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 even has a **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.

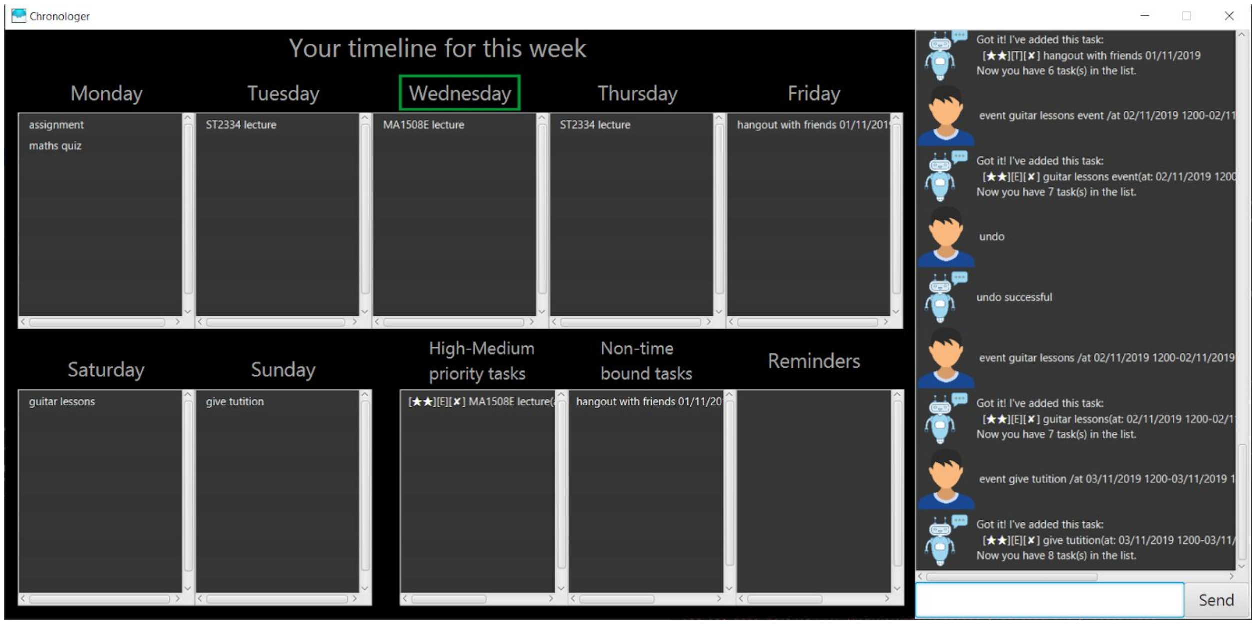
2) Quick start

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.)*



Monday: assignment, maths quiz
Tuesday: ST2334 lecture
Wednesday: MA1508E lecture
Thursday: ST2334 lecture
Friday: hangout with friends 01/11/201
Saturday: guitar lessons
Sunday: give tuition
High-Medium priority tasks: [★★][E][✘] MA1508E lecture(
Non-time bound tasks: hangout with friends 01/11/20

```
Got it! I've added this task:
  [★★][T][✘] hangout with friends 01/11/2019
Now you have 6 task(s) in the list.
event guitar lessons event /at 02/11/2019 1200-02/11
Got it! I've added this task:
  [★★][E][✘] guitar lessons event(at: 02/11/2019 1200
Now you have 7 task(s) in the list.
undo
undo successful
event guitar lessons /at 02/11/2019 1200-02/11/2019
Got it! I've added this task:
  [★★][E][✘] guitar lessons(at: 02/11/2019 1200-02/1
Now you have 7 task(s) in the list.
event give tutition /at 03/11/2019 1200-03/11/2019 1
Got it! I've added this task:
  [★★][E][✘] give tutition(at: 03/11/2019 1200-03/11/
Now you have 8 task(s) in the list.
```

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

3.11) 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 29/9/2019 1900
  B. deadline pay bills /by 05/06/2019 0800

3.12) 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 29/9/2019 1900 - 30/9/2019 1600
  B. event graduation /at 08/09/2019 1500 - 09/09/2019 1700

3.13) 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

3.14) 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:

A. Todo homework /for 4

3.15) 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:

A. Todo homework /between 29/9/2019 1900 - 30/9/2019 1600

3.16) Adding an assignment: assignment

- Adds an assignment task to the task manager list.
- Format: assignment </m> <module code of assignment> </by> <end_datetime>
  - Time must be in 24 hour format (dd/mm/yyyy HHmm)
  - Module code must be given in one word

Example:

A. Assignment /m 2040c /by 28/10/2019 2000

3.17) Adding an examination: exam(v1.4)

- Adds an examination task to the task manager list.
- Format: exam </m> <module code of examination> </at> <start_datetime - end_datetime>
  - Time must be in 24 hour format (dd/mm/yyyy HHmm)
  - Module code must be given in one word

Example:

A. exam /m 2040c /by 28/10/2019 1000 - 28/10/2019 1200

3.18) 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

3.21) 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

3.31) 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

3.32) 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

3.33) 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.
- Example:

A. remind 2 /in 3

3.34) 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

3.35) 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

3.41) Postpone a deadline: postpone

- Postpone a deadline to another timeslot.
- Format: postpone <list no> <date> <time>
- Postpone the deadline 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.
- Chronologer will inform the user if the date and time specified is clashing with another task.

  Example:

  A. postpone 2 16/07/2019 1900

3.42) Postpone an event: postpone

- Postpone an event 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.
- 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 25/07/2019 0800 - 26/07/2019 1900

3.43) 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.
- 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

3.51) 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

3.52) Viewing timeline

- The GUI will automatically display any task within the current week.

Section 3.6) Search related commands

3.61) 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

3.62) 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] *(v1.4)*
  - <unit_of_time> supported: mins, hours, days, weeks *(v1.4)*
  - 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)*

3.63) Search for next free time slot: schedule

- Find all free time slots within the user's schedule to fit in a 'Todo' task of a certain duration by a specified deadline.
- The command can accept the deadline input as either index or as a raw date.
- Format**:** schedule <index no. of todo> /by <index no. of deadline>
    - All index inputs must be a positive integer no greater than the total number of tasks in the list
    - <index no. of todo> must select a todo with a duration value.
    - <index no. of deadline> must select a deadline.
- Format**:** schedule <index no. of todo> /by <deadline date>
    - All index inputs must be a positive integer no greater than the total number of tasks in the list
    - <index no. of todo> must select a todo with a duration value.
    - <deadline date> must be of format dd/MM/yyyy HHmm and no earlier than the current time.
- Returns all free time periods in chronological order for the user's consideration.
- If there is no free time periods otherwise, Chronologer will informs the user of it.

  Example:

  A. schedule 5 /by 2
  B. Schedule 5 /by 08/01/2001 0800

Section 3.7) Storage commands

3.71) Autosave:

- Program will automatically save your tasklist under src/DukeDataBase/ArrayList by default if any command changes the schedule.
- Program will also initialize the tasklist stored under src/DukeDataBase/ArrayList every time it starts.
- However,the user can also manually save and load the tasklist.

3.72) Manual save: save (V1.4)

- Save the current tasklist.
- Format: save <file name>
- The file will be automatically saved under src/DukeDataBase/<file name>

3.73) Manual load: load (V1.4)

- Load a tasklist from DukeDataBase directory.
- Format: load <file name>

- Chronologer will load the tasklist stored under src/DukeDataBase/<file name>.
- [Warning!]Autosave feature will continue on the loaded file and not the default ArrayList.

Section 3.8) Export command

3.81) Exporting an ICS file: export

- Create an ICS file which can be used to import your tasklist to other applications that support calendar files.
- Format: export
- A new ICS file will be created under src/DukeDatabase/.
  Example:
  - export

3.82) (v1.4)Exporting only certain type of tasks: export

- Create an ICS file which only consist of the tasks included.
- Format: export **<task type>**
- Task type supported: Todo with period,deadline,event
  Example:
  - export deadline
  - export event
  - export todo

Section 3.9) Undo/Redo command

3.91) 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
- Simply use normal short-cut of ctrl+z to perform an undo. (*v1.4)*
- If there are no more undo commands possible, the user will be notified.

3.92) To undo the last 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
- (v1.4) Simply use normal short-cut of ctrl+y to perform a redo. (*v1.4)*
- If there are no more redo commands possible, the user will be notified.

3.93) (v1.4) The History: undo

- The last 5 changes from a particular usage will be stored into persistent storage, to allow the user to undo from launch of Chronologer.
- Format: undo
- If there are no more redo commands possible, the user will be notified.

3.94) (v1.4) The History (version storage):save state

- This allows the user to store 3 versions of the task manager at any one time to load onto the system and use.
- Format: save state
- If there are no states saved, or 3 versions already saved, the user will be notified.

Section 3.10) Terminating program

3.101) Terminating program: bye

- Terminates the program.
- Format**:** bye