

README FILE
ENMG644 – Final Project

Group 9 – Reem Nakouzi – Wafic El Ariss – Shadi AbdulAziz – Georges El Rahi

Description:

This software will allow the user to manage iterations in an agile project and track their progress. First, the iteration needs to be defined with its respective user stories and tasks. Then, the progress can be tracked through the big board. The software will show the big board by listing the user stories in backlog, user stories in progress, and completed user stories along with the burn-down chart of that iteration. Scrum master acting as the project manager will have the role of creating an iteration and adding user stories and tasks to it. Developer can change the status of a task assigned to him from “Open” status to “In Progress” to “Completed”.

Installation:

Dependencies

Prerequisites to this project are to install python3 and pip3, using the following two links:

- Python 3.8+: <https://www.python.org/downloads/>
- Pip3: <https://pip.pypa.io/en/stable/installing/>

In order to run the code, the user needs to install the following:

1. Matplotlib library for plotting the burn-down chart

```
pip3 install matplotlib
```

2. Pandas library to create and manage databases as dataframes

```
pip3 install pandas
```

3. Openpyxl xlswriter xlrd packages for reading and writing on Excel

```
pip3 install openpyxl xlswriter xlrd
```

Installing the code

Get Project using Git

```
git clone https://github.com/wafts95/ENMG_644.git
```

Usage:

After following the installation procedures, run the **Driver.py** script from a terminal.

The software will first initialize and then prompt the user to select what type of user they are:

```
Initializing application...

Application Loaded Successfully!

Greetings, and welcome to your journey towards success...
My name is Trella, and I will be your digital Agile assistant!

Please Select a Role:
1- Scrum Master
2- Developer
```

Throughout the run, the user must select the number of the desired option.

Passwords and Validation

1. Scrum Master:

Must log in using a general Scrum Master password (“enmg644”).

The password input will not be shown, to secure privacy.

Note: On IDLE, the feature to hide the password input will not work, and will return the following Warning:

```
Warning (from warnings module):
  File "/Library/Frameworks/Python.framework/Versions/3.9/lib/python3.9/getpass.py",
    line 62
      passwd = fallback_getpass(prompt, stream)
GetPassWarning: Can not control echo on the terminal.
Warning: Password input may be echoed.
Please enter the Scrum Master password:
```

The program will still proceed, and will accept the input, however it will not hide said input.

2. Developer:

Must enter a valid username (case sensitive), which can be found in the database. The available users currently are:

- Reem
- Shadi
- George
- Wafic
- Ali

At least one Developer needs to be predefined in the users' database (in **user.xlsx**) in order to log in as a developer.

Depending on the user role, they will see the available options for them. The user should choose the number of the desired option to work with.

1- Scrum Master Options:

```
Available Options:
0- Save & Exit
1- Create Iteration
2- Create User Story
3- Create Task
4- Delete User Story
5- Delete Task
6- Delete Iteration
8- Generate Iteration Report
9- View Project

Enter option: █
```

2- Developer Options:

```
Available Options:
0- Save & Exit
7- Update Task Status
8- Generate Iteration Report
9- View Project

Enter option: █
```

After selecting an option, the user must enter the information that is requested by the system. For each functionality, the software will ask the user to input the attributes of the object he wants to create, delete, or update. If the user enters an invalid option or input, an error message is displayed asking the user to try again.

Whenever the Back option is shown, the user can use it to go back to the main menu of his role.

In Create and Delete user story option, the user must select user story's respective iteration first. Same goes for Create, Delete, and Update Task Status, where the user must select the iteration and user story of that task.

In Create Iteration, the user will be asked to enter the developers' names separated by (-). All new developers' names assigned to an iteration will be added to the User database.

For Update Task Status option, each developer will be able to update only the tasks assigned to him. If the user is not a part of an iteration or there are no tasks assigned to him, the software will display a message.

To generate a report and Burn-Down Chart, a user must first select an iteration. The summary of the iteration will appear on the Terminal or Shell showing the user stories in backlog (all tasks are open), the user stories that are in progress, and the completed user stories. It will also show the velocity of the team which is the sum of the completed user stories size. Also, the Burn-down plot will pop-up. The chart will show the ideal vs. actual lines. The x-axis represents the duration of the iteration (input by the user) and the y-axis is the sum of all task efforts in that iteration. The ideal line will connect the maximum effort with the maximum duration of the iteration. A new point will be plot on the actual line every time a task status is changed to completed where the total remaining effort is decreased by the effort of that task.

To view the project, the user interface (UI) a user will asked to enter the desired Iteration to view, which will display its details and children User Stories. Subsequently, the UI will ask the user to enter the desired User Story to view from the previously selected Iteration, which will display the details and children Tasks of the former. Finally, the UI will ask the user to enter the desired Task from the previously selected User Story, which will display the details of the former.

To exit and save all the data created and edited in any run, the user much choose option 0 "**Save & Exit**". All the data will be written to the database.

Design Principles and Software Documentation:

This Agile project management software is organized into three main components:

- 1- The Database of information
- 2- The Object Classes
- 3- The main system Driver and its accessory scripts

The concepts of Object-Oriented Programming were followed to design the components of the project, which will be reflected in the classes that were developed. The user will observe that, throughout the code, the Single Responsibility Principle (SRP) were respected when possible, by developing class methods that are clear, simple and perform a very well-defined behavior. Coupling was minimized by offering methods that modify class attributes, and refrained where possible from changing class attributes directly from the outside (used private attributes). Moreover, in the context of minimizing coupling, by following SRP, dependencies were minimized where possible from calling different methods within other methods.

The database of information is made up of four tables that capture the information on the following four project elements:

- 1- Iterations DB:
 - a. Iteration name
 - b. Duration
 - c. Start Date
 - d. Assigned developers
- 2- User Stories DB:
 - a. Name
 - b. Size
 - c. Associated Iteration
- 3- User Story Tasks DB
 - a. Name
 - b. Effort
 - c. Start
 - d. End
 - e. Assigned developer
 - f. Parent User Story
 - g. Progress Status
- 4- Project Users DB
 - a. Developer names
- 5- Password DB
 - a. Scrum Master login password

The defined classes represent the different “objects” that were identified to be part of the system, the corresponding methods represent the actions that these objects can perform, and the attributes of the classes represent the properties that the objects can hold. These classes are:

1- Iteration

An iteration holds the properties presented in the corresponding database, and as such the class `Iteration()` contains an attribute for each one of the `IterationDB` columns.

In this project, and as part of an iteration, it is required to have the ability to create new user stories, delete existing user stories, and print the report of the iteration. Consequently, in the `Iteration()` class, a method for each one of these three actions that pertain to a project’s iteration was created.

2- User Story

A user story holds the properties presented in the corresponding database, and as such the class `UserStory()` contains an attribute for each one of the `UserStoryDB` columns.

In this project, and as part of a user story, it is required to have the ability to display the parent iteration, to update the effort estimates of the story, to update the completion status of the story, to add a task to the story and finally to delete a task from the story. Consequently, in the `UserStory()` class, a method for each one of these five actions that pertain to a project’s user story was created.

3- Task

A task holds the properties presented in the corresponding database, and as such the class `Task()` contains an attribute for each one of the `TaskDB` columns.

In this project, and as part of a user story's task, it is required to update the completion status of the task. Consequently, in the Task() class, a method for this action that pertain to a user story's task was created.

4- Developer

The developer class holds the name of the developer as attribute, and can print the developer name using the print_developer() method.

Finally, the main program is controlled by a high level Driver.py script, which loads the application by using the methods inside the script initializer.py, displays to the user the available options, takes-in the user input, and manages passing the information to and from the different object classes, as well as performing internal computations for the operation of the software.