Student: Travis Uhrig

Capstone Project Proposal: Command Line Kanban Board

Description (What)
A minimalist task management kanban tool, like Trello, written in C++ for the windows command line interface. It allows users to create, assign, and manage a task's title, description, due dates, and stages. This tool offers a simple alternative to complicated GUI tools.

Intended Users (Who and Why)
Designed primarily for small project teams and managers who need to track multiple tasks, delegate them to team members, and monitor progress. It also serves individuals for personal task management, helping them keep track of their projects.

Data
The program accepts task-related data like task title, description, due date, stage, and assigned user. It also accepts board and user data when new users and boards are created.

Advanced Concepts
- Classes: Utilized to define tasks, users, and kanban boards.
- Constructors and Destructors: For creating tasks, users, and boards, and destructing when tasks are completed, users removed, or boards deleted.
- Exception Handling: Database interactions can fail so the project will need to handle those failures and errors gracefully.

Algorithm
The kanban program starts with a user being selected, either by creating a new user or loading an existing user. The user then selects or creates a board, and in that board the user can create or manage tasks. The users, boards, and tasks are all saved and loaded from a database. The database is updated every time a change is made to a task, board, or user.

Tasks can be updated and moved through different stages. The transition to each new stage triggers checks to ensure the task meets the necessary criteria, with missing requirements causing an error message.

Users, boards, and tasks can be deleted. When a board is deleted, all of its tasks are also deleted.

Exception Handling
Detailed exception handling protects the user experience, the program data, and the stability of the program even when unexpected errors occur. Cases handled include:
- Null Input: Stops processing and requests valid input for null user input.
- Invalid Date Format: Prompts correction for dates not in the MM/DD/YYYY format.
- Non-existent User: Flags non-existent users during task assignments.

- Stage Requirements: Verifies if a task meets the necessary requirements to move to a new stage.
- Database Error: Checks for successful data read/write operations to the database. In case of a failure, it notifies the user of a database error.
- Unknown Exception: In case of an unforeseen issue, it outputs error info and terminates to prevent potential damage.

Functionality Summary

- Three main entities exist in the program, each having it's own class: Users, Tasks, and Boards. There is also a 4th class called Database which handles saving and loading data to the SQLite DB.
- User:
    - Users can create Boards and Tasks.
    - Users can be assigned to multiple Tasks.
    - Users can update any Task, even if it is not assigned to them.
    - A replacement User must be given when you delete a User. They are assigned all of the tasks previously assigned to the deleted User.
    - To reduce complexity, no user authentication is included in the program.
- Task:
    - Tasks can only be assigned to one User at a time.
    - Tasks are added to the current board's task list when created.
    - Tasks are created with a title and start in the "Backlog" stage.
    - The variable difficultyScore on tasks can be an int from 1 to 5.
    - Tasks move through, in order, the stages "Backlog", "To Do", "In Progress", "Done", and "Archive". Tasks cannot skip a stage.
    - Task must meet the next stage's requirements to move to the next stage:
        - Tasks need a description and a difficulty score to move to the "To Do" stage.
        - Tasks need a description, difficulty score, assigned user, and a due date to move to the "In Progress" stage.
        - Tasks don't need any additional requirements to move to the "Done" or "Archive" stages.
- Board:
    - Any User can delete any Board.
    - Users can switch between boards, working with only one board at a time.
    - Tasks are ordered in their stages on their board by their id and due date.
    - Deleted Boards get all of their Tasks deleted as well.
- Database:
    - Created Users, Boards, and Tasks are saved to a SQLite database.
    - Any updates are also saved to the SQLite database.
    - When a User, Task, or Board is deleted, what really happens is it gets marked inactive in the database. Loading records from the db also ignores anything marked inactive.