Progress Report #6	
Course Code: CPE201L	**Program:** Computer Engineering
Course Title: Data Structures and Algorithm	**Date Performed:** October 18 , 2025
Section: CpE - 2A	**Date Submitted:** October 18, 2025
Members: Balana, Jerkielle Roen O. Balaoro, Judge Wayne B Barbas, Steven Jade Dispo, Lei Andrew Sorellano, John Kenneth	**Instructor:** Engr. Maria Rizette Sayo

1. Objectives

- Create a file-based system that allows multiple users to use the app at the same time without issues.
- Add a file-backed queue that saves its data in a JSON file, so nothing gets lost even if the app closes or restarts.
- Make the "Active Now" user count more accurate by tracking when users were last active and automatically removing inactive ones.
- Improve the user interface by showing a live view of the current message queue, so users can easily see how the FIFO process works in real time.

2. Discussion

In this progress report, we fixed a major problem to make the chatbot work better for multiple users. The last system that we made last week was complicated and lost all waiting messages if the app restarted. We replaced it with a simple, stable system that saves the message queue to a file, so nothing is lost.

After we fix the current problem, the new version now uses a JSON file that saves every messages of the users that means the queue data isn't deleted and everyone shares the same queue list. We also improved the "Active Now" to be more accurate in seeing the current number user using the chatbot.

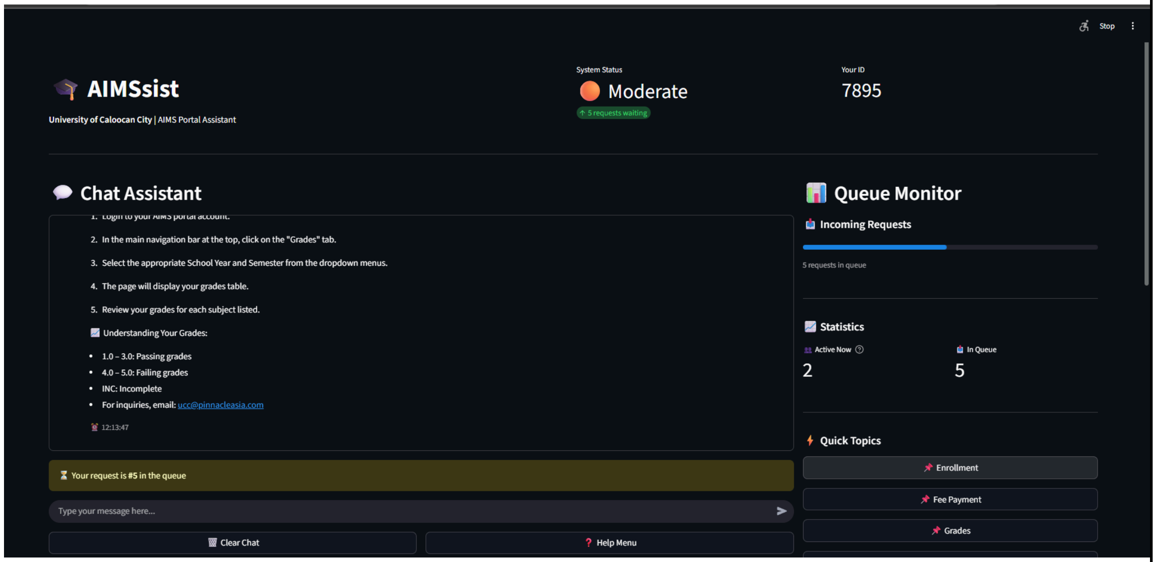
3. Materials and Equipment

- PyCharm
- Streamlit
- Python
- Github

4. Procedure

1. Replaced the architecture by removing complex background threads and switch to more stable file-based system to handle the queue.
2. Making the Queue persistent by implementing a save and load the entire queue state to a JSON file.
3. Improved the “Users” to “Active Now” to provide an accurate count of currently online users.
4. Enhancing the UI by removing unnecessary buttons and text and added a “Current Queue” display to monitor panel so that all users can see a live view of their queue while waiting in line.

5. Output



2. In the main navigation bar at the top, click on the "Grades" tab.
3. Select the appropriate School Year and Semester from the dropdown menus.
4. The page will display your grades table.
5. Review your grades for each subject listed.

Understanding Your Grades:

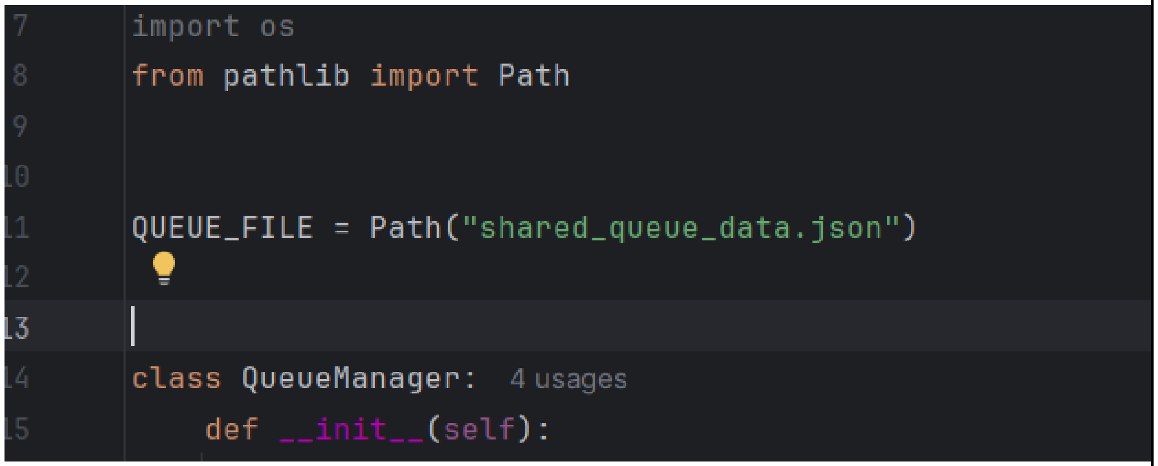
- 1.0 – 3.0: Passing grades
- 4.0 – 5.0: Failing grades
- INC: Incomplete
- For inquiries, email: ucc@pinnacleasia.com

12:13:47

Your request is #5 in the queue

Figure 1. True Multi User Functionality

When the application is accessed through multiple devices in the same network for now. All users interact with the exact same queue and it will show the message from one device's current queue**.**



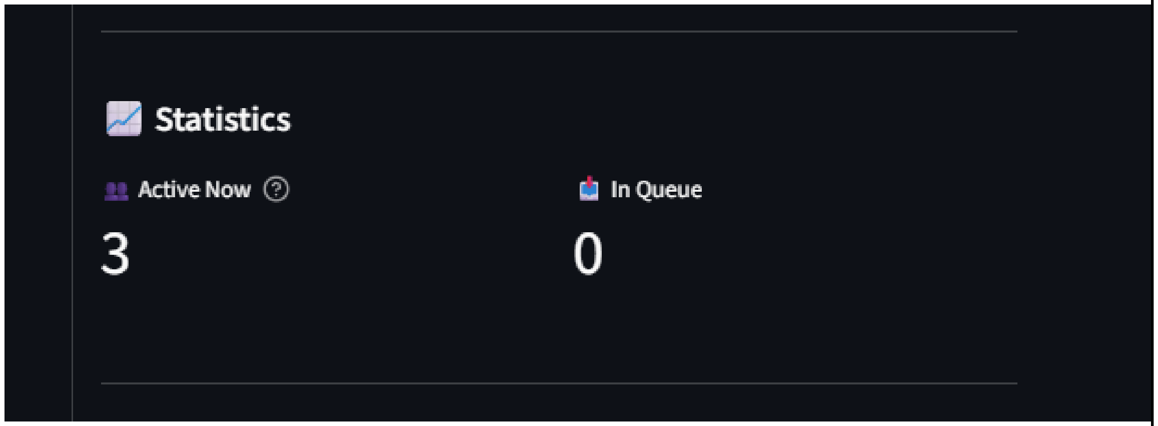
```python
import os
from pathlib import Path


QUEUE_FILE = Path("shared_queue_data.json")


class QueueManager:  4 usages
    def __init__(self):
```

Figure 2. Persistent State

The shared_queue_data.json file stores the entire queue. If the Streamlit is ever stopped and restarted, any message that are waiting in queue are reloaded and will continue to process.



Statistics

Active Now
3

In Queue
0

Figure 3. Accurate "Active Now" Counter

The UI now displays a live count of users who have been active in 30 seconds which will correctly decreases as users go idle.

6. Conclusion

In conclusion, this part of our progress successfully transitioned our program the AIMSsist chatbot to a more responsible and trusted program. By removing the hard threading model and putting a good queue using a JSON file, we created a stable, multi-user application where the queue state is still in function. This new system effectively shows the pillar principles of a shared queue data structure in a way that is both functional and easy to understand.

The project now fully meets the course objectives of our goal. The chatbot not only implements a FIFO queue but does a realistic, multi-user environment with features like an accurate "Active Now" user count and a transparent display of the queue's status. The application is now in near final state, clearly showcasing how data structures are used to solve real-world application problems in software development using data structure.