Telerik Academy

Task Management System

Project Description

Design and implement a **Tasks Management** console application that will help a small team of developers to organize and manage their work tasks.

Functional Requirements

The application must support multiple **teams**.

Each team must have a **name**, **members,** and **boards**.

- The name must be unique in the application.
- The name is a string between 5 and 15 symbols.

Each member must have a **name**, list of **tasks** and **activity history**.

- The name must be unique in the application.
- The name is a string between 5 and 15 symbols.

Each board must have a **name**, list of **tasks** and **activity history**.

- Name must be unique in the team.
- Name is a string between 5 and 10 symbols.

There are 3 types of tasks: **bug**, **story,** and **feedback**.

Bug

Bugs must have an ID, a title, a description, a list of steps to reproduce it, a priority, a severity, a status, an assignee, a list of comments and a list of changes history.

- Title is a string between 10 and 50 symbols.
- Description is a string between 10 and 500 symbols.
- Steps to reproduce is a list of strings.
- Priority is one of the following: **High**, **Medium**, or **Low**.
- Severity is one of the following: **Critical**, **Major**, or **Minor**.
- Status is one of the following: **Active** or **Fixed**.
- Assignee is a member from the team.
- Comments is a list of comments (string messages with an author).
- History is a list of all changes (string messages) that were done to the bug.

Story

Stories must have an ID, a title, a description, a priority, a size, a status, an assignee, a list of comments and a list of changes history.

- Title is a string between 10 and 50 symbols.
- Description is a string between 10 and 500 symbols.

- Priority is one of the following: **High**, **Medium**, or **Low**.
- Size is one of the following: **Large**, **Medium**, or **Small**.
- Status is one of the following: **Not Done**, **InProgress**, or **Done**.
- Assignee is a member from the team.
- Comments is a list of comments (string messages with author).
- History is a list of all changes (string messages) that were done to the story.

Feedback

Feedbacks must have an ID, a title, a description, a rating, a status, a list of comments and a list of changes history.

- Title is a string between 10 and 50 symbols.
- Description is a string between 10 and 500 symbols.
- Rating is an integer.
- Status is one of the following: **New**, **Unscheduled**, **Scheduled**, or **Done**.
- Comments is a list of comments (string messages with author).
- History is a list of all changes (string messages) that were done to the feedback.

Important

Each task must have a unique ID. For example, if there is a bug with ID = 1 then a story or feedback with ID = 1 cannot exist.

Operations

The application must support the following operations:

- Create a new person.
- Show all people.
- Show person's activity.
- Create a new team.
- Show all teams.
- Show team's activity.
- Add person to team.
- Show all team members.
- Create a new board in a team.
- Show all team boards.
- Show board's activity.
- Create a new Bug/Story/Feedback in a board.
- Change the Priority/Severity/Status of a bug.
- Change the Priority/Size/Status of a story.
- Change the Rating/Status of a feedback.

- Assign/Unassign a task to a person.
- Add comment to a task.

- Listing:
    - List all tasks (display the most important info).
        - Filter by title
        - Sort by title
    - List bugs/stories/feedback only.
        - Filter by status and/or assignee
        - Sort by title/priority/severity/size/rating (depending on the task type)
    - List tasks with assignee.
        - Filter by status and/or assignee
        - Sort by title

Use cases

Use case #1

One of the developers has noticed a bug in the company's product. He starts the application and goes on to create a new Task for it. He creates a new Bug and gives it the title "The program freezes when the Log In button is clicked." For the description he adds "This needs to be fixed quickly!", he marks the Bug as High priority and gives it Critical severity. Since it is a new bug, it gets the Active status. The developer also assigns it to the senior developer in the team. To be able to fix the bug, the senior developer needs to know how to reproduce it, so the developer who logged the bug adds a list of steps to reproduce: "1. Open the application; 2. Click "Log In"; 3. The application freezes!" The bug is saved to the application and is ready to be fixed.

Use case #2

A new developer has joined the team. One of the other developers starts the application and creates a new team member. After that, he adds the new team member to one of the existing teams and assigns all Critical bugs to him.

Use case #3

One of the developers has fixed a bug that was assigned to him. He adds a comment to that bug, saying "This one took me a while, but it is fixed now!", and then changes the status of the bug to Done. Just to be sure, he checks the changes history list of the bug and sees that the last entry in the list says, "The status of item with ID 42 switched from Active to Done."

Technical Requirements

- Follow the **OOP best practices**:
  - Use data encapsulation.
  - Use inheritance and polymorphism properly.
  - Use interfaces and abstract classes properly.
  - Use static members properly.
  - Use enumerations properly.
  - Aim for strong cohesion and loose coupling.
- Follow guidelines for writing **clean code**:
  - Proper naming of classes, methods, and fields.
  - Small classes and methods.
  - Well formatted and consistent code.
  - No duplicate code.
- Implement user **input validations** and display meaningful messages.
- Implement proper **exception handling**.
- Prefer using LINQ when working with collections.
- Cover the core functionality with unit tests (**at least 80%** code coverage of the models and commands).
  - There is no need to test the printing commands.
- Use **Git** to keep your source code and for team collaboration.

Teamwork Guidelines

Please see the Teamwork Guidelines document.