TIC2002

INTRODUCTION TO SOFTWARE ENGINEERING

AY19/20 Semester 1

Duke Project Report

November 17, 2019

NAME	Li Shihao
MATRIX NO.	A0165362E
GITHUB USERNAME	asmaww
EMAIL	e0166067@u.nus.edu

User Stories

1. Duke task checklist supports multiple users to use the application;

2. As users who prefer faster entries, it would be a great fit for this kind of audience to be able to quickly note down and manipulate tasks using various commands, which are triggered by a few key strokes. It's faster than clicking buttons by mouse in GUI;

3. When the next time users start the program, Duke should still remember the tasks that users had left from last time;

4. There are different types of task that fit in different use cases;

5. There are commands that mark task status such as completion and provide views of list of tasks;

6. System should be fast, reliable and provide message to guide correct user behaviour.

Non-functional requirements

1. Message from the system should be fun and intimate;

2. The system should be smart to guide the users with meaningful alert of what to do rather than just exit;

3. Minimum visual elements/clusters which create distraction for users want to be fast. The aesthetic aims for simplicity and tidiness.

Level-1 The output Duke shows when launching the program

```
Knock knock  Σ ﾉ( ･ᴗ ･ ? )        Knock knock  Σ ﾉ( ･ᴗ ･ ? )
  ____        _                      ____        _
 |  _ \ _   _| | _____              |  _ \ _   _| | _____
 | | | | | | | |/ / _ \             | | | | | | | |/ / _ \
 | |_| | |_| |   <  __/             | |_| | |_| |   <  __/
 |____/ \__,_|_|\_\___|             |____/ \__,_|_|\_\___|
          \     /                            \     /
     Hey! \   / here, living in a ...    Hey! \   / here, living in a ...
     pod... 🏺                           pod... 🏺
     Who is there summoning me?         Who is there summoning me?
li shihao                           Linus T
     Welcome back, li shihao            New master registered \^o^/
```

Figure 1: Screenshot - Start Greeting Page Existing/New User

Level-4 Describe the commands for adding different types of tasks

```
todo do laundry
    Got it. I've added this task:
      [T][×] do laundry
    Now you have 11 tasks in the list.
deadline finish milk /by 2019-11-29
    Got it. I've added this task:
      [D][×] finish milk (by: Nov 29)
    Now you have 12 tasks in the list.
event sleep after exam /at 2019-12-04 3:30-7
    Got it. I've added this task:
      [E][×] sleep after exam (at: Dec 04 03:30-07:00)
    Now you have 13 tasks in the list.
```

Figure 2: Screenshot - `todo, deadline, event`

Level-2 Describe the commands for listing tasks

```
Knock knock  Σ ﾉ( · ౪ · ? )
 ____        _
|  _ \ _   _| | _____
| | | | | | | |/ / _ \
| |_| | |_| |   < __/
|____/ \__,_|_|\_\___|
        \     /
    Hey! \   / here, living in a ...
    pod... 🏺
    Who is there summoning me?
li shihao
    Welcome back, li shihao
list
    Here are the tasks in your list:
    1.[T][×] read book Algorithms
    2.[D][✓] return book (by: Feb 28)
    3.[E][✓] project meeting (at: Oct 11 02:00-04:00)
    4.[E][✓] project management (at: Dec 25 02:00-13:15)
    5.[T][×] join sports club
    6.[T][×] meeting at 5F when free
    7.[D][✓] duke (by: Nov 17)
    8.[D][×] exam (by: Dec 04)
    9.[E][×] conf.call with customer (at: Nov 18 10:30-12:00)
    10.[D][×] security report (by: Nov 20)
|
```

Figure 3: Screenshot - `list`

Level-3 Describe the commands for marking/unmarking tasks as done.

Level-5 Describe what kind of errors Duke can handle. E.g., give different types of incorrect commands it can handle

Level-6 Describe the commands for deleting tasks

Level-7 Give a sample of the tasks as they are stored in the hard disk

Level-8 Explain how Duke uses dates/times

Level-9 Describe the commands for searching for tasks.

Level-10 Individual feature: If you implemented an individual feature, describe that feature

Other features Describe other features you implement

A-MoreOOP Give a class diagram to match your code

A-MoreOOP Give at least one object diagram illustrating the state of your program at some point

A-MoreOOP Give at least one sequence diagram illustrating an object interaction in your product

A-JavaDoc: Give at least 2 javadoc comments from you code

A-JUnit: Give 2-3 JUnit test methods from your code

A-Assertions: Give at least 2 code segments that contain assertions you added to your code

Suggested test commands Give a list of commands a tester can execute in sequence to examine your product. Cover all features in a reasonable order: