Chronologer User Guide

1) Introduction

Chronologer is a task manager designed for **students to handle convoluted and unclear management of module information** frequently faced throughout the semester. It is optimized for those who are familiar with **Command Line Interfaces(CLI)** and has a **Graphical User Interface (GUI)** that displays clear and intuitive information. With Chronologer, students no longer have to fret about lagging behind in their work and continue to be on top of their taskings.

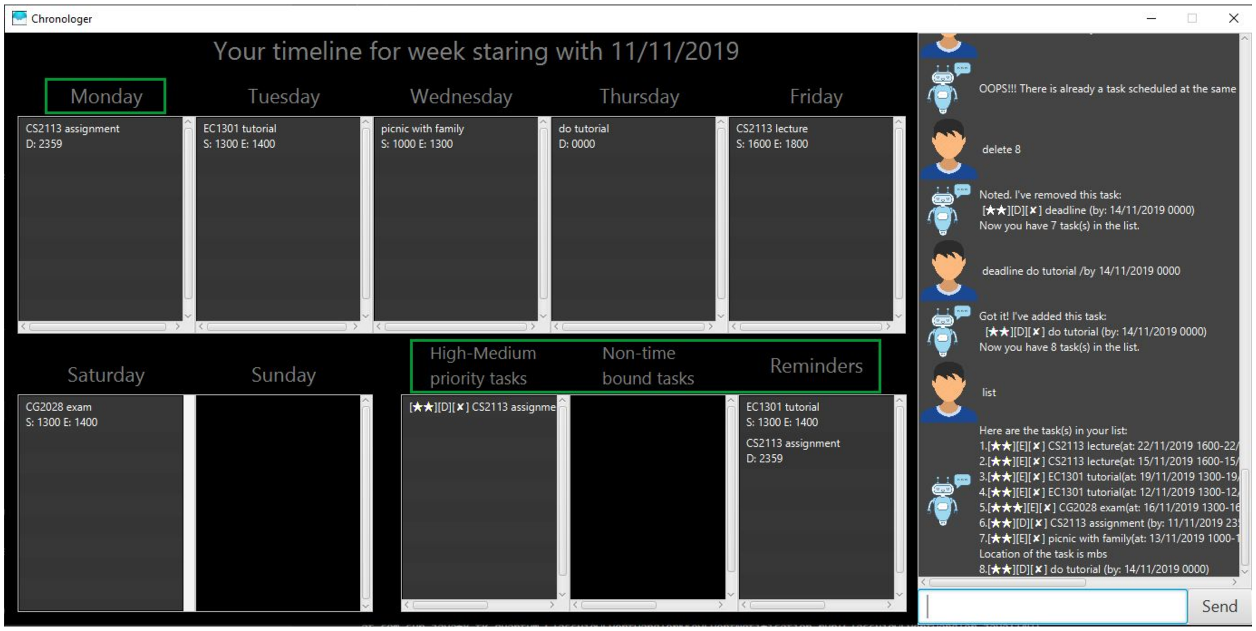
2) Quick start

01. Ensure you have Java 11 installed on your computer.

02. Download the latest Chronologer.jar file here.

03. Copy the file to a home folder you want to use as the main directory.

04. Double-click the file to start the app. (*The GUI should load in a few seconds.)*



Your timeline for week staring with 11/11/2019

Monday	Tuesday	Wednesday	Thursday	Friday
CS2113 assignment D: 2359	EC1301 tutorial S: 1300 E: 1400	picnic with family S: 1000 E: 1300	do tutorial D: 0000	CS2113 lecture S: 1600 E: 1800

Saturday	Sunday	High-Medium priority tasks	Non-time bound tasks	Reminders
CG2028 exam S: 1300 E: 1400		[★★][D][✘] CS2113 assignme		EC1301 tutorial S: 1300 E: 1400 CS2113 assignment D: 2359

```
OOPS!!! There is already a task scheduled at the same

delete 8

Noted. I've removed this task:
 [★★][D][✘] deadline (by: 14/11/2019 0000)
Now you have 7 task(s) in the list.

deadline do tutorial /by 14/11/2019 0000

Got it! I've added this task:
  [★★][D][✘] do tutorial (by: 14/11/2019 0000)
Now you have 8 task(s) in the list.

list

Here are the task(s) in your list:
1.[★★][E][✘] CS2113 lecture(at: 22/11/2019 1600-22/
2.[★★][E][✘] CS2113 lecture(at: 15/11/2019 1600-15/
3.[★★][E][✘] EC1301 tutorial(at: 19/11/2019 1300-19/
4.[★★][E][✘] EC1301 tutorial(at: 12/11/2019 1300-12/
5.[★★★][E][✘] CG2028 exam(at: 16/11/2019 1300-16
6.[★★][D][✘] CS2113 assignment (by: 11/11/2019 23
7.[★★][E][✘] picnic with family(at: 13/11/2019 1000-1
Location of the task is mbs
8.[★★][D][✘] do tutorial (by: 14/11/2019 0000)
```

05. Type the command in the command box and press the ENTER key to execute that command.

06. Refer to the features section for details of each command.

3) Features

Command format:

- *Words enclosed by <> brackets are the parameters to be provided by the user.*
- *Inputs enclosed by these [ ] are optional fields.*
- *Each command is not case sensitive.*
- *The words highlighted with turquoise are the keywords for each command.*

Section 3.1) Adding and deleting task commands

3.1.1) Adding a deadline task: deadline

- Adds a deadline task to the task manager on specific date and time.
- Format: <deadline> <description></by> <date> <time>
- Time must be in 24 hour format (dd/mm/yyyy)
- /by flag is required to separate the date-time components from the deadline description
- Chronologer will detect clashes with another deadline at the same time slot.

  Examples:

  A. deadline CS2113 homework /by 29/09/2019 1900
  B. deadline pay bills /by 05/06/2019 0800

3.1.2) Adding an event task: event

- Adds an event task to the task manager on specific date and time.
- Format: <event> <description></at> <start_datetime - end_datetime>
- Time must be in 24 hour format (dd/mm/yyyy)
- /at flag is required to separate the date-time components from the event description
- Chronologer will detect clashes with another event around the same time range.

  Examples:

  A. event carnival /at 29/09/2019 1900 - 30/09/2019 1600
  B. event graduation /at 08/09/2019 1500 - 09/09/2019 1700

3.1.3) Adding a dateless todo task: todo

- Adds a todo task to the task manager.
- Format: <todo> <description>
- No task will clash with a todo due to its less strict nature in real life.

  Examples:

  A. Todo homework

3.1.4) Adding a todo task with duration: todo

- Adds a todo task with duration to the task manager.
- Format: <todo> <description></for> <duration>
- Time duration is in hours

  Examples:
  1. Todo homework /for 4

3.1.5) Adding a todo task with period: todo

- Adds a todo task with period to the task manager.
- Format: <todo> <description></between> <start_datetime - end_datetime>
- Time must be in 24 hour format (dd/mm/yyyy)
- /between flag is required to separate the date-time components from the Todo description

  Examples:
  1. Todo homework /between 29/09/2019 1900 - 30/09/2019 1600

3.1.6) Adding an assignment: assignment

- Adds an assignment task to the task manager list.
- Assignments are a type of deadline and Chronologer will inform the user if any clashes will occur.
- Format: assignment </m> <module code of assignment> </by> <end_datetime>
  - Time must be in 24 hour format (dd/mm/yyyy HHmm)
  - A *valid* module code must be given in one word

  Example:
  1. Assignment /m cs2040C /by 28/10/2019 2000

3.1.7) Adding an examination: exam

- Adds an examination task to the task manager list.
- Exams are a type of deadline and Chronologer will inform the user if any clashes will occur.
- Format: exam </m> <module code of examination> </at> <start_datetime - end_datetime>
  - Time must be in 24 hour format (dd/mm/yyyy HHmm)
  - A *valid* module code must be given in one word

  Example:
  1. exam /m MA1508E /at 28/10/2019 1000 - 28/10/2019 1200

3.1.8) Adding an lecture: lecture

- Adds weekly recurring lecture tasks to the task manager list.
- Lectures are a subclass of event and Chronologer will inform the user if any clashes occur.
- Format: lecture </m> <module code of examination> </at> <day_of_week start_time-end_time> </till> <end_date>

    - Day of week must be the full name of day (e.g. mondays)
    - A *valid* module code must be given in one word
    - Time must be in 24hours format (HHmm-HHmm)

  Example:

    A. lecture /m Ec1301 /at mondays 1200-1400 /till 12/12/2019

3.1.9) Adding a tutorial: tutorial

- Adds weekly recurring tutorial tasks to the task manager list.
- Tutorial are a subclass of event and Chronologer will inform the user if any clashes occur.
- Format: tutorial </m> <module code of examination> </at> <day_of_week start_time-end_time> </till> <end_date>
    - Day of week must be the full name of day (e.g. mondays)
    - A *valid* module code must be given in one word
    - Time must be in 24hours format (HHmm-HHmm)

  Example:

    A. tutorial /m cs2040c /at tuesdays 0800-1000 /till 25/12/2019

3.2.0) Deleting a task: delete

- Deletes a task from the task manager list.
- Format: delete <list number where task is located>
- Deletes the task at the specified list no.
- If there are no tasks on that list no, Chronologer will inform the user.
- List no must be within the range of the task manager current list.
- List no must be an Integer and positive.

  Example:

    A. delete 2
    B. Delete 3

Section 3.2) Completing a task

3.2.1) Completing a task: done

- Mark a task on the task manager as completed
- Format: <done> <list no>
- List no must be within the range of the task manager current list.

  Examples:

    A. Done 1
    B. Done 2

Section 3.3) Editing/Setting task properties commands

3.3.1) Edit task description: edit

- Allow users to change their task description to reflect any real life changes.
- Format: edit <list no> <new description>

- Chronologer will inform the user if there are no tasks on that list no or if the new description is empty.
- List no must be positive,an integer and within range of Chronologer current list.
  Example:
  A. edit 1 Study maths
  B. edit 20 watch Joker

3.3.2) Set task priority: priority

- Sets a particular task priority level.
- Each task has a priority level of Medium by default.
- Format: priority <list no> <priority level>
- Priority levels: High,Medium,Low.
- Tasks with a priority level that isn't low will be displayed in the GUI.
- Example:
  A. Priority 2 high
  B. Priority 3 low
  C. Priority 1 medium

3.3.3) Set task reminder: reminder

- Sets a reminder for a task which reminds the user in a specified days before the task's date.
  Each task has a reminder of 3 days by default.

- Format: reminder <list no> /in <days>
  - <list no> must be a positive integer no greater than the total number of tasks in the list.
  - <days> must be a positive integer

- If the current time has gone past the specified reminder days before a task's date, the task will be displayed in the GUI's reminder window (bottom left corner).
- Example:
  A. remind 2 /in 3

3.3.4) Add/Modify location: location

- Sets location of particular task.
- Each tasks won't have a location by default.
- Format: location <list no > <location>
- List no must be positive,an integer and within range of Chronologer current list.
  Example:
  A. Location 3 NUS LT3
  B. Location 2 Hougang MRT

3.3.5) Add/Modify comment: comment

- Adds/modifies comment of particular task.
- Each tasks won't have a comment by default on creation.
- Format: comment <list no > <comment>
- List no must be a positive integer within range of Chronologer current list.

  Example:

  A. Comment 1 Do by midnight
  B. Comment 2 Random stuff

Section 3.4) Postpone commands

3.4.1) Postpone a deadline: postpone

- Postpone a deadline to another timeslot.
- This command syntax applies to exams and assignment tasks.
- Format: postpone <list no> <date> <time>
- List no must be a positive integer not greater than the total number of tasks in the task manager.
- Chronologer will inform the user if the date and time specified is clashing with another task.
- Time postponed must be later than the current one.

  Example:

  A. postpone 2 13/12/2019 1900

3.4.2) Postpone an event/tutorial/lecture: postpone

- Postpone an event/lecture/tutorial to another time range.
- Format: postpone <list no> <date> <start_datetime - end_datetime>
- Postpone the event at the specified list no to the corresponding date time.
- Time postponed must be later than the current one.
- List no must be a positive integer not greater than the total number of tasks in the task manager.
- Chronologer will inform the user if the date and time specified is clashing with another event.

  Example:

  A. postpone 4 25/11/2019 0800 - 26/11/2019 1900

3.4.3) Postpone a todo task with period: postpone

- Postpone a todo task with period to another time range.
- Format: postpone <list no> <date> <start_datetime - end_datetime>
- Postpone the todo at the specified list no to the corresponding date time.
- List no must be a positive integer not greater than the total number of tasks in the task manager.
- Time postponed must be later than the current one.
- Todo tasks won't clash with others

  Example:

  A. Postpone 2 24/06/2019 1900 - 25/06/2019 1900

Section 3.5) View task schedule/timeline commands

3.5.1) Viewing Schedule: view

- View the schedule for a specific date.
- Format: view <date>
- Schedule shown will be automatically sorted from earliest to latest task.
- If there are no task on that date, Chronologer will inform the user.

  Example:

  A. view 22/9/2019

3.5.2) Viewing timeline: week

- View the schedule for a given week. (To note all weeks of a semester are numbered from 0-18, eg. recess week is week 7)
- Format: week <week number>/<week current>
- week current is a special command to return to your current week from any week.
- If the week selected is invalid, Chronologer will inform the user.

  Example:

  A. week 1

Section 3.6) Search related commands

3.6.1) Locating tasks by keyword: find

- Find tasks which contain the corresponding keyword.
- Format: find <keyword>
- Keyword is case sensitive.
- Both full words and sub strings are checked for.
- Returns a list of tasks that contains the keyword.

  Example:

  A. Find PC1222

  B. Find Minecraft

3.6.2) Search for next free time slot: search

Find the next free time slot of a certain duration.

Format**:** search <duration in hours>

- ○ Duration given must be rounded to the nearest hour.

Format**:** search <duration> [unit_of_time] *(coming in v2.0)*

- ○ [unit_of_time] supported: mins, hours, days, weeks *(v2.0)*
- ○ Default [unit_of_time] is hours if not provided by user.

Returns the next free time slot of that duration.

Example:

A. search 4
B. search 4 weeks *(v1.4)*

Section 3.7) Scheduling feature commands

3.7.1) Schedule a Todo with duration by a Deadline task: schedule

Find all free periods within the timeline that can accomodate the Todo's duration by the Deadline.
Format: schedule <INDEX_OF_TODO> [/by <INDEX_OF_DEADLINE>]`

- <INDEX_OF_TODO> and <INDEX_OF_DEADLINE> must be a positive integer and not be greater than the size of existing tasks in the list.
- Deadline selected must have a deadline date after the present time
- The duration of the Todo selected must be smaller than the duration between now and the Deadline
- [/by <INDEX_OF_DEADLINE>] can be omitted, the Scheduling feature will instead find all free periods for the Todo up to a hard-limit of 30 days from the present time.

If at least one free period has been found, Chronologer will display, in chronological order, all periods that the user can do the Todo in while still complying with the Deadline.

If no free period could be found, Chronologer will inform the user that no period long enough to complete the Todo by the deadline could be found and suggests the user to consider freeing up their schedule.
Additionally, all low-priority events within the searched timespan are displayed below for the user's consideration to delete or reschedule in order to free up more time.

Example input:
A. “schedule 6 /by 4”

Example outputs:
“You can schedule this task from now till 08/11/2019 1400”

“There is no free slot to insert the task. Consider freeing up your schedule.”

3.7.2) Schedule a todo with duration by a date

Alternatively, the scheduling feature can also be done with a raw date-time input as a deadline.
Format: `schedule <INDEX_OF_TODO> [/by <DATE_TIME>]`

- <INDEX_OF_TODO> must be a positive integer and not be greater than the size of existing tasks in the list.

- <DATE_TIME> must be of the format dd/MM/yyyy HHmm
- <DATE_TIME> must be after the present time.
- The duration of the Todo selected must be smaller than the duration between now and the chosen date
- [/by <DATE_TIME>] can be omitted, with the same hard-limit as mentioned in 3.6.1

Likewise in **3.7.1**, Chronologer will display the results to the user based on whether there are any free periods of time or not.

Example input:

A. `schedule 6 /by 08/08/2019 0800`

3.7.3) Schedule a todo with period (coming in v2.0)

A Todo with period can be chosen for scheduling. Chronologer will search all the free periods of time within the Todo's period.
Format: 'schedule <INDEX_OF_TODO_WITH_PERIOD> [-L]'

- <INDEX_OF_TODO_WITH_PERIOD> must be a positive integer and not be greater than the size of existing tasks in the list.
- [-L] can optionally be included in for Chronologer to only display the largest period of time within the Todo's timespan.

Likewise in **3.7.1**, Chronologer will display the results to the user based on whether there are any free periods of time or not.

Example input:

A. ‘schedule 6’

Section 3.8) Storage commands

3.8.1) Autosave:

- Program will automatically save your tasklist under src/ChronologerDatabase/ArrayList by default if any command changes the schedule.

Section 3.9) Export commands

3.9.1) Exporting an ICS file: export

- Create an ICS file which can be used to import your tasklist to other applications that support calendar files.
- Only deadlines,events and todo tasks with period will be exported as most calendar applications won’t support dateless tasks.
- Note that the start date of deadlines are interpreted as the time when the export command is inputted.
- Format: export <file name>
- A new ICS file will be created under src/ChronologerDatabase/file name.

- Chronologer will inform user if no file name provided.
  Example:
  - Export calendar
  - Export schedule

3.9.2) Exporting only certain type of tasks: export

- Create an ICS file which only consist of the tasks included.
- Format: export <file name> </task type flag>
- Task type flags:
  - ❖ -d = Deadline,assignments and exams.
  - ❖ -e = Event,lectures and tutorials
  - ❖ -t = Todo with period
- Can include multiple flags in the instruction. If there are no task flags, all supported task type will be exported by default.
  Example:
  - Export calendar -d
  - Export MySchedule -e
  - Export testCalendar -d -e
  - Export test -e -t

Section 3.10) Undo/Redo commands

3.10.1) To undo the last command: undo

- Any changes made to the tasks, such as adding and deleting will be undone and the task manager will revert to a previous state.
- Format: undo
- If there are no more undo commands possible, the user will be notified.

3.10.2) To perform a redo command: redo

- Any changes made to the tasks by an undo command, will be reversed and reverted back to the state before the undo command was executed.
- Format: redo
- If there are no more redo commands possible, the user will be notified.

Section 3.11) Store/Restore commands

3.11.1) To store a state: store

- This allows the user to store 3 versions of the task manager at any one time to load onto the system and use.
- Format: store <version number>
- Overwrites the previous version 1 with the current version of the task manager.
  Example:
  A. store 1

3.11.2) To restore from a state: restore

- This allows the user to restore from 3 versions of the task manager at any time to load onto the system and use.
- Format: restore <version number>
- Overwrites the current task manager with the stored version 1 of the task manager.
  Example:
  A. restore 1

Section 3.12) Command Line History

3.12.1) To move up/down your command history: up/down

- By simply pressing the up arrow or the down arrow the user will be able to move through their command history.

Section 3.13) Manual command

3.13.1) To see the latest version of this guide: manual

- Allows the user to view the latest version (online) of the user guide from our repository.
- Format: manual

Section 3.14) Help command

3.14.1) To see a quick help menu: help

- Allows the user to view a short quick help menu within the application.
- Format: help

Section 3.15) Theme command

3.15.1) To change themes: theme

- Allows the user to change the current theme of the application between two choices: dark mode/light mode.
- Format: theme <mode of choice>
  Example:
  A. theme light
  B. theme l (to note: both imply light mode)

Section 3.16) Terminating program

3.16.1) Terminating program: bye

- Terminates the program.
- Format: bye

4) Testing the program

4.1) Load the program with a prefilled schedule: tester

- Loads Chronologer with a prefilled schedule for testing purposes.
- Format: tester

4.2) Clear the schedule (WARNING: irreversible): sudo-clear

- Clear the whole schedule to get a blank slate for testing purposes.
- Format**:** sudo-clear