

J.listee

overdue

1

SE lecture
@icube #lecture

every Friday
16:00-18:00

today

2

SE lecture
@icube #lecture

Feb 14 16:00
Feb 14 24:00

3

SE lecture
@icube #lecture

every Friday
16:00-18:00

reserved

4

SE lecture
@icube #lecture

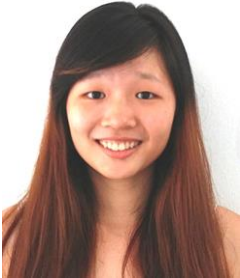
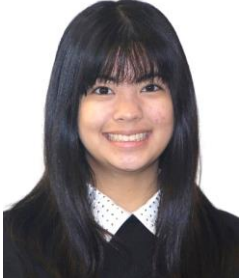


Apr 1 13:00
Apr 1 14:00

Apr 3 13:00
Apr 3 14:00

Apr 4 13:00
Apr 4 14:00

type your command here

Supervisor: Karan Kamath Extra feature: GoodGui

 <p>Her Kai Lin</p> <p>Team lead, Scheduling and tracking</p>	 <p>Chloe Odquier Fortuna</p> <p>Git expert, Integration</p>	 <p>Boh Tuang Hwee, Jehiel</p> <p>Code quality</p>	 <p>Zhu Bingjing</p> <p>Documentation and UI</p>
----------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------

User guide

TABLE OF CONTENTS

Quick start

Feature details

- ✧ Adding tasks
- ✧ Updating tasks
- ✧ Deleting tasks
- ✧ Searching tasks
- ✧ Marking as done/undone
- ✧ Undoing/redoin
- ✧ Exiting

Cheat sheet

QUICK START

1. Configure java environment on your computer. You can download JDK [here](#).
2. Run J.Listee.jar
3. Try to type the following commands in the command box and see what's happening:

add my first task @icube (Feb 14) #cs2103

update 1 updated first task

done 1

undo

postpone 1 2

show #cs2103

delete 1

Now you can add more tasks and start to manage them using J.Listee!

You can also use the cursor keys (↑ and ↓) to scroll through the tasks and select a specific task.

FEATURE DETAILS

✧ Adding tasks

Format:

```
add <task description> [[([recurring date] [<start time> -] <end time>)] [@ location]
[#<tag>[#<tag> [...]]]
```

Description:

This command enables you to add a task into J.Listee. The task will be assigned a task number when listed to you. After adding the task, the list of all tasks will automatically be displayed in order of urgency (overdue, today, tomorrow, timed, untimed, reserved).

Must-have arguments:

Task description

Optional arguments:

recurring date, start time, end time, location, tags (tags cannot contain whitespace)

✧ Updating tasks

Format:

```
update <task number> [<new task description>] [- ()] [[<new start time> -] <new end
time>]] [-@] [@ new location] [-#<existing tag> [#<new tag> [...]]]
```

Description:

This command enables you to update an existing task from J.Listee.

Must-have arguments:

a task number indicating the specific task

Optional arguments:

new task description, removing any time, new start time, new end time, removing existing location, new location, removing specific tags, new tags.

✧ Postponing tasks

Format:

Postpone <task number> <number of days>

Description:

This command enables you to postpone the deadline of an existing task for some days.

Must-have arguments:

task number and the number of days

Shortcut:

Select the specific task and press *p*, the deadline of the task will be postponed for 1 day.

✧ Deleting tasks

Format:

delete <1st task number> [, <2nd task number> [...]]

delete <task group> *all/done/undone/overdue/today/tomorrow/reserved

Description:

This command enables you to delete existing tasks from J.Listee.

Must-have arguments:

group name or at least one task number

Optional arguments:

more task number separated by English comma

Shortcut:

Select the specific task and press *Backspace* or *Delete*, the task will be deleted.

✧ Searching tasks

Format:

show <keywords> *part of description, time, location, tags

show <task group> *all/done/undone/overdue/today/tomorrow/reserved

show ([[<start time>-] <end time>])

Description:

This command enables you to see specific tasks in a time period.

Must-have arguments:

Keywords separated by whitespace/group name/an end time

Optional arguments:

Start time

✧ Reserving time slots

Format:

reserve <task description> (<start time1>- <end time1>) [(<start time 2>- <end time2>)
[<start time 3>- <end time3> [...]]] [@ location] [#<tag>[#<tag> [...]]]

Description:

This command enables you to reserve multiple time slots for later use.

Must-have arguments:

Task description, start time1, end time1

Optional arguments:

Start time2, end time2, location, tags

✧ Confirming time slots

Format:

confirm <task number> [new description] (<time slot number>) [-@] [@ new location] [-
#<existing tag> [#<new tag> [...]]]

Description:

This command enables you to confirm one of the time slots you have reserved and update its information if you need.

Must-have arguments:

Task number, time slot number

Optional arguments:

new task description, removing existing location, new location, removing specific tags, new tags.

✧ Marking as done/undone

Format:

done <1st task number> [, <2nd task number> [...]]

done <task group> *all/done/undone/overdue/today/tomorrow/reserved

undone <1st task number> [, <2nd task number> [...]]

undone <task group> *all/done/undone/overdue/today/tomorrow/reserved

Description:

This command enables you to mark existing tasks as done/undone

Must-have arguments:

group name/at least one task number

Optional arguments:

more task number separated by English comma

Shortcut:

Select the specific task and press *d*, if the task is already marked as done, it will be marked as undone; otherwise it will be marked as done.

✧ Undoing/redoin

Format:

undo

redo

Description:

This command enables you to undo or redo the most recent operation.

Shortcut:

Press Ctrl + z to undo the last operation. Press Ctrl + y to redo the last operation.

✧ Exiting

Format:

exit

Description:

This command enables you to exit.

Shortcut:

Press Esc to exit

CHEAT SHEET

#	Command	Example
1	add <task description> [[[<start time> -] <end time>]] [@ location] [#<tag>[#<tag> [...]]]	add Computer class (every Friday 16:00-18:00) @icube #lecture
2	update <task number> [<new task description>] [- ()] [[[<new start time> -] <new end time>]] [-@] [@ new location] [-#<existing tag> [#<new tag> [...]]]	update 1 SE lecture update 2 -@ update 3 -#lecture #assignment
3	Postpone <task number> <number of days>	postpone 2 3
4	delete <1st task number> [, <2nd task number> [...]] delete <task group>	delete 1,2,3 delete today
5	show <keywords> show <task group> show ([[<start time>-] <end time>])	show #lecture, computer show today show (Apr 1-Apr 3)
6	reserve <task description> (<start time1>- <end time1>) [[(<start time 2>- <end time2>) [<start time 3>- <end time3> [...]]] [@ location] [#<tag>[#<tag> [...]]]	reserve meeting with boss (Tue 15:00-Tue 16:00) (Thu 15:00- Thu 16:00) @lt13 #meeting
7	confirm <task number> [new description] (<time slot number>) [-@] [@ new location] [-#<existing tag> [#<new tag> [...]]]	confirm 2 meeting with manager (2) -@
8	done/undone <1st task number> [, <2nd task number>	done 1,2,5

	[...] done/undone <task group>	done today undone done
9	undo redo	undo redo
10	exit	exit

**Group name: all/done/undone/overdue/today/tomorrow/reserved*

**Keywords: part of description, time, location, tags*

Appendix A: User stories. As a user, ...

[Likely]

ID	I can ... (i.e. Functionality)	so that I ... (i.e. Value)
newFile	specify a specific folder as the data storage location	can use a cloud syncing service to control his to-do list and access task data from multiple computers
addTimed	add a task with a start time or end time/due or recurring time	can record tasks that must be done in a period of time
addUntimed	add a task by specifying a task description only	can record tasks that I want to do some day
setLocation	set location for a task	can see where to do the task in the list
setTag	set tags for a task	can add the task to some category
updateATask	change the details of a task	can update the latest information
postponeATask	postpone a task by setting deferred time	can postpone a task more easily
markAsDone	mark a task as done	can pay attention to other ongoing tasks
markAsUndone	mark a task as undone	can resume a task that has been closed
deleteTasks	delete a task	can remove tasks that I no longer need to do
searchByGroup	see the list of my tasks by group (all, undone, done, today, tomorrow, overdue, reserved)	can see some specific tasks
searchByKeywords	search my tasks by keywords (including task description, time,	can find some specific tasks easily

	tags and locations)	
searchPeriod	search my tasks that must be done during a certain period time	can see my schedule during a certain period of time
undoLast	undo last operation	can quickly recover from a recent mistake
redoLast	redo the last cancelled operation	can quickly recover from a recent undo mistake
sortByUrgency	always display the whole list in order of urgency (overdue, today, tomorrow, timed, untimed, reserved)	can decide what to do better
quickLaunch	launch the app by shortcut	can activate it more quickly
GUI	view the tasks in a graphical user interface	can see the list more directly
useSynonym	use some natural variations of the command format	can type faster
reserveTimeSlots	reserve time slots for later use	can notice the reservation and won't arrange other tasks by mistake.
confirmTimeSlot	confirm one of the reserved time slots	can deal with tasks with uncertainty more easily
offline	use the app without internet connectivity	can modify its tasks list anywhere and everywhere.

[Unlikely]

ID	I can ... (i.e. Functionality)	so that I ... (i.e. Value)
guidance	see a tutorial when using the app the first time	can learn how to use the app better and quicker

suggestCommand	see suggestions when I'm typing my command	can no longer memorize commands' format
printList	print the daily/weekly/monthly list of tasks	can take the hardcopy along anywhere
attachFile	attach files to tasks	can be more efficient in doing things
createWorkFlow	create workflow	can break it into sub-tasks and set different completion date when he has a task to be done before a specific time

Appendix B: Non Functional Requirements

1. The software should run on Windows 7 or later with at least a Pentium III processor, at least 512MB of RAM and approximately 20MB of free space.

Appendix C: Product survey

Product: Todotxt.net **Documented by:** Zhu Bingjing

Strengths:

1. it's a desktop application which can be activated quickly
2. Portable and offline
3. One-shot approach. Users can type all the information at a time.
4. Quite powerful and various types of operation with keyboard shortcuts.
5. Can deal with tasks without a certain deadline or start time
6. Mark things done easily by pressing X
7. Enable priority so that users can decide what to do more easily
8. Postponing a task by a day can easily be done by pressing Ctrl+Alt+Up

9. Automatically translate due date, e.g. set due: Friday→ due:2016-02-05

Weaknesses:

1. Cannot deal with events with a start time and end time.
2. Cannot withdraw last operation.
3. Lack of a good way for users to find suitable slots
4. Do not facilitate multiple slots to be “blocked” when the exact timing of a task is uncertain
5. Too many operations with keyboard shortcuts that users can hardly remember and the shortcuts cannot be d by users.
6. Some of the functions are redundant or trivial or useless,.

Redundant

e.g. No need to use ***filters*** in searching. Keyword search is enough.

Trivial

e.g. the function ***postpone*** which requires inputting number of days I think is already involved in the function update.

Useless

e.g. the function ***show calendar...***
7. Always listing all the tasks don’t seem to be neat and simple

Product: OmniFocus **Documented by:** Her Kai Lin

Strengths:

1. Able to create different project, and then subtask of projects
2. Different view for different context
3. Can sort with due dates or flag priority
4. Set due date, estimation time to complete.

5. attach files with regards to the project
6. defer date if can't complete on time / change
7. mark completed / completed time

Weaknesses:

1. buttons too small to press
2. cannot specify custom recurring task
3. very troublesome to key, have to press a lot of buttons to change information.
4. can be confusing If did not read the manual
5. no alarms to inform

Product: Google Calendar **Documented by:** Boh Tuang Hwee, Jehiel

Strengths:

1. Calendar view
2. Daily/monthly/weekly view
3. Separates Timed events from floating tasks
4. Quick add events
5. Enable/disable floating tasks panel
6. Has agenda list
7. Able to set recurring events
8. 1 click to mark tasks as done
9. Search events/tasks by keyword
10. Allows printing
11. Small calendar at side for easy selection of days
12. Drag and drop to edit event timings
13. Extension of event duration by dragging box

Weaknesses:

1. Requires internet access
2. View can get cluttered if there are many events

Product: Remember the Milk **Documented by:** Chloe Fortuna

Strengths:

1. Is able to sort the list by priority, due date, and task name
2. Is able to search for tasks using keywords
3. The interface shows how many tasks are overdue, left to do, and completed
4. Is able to create different lists based on category
5. Highlights which tasks are overdue so it is easy to see
6. Is able to use the keyboard to navigate through the lists
7. Allows the user to input tags for each task for easier searching and categorizing
8. Has an easy undo option
9. Is able to share and print the list
10. Is able to sync with the phone in order to see the list on the go
11. Is able to send reminder notifications for the user
12. Is able to create a task via email

Weaknesses:

1. The keyboard shortcuts are very difficult to remember
2. You have to make an account to use this program
3. You can only access the list when you have Internet
4. You have to manually edit a task after adding it. You can't store all the information with only one click
5. You have to wait through a long loading screen before accessing the list

- | |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <ol style="list-style-type: none">6. Creating a task via email has a lot of constraints7. Many features are limited unless you pay for the full version |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|