Internship Assignment Presentation

Simple Console-Based Task Management App

By: Nikolaos Rossis

General View of Task Management App

Users can successfully **add**, **edit**, **delete**, and **view** tasks.

Tasks include the attributes: **title**, **description**, **priority**, and **due date**.

Different Users can enter the program and write their own tasks.

They are identified with their own **userID** and can act on their previous tasks once have been identified correctly.

Classes Called **UserManager** and **TaskManager** are used to handle operations on their respective class they are managing.

I/O Computing

JSON files are used in order to save data. Users.json is responsible for registering Users. Then each user gets his own json file by adding his ID at the end of the filename. All User Task Files are stored in a folder called UserTasks.

Example:

> New User with ID 123456 is created

> ID with rest of User attributes gets added to Users.json

> New .json file with name Tasks123456.json is created inside the UserTasks folder

Explanation of App’s classes

The **User** class contains attribute: id, name, ReadPermission and WritePermission *(see slide 9 for the Permissions)*. The class is used to register a User and retrieving User objects in order to act upon their tasks.

The **Task** class contains attributes: taskID, title, description, priority, due date. Just like the User the class is used to register Tasks and retrieving them.

Explanation of App’s classes (2)

The **TaskManager** Class is responsible for every operation involving the Tasks, and their updating to the JSON file. It stores Tasks in a Dictionary called Tasks. It serializes the Dictionary into a String with JSON Format and writes it into the Json File.

The **UserManager** Class works exactly the same way as **TaskManager** but with fewer actions since Users only need be registered or retrieved from the Json.

Both inherit from the **Manager** Class.

Error Handling

A function called **errorChecking**(input, string TypeOfError) is responsible for handling correct input from the user. It implements a switch case through string TypeOfError for different cases of user input. It uses the TryParse function in order to handle correct parsing and general True/False statements to comply with the Apps Requirements

Unit Testing

Simple Unit tests are used in order to check that the addition and deletion of the Tasks happens correctly.

Type of Unit Test : NUnit

Tasks12345678.json is a file used only for Unit Testing!

TaskManagerApp REST API

A simple REST Api with http get, post, put and delete operations has been implemented.

Differences between App and API: API doesn’t use User Identification. It completes the task actions successfully and writes each task in one .json file called Tasks.json.

Non-Implemented Ideas & Code Fixes

Due to having little time to focus entirely on the project through these days I was not able to polish the code as much as I would like to and implement some extra ideas I had about the use of the program

Non-Implemented Ideas involve:

- User identification through Username and Password
- Use of an encrypting algorithm for user data safety
- Read/Write permissions. Users would only be able to Read and Write on their personal files while the Admin could Read (but not Write) all the other User Tasks.

Non-Implemented Ideas & Code Fixes (2)

Code Fixes:
- Giving **Manager** Class more functions as abstract and implementing them later from **UserManager** and **TaskManager** since they both do similar operations through Functions with nearly identical code but different names.

- Put code written in the program Main function into other Functions for better readability.
- Adding more comments
- Adding proper type of Task variables. (Ex. DateTime for DueDate or int for priority)

Last Notes

The program runs in the Android Studio 2022 console and by running the .exe

Change Paths to your appropriate paths for correct use of the program.

Tasks12345678.json is a file used only for Unit Testing!

Thank you

Vimachem

for the opportunity!