HowTodoit

- a simple to do list app written in java

1.0 Introduction

HowTodoit is a to-do list app written in java. The user will be able to interact with a virtual task manager (via command line) that is able to load data from and save data into local history (using Java's Serializable interface).

2.0 Statement of scope

The app will be built using gradle to a JAR file and run in JRE 11. It initially only has a CLI (command line interface). We will add a GUI (and more features) in later phases of the project if time allows.

3.0 Architectural and component-level design

Task:

- The user can add tasks with a (unique) name, a due date, and a description.
- Tasks can be modified (name, due date, description, project) and completed.

Project:

- A project is basically a directory that stores tasks; a task must be in exactly one project.
- By default, all tasks created will be added to a project named "Inbox" at first; it's a default project that cannot be modified or deleted.
- The user can create projects with a (unique) name. User-created projects can be modified (name, tasks) and deleted. When deleted, all tasks stored inside this project will be moved back to "Inbox".

Label:

- A label is similar to a project except that a task can be in zero or more labels.
- The user can add labels with a (unique) name. User-created labels can be modified and deleted.
- The user can add tasks to labels. In particular, they can add tasks to a label named "Starred", which cannot be modified or deleted.

4.0 User interface design

The user can use the following (case-sensitive) commands to interact with the system. Note that arguments are separated by the character ";".

General commands:

- *upcoming* Show all upcoming tasks in all projects in chronological order.
- *exit* Quit the program and save data.

Commands on projects or labels:

- *mkproj;<name>* Create a project called <name>.
- *rnproj;<name1>;<name2>* Change a project’s name from <name1> to <name2>.
- *delproj;<name>* Delete the project called <name> and move all its tasks to Inbox.
- *vproj;<name>* View tasks from a project called <name> in chronological order.
- *lsprojs* Show all projects in alphabetical order but with Inbox at the bottom.

- *mklab;<name>* Create a label called <name>.
- *rnlab;<name1>;<name2>* Change a label’s name from <name1> to <name2>.
- *dellab;<name>* Delete a label called <name>.
- *vlab;<name>* View tasks from a label called <name> in chronological order.
- *lslabs* Show all labels in alphabetical order but with Starred at the top.

Commands on tasks:

- *newtask;<name>;<time>;<desc>* Create a new task called <name> with due date <time> and description <desc> and add it to Inbox; a valid example of <time> would be “2021 1015 2359”.
- *completetask;<name>* Remove a task called <name> from its project.
- *star;<name>* Add a task called <name> to Starred.
- *unstar;<name>* Remove a task called <name> from Starred.
- *rename;<name1>;<name2>* Change the name of a task from <name1> to <name2>.
- *redesc;<name>;<desc>* Change the description of a task called <name> to <desc>.
- *retime;<name>;<time>* Change the due date of a task called <name> to <time>.
- *reproj;<task name>;<proj name>* Move a task called <task name> to a different project called <proj name>.
- *addtasklab;<task name>;<lab name>* Add a task called <task name> to a label called <lab name>.
- *deltasklab;<task name>;<lab name>* Remove a task called <task name> from a label called <lab name>.