**V0.0 Deliverables**

**Suggested sequence :** document product survey --> document user stories --> prioritize use stories --> select user stories to include in your product --> design the product --> write the user guide

Fill up Glossary and NFRs in parallel to the above.

Submission: Proper submission will be in Week 8 (after converting to markdown and pushing to the repo). In tutorial 7, just show your intermediate drafts to the tutor.

Product Survey

1. Todo.txt

Strength

* It is offline and easy to start.
* Users can quickly add a task by typing a command.
* Users can add context to a task.
* Users can add priority to a task. Prioritized tasks are shown in different colors.
* Users can mark a task as complete.

Weakness

* It does not support for events and deadlines.
* It does not support for undo operations.
* The UI is too simple.
* Todo.txt

     2.   Wunderlist

Strength

* It is offline.
* It can sync to any android, ios or windows phone.
* Users can quickly add a task through the text box at the home page.
* Users can add context to a task like alarm as remainder.
* Users can add priority to a task.
* Users can mark a task as complete.
* Users can create folder and save different tasks under specific folder.
* It display tasks in different deadline folder. eg. This week, This month

Weakness

* It does not support for events.
* It does not support for undo operations.
* No calender view of tasks.
* Unable to design the colour theme or changing of background image.

User stories

Likely to be implement these features by V.0.5.:

* As a new user I can view usage instructions, so that I can learn how to use various commands.<help>
* As a user I can add a task by specifying a task description only, so that I can record tasks that need to be done ‘some day’.<add>
* As a user I can set a deadline or a start time and end time for a task, so that I can schedule my timetable better.<add>
* As a user I can find upcoming tasks, so that I can decide what needs to be done soon.<find>
* As a user I can edit a task, so that I can update my tasks.<edit>
* As a user I can delete a task, so that I can get rid of tasks that I no longer care to track.<delete>
* As a user I can undo operations, so that I can recover from mistakes.<undo>
* As a user I can have some flexibility in the command format, so that I can do operations in an intuitively way.<flexiCommand>
* As a user I can mark a task as done, so that I ……?
* As a user I can find a task by some keywords from the task description, so that I can locate a task without having to go through the entire list.<find>

Nice to have features:

* As a user I can sort the list of tasks, so that I can decide a task to do next.<sort>
* As a user I can add a recurring task, so that I don’t need to manually add it everytime.<add>
* As a user I can add tags, so that I can add more context to my tasks.<tag>
* As a user, I can add priority to a task, so that I can focus on more important tasks.
* As an advanced user I can specify a specific folder as the data storage location, so that I can choose to store the data file in a local folder controlled by a cloud syncing service, allowing me to access task data from multiple computers.<storage>
* As a user with many tasks, I can filter tasks for attributes such as start time, deadline, tag, and priority, so that I can locate some tasks easily.<filter>

Non Functional Requirements

1. Should work on any mainstream OS (add link) as long as it has Java 8 or higher installed.
2. Should come with automated unit tests and open source code.
3. Should favor DOS style commands over Unix-style commands. (??)
4. Should have a clean GUI

Glossary

Mainstream OS

Windows, Linux, Unix, OS-X

**Must-have features:**

1. Support for events (i.e., has a start time and end time), deadlines (tasks that have to be done before a specific deadline), and floating tasks (tasks without specific times). Note that these category names are temporary names only. You should figure out what are the best names for these categories. Are these names intuitive to end-users? Should they even care about consciously categorizing tasks?
2. CRUD (i.e., Create, Read, Update, Delete) support for tasks.
3. Undo operations (at least for the most recent action)
4. Some flexibility in the command format: Support a few natural variations of the command format.
5. Simple search: A simple text search for finding an item if the user remembers some keywords from the item description.
6. Some way to keep track of which items are done and which are yet to be done.
7. The ability to specify a specific folder as the data storage location. With this feature, Jim can choose to store the data file in a local folder controlled by a cloud syncing service (e.g. dropbox), allowing him to access task data from multiple computers.

**Nice to have features :** (you may implement 0 or more of these features)

Difficulty-low

* AutomatedTesting : Achieve a very high level of automated testing.
* RecurringTasks: Support for managing recurring tasks. e.g. CS2103 lectures are a recurring task from week 1 to week 13 except in recess week.

Difficulty-medium

* PowerSearch : More powerful and intelligent search. e.g. search for empty slots, near-match search, auto-complete (similar to Google search box), filters for other attributes (e.g. start time).
* FlexiCommands: High flexibility in command format. e.g., non-strict ordering of keywords, ability to specify aliases for commands, support for more ‘natural’ language input, multiple undo/redo, How flexible should the format be? As flexible, intuitive, and user-friendly as you can make it.

Difficulty-high

* GoodGui: A good GUI to give visual feedback to the user e.g., in-built guidance for new users, good visual feedback for actions, feedback while typing commands, hotkey to activate/hide, notifications, etc. conveying more information visually (e.g. more important/urgent tasks stand out from the rest, longer events look different from shorter events, easy to see free slots, etc.)
* GoogleIntegration: Google Calendar integration. e.g. upload todo items to GCal, two-way sync, Support for GCal quick add command format, etc. You may also use the Google Tasks API for this.
* Propose your own (subject to approval)