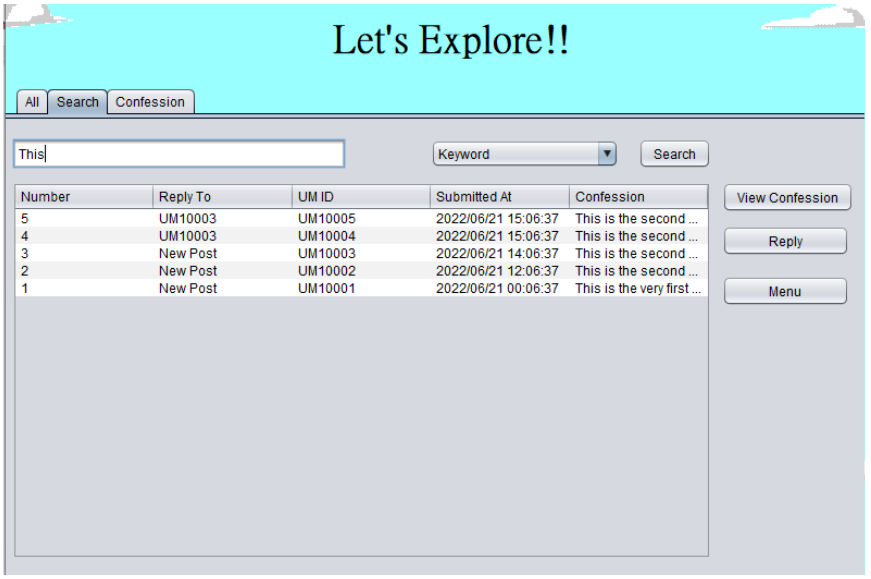
For the confession part, the user can use the previous button to go up in the confession post, or the next button to go to the older confession post. When it arrive at the latest page, the previous button will pop up a window with notification “This is the latest confession post”, meanwhile at the oldest post, the next button will pop a window with notification “ This is the last confession post“



This is for viewing only one confession post at each time

- Search Published Confession Posts

  For the Search tab, the user can use the combo box to 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.



This is for searching by using certain condition

- Waiting / Queueing List

  For the waiting list, we used java.util.Timer and also java.util.TimerTask to schedule tasks in Java and form a Timer API. So, with this we can make the waiting list work.

  Later, we will enqueue all the information in the Queue.txt file into a queue so that we can implement the First In First Out (FIFO) operation If the number of elements in the queue is less than or equal to 5, pop the data every 15 minutes. If the number of elements in the data structure is less than or equal to 10, pop the data every 10 minutes and if the number of elements in the data structure is more than 10, pop the data every 5 minutes.

- Spam Checking

  For spam checking, the algorithm that we use is comparing two strings. The program will first get the length of both strings and then the decimal number of each character in the string based on the ASCII table. We realise that if the sum of the number is the same, it could be considered spam. Therefore, from the total number the string has, we can calculate the similarity percentage of both strings. For this project, we set the criteria for spam if the similarity percentage is more than 80%.

- Batch Removal

  For the batch removal, we used ArrayList and Linear Search to delete a confession post and all the confession posts that relate to it.

  For the ArrayList, we used it to store the postID and reply postID that we want to delete. By using Linear Search, we will check line by line in the Confession.txt file (in which the confession information is stored) and then save the line that does not have neither postID nor reply postID in its line to a temporary file. The ArrayList will help us in checking if it contains the deleted post ID by using the contains() function.

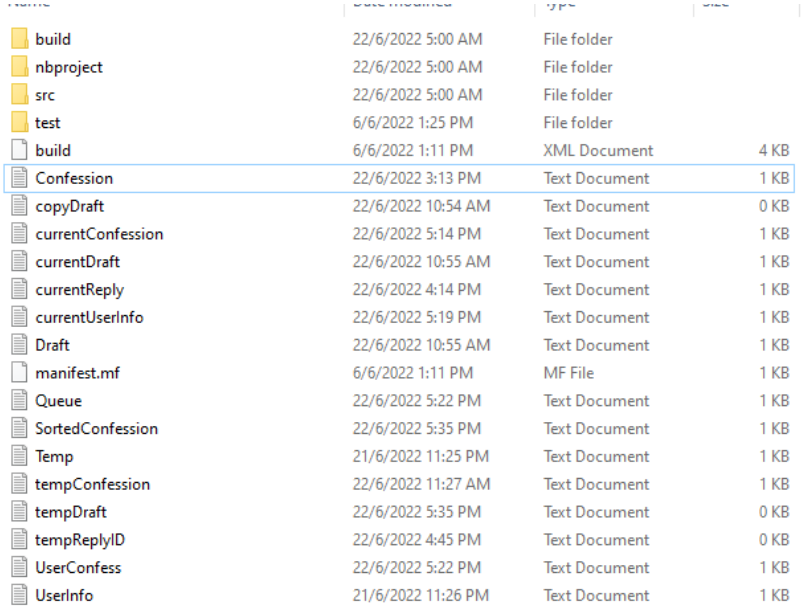
  After the checking is done, we will rewrite the Confession.txt file using the temporary file. Later, the confession post that we already deleted will not be saved. Thus, we can do the batch removal by using this method.

This is the GUI and the Confession.txt before batch removal

This is the GUI and the Confession.txt after batch removal

- Save and Load Files

For this project, we had used many text files as our database. So, we could easily access the files from anywhere inside the program to be used for certain actions in our program. The information saved in the text file each has a different format depending on how intended to access the files. The information saved will not be lost, so we can reuse the information for later time in the project even after we close the program.



These are the text files that we used in our project.

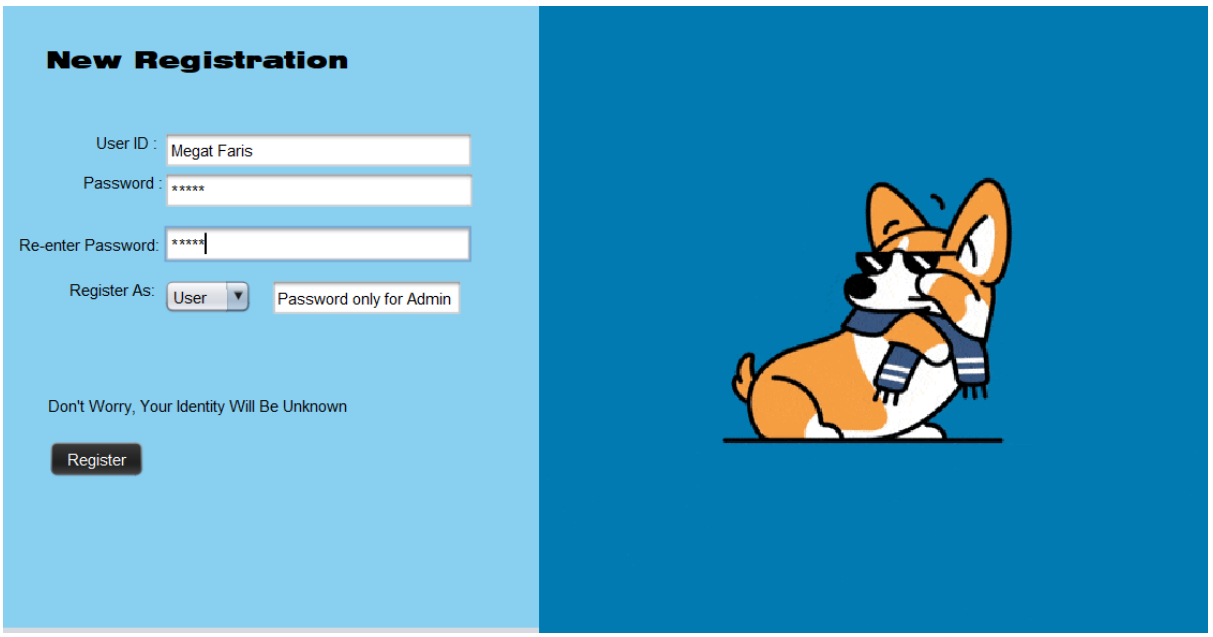
5. Extra Features

- Graphical User Interface(GUI)

  In order to make this app more user-friendly, we add the Java Swing which is GUI's toolkit that includes the GUI components to the program. The Java Swing library is built on top of the Java Abstract Widget Toolkit (AWT). All components in Java Swing are JComponent which can be added to the container classes.

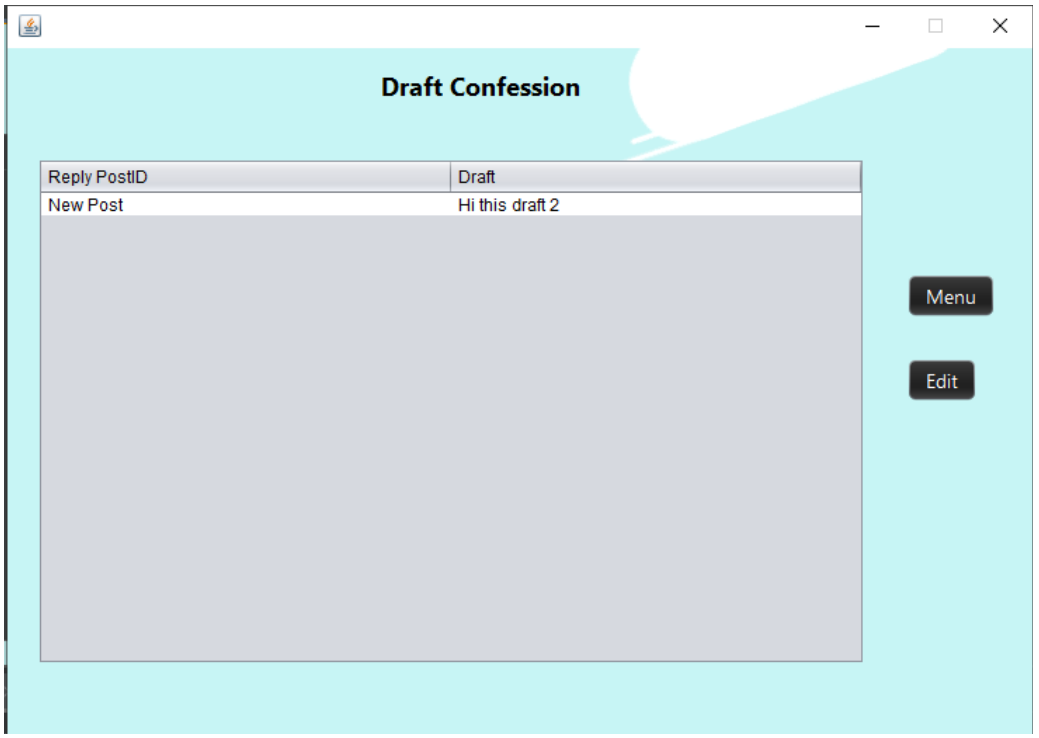
- Registration

For a new User, they need to pick their unique User ID and set a password. They also need to re-enter their password to double-check the password they entered just now. If the user ID that they picked already exists, a message will be displayed saying that the user id has already been taken. The same goes for the Admin registration.The one and the only difference between user and admin registration is that the admin needs a password to complete the registration for an admin. Upon the completion of registration for both user and admin, their user ID and password will be stored in their text files respectively.



- Draft

For the draft confession page, users can select the draft they saved and edit it. The ”edit” button will bring users to the page for submitting a confession post page. Then, the user can decide it there whether they want to save it as draft or publicise it.

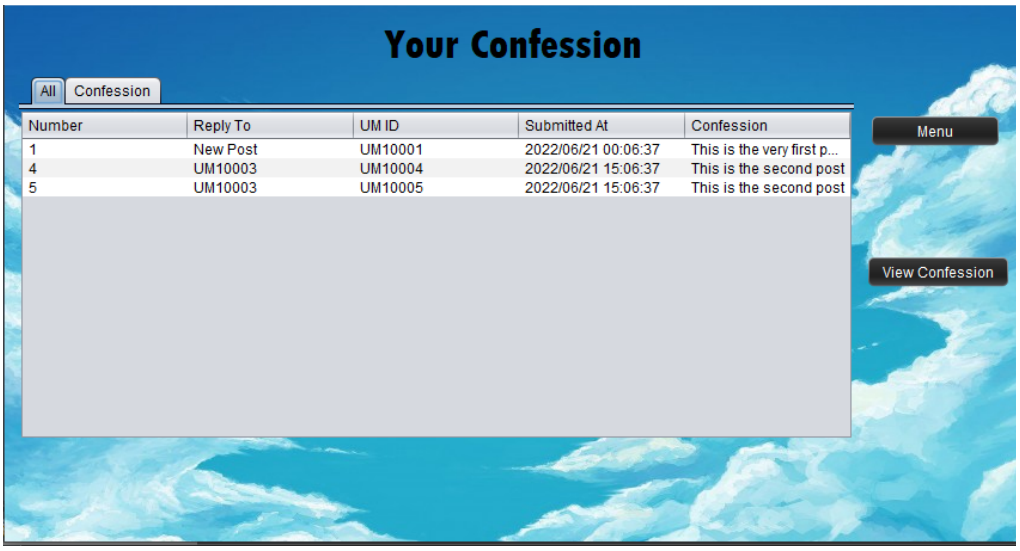


- Login

For the login page, the user will enter the user ID, their password, and the status of whether they entered as the user or Admin. If they do not have an account yet, they can click the Register Now button for the new user to register. In order to give the user some protection, we provide the login feature that allows everyone to open their account. This feature is also helpful since we provide features like draft and view your confession. The login feature will prevent people from seeing your personal confession.

- View Your Confession

  Here, we can see the previous confession of the user that logged into the program. All of the confessions that have been stored can be accessed by searching for the specific user ID. So, the user can use this feature to recall what they have confessed before.



6. Conclusion

In conclusion, the task we were assigned in this assignment was relatively more challenging than the usual tutorial lab we have done before. It is because the project needs to implement many data structures. Plus, the smoothness of our project progress is not so good because we can't do a physical meeting as our group members live in different places.

Even so, we discussed our project frequently through the video call via Microsoft Teams and kept up with each member's work updates via WhatsApp group chat. Thankfully, our members were full of spirit and always did their work on time. As a result, we can search for the code bugs to prevent any error that might have us emotional damage later when we compile and run the project.

Despite this, there were some challenges we faced during this project. Firstly, it was hard for us to share the codes as we didn't create an online code repository for our codes to update the codes online. Instead, we just sent a zip file repeatedly in the WhatsApp Group Chat whenever one of us wanted to update the codes to other group members. Also, we encountered problems in implementing Java Swing at first because the output was not as we wished it to be. Hence, we spent a lot of time figuring out the problems and solving them by referring to resources such as websites and YouTube tutorials. Then, we implement what we have learned and debug the codes so that the output is the same as we initially planned.

In the end, we gained a lot of new knowledge while completing this project, such as implementing data structures in real-life applications, Java Swing as Graphical User Interfaces, and text files as a database.

7. References

1. GUI (Java Swing) :
    - *Java Swing Tutorial - javatpoint*. www.javatpoint.com. (n.d.). Retrieved June 22, 2022, from https://www.javatpoint.com/java-swing

2. Data Structures:
    - Liang, Y. D. (2019). *Introduction to java programming: Brief version*. Pearson Education Limited.
    - *Data Structure and algorithms tutorial*. Tutorials Point. (n.d.). Retrieved June 22, 2022, from https://www.tutorialspoint.com/data_structures_algorithms/index.htm