TIC2002 Introduction to Software Engineering  
AY2022/23 Semester 1

**Project report**

**Name**: Yao Liang

**Student Number**: A0227092H

**GitHub Username**: YaoLiangA0227092H

**GitHub Link:** <https://github.com/YaoLiangA0227092H>

|  |
| --- |
| **User stories** Give some user stories (5-10) that match your project.   1. As a user, I can view existing task list. 2. As a user, I can add to-do tasks with description, so that I can view it later. 3. As a user, I can add deadline tasks with description and deadline datetime, so that I can view it later. 4. As a user, I can add event tasks with description and event datetime, so that I can view it later. 5. As a user, I can mark a task as completed. 6. As a user, I can unmark a task as incomplete. 7. As a user, I can delete a task from list. 8. As a user, I can find a task by keyword, so that I can only view related tasks. 9. As a user, I can sort task list by name or date. 10. As a user, I can exit the chatbot. |
| **NFRs** Give some non-functional requirements (3-5) for your project.   1. Duke should give response on user’s input with in 2 seconds. 2. Duke should be usable for every user that have a computer with java environment installed. 3. Duke should work on both 32-bit and 64-bit environments. |
| **Launch** Show the output Duke shows when you launch the program.  Graphical user interface, text, application  Description automatically generated |
| **Todo, Deadlines, Events** Give examples of command(s) and expected outputs for adding different types of tasks.  Graphical user interface, text, application, chat or text message  Description automatically generated |
| **List** Give examples of command(s) and expected outputs for listing tasks.  Text  Description automatically generated |

|  |
| --- |
| **Mark, Unmark** Give examples of command(s) and expected outputs for marking/unmarking tasks as done.  Text  Description automatically generated |

|  |
| --- |
| **Errors** Describe what kind of errors Duke can handle. E.g., give different types of incorrect commands (and the expected outputs) it can handle  Timeline  Description automatically generated  Graphical user interface, text, application, chat or text message  Description automatically generated |

|  |
| --- |
| **Delete** Give examples of command(s) and expected outputs for deleting tasks.  Text, chat or text message  Description automatically generated |
| **Save** Give a sample of the tasks as they are stored in the hard disk.  e.g.,  T | 1 | read book  D | 0 | return book | June 6th  E | 0 | project meeting | Aug 6th 2-4pm  T | 1 | join sports club |

|  |
| --- |
| **Dates** Give examples (i.e., screenshots of commands and outputs) of how your Duke uses dates/times in meaningful ways i.e., what commands can use dates/times in meaningful ways? E.g., is it possible to list deadlines/events on a specific day? Does it detect invalid dates given by the user?  Timeline  Description automatically generated |
| **Find** Give examples of command(s) and expected outputs for searching for tasks.  Text  Description automatically generated |

|  |
| --- |
| **GUI**/**individual feature** If you implemented a GUI, give some screenshots. If you implemented an individual feature, describe that feature.  **GUI** – Please refer to any of the screenshot in the report  **C-Sort**  Text  Description automatically generated  **C-Help**  Graphical user interface, text, application  Description automatically generated |

|  |
| --- |
| **Other features** Describe other features you implement (i.e., not described above), if any e.g., optional increments.  N.A. |
| **A-MoreOOP** Give a class diagram to match your code. Do not use auto-generated diagrams. Omit less important classes, attributes, methods. Follow the given UML notation exactly.  Diagram  Description automatically generated  Give at least one object diagram illustrating the state of your program at some point. It should include at least one object of each of the important classes in your project.    Diagram  Description automatically generated  Give at least one sequence diagram illustrating an object interaction in your product.  Diagram  Description automatically generated |
| **A-JavaDoc**: Give at least 2 javadoc comments from you code.  Graphical user interface, text, application, email  Description automatically generatedGraphical user interface, text, application, email  Description automatically generatedGraphical user interface, text, application, email  Description automatically generatedGraphical user interface, text, application, email  Description automatically generated |
| **A-JUnit**: Give 2-3 JUnit test methods from your code.  Graphical user interface, text, application, email  Description automatically generatedGraphical user interface, text, application  Description automatically generated |
| **A-Assertions**: Give at least 2 code segments that contain assertions you added to your code.  Text  Description automatically generatedGraphical user interface, text, application  Description automatically generatedGraphical user interface, text, application, email  Description automatically generated |
| **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. E.g.,  list  todo Submit TIC2002 project report  deadline Submit TIC2002 JAR file /by 13-11-2022 2359  event Attend TIC2002 final exam /at 30/11/2022 1200  mark 2  unmark 2  delete 2  find book  sort by name  sort by date  bye |