Universidad de las Fuerzas Armadas “ESPE”

CLASS NAME: OBJECT ORIENTED PROGRAMMING
TEACHER: EDISON LASCANO
NRC: 14575
TOPIC: TEST CASE

Coding Ninjas

MEMBERS: - Lopez David
-Revilla Antonio
-Paspuel Oliver
- Ñato Carlos

TEST CASES

CASES MENÚ

1. Create a new account
    - Input: The options Register as an administrator and Register as a developer are displayed to select
    - Expected: You can select any of the two options to register

2. Register as an administrator
    - Input: A username and password are registered, additionally the name of the organization to which the administrator belongs is registered.
    - Expected: A new user is created correctly and the option to start the session in the program is displayed.

3. Register as administrator with existing data
    - Input: A username and password are registered, which already exist
    - Expected: A notice is generated indicating that the data entered is not valid

4. Register as an developer
    - Input: The organization code is requested, and the corresponding username and password are requested.
    - Expected: A new developer is created successfully and you can log in with those credentials

5. Register as a developer with existing data
    - Input: The organization code is requested, and an existing username is entered.
    - Expected: You receive a notice that you cannot register a new account with those credentials

6. Login to an existing account
    - Input: User and password data are entered correctly

- Expected: The Landing Page "Program Start" is entered normally

7. Login to an existing account
    - Input: The username and password data of an unregistered user are entered.
    - Expected: An alert is given that the account does not exist and the data is requested again.

CLASS MESSAGE

1. Show message in task menu

    - Input: Task JSON file read
    - Expected: A message indicating the status of your tasks is displayed on the screen.

2. Show message when all tasks are complete
    - Input: reading the logged-in user's task JSON file
    - Expected: a message is displayed on the screen indicating that there are no pending tasks

3. Show message when there are incomplete tasks
    - Input: reading the logged-in user's task JSON file
    - Expected: a message is displayed on the screen indicating that there are tasks for the logged in user that have yet to be completed or are pending.

CASES TASK

1. Create a Task Successfully:
    - Input: Valid data for a new task (name, description, due date).
    - Expected Output: The task is created successfully and stored in the file.

2. Complete an Existing Task.
    - Input: Valid ID of an existing task.
    - Expected Output: The task is marked as completed and updated in the file.

3. Show Tasks for Today:
    - Input: List of tasks with at least one task for today.
    - Expected Output: The task is printed correctly.

4. Show All Tasks:
    - Input: List of tasks with at least one task.
    - Expected Output: The list of tasks is printed correctly.

5. Due Date Before Creation Date:
    - Input: Task data with a due date before the creation date.
    - Expected Output: An error message is displayed, and the task is not created.

6. Show Tasks with No User Logged In:
    - Input: No user logged in.
    - Expected Output: A message indicating that no user is logged in is displayed, and the user is returned to the main menu.

7. Complete a Non-Existent Task:
    - Input: ID of a task that does not exist.
    - Expected Output: A message indicating that the task was not found is displayed, and the user is returned to the main menu.

8. Show Tasks for Today with No Tasks Pending:
    - Input: List of tasks without tasks for today.
    - Expected Output: A message indicating that there are no tasks for today is displayed, and the user is returned to the main menu.

9. Show All Tasks with No Tasks Created:
    - Input: Empty list of tasks.
    - Expected Output: A message indicating that there are no tasks is displayed, and the user is returned to the main menu.

10. Read Tasks from a File:
    - Input: JSON file with task data.
    - Expected Output: The file is read correctly, and a list of tasks is obtained.

11. When you enter Task and have pending tasks:
    - Input: List of tasks with at least one pending task.
    - Expected Output: The list of pending tasks is printed.

12. When you enter Task and have all tasks completed:
    - Input: List of tasks with all tasks completed.
    - Expected Output: A message indicating that all tasks are completed is printed.

13. When you have tasks with a due date equal to the current date, "today's tasks" are printed:

    - Input: List of tasks with a due date equal to the current date.
    - Expected Output: "Today's tasks" are printed correctly.

14. When you try to complete a task with an invalid ID:
    - Input: Invalid task ID.
    - Expected Output: An error message is displayed, and the user is returned to the main men

15. When you show all tasks for a user with no tasks:
    - Input: User ID with no associated tasks.
    - Expected Output: A message indicates that there are no tasks for the current user, and the user is returned to the main menu.

16. When you create multiple tasks consecutively:

- Input: Creating multiple tasks in sequence.
- Expected Output: Each task is created successfully, and the tasks are correctly stored in the file.

17. When you show tasks with a null user ID:
    - Input: User ID is set to null.
    - Expected Output: A message indicates that no user is logged in, and the user is returned to the main menu.

18. When you create a task with an empty name:
    - Input: Empty string for the task name.
    - Expected Output: An error message is displayed, and the task is not created.

19. When you create a task with a long description:
    - Input: Long description for the task.
    - Expected Output: The task is created successfully, and the long description is stored correctly.

20. When you create a task and then complete it:
    - Input: Creating a task and then completing it.
    - Expected Output: The task is marked as completed, and the change is reflected in the file.

CASES MEETING

1. Successful Meeting Creation Test:
    - Input: Valid title, valid start and end times, valid participants.
    - Expected Output: The meeting is created successfully, assigned an ID, and added to the meetings list.

2. Meeting Creation without Participants Test:
    - Input: Valid title, valid start and end times, no participants.
    - Expected Output: The meeting is not created, and a message indicates that participants are required.

3. Successful Meeting Update Test:
    - Input: Select an existing meeting, provide a new title, valid start and end times.
    - Expected Output: The meeting is successfully updated with the new information.

4. Meeting Update with Invalid Time Test:
    - Input: Select an existing meeting, provide a new title and invalid start time.
    - Expected Output: The meeting is not updated, and an error message is displayed.

5. Successful Meeting Deletion Test:
    - Input: Select an existing meeting to delete.
    - Expected Output: The meeting is removed from the meetings list

    .

6. Non-existent Meeting Deletion Test:
    - Input: Select an index that does not correspond to any meeting.
    - Expected Output: A message is displayed indicating that the meeting does not exist

    .

7. Meeting Listing Test:
    - Input: Display the list of meetings.
    - Expected Output: The list of meetings is printed in a formatted manner.

8. Load Meetings from JSON Test:
    - Input: "meetings.json" file with valid meetings.

- Expected Output: Meetings are loaded from the JSON file and reflected in the meetings list.

9. Invalid Date Handling Test:
    - Input: Create a meeting with a start time earlier than the current time.
    - Expected Output: The meeting is not created, and an error message is displayed.

10. Update with Out-of-Range Date Test:
    - Input: Attempt to update a meeting with a start date more than 30 days in the future.
    - Expected Output: The meeting is not updated, and an error message is displayed.