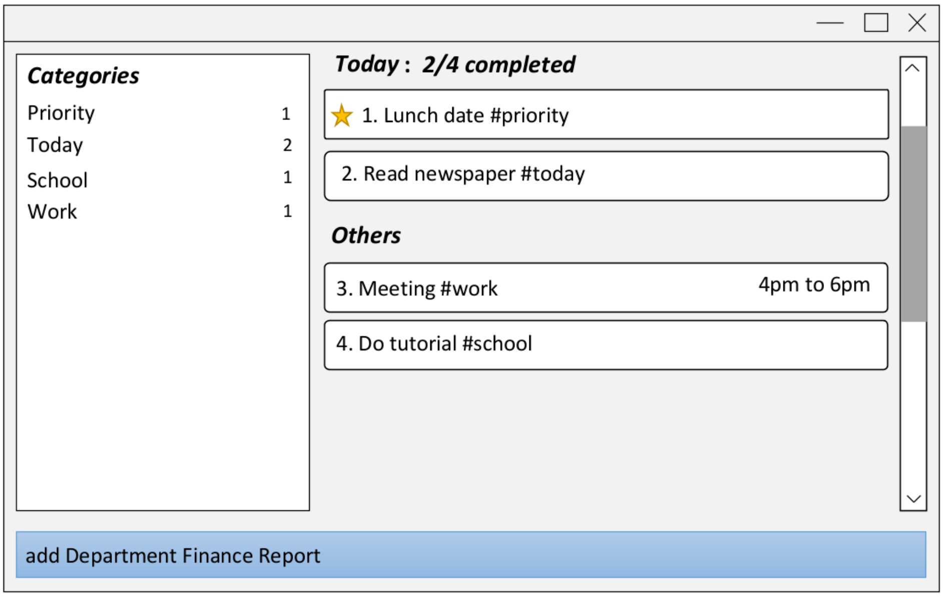
[W09-4j][V0.0]

GRIDtask

Before adding a new task:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

add Department Finance Report

Supervisor: Michelle Tan **Extra feature**: Autocompletion

Roles and Responsibilities

Trinh Ngoc Khanh Team Leader Code Quality	**Gilbert Chua** Documentation	**Quek Ruo Ling** Integration and Testing	**Tan Ching** Scheduling and tracking Deliverables and deadlines

User Guide

About the product

Getting bogged down by all those seemingly never-ending tasks? Always wanted something simple to use yet helps you get the job done on time? Fear no more, GRIDtask will help you **G**et **RID** of all your **task**s from now on!

GRIDtask is a software that allows you to keep track of your tasks in your to-do list through many basic functions such as add, delete, etc. There are also additional features such as grouping your tasks based on different categories to aid in organisation and inform you of tasks of higher importance.

Contents

Feature Details

Adding tasks

There are three types of task you can add into your to-do list:

1. Events with start and end timing
2. Floating task (a task with no specific deadline)
3. Task with deadlines

To add a floating task without any dates or timing involved, use this command format:

add <task description>

Example: "add watch lord of the rings marathon"

To add events with start and end timings, use this command format:

add <task description> start <date/time> end <date/time>

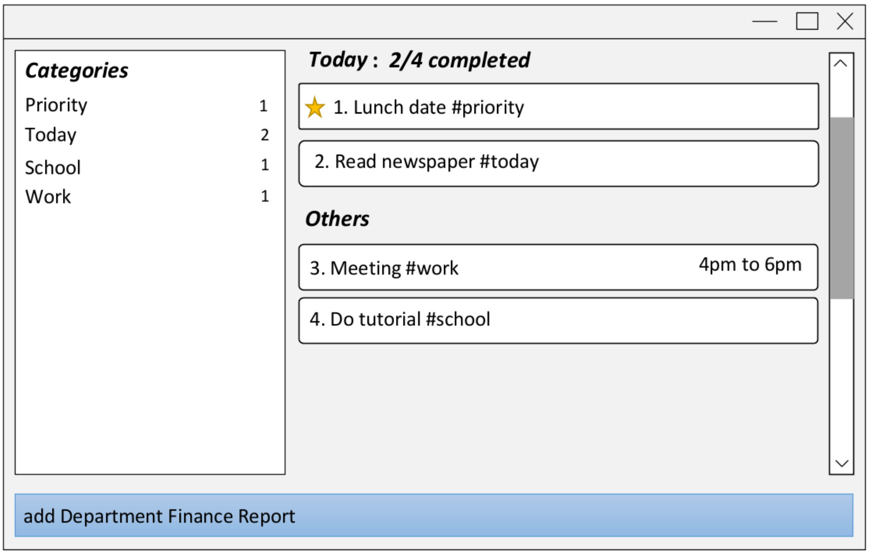
Example: "add meeting start 4pm end 6pm"

To add tasks with deadlines, use the following command format.

add <task description> due <date/time>

Example: "add cs2103 assignment due 03/02/2016"

Before adding a new task:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

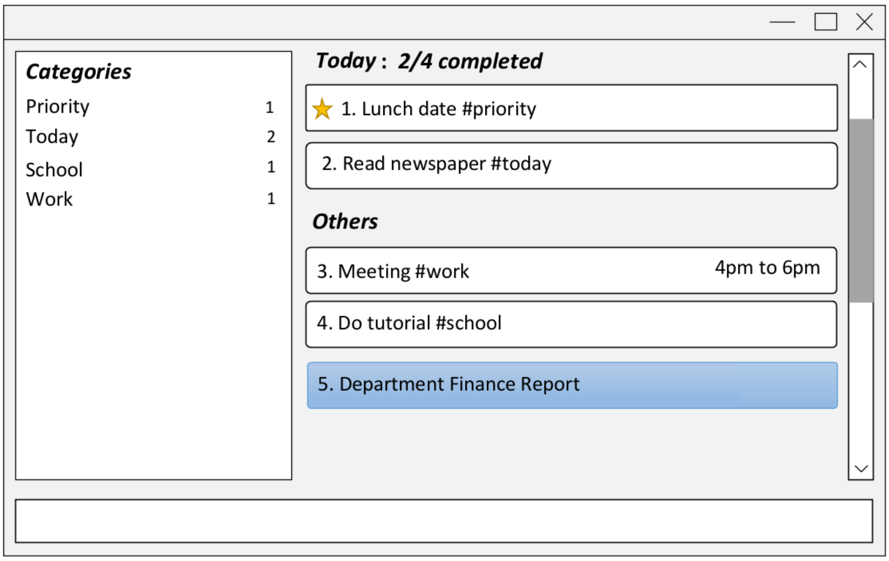
Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

add Department Finance Report

After adding a new task:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

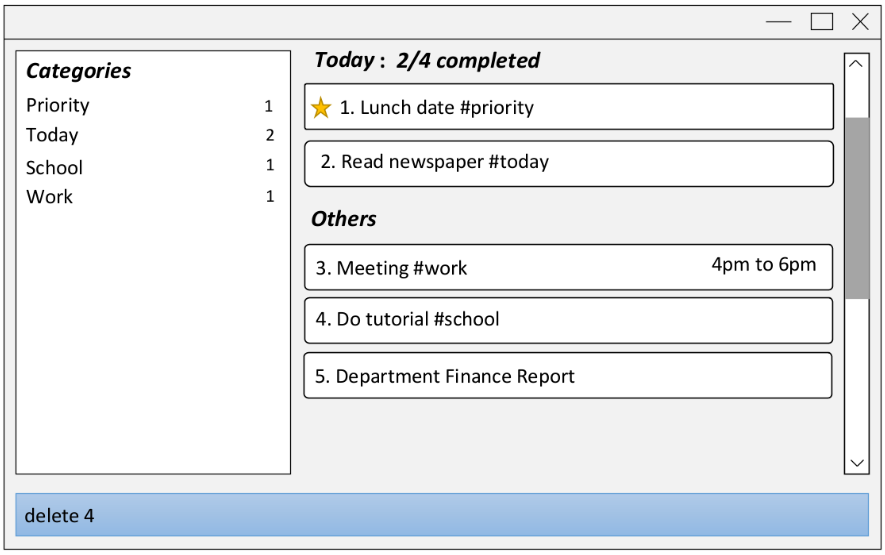
Deleting tasks

You can delete a task if you do not have to do it anymore. To delete a task, use the following command:

delete <task ID>

where <task ID> is the ID of the task to delete

Before deleting task 4:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

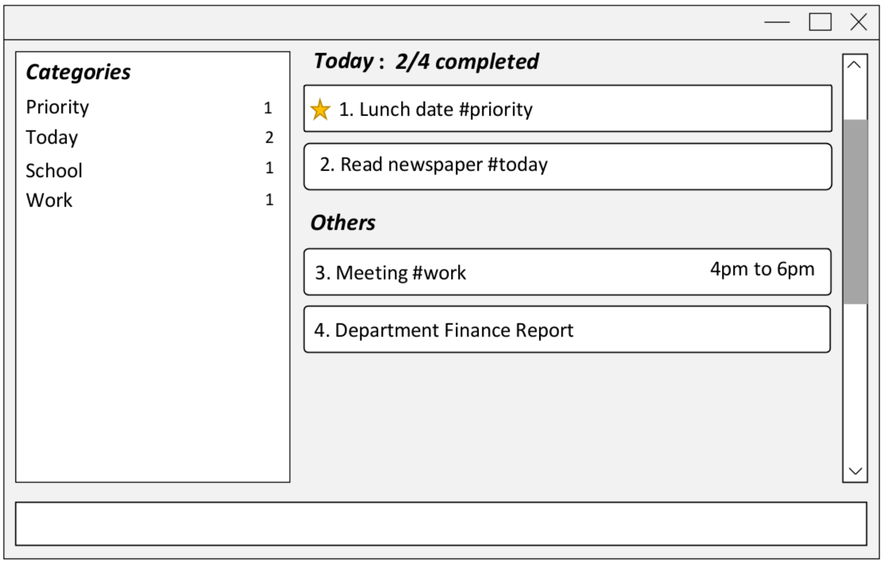
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

```
delete 4
```

After deleting the task:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Department Finance Report

Editing tasks

You can edit the details of a task.

To edit a task with a new description, use this command format:

edit <task ID> <new description>

where <task ID> is the ID of the task to delete

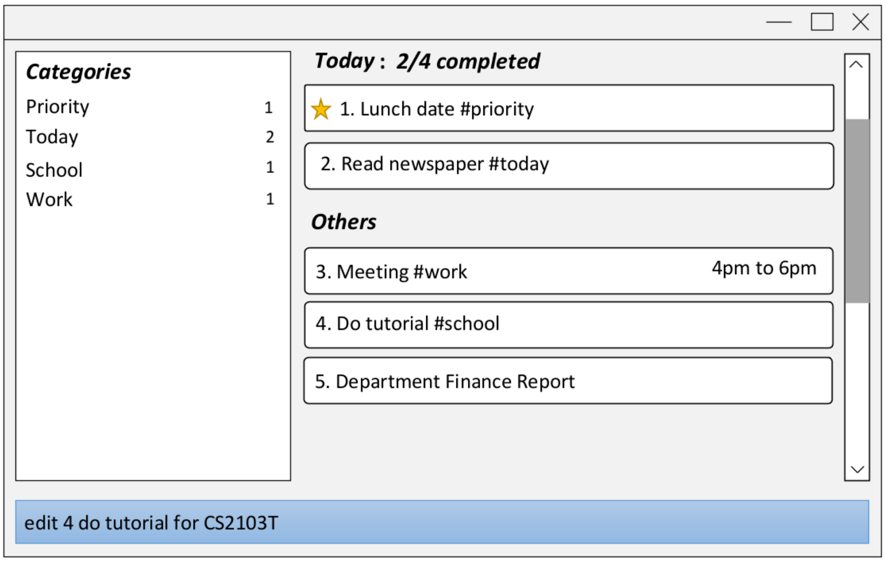
To update a task with a new timing, use this command format:

edit time <task ID> <new time/date>

where <task ID> is the ID of the task to delete

Example: **edit time 15 11/2/2016**

Before editing task 4:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

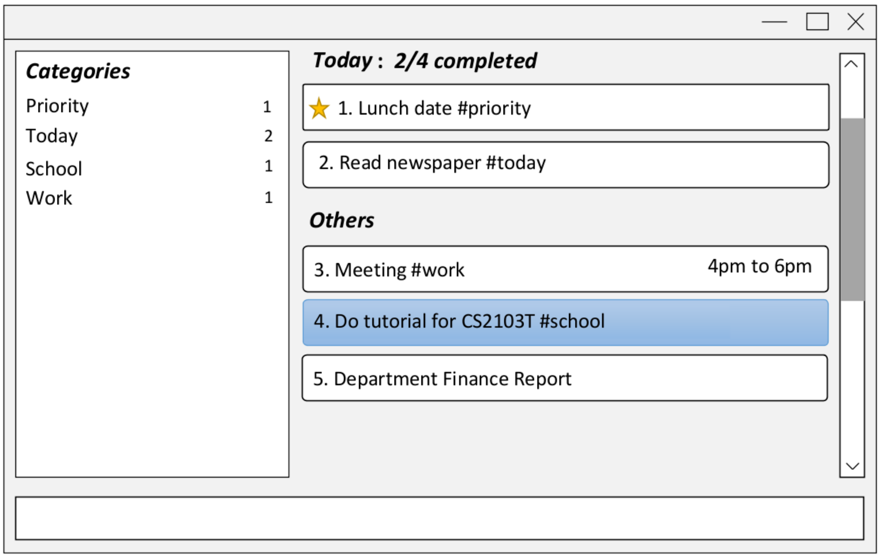
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

edit 4 do tutorial for CS2103T

After editing the task:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial for CS2103T #school

5. Department Finance Report

Searching Tasks

You can also search for your task in the list through the following three ways:

- category
- date
- keyword of the task description

To search for tasks with the same category, use this command format:

search #categoryName

Example: search #family

The search results will show all the tasks that belong to the particular category.

To search tasks using the date, use this command format:

search <date>

Example: search 11/02/2016

The search results will show all the tasks containing this date.

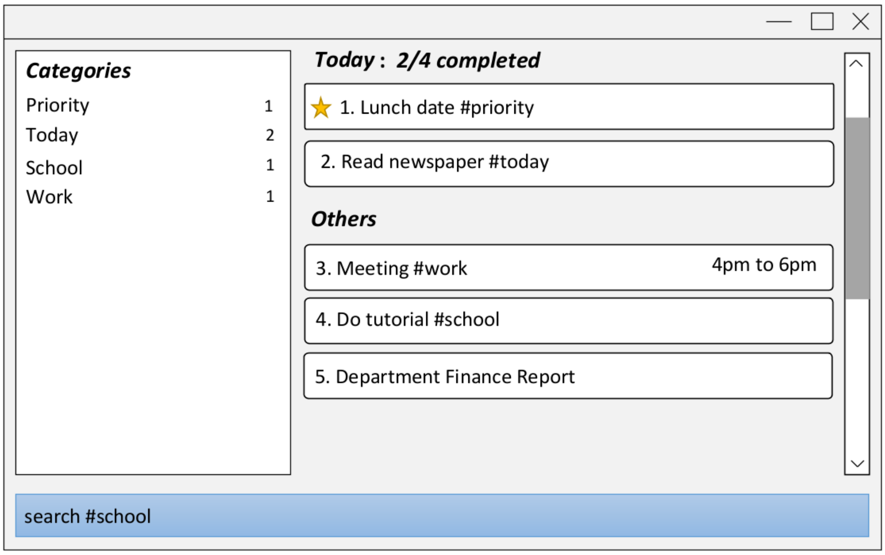
To search for a specific task using keywords in the task description, use this command format:

search <keyword>

Example: search meeting

The search results will show all the tasks containing the keyword.

User tries to search for task that belong to the category - school:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

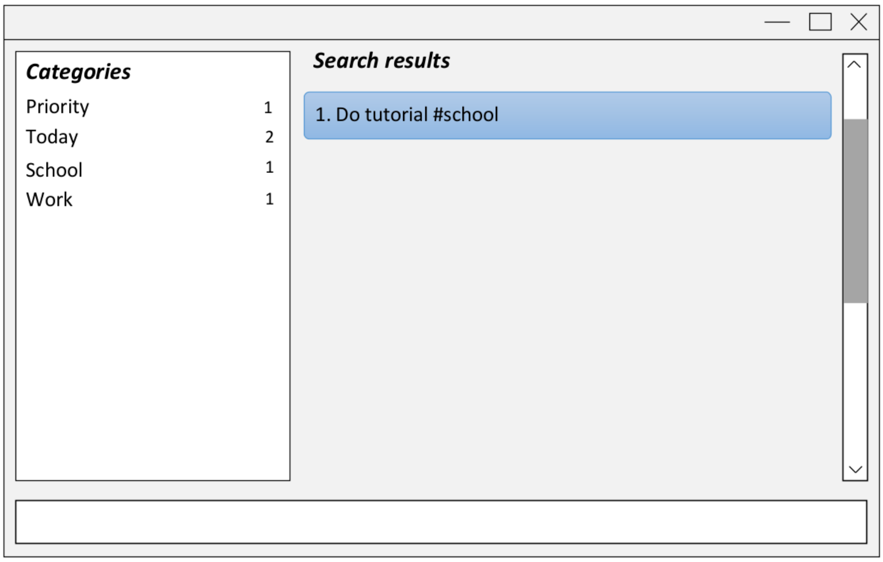
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

search #school

Search results shows all the task that belong to the category – school:



Categories

Priority	1
Today	2
School	1
Work	1

Search results

1. Do tutorial #school

Undo and Redo

Made a mistake? To cancel the previous command, use this command format:

undo

Changed your mind again? To repeat the cancelled command, use this command format:

redo

Saving

Data is automatically saved every time you make a change in any task list. The change can be any command that causes an update (add/edit/delete/undo/redo).

To save data to a specific file, use this command format:

save <name of file>

GRIDtask will check if such a file with the name <textfilename> exists within the same directory as the program.

If it exists then the data is saved to the file, overwriting any previous data inside the file.

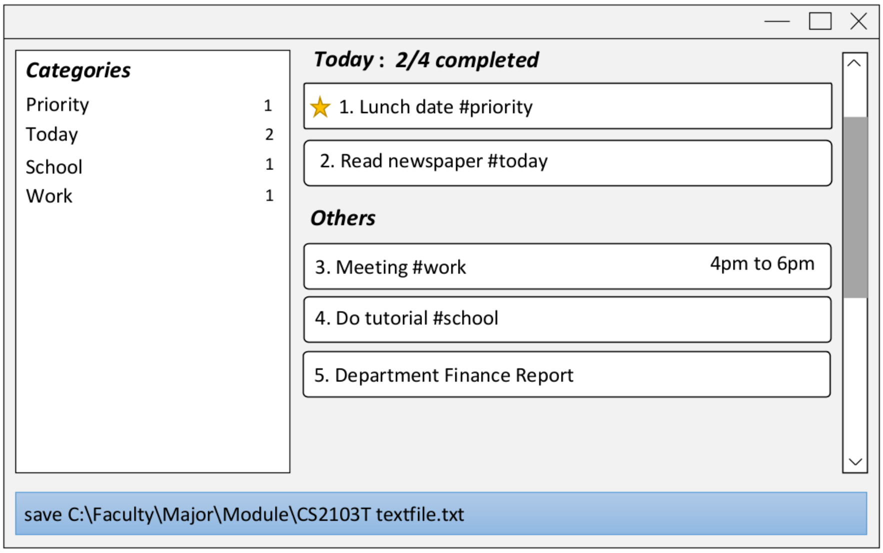
If it does not exist, then GRIDtask will create a new file with <textfilename> as the file name within the program's directory.

Also, it is possible for you to specify the directory that you want to save the data into. Use this command format:

save <directory> <name of file>

GRIDtask will search for this directory first. If the directory exists, GRIDtask save the file in that directory. Otherwise, GRIDtask outputs an error message saying that the directory is invalid.

Specify path name when saving:



```
save C:\Faculty\Major\Module\CS2103T textfile.txt
```

Loading

To load a file, use this command format:

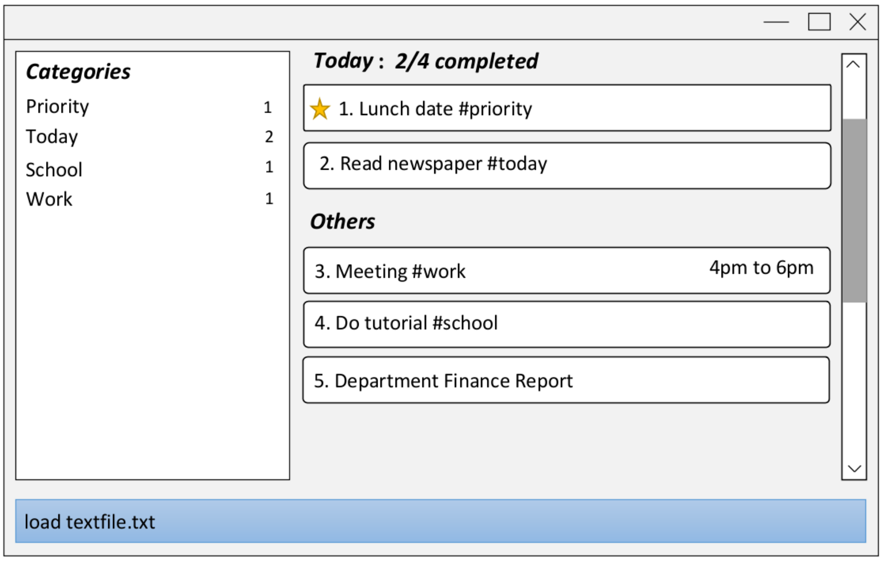
load <name of file>

GRIDtask searches for an existing file with this name within the program directory.

If the file exists, GRIDtask will use the data inside the file, and you may continue updating the file through the basic updating commands.

If the file does not exist, GRIDtask will return an error message, and asks you if you would like to create a new file with this name.

Loading:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

```
load textfile.txt
```

Also, it is possible for you to specify the directory that you want to load the data from. Use this command format:

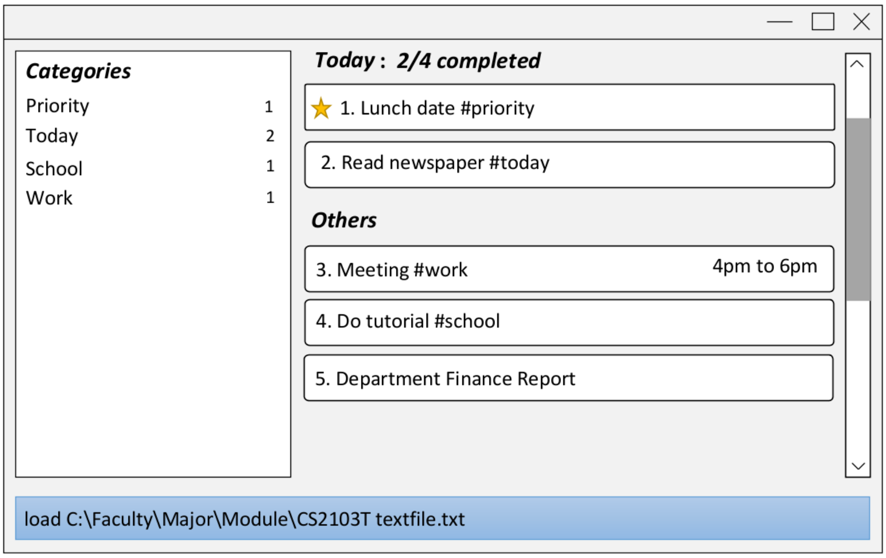
load <directory> <textfilename>

Similar to the saving feature, GRIDtask will search for this directory first.

If it exists then GRIDtask will load this file.

If it does not exist, then GRIDtask will output an error message saying that the directory is invalid.

Specify path name for loading:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

```
load C:\Faculty\Major\Module\CS2103T textfile.txt
```

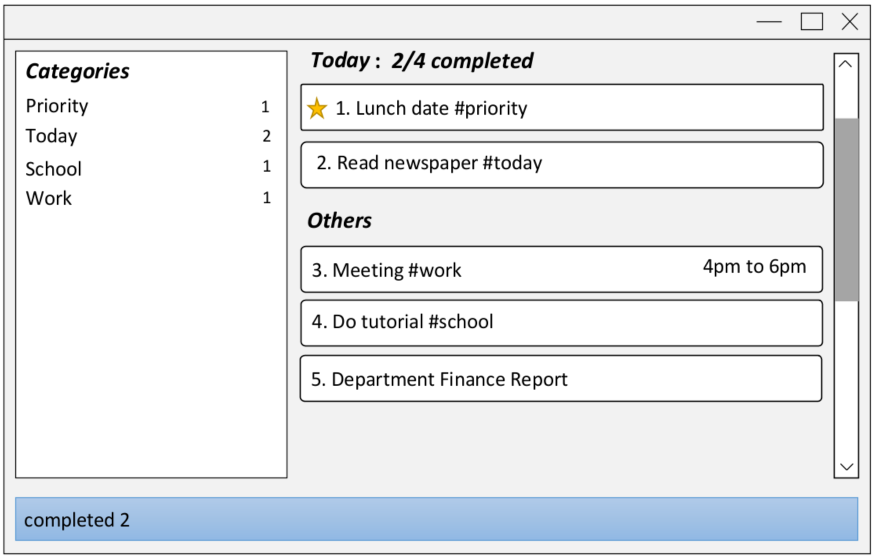
Completing Tasks

Once you have completed your task, you can mark it as completed. Use this command format:

completed <task ID>

Example: completed 2

If the user completes task 2 and wants to remove it from the list:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

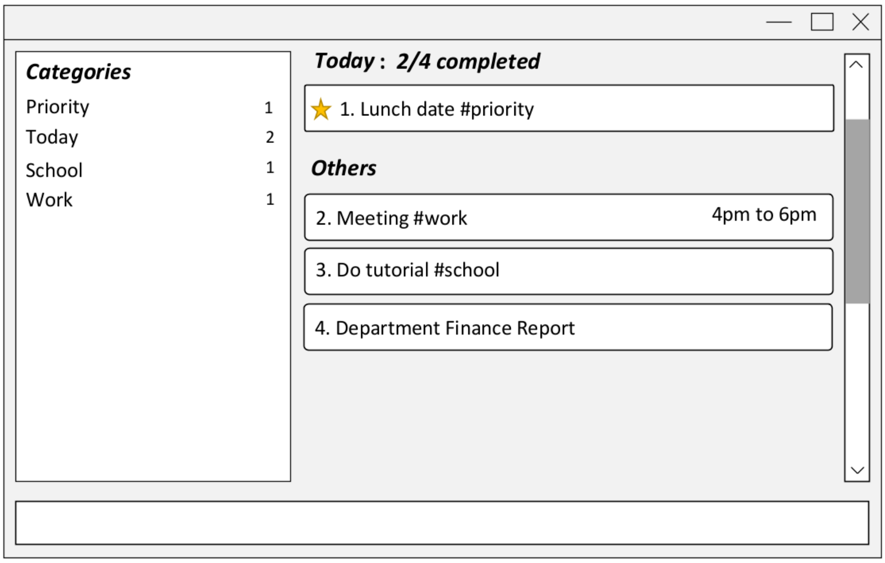
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

```
completed 2
```

Updated task list after the user enters the command:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

Others

2. Meeting #work 4pm to 6pm

3. Do tutorial #school

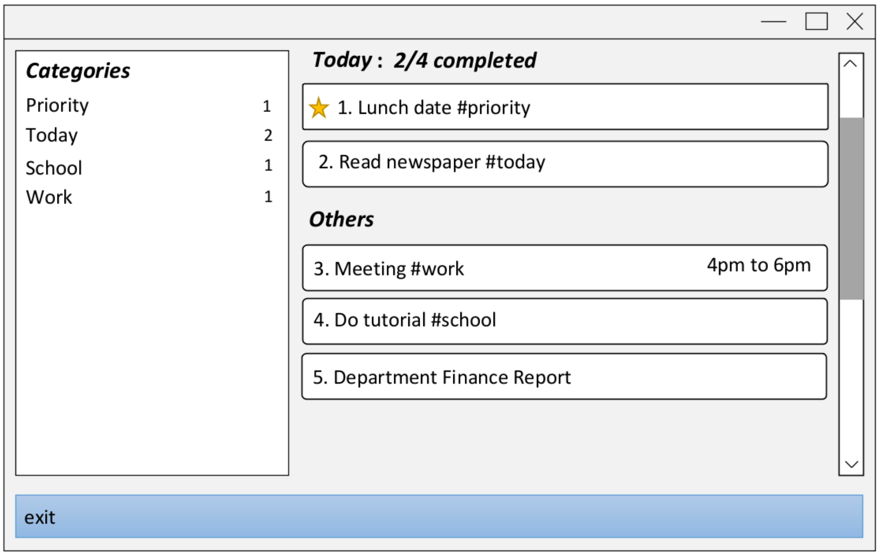
4. Department Finance Report

Exiting

To quit GRIDtask, use this command format:

exit

Exiting the program:

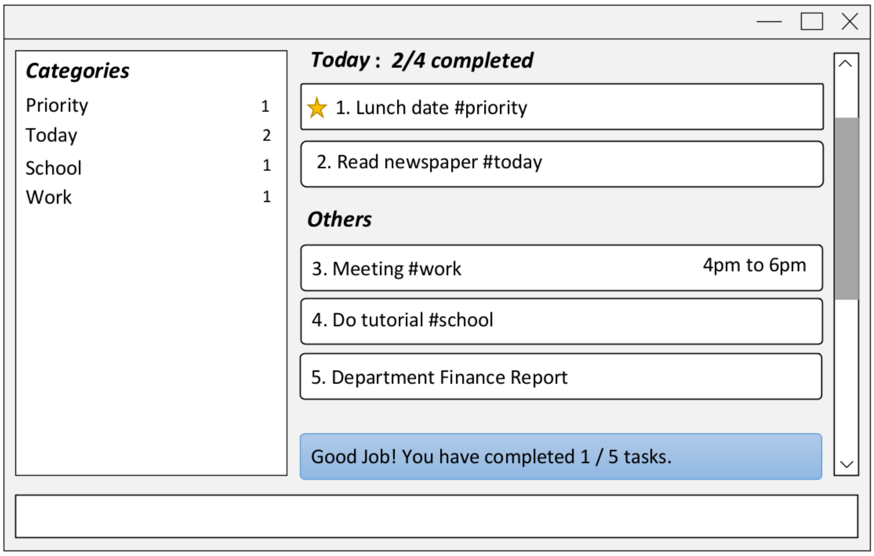


exit

Task Progression

GRIDtask will tell you how many tasks you have completed so far. Feedback will appear whenever you complete a task. This will give you a sense of satisfaction as well as an idea of how much work is left.

After the user completed the task:



Good Job! You have completed 1 / 5 tasks.

Categories

You can make your own categories for easy organisation using this command format:

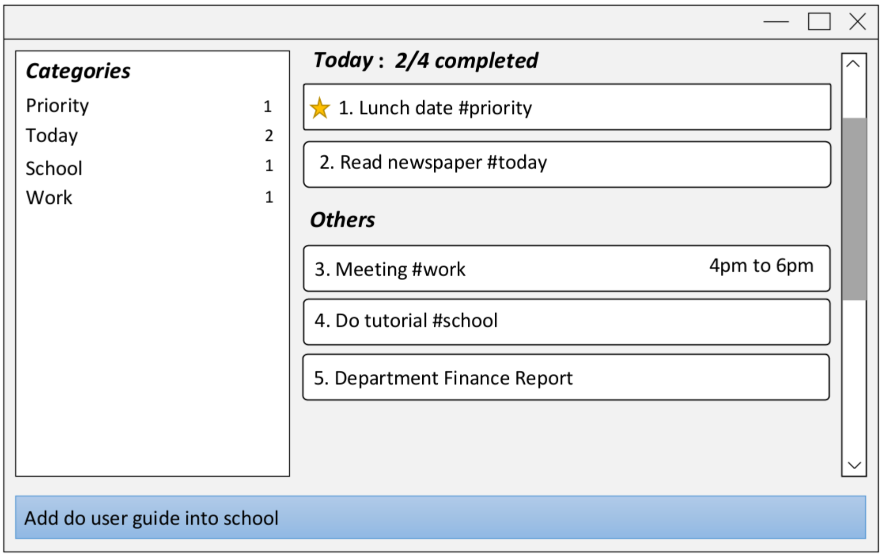
addcat <category name>

View all categories in the left panel.

To use a category when adding a task, use this command format:

add <task description> into <category>

Adding a task into a category:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

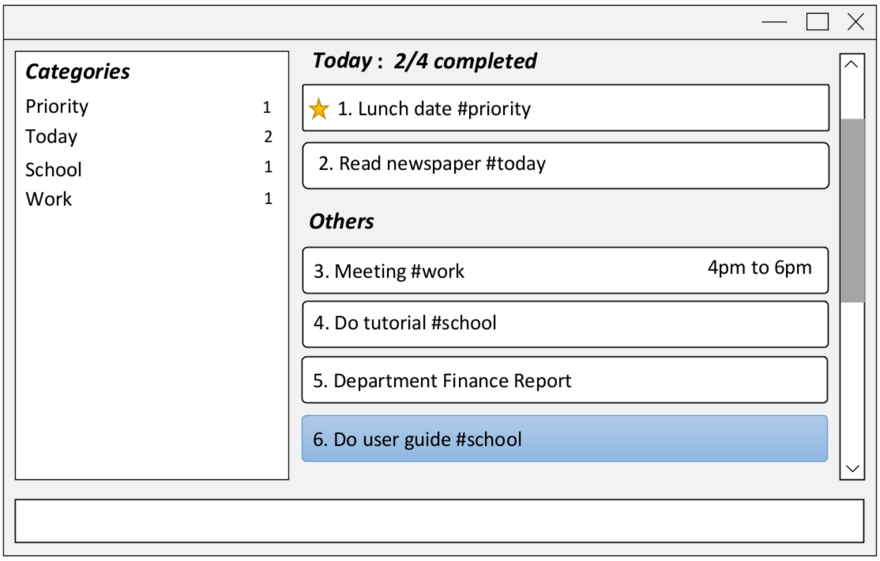
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

Add do user guide into school

After adding the task into the category, it is indicated by the hashtag:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

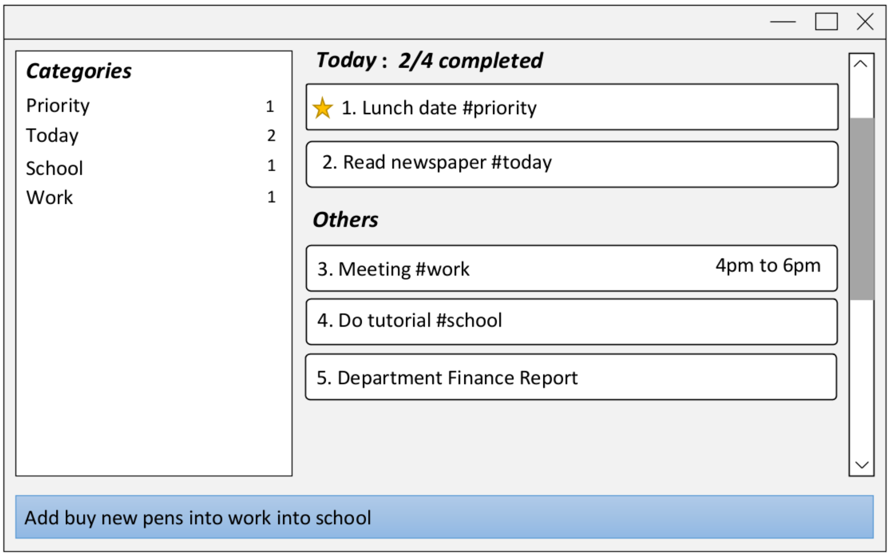
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

6. Do user guide #school

You can put a task into more than one category:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

Add buy new pens into work into school

Rename a category using this command format:

editcat <category name>

Delete a category using this command format:

deletecat <category name>

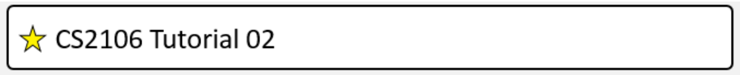
GRIDtask will let you choose if you want to delete all the tasks under that category, or keep them (all other categories will remain intact).

Priority tasks

Priority is a special type of category. Use it to mark tasks that are important using this command format:

add <task> into Priority

An important task will appear in the main panel with a star, like so:



☆ CS2106 Tutorial 02

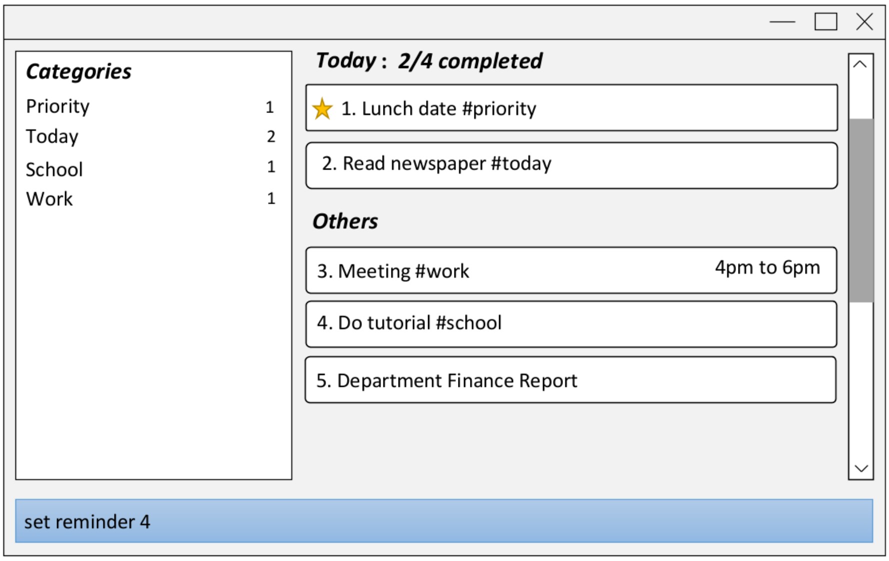
Reminders

To set a reminder for a task, use this command format:

set reminder <task ID>

GRIDtask will list out the possible times for the reminder:

Setting reminders for a task:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

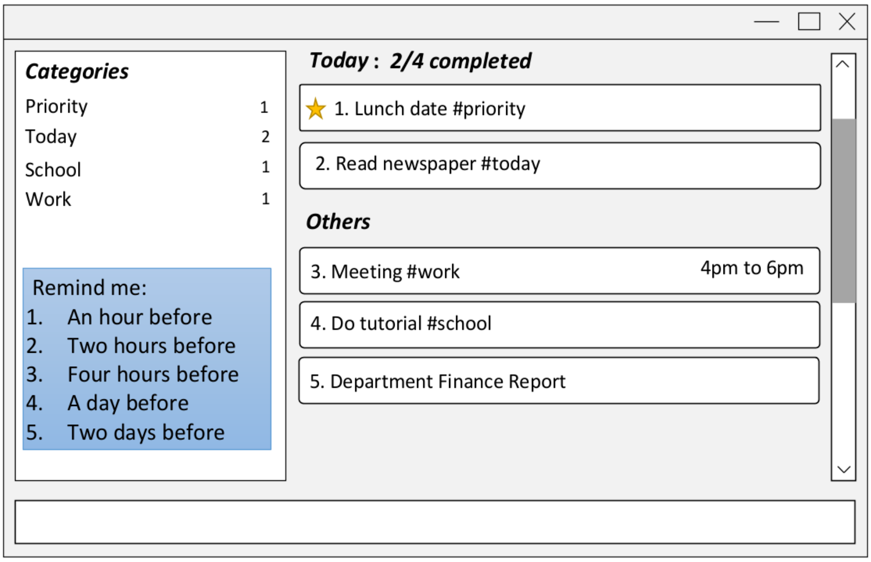
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

set reminder 4

Setting reminders for a task:



Categories

Priority	1
Today	2
School	1
Work	1

Remind me:

1. An hour before
2. Two hours before
3. Four hours before
4. A day before
5. Two days before

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

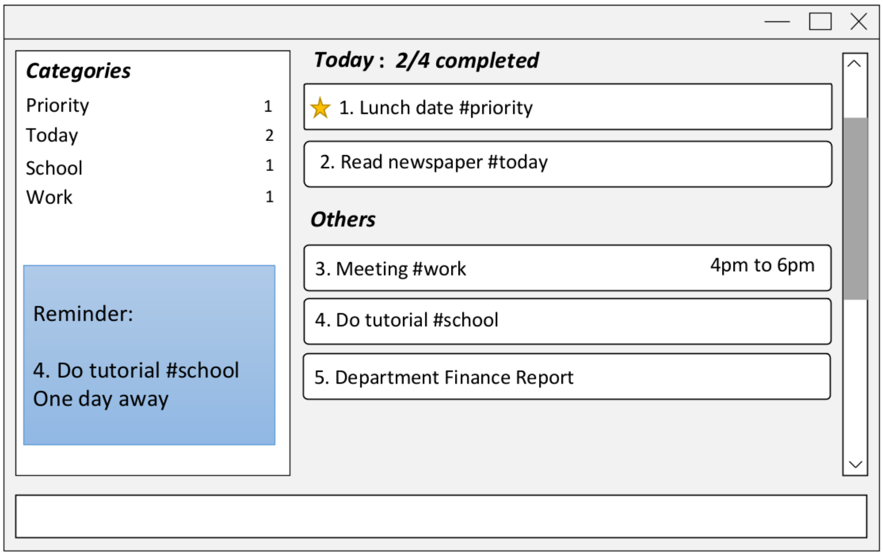
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

When a reminder time is due, a notification will pop up on the screen.

Setting reminders for a task:



Categories

Priority	1
Today	2
School	1
Work	1

Reminder:

4. Do tutorial #school
One day away

Today : 2/4 completed

☆ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

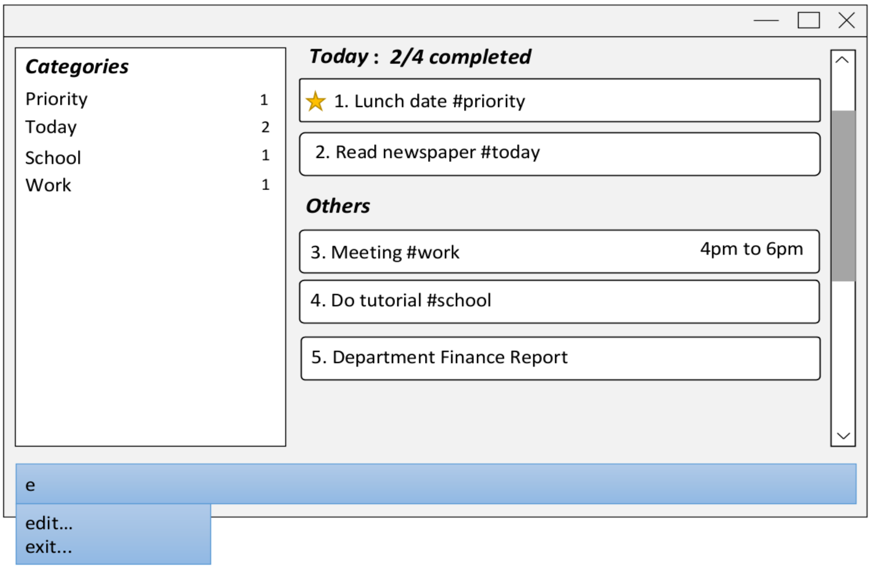
4. Do tutorial #school

5. Department Finance Report

Auto completion

This feature is used to facilitate faster speed for you when keying your task and also make the software more convenient for you.

Auto completion:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

☆ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

e

Access completed tasks

You can access completed tasks using this command format:

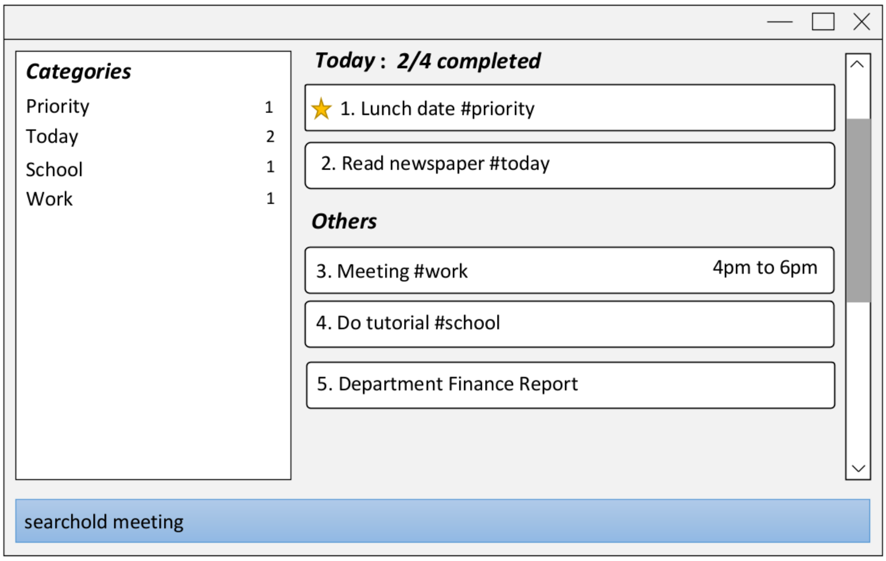
searchold …

where the remaining text is the search parameter. It works like search, but just on the set of completed tasks.

Example: searchold <keyword>

This searches for completed tasks that contain <keyword> in their description.

Searching for completed task using the keyword:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

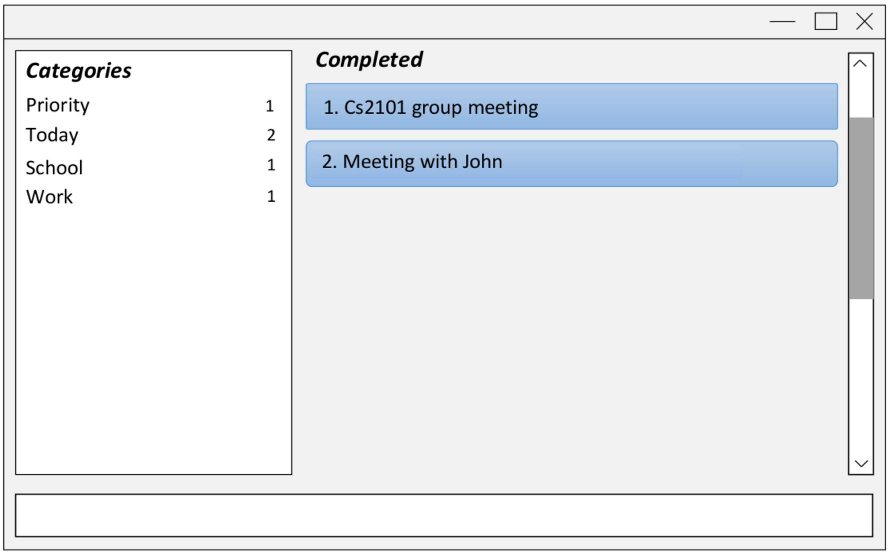
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

searchold meeting

Search results for completed tasks:



Categories

Priority	1
Today	2
School	1
Work	1

Completed

1. Cs2101 group meeting

2. Meeting with John

To search for completed tasks of a specified date, use this command format:

searchold <date>

Example: searchold 22/11/2010

To search for completed tasks belonging to a particular category, use this command format:

searchold #category

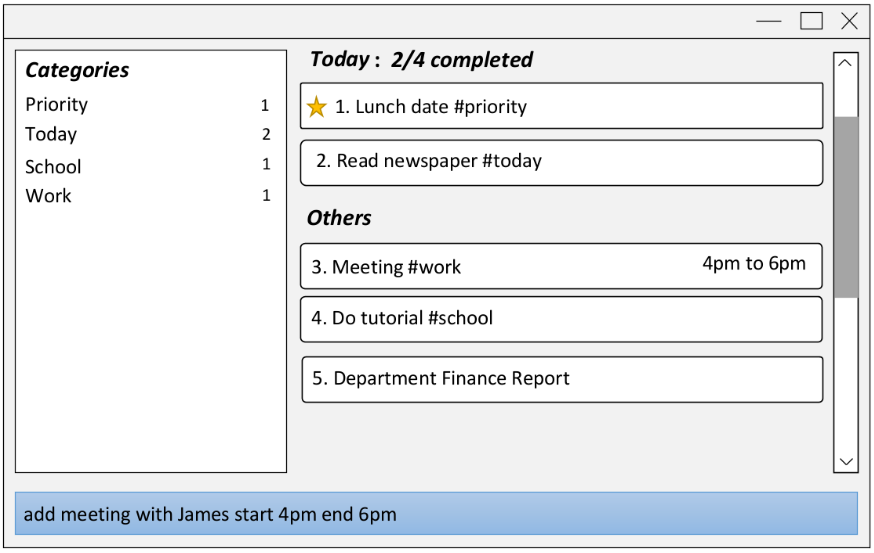
Example: searchold #work

Clashing Tasks

If you add a new task or make changes to a task and it causes a clash in timing, GRIDtask will prompt you if you want to continue.

***Example*:** when adding a new task results in a clash

After adding a new task that clash:



Categories

Priority	1
Today	2
School	1
Work	1

Today : 2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

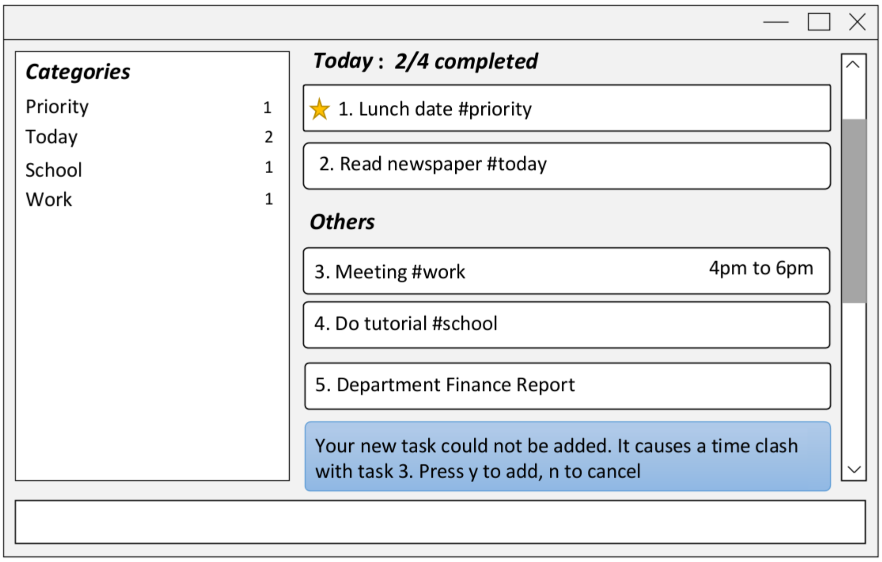
3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

add meeting with James start 4pm end 6pm

Feedback the user the clash of timing:



Categories	
Priority	1
Today	2
School	1
Work	1

Today* : *2/4 completed

★ 1. Lunch date #priority

2. Read newspaper #today

Others

3. Meeting #work 4pm to 6pm

4. Do tutorial #school

5. Department Finance Report

Your new task could not be added. It causes a time clash with task 3. Press y to add, n to cancel

Tutorial Guide

This tutorial guide will teach you the basic commands for GRIDtask, such as adding, deleting, and editing tasks. When you open the program for the first time, the tutorial will be shown to you.

You can access it subsequently using this command format:

help

Cheatsheet

Command	Description
add <task description>	add a floating task into the list
add <taskDescription> start <date/time> end <date/time>	add a task with start and end timings
add <taskDescription> due <date/time>	add a task with a deadline
add <task description> into <categoryName>	add a task into a category
add <task description> into <category name> into <category name> ...	add a task into multiple categories
delete <task ID>	delete a task
edit <task ID> <new description>	edit the description of a task
edit time <task ID> <new time/date>	edit the timings of a task
search #<category name>	search tasks by their category
search <date>	search tasks by their deadline or date
search <keyword>	search tasks containing the keyword
completed <task ID>	mark the task as completed
load <text file name>	load an existing file, or create a new file with that name if file does not exist within the program's directory
exit	exit the software
addcat <category name>	create a new category
editcat <category name>	rename a category
deletecat< category name>	delete a category
searchold <keyword>	search for completed tasks that contains the keyword
searchold <date>	search for completed tasks that is associated with the date
searchold #category	search for completed tasks that belong to a particular category

Appendix A: User Stories

[Likely]

ID	I can … (i.e. Functionality)	so that I … (i.e. Value)
add	create a new task	record tasks that I intend to do
edit	modify the task description	update the task if there is any changes
delete	delete a task	can remove unnecessary tasks
undo	undo the most recent command	can correct my previous error without much effort
redo	redo the most recent command	can get back what I have accidently deleted/lost
automatically save	sync my entries across devices	do not have to update the list manually when I use the software on multiple devices
save	save from a file	still have offline access to my data
load myfile.txt	load from a file	still have offline access to my data
addDate	can add deadlines to task	am aware of how urgent the task is
addReminder	add reminders to my tasks	be reminded of upcoming tasks or events
category - Completed	access completed tasks	know what has been done
feedback on completion rate	see my completion rate	know how productive I am
addCategory	set my own categories	can be more organised
category - Important	set high priority tasks	know which are more important
display	see tasks that need to be done next	know what task is due next

search	search for specific entries	can find them easily
searchKeyword	search tasks by description	search for the task without the need to remember the whole task description
searchDate	search tasks by date	access all the tasks with the same deadline/date
searchCategory	search by categories	get all the tasks belonging to the category
searchNearMatch	find results that are close but not exactly what I search for	can search for things that I don't remember exactly
autoComplete	have my commands suggested before I finished typing	save time and have to type less
tutorial guide	understand more about the software by reading the tutorial	know how to use the software to meet my needs

[Unlikely]

ID	**I can … (i.e. Functionality)**	**so that I … (i.e. Value)**
subtask	create subtasks under a task	add additional details or pre/post-tasks to my main task
editMultipleTasks	edit a few tasks at the same time	save time to edit my to-do list
addRecurringTask	set recurring tasks and events	do not have to key them in again
informClash	see when events clash in timing	can reschedule them appropriately
shortcut	use a shortcut to activate the software	can access it quickly
editRecurringTask	edit a single recurring task without affecting the rest	give it slightly different details
duplicate	check if there's any duplicates in the list	do not add redundant tasks
customisation	customise my own commands	am more comfortable and familiar with the software

Appendix B: Non-functional Requirements

Usability

GRIDtask should have a reasonable start up time.

GRIDtask should be able to handle all invalid inputs.

The user is able to use GRIDtask without any programming knowledge.

Compatibility

GRIDtask is written in Java.

GRIDtask runs on Windows and OS X platforms with minimum Java version 7 installed.

GRIDtask is an open source

Appendix C: Product Survey

Note: Highlighted points are not applicable to our software or in the context of the project.

Product: Todoist (Android)

Documented by: Gilbert Chua

Strengths:

- Beautiful UI
- Creating tasks is simple
- Auto-completion for some words such as '*today*', '*tomorrow*', '*friday*', '*next week*'
- Given a string such as '*next friday, 20 feb 9am*', it can auto-detect which date and time the user has specified
- Subtasks can be added to existing tasks
- Select multiple tasks simply by tapping on each task. No need to tap an additional button to activate multiple selection
- There are a variety of categories available by default. The user can even create custom categories

Weaknesses:

- Does not inform the user whether the task has been created when the user submits the task
- Unintuitive, requires the user to explore features
- Features such as setting reminders and writing additional descriptions require a premium account
- Once created, tasks cannot be recategorised

Product: Any.do (Web and iOS)

Documented by: Tan Ching

Strengths:

- The user can create custom categories based on preference (eg: work / movies to watch / personal / all) [can also delete / edit list]
- The user can set the timing of an event
- Subtasks, notes and attachments can be added to a task
- The user can set recurring tasks (once/daily/weekly/monthly/yearly)
- Able to mark as priority and view
- Able to set reminder / send notification for mobile app
- Able to sign in using facebook account (convenient)
- Adding and deleting tasks in different categories are easy as only requires one click (there is a 'add' button for each category and completed button for each task)
- Feedback on the total number of tasks completed - > "great job" / able to share on facebook
- Tasks can be recategorised
- Able to view the list by three ways
    - time (today/tomorrow/upcoming/someday)
    - priority (high / normal)
    - list tasks according to their category
- Tasks can be exported or printed
- Tasks can be synced across different devices
- Can check task that are completed and clear the list
- Search for tasks

- Provides help to the user (mobile version)
    - Short tutorial on how to use the app (eg: swipe right to complete / tap and hold to drag me around)

Weaknesses:

- Some features such as choosing different themes are premium
- Extra step is needed to delete a task after it has been marked as completed (mobile version)
- No undo function

Product: Wunderlist (Windows)

Documented by: Quek Ruo Ling

Strengths:

- Intuitive for a new user due to its simplicity
- Clean and aesthetically pleasing interface
    - Feedback on actions using elements such as highlighted buttons and hovertext
    - Animations are used as visual aids and do not slow down the user
- Hashtags can be used to label related tasks. They stand out from the rest of the text as they are coloured and underlined
- Search function
- The user can rearrange tasks in the list
- Support for due dates, reminders, recurring tasks, subtasks, notes and attachments

Weaknesses:

- Sends email reminders by default, cluttering up the user’s inbox
- Setting the due date and reminder for a task is troublesome
    - No way to set it during task creation, the user has to click into the task
    - The user has to select the date and time from a calendar and time-picker. No way to key in the date and time directly
- Hashtags are not used to full potential. They could have functioned as categories
- Subtasks are not visible from the main view of tasks
- Slow to navigate with just a keyboard, limited keyboard shortcuts
- No undo function
- A confirmation dialog appears every time the user deletes a task

- The user’s rearrangement of tasks is not saved, so once sorted by the program, the configuration is lost

Product: Google Calendar (Web-based)

Documented by: Trinh Ngoc Khanh

Strengths:

- Free to use
- Can sync across devices and access everywhere with Internet
- It is global, events do not change time even if you change time zone.
- Can sync with other Google services
- Can share with other people and use within an organization
- Can have multiple calendars for different categories
- Have reminders
- Great customisation features: color code events and calendars, customised background

Weaknesses:

- Does not work without Internet connection (has offline access but with read-only access)
- Has to sign up for a Google account
- If there are too many tasks with long descriptions, it can be messy and difficult to read
- Other people may access your calendar if you are not familiar with privacy settings.