Use Case Title: Server Log Reader for Error & Warning Detection

Description: A Python-based utility scans through large server log files line-by-line to identify and extract any lines containing `[ERROR]`, `[WARNING]`, or `[FAILURE]` tags. The tool records the matching log entries along with their corresponding line numbers for easier debugging and reporting.

Actors:

- Primary User: System Administrator / DevOps Engineer
- Secondary User: Developer / Support Engineer

Preconditions:

- The server log file is accessible for reading (local or network location).
- The log format consistently uses tags `[ERROR]`, `[WARNING]`, and `[FAILURE]` to indicate issues.

Trigger: The administrator runs the script with the path to the log file as an argument.

Main Flow:

1. The utility opens the specified server log file in read mode.
2. It iterates through the file line-by-line to minimize memory usage (handles very large logs).
3. Each line is checked using a regular expression to detect `[ERROR]`, `[WARNING]`, or `[FAILURE]`.
4. If a match is found, the line number and full log line are recorded.
5. After scanning the entire file, the results are saved to an output file or displayed on screen for review.

Postconditions:

- A list of problematic log entries with line numbers is generated for further investigation.

Example Output:

```
Line 1023: [ERROR] Connection to database failed.
Line 2877: [WARNING] Disk space low: 5% remaining.
Line 3499: [FAILURE] Scheduled backup did not complete.
```

Business Value:

- Speeds up troubleshooting by quickly pinpointing problematic log entries.
- Reduces manual searching through massive logs.
- Improves incident response time and system reliability.