Documentation for “Finito!”

Objective

The primary goal of this project is to develop a task management application named "Finito!" that allows users to create, manage, and organize tasks efficiently. The application aims to provide a user-friendly interface for managing tasks, including features such as setting deadlines, prioritizing tasks, and tracking completion status. The project also aims to implement a system for storing and retrieving task data, ensuring that users can access their tasks across sessions.

Project Description

The task manager application is designed to help users keep track of their tasks and deadlines. The application includes the following key features:

- **Task Creation**: Users can create new tasks with descriptions, deadlines, and priority levels (low, medium, high).
- **Task Management**: Users can view, edit, and delete tasks. They can also mark tasks as completed.
- **Task Organization**: Users can categorize tasks into folders, making it easier to manage related tasks.
- **Statistics Panel**: The application provides a statistics panel that displays information about the total number of tasks, completed tasks, and tasks with deadlines.
- **Quote Display**: The application randomly displays motivational quotes to inspire users.

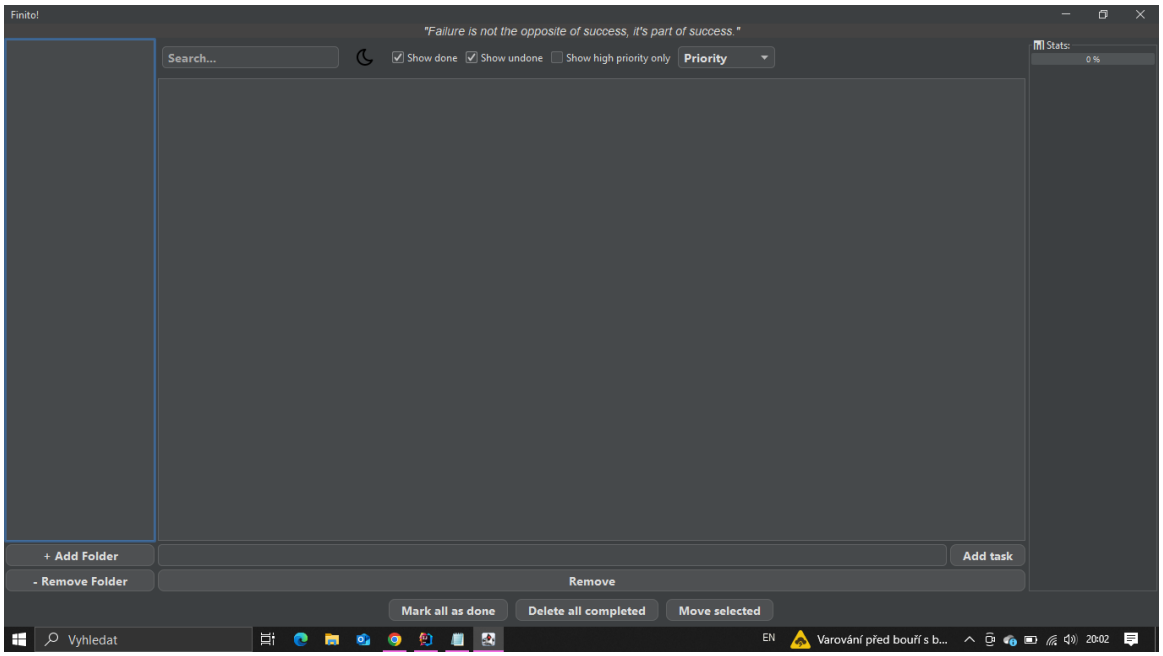
Functionality Overview

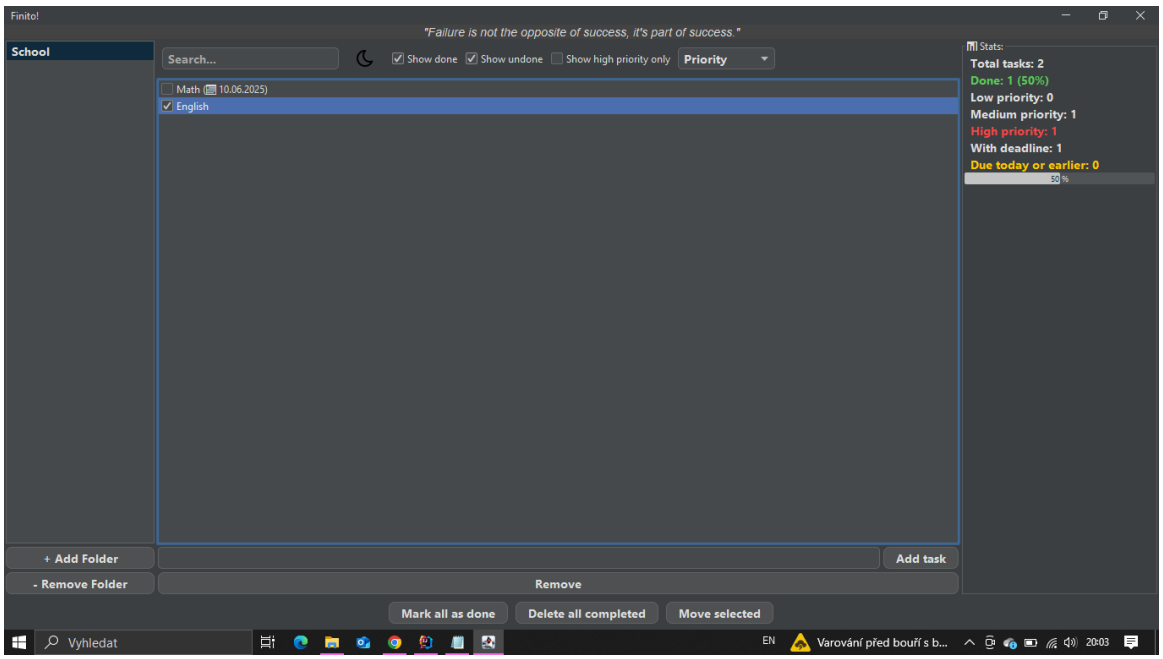
The application operates through a graphical user interface (GUI) built using Java Swing. Users interact with the application through buttons, text fields, and lists. The main components of the application include:

- **Main Frame**: The primary window where users can see their tasks and folders.
- **Sidebar**: Contains a list of task folders and buttons for adding or removing folders.

- **Content Panel**: Displays the tasks within the selected folder and provides options to add or remove tasks.
- **Statistics Panel**: Shows statistics related to the tasks in the selected folder.

Screenshots

- **Main Interface**:
  - 

- **Statistics Panel**:
  - 

System Requirements

- **Java Version**: The application requires Java Development Kit (JDK) version 21 or higher.
- **Libraries**: The application uses the following libraries:
    - **FlatLaf**: A modern look and feel for Swing applications.
    - **JUnit**: For unit testing the application.

Basic Structure

The application is structured around several key classes:

- **Task**: Represents a single task with attributes such as description, deadline, priority, and completion status.
- **TaskFolder**: Represents a folder that can contain multiple tasks.
- **DataManager**: Handles the serialization and deserialization of task folders to and from a file.
- **Quotes**: Manages the loading and retrieval of motivational quotes from a text file.
- **StatisticsPanel**: Displays statistical information about tasks in the selected folder.
- **TaskManagerGUI**: The main GUI class that manages user interactions and updates the interface.

Communication Between Components

- The `TaskManagerGUI` class interacts with `Task`, `TaskFolder`, and `DataManager` to manage tasks and folders.
- The `StatisticsPanel` updates its display based on the tasks in the currently selected folder, using data from the `TaskFolder` class.

Testing Data

To ensure the application functions correctly, the following testing strategies are recommended:

- **Unit Tests**: Implement unit tests for the `DataManager`, `Task`, and `Quotes` classes to verify their functionality.
- **Functional Tests**: Manually test the GUI to ensure that all buttons and features work as expected.
- **Example Tests**:

- Test adding a task with a valid deadline and check if it appears in the task list.
- Test marking a task as completed and verify that the statistics update accordingly.
- Test loading quotes from a file to ensure the application handles missing or empty files gracefully.

User Manual

The application is designed to be intuitive and user-friendly. Users can perform the following actions:

- **Add a Task**: Enter a task description in the text field and click the "Add task" button. Optionally, set a deadline and priority.
- **Remove a Task**: Select a task from the list and click the "Remove" button.
- **Edit a Task**: Double-click on a task to open the editor dialog, where users can modify the task's details.
- **View Statistics**: The statistics panel on the right side of the main window displays information about the tasks in the selected folder.

Conclusion

Throughout the development of the "Finito!" task manager application, several challenges were encountered, including:

- **Data Serialization**: Ensuring that task data is saved and loaded correctly required careful handling of file I/O operations.
- **User Interface Design**: Creating an intuitive and visually appealing GUI was a significant focus, leading to the adoption of the FlatLaf library for a modern look.
- **Testing**: Developing a comprehensive testing strategy was essential to ensure the reliability of the application.

Overall, the project provided valuable experience in software development, particularly in Java programming, GUI design, and data management. The final product is a functional task manager that meets the initial project goals and offers a pleasant user experience.