

TA-Tracker - User Guide

1. Introduction	2
2. Quick Start	3
3. About the User Guide	4
3.1. Common Symbols	4
3.2. Command Format	4
3.3. Common Parameters	5
3.3.1. Duplicate Parameters	8
3.3.2. Rearranging Parameters	9
4. Layout	9
4.1. Student View	10
4.2. Session View	11
4.3. Claims View	12
5. Usage	13
5.1. Keyboard Shortcuts	13
5.2. Syntax Highlighting	13
5.3. Hints	14
6. Commands	15
6.1. Navigation	15
6.1.1. Viewing help : <code>help</code>	15
6.1.2. Switching tabs : <code>goto</code>	16
6.1.3. Exiting the program : <code>exit</code>	17
6.1.4. Listing all sessions : <code>list</code>	17
6.2. Student View	18
6.2.1. Adding a Module : <code>module add</code>	18
6.2.2. Deleting a Module : <code>module delete</code>	19
6.2.3. Editing a Module : <code>module edit</code>	20
6.2.4. Adding a Group : <code>group add</code>	21
6.2.5. Deleting a Group : <code>group delete</code>	22
6.2.6. Editing a Group : <code>group edit</code>	23
6.2.7. Adding a Student : <code>student add</code>	24
6.2.8. Deleting a Student : <code>student delete</code>	25
6.2.9. Editing a Student : <code>student edit</code>	26
6.2.10. Sorting a Group : <code>sort group</code>	29
6.2.11. Sorting a Module : <code>sort module</code>	30
6.2.12. Sorting Everything : <code>sort all</code>	31
6.2.13. Filter : <code>filter</code>	32
6.3. Session View	33
6.3.1. Adding a Session : <code>session add</code>	33

6.3.2. Deleting a Session : <code>session delete</code>	35
6.3.3. Editing a Session : <code>session edit</code>	36
6.3.4. Marking a Session as Done : <code>session done</code>	37
6.3.5. Filter : <code>filter</code>	38
6.4. Claims View	40
6.4.1. Filter : <code>filter</code>	40
6.4.2. Changing the hourly rate : <code>setrate</code>	40
6.5. Statistics Window	41
6.5.1. Generate Statistic Report : <code>report</code>	41
7. Glossary	43
8. FAQ	44
9. Command Summary	44
9.1. Navigation	44
9.2. Student View	44
9.2.1. Module commands	44
9.2.2. Group commands	44
9.2.3. Student commands	45
9.2.4. Others	45
9.3. Session View	45
9.3.1. Session commands	45
9.3.2. Others	45
9.4. Claims View	45
9.5. Statistics	45

By: Team W17-4 Since: Jan 2020 Licence: MIT

1. Introduction

Are you a **Teaching Assistant** in NUS School of Computing (NUS SOC) who is tired of having to fill out the **Teaching Support Student (TSS) Claims Form** accurately at the end of the semester?

Do you wish there was a **desktop application** that can help you **keep track of your teaching duties**, such as your **claimable hours** and **student ratings**, all in **one place**?

Look no further, because **TA-Tracker** is the tool just for you!

No more saving your claimable hours in an **unappealing spreadsheet**.

No more keeping notes about your students in a **cluttered diary**.

TA-Tracker is a **productivity tool** that is made for all you **NUS SOC Teaching Assistants (TAs)** who need to easily **fill out forms**, such as the **TSS Claims Form**, at the end of the semester.

TA-Tracker has features that will help you easily **track and manage** your **claimable hours** and **students in one place**.

What's more, TA-Tracker has:

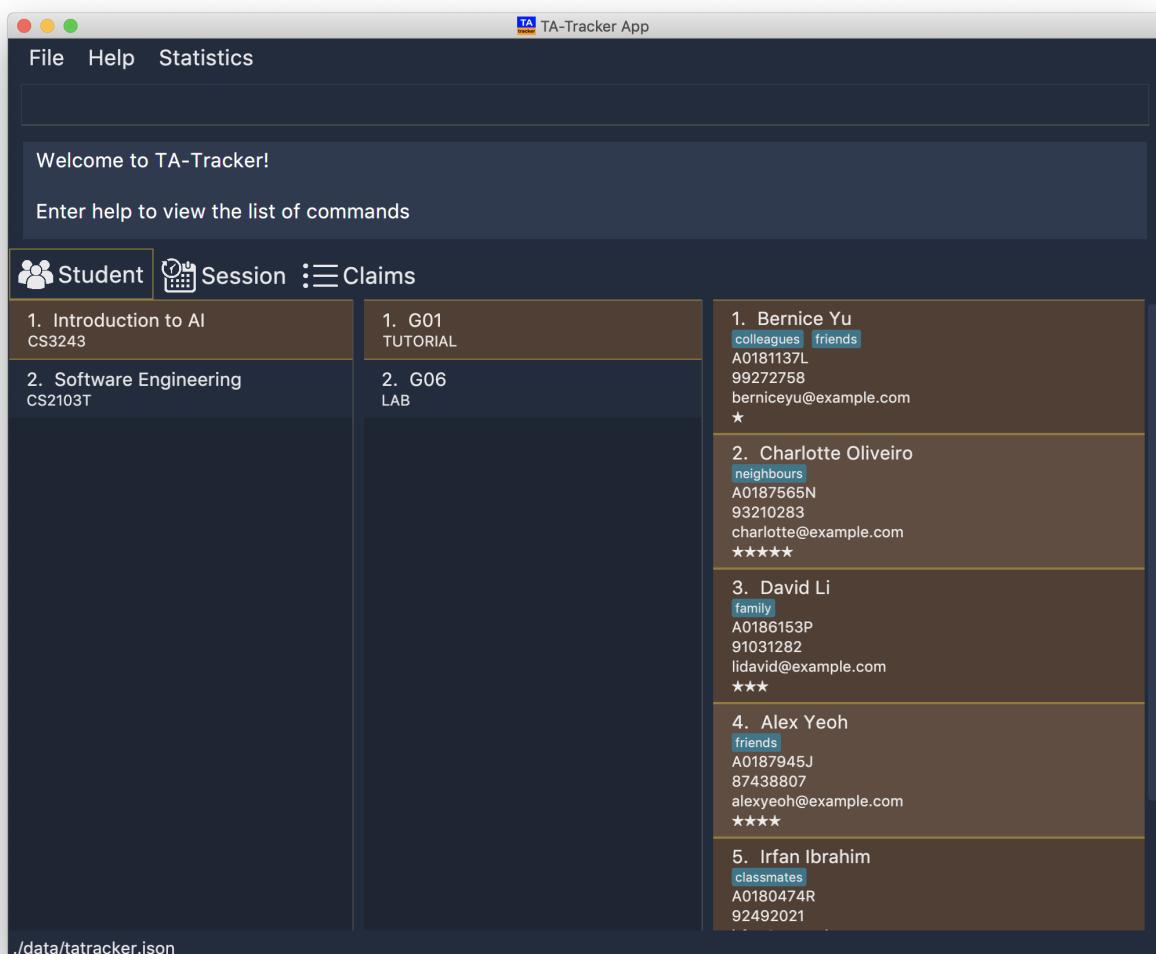
- A **Command Line Interface (CLI)** for all you TAs who can type fast and prefer to use a keyboard, and
- A **Graphical User Interface (GUI)** so that you can easily view all your **teaching duties**.

If you are interested, jump to [Section 2, “Quick Start”](#) to discover how easy it is to fill out forms using TA-Tracker.

2. Quick Start

This section gives you a step-by-step explanation on how you can download and open the application.

1. Ensure that you have Java [11](#) or above installed in your Computer.
2. Download the latest [tatracker.jar](#) [here](#).
3. Copy the file to the folder you want to use as the home folder for your TA tracker.
4. Double-click the file to start the app. The GUI should appear in a few seconds.



This is how the GUI may look like when the TA-Tracker is opened.

- Type a command in the command box, and press **Enter** to execute it.
e.g. typing **help** and pressing **Enter** will open the help window.



This diagram explains what a **Result Display** and **Command Box** are.

- Refer to [Section 5, “Usage”](#) for details of each command.

NOTE

TA-Tracker data is saved periodically so you don't have to worry about saving your data manually.

3. About the User Guide

This User Guide introduces you to TA-Tracker's features and shows you how you can use the TA-Tracker to make your life as a Teaching Assistant easier.

3.1. Common Symbols

This section shows you the symbols commonly used in this guide.

NOTE

This symbol indicates that there is something that you should take note of.

TIP

This symbol indicates that a tip is being mentioned.

CAUTION

This symbol indicates that there is something you should be careful to avoid.

3.2. Command Format

This section shows you how all the commands in this guide have been formatted.

Format	Meaning	Example
lower_case/ Any lower case letters, followed by a forward slash	These are prefixes . They are used to separate the different parameters of a command.	These are prefixes: n/, d/, t/ Note that prefixes cannot have spaces : n / is not a prefix , and will not be recognized.

Format	Meaning	Example
UPPER_CASE Words in upper case	<p>These are parameters.</p> <p>You will need to supply parameters in order to complete certain commands.</p>	<p>You can create a student with the name John Doe using the student add command.</p> <p>Suppose the student add command looks like this:</p> <p>student add n/NAME</p> <p>Simply replace NAME with John Doe to create the student John Doe:</p> <p>student add n/John Doe</p>
[UPPER_CASE] Words in upper case, surrounded by square brackets	<p>These are optional parameters.</p> <p>Certain commands can be used without these parameters.</p>	<p>Suppose a command contains two parameters next to each other:</p> <p>n/NAME [t/TAG]</p> <p>The first parameter NAME is compulsory. The second parameter TAG is optional.</p> <p>Since a TAG is optional, you will be able to use the command with these inputs:</p> <ul style="list-style-type: none"> • n/John Doe t/Fast learner, or • n/John Doe
UPPER_CASE... [UPPER_CASE]... An ellipsis ... following any words in upper case	<p>These are parameters that can be used multiple times or none at all.</p>	<p>The following parameter can be used multiple times:</p> <p>t/TAG...</p> <p>This means that it can be:</p> <ul style="list-style-type: none"> • Left empty (i.e. 0 times): t/ • Used one time (i.e. 1 time): t/friend • Used multiple times (i.e. 2 or more times): t/friend t/family

3.3. Common Parameters

This section lists and explains what the parameters commonly used in the commands mean.

Parameter	Explanation	Examples
TAB_NAME	This refers to the different tab names.	student - to indicate the Student View session - to indicate the Session View claims - to indicate the Claims View
INDEX	Indicates the position of an item in a list	1 - refers to the first item in a list
MOD_CODE	Refers to the unique code given to the module. You can personalise this and give it your own code. However, we recommend you use the module's official code.	CS2103T
GROUP_CODE	Refers to the unique code given to the group. You can personalise this and create your own code. However, we recommend that you use the group's official code assigned by SoC	G06
NEW_GROUP_CODE	Refers to the new group code of a group when you edit a group.	G05
GROUP_TYPE	Refers to the group type.	lab, tutorial, recitation or other
NEW_GROUP_TYPE	Refers to the new group type you want a group to be.	lab, tutorial, recitation or other
MATRIC_NUMBER	Refers to the matriculation number of a student. It must start with an A, have 7 numbers in between and end with an alphabet.	A0123456X
NAME	Depending on the command, this could either refer to the student's name or the module's name.	John Doe or Software Engineering

Parameter	Explanation	Examples
SORT_TYPE	Refers to how you want to sort the Student View .	alpha , alphabetical or alphabetically - to sort alphabetically. rating asc - to sort by rating in ascending order. rating desc - to sort by rating in descending order. matric - to sort by matriculation number.
SESSION_TYPE	Refers to the type of session.	consult - consultation tutorial - tutorial lab - lab grading - grading assignments, projects or assessments prep - class preparation todo - other tasks and notes
NOTES	Refers to any extra description	This time was spent correcting 40 assignments
START	Indicates the starting time. Must be in the HH:mm format.	10:42
END	Indicates the ending time. Must be in the HH:mm format.	16:42
DATE	Indicates the date of a session. Must be in the yyyy-MM-dd format.	2020-06-20
WEEK	Indicates the recurring period of a session.	1 - represents a session that repeats every week.
RATING	Indicates the rating of a student. A rating is as a number between 1 (Poor) to 5 (Excellent).	3 - represents an average student rating.

Parameter	Explanation	Examples
EMAIL	<p>Indicates the email of a student.</p> <p>Emails should be in local-part@domain format.</p> <p>The local-part can only contain:</p> <ul style="list-style-type: none"> • Alphanumeric characters, and • The following special characters: !#\$%&'*+/=?`{ }~^.- <p>The domain can only contain: * Alphanumeric characters * * The following special characters in between: . -</p> <p>The domain name must also:</p> <ul style="list-style-type: none"> • Have at least 2 characters • Start and end with alphanumeric characters 	johnDoe97@example123.com

3.3.1. Duplicate Parameters

You can specify the same parameter **more than once** in a command. Depending on the command, you will see **different outcomes**.

Here is a table explaining the different outcomes:

Parameter	Explanation	Example
Can be used multiple times	All occurrences will be used when executing the command	<p>In the student add command, you can create a student with multiple tags.</p> <p>Therefore, you can input multiple tags by chaining them: t/needs help t/aka henry</p> <p>This will give a student the tags needs help and aka henry.</p>

Parameter	Explanation	Example
Can only be used once	The very last occurrence will be used when executing the command	<pre>module add m/CS2103T n/SE n/Software Engineering</pre> <p>If you execute this command, you will create a module with the name Software Engineering.</p> <p>The name SE will be ignored.</p>

3.3.2. Rearranging Parameters

Parameters can be in **any order** if they have **prefixes**.

For example, if a command needs a **n/NAME** and **p/PHONE_NUMBER**, you can specify them in any order:

- **n/NAME p/PHONE_NUMBER**, or
- **p/PHONE_NUMBER n/NAME**

However, if the command has a parameter **without a prefix**, that parameter **must** be the very **first parameter**.

For example, if a command needs an **INDEX**, it must be the **first parameter**:

- **INDEX n/NAME** is valid, but
- **n/NAME INDEX** is invalid

4. Layout

This section gives you a brief overview of the layout of the **TA-Tracker**.

TA-Tracker is divided into three **tabs** representing the different **Views**:

- The **Student View** under the **student tab**,
- The **Session View** under the **session tab**, and
- The **Claims View** under the **claims tab**

When you switch to a **tab**, that **tab** will be highlighted in orange.

Furthermore, when you enter a new command, you will be automatically switched to the relevant **tab** so that you can instantly see the result of the command.

- You can **select** a **tab** to show a different **View**. This **tab** will be highlighted in **blue**.
- If you are **switched** to a **tab** when you **enter a command**, that **tab** will be highlighted in **orange**.

This should help you easily remember where you last made a change in **TA-Tracker**.

- You will sometimes see the **tabs** highlighted in both **orange** and **blue**.

The **orange tab** will remain highlighted even if you select another **tab**.

- You may notice that the **orange** and **blue** highlights for the **tabs** are **not the same size**.

NOTE

This is to **prevent** the highlights from **overlapping** each other, allowing you to see them better.



This shows what happens when you last made a change in the **student tab** and then clicked on the **claims tab**.

4.1. Student View

Under the **student tab**, the **Student View** is used to show you the students that you're teaching. The students have been grouped according to the **module** and **group** they belong to. The **Student View** has been divided into three columns.

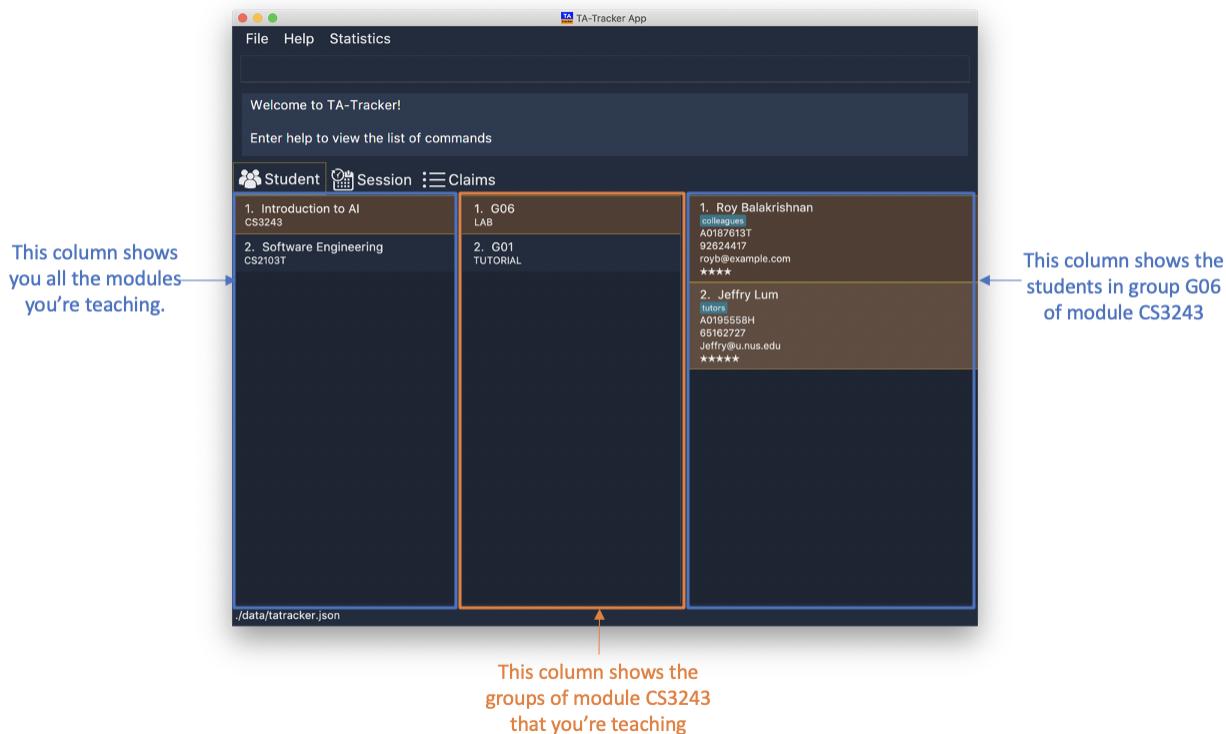
1. The first column shows you a **list of all the modules** that you are a teaching assistant for.
2. The second column shows you a **list of all groups** that you're a teaching assistant for **in a module of your choice**. If you haven't chosen anything, you will be shown the groups of the module in the **first index** in the list of modules by default.
3. The third column shows you a **list of all students** in the **group of your choice**. If you haven't chosen anything, you will be shown the students of the group in the **first index** in the list of groups by default.

The purpose of the **Student View** is to help you keep track of your students. It will show you information such as:

- **NAME:** student name

- **MATRIC_NUMBER**: matriculation number
- **RATING**: ratings you have given the student, represented by stars (on a scale from 1 - 5)
- **EMAIL** and **PHONE**: contact details
- **TAG**: any additional information you have about the student

NOTE The **RATING** value must be a positive integer.



This is an example of what the Student View might look like.

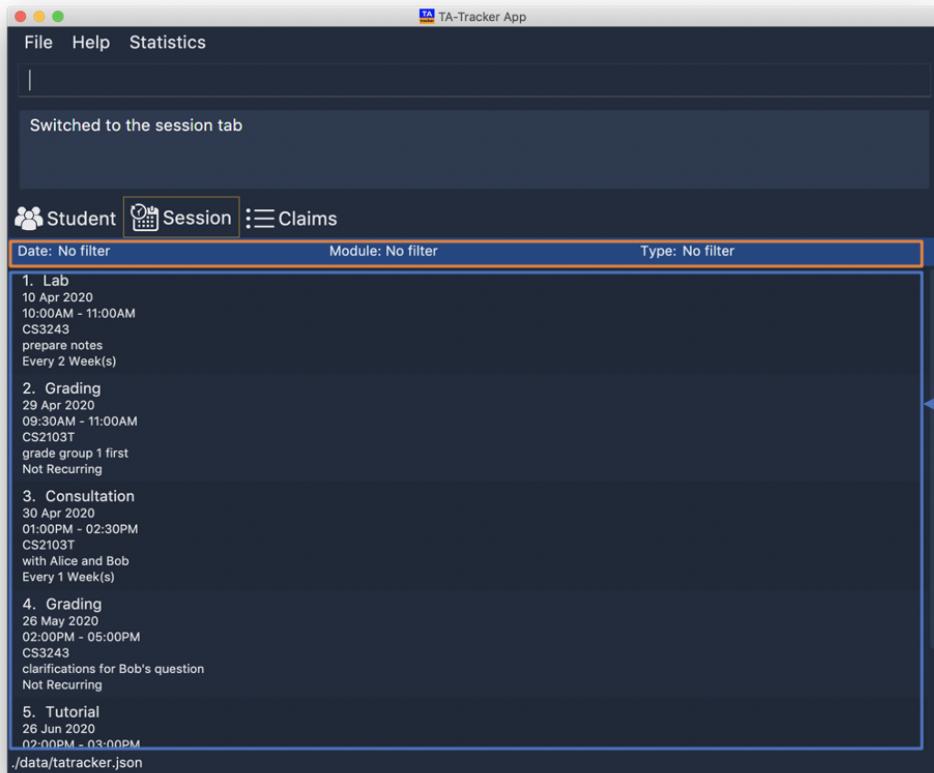
4.2. Session View

Under the **session tab**, the **Session View** contains a list of the upcoming sessions that you haven't done yet.

The sessions are automatically sorted by date.

The purpose of the **Session View** is to help you keep track of your upcoming teaching duties. It will show you information such as:

- **SESSION_TYPE**: the type of session
- **DATE**: the date that the session will occur on
- **START** and **END**: the start time and end time of the session
- **MODULE**: the module that the session is under
- **NOTES**: any additional information
- **WEEK**: the recurring period of the session



This is an example of what the Session View might look like.

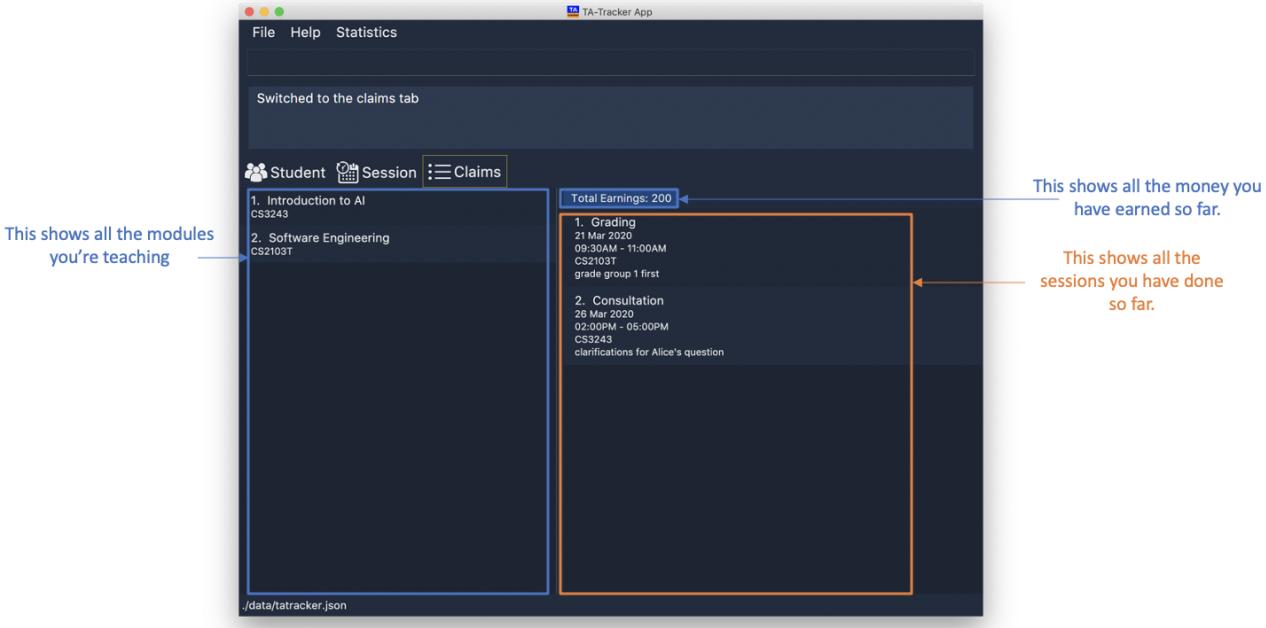
4.3. Claims View

Under the **claims tab**, the **Claims View** contains a list of all the claimable teaching duties you have completed so far.

The purpose of this view is to allow you to keep track of all your claims so you can easily enter it into the TSS claims form at the end of the semester.

The **Claims View** has been divided into two columns.

1. The first column shows you a **list of all the modules** that you are a teaching assistant for.
2. The second column shows you a **list of all the sessions** that you have **marked as done**.



This is an example of what the Claims View might look like.

5. Usage

This section describes how you can interact with **TA-Tracker**.

5.1. Keyboard Shortcuts

You can navigate everything in TA-Tracker with just a **keyboard**!

Here are some keyboard shortcuts that you can use:

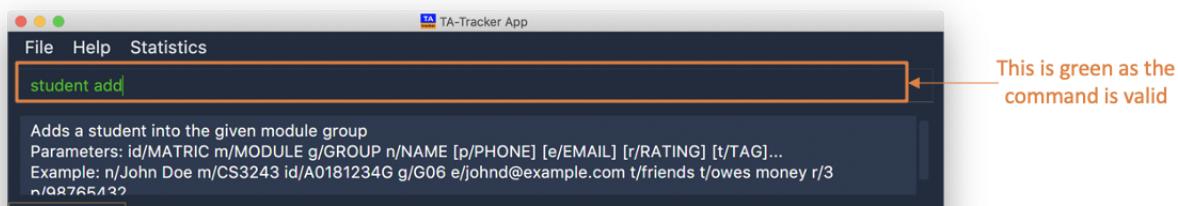
Shortcut Key	Usages
<code>Esc</code>	<ul style="list-style-type: none"> • Toggle between the command box and the Views. • Close a popup window.
<code>↑ up</code> <code>↓ down</code> arrow keys	<ul style="list-style-type: none"> • Scroll through a list (you must toggle out of the command box in order to use this shortcut)
<code>← left</code> <code>→ right</code> arrow keys	<ul style="list-style-type: none"> • Navigate between different lists in the same View. (you must toggle out of the command box in order to use this shortcut)

5.2. Syntax Highlighting

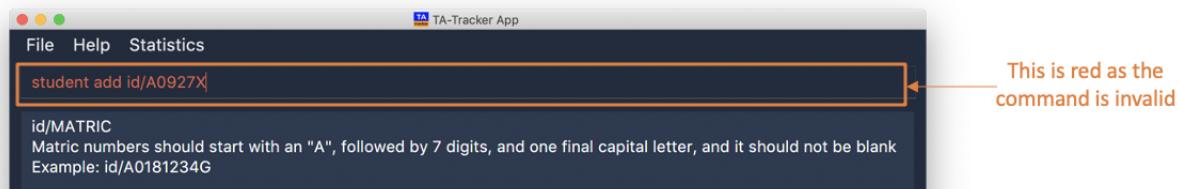
When you type a command into the **command box**, your input will be **highlighted**.

Here is a table explaining the meaning of each colour:

Colour	Meaning
green	Valid input.
red	Invalid input.
white	The default font colour.



This is an example of a valid input.



This is an example on an invalid input.

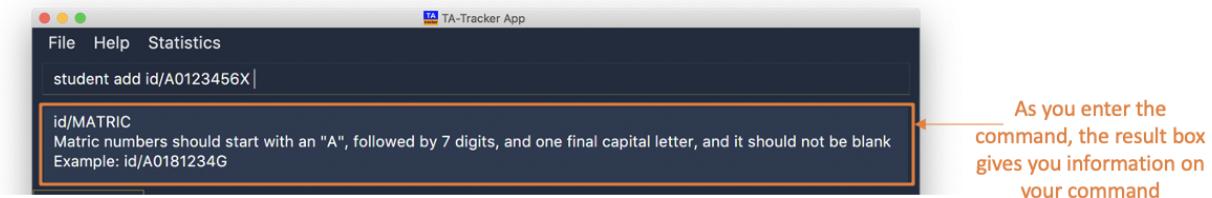
5.3. Hints

When you type out a command, you will also see **hints** about how to use the command.

Here is a table describing the different types of hints that you will encounter:

Hint	Trigger
Showing the command usage	<ul style="list-style-type: none">When you first type out the command word.After typing two whitespaces in a row.

Hint	Trigger
Showing the parameter usage	<ul style="list-style-type: none"> When you first type out a valid prefix. When there is an invalid prefix in your command.



This is an example of what a hint may look like.

6. Commands

This section explains how to use all the **CLI** commands in **TA-Tracker**.

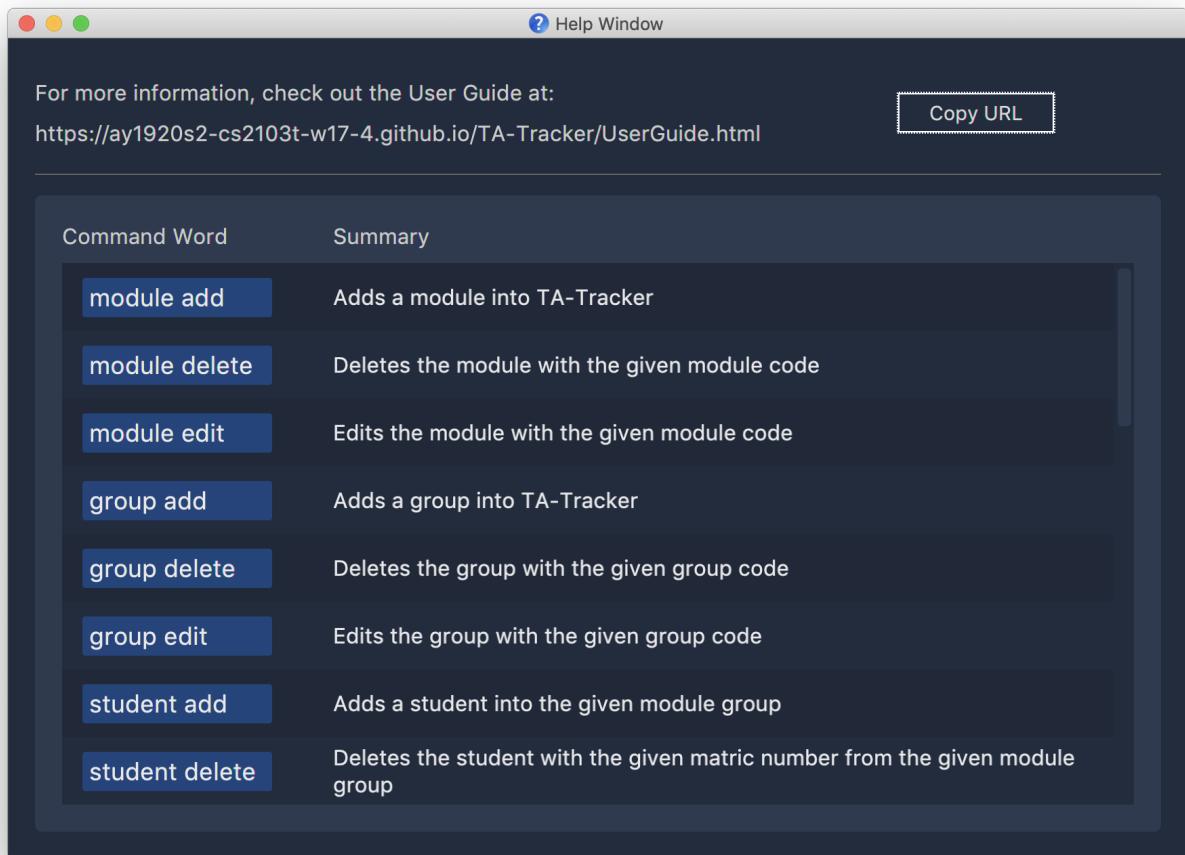
6.1. Navigation

These commands are used to control the different **windows** in **TA-Tracker**, as well as switching between the different **Views**.

6.1.1. Viewing help : **help**

You can open the **help window** with this command. You can close the **help window** by pressing the **ESC** key on your keyboard.

Format: **help**



This is what the **help** window looks like.

6.1.2. Switching tabs : **goto**

You can switch to different **tabs** with this command to show their associated **view**.

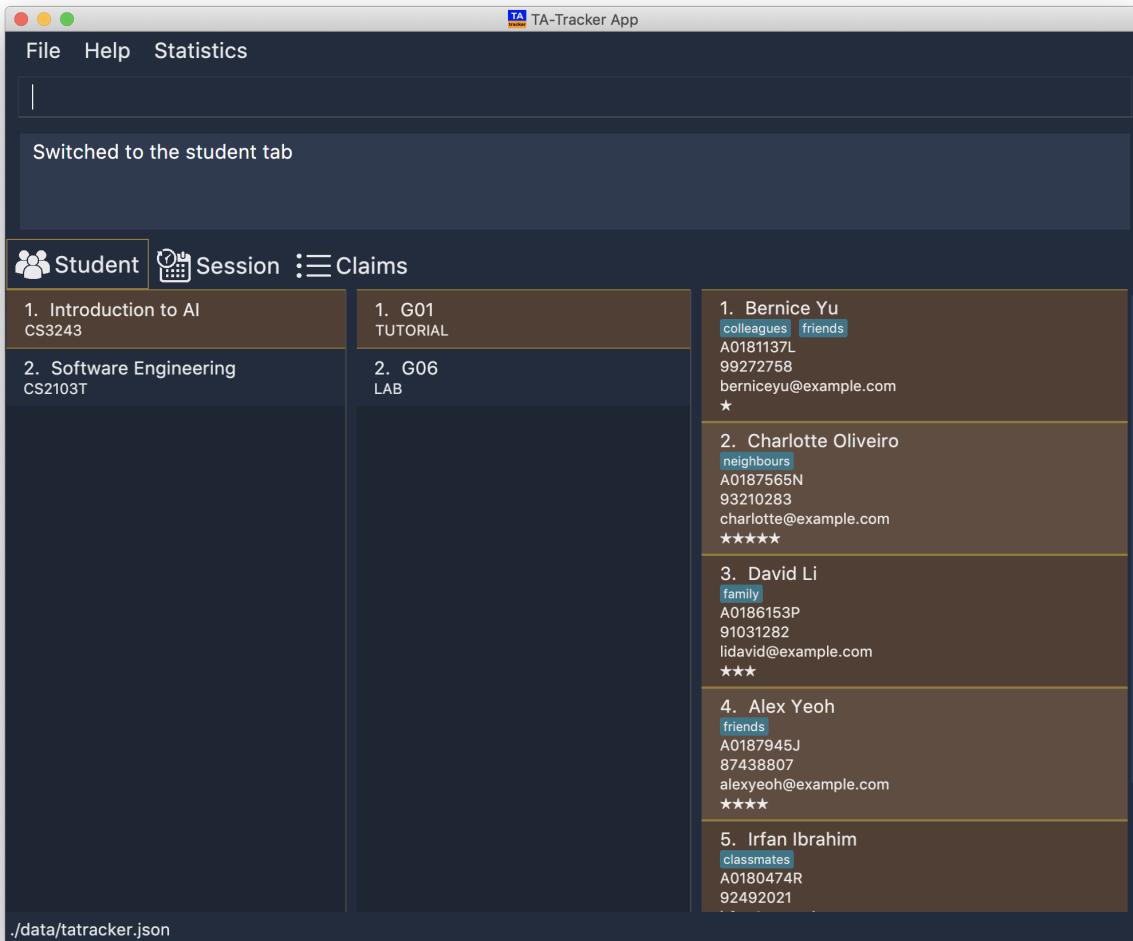
Format: **goto TAB_NAME**

NOTE

- You cannot switch to a **tab** that does not exist in TA-Tracker

Example:

goto student



This command takes you to the `student tab`.

6.1.3. Exiting the program : `exit`

You can exit the program with this command.

Format: `exit`

6.1.4. Listing all sessions : `list`

`list` command shows you all the sessions under Session View and Claims View.

TIP

- You can use this command to display all your sessions after using the `session filter` or `claims filter` command.

6.2. Student View

This section explains all the commands you can use in the **Student View**.

6.2.1. Adding a Module : `module add`

You can use this command to add a new module to the TA-Tracker.

When a new module is created, the **Student View** will show the groups and students of the new module. (Initially, a new module doesn't have any students or groups till you add some). If you were on a different `tab`, you will automatically be switched to the `student tab`.

Format: `module add m/MOD_CODE n/NAME`

- You **can't** have two modules with the **same module code** in the TA-Tracker.
- `MOD_CODE` and `MOD_NAME` can't be an empty string or a sequence of spaces.
- You can give a module your own **custom MOD_CODE** or **NAME** if you find it easier to remember. However, we recommend you use the **official module code and name** for it.
- If the module name or module code are very long, you can **increase the width** of the module name list by dragging the edge with your cursor. However, we suggest not giving the modules long codes or long names. The standard official names are usually of a good length.

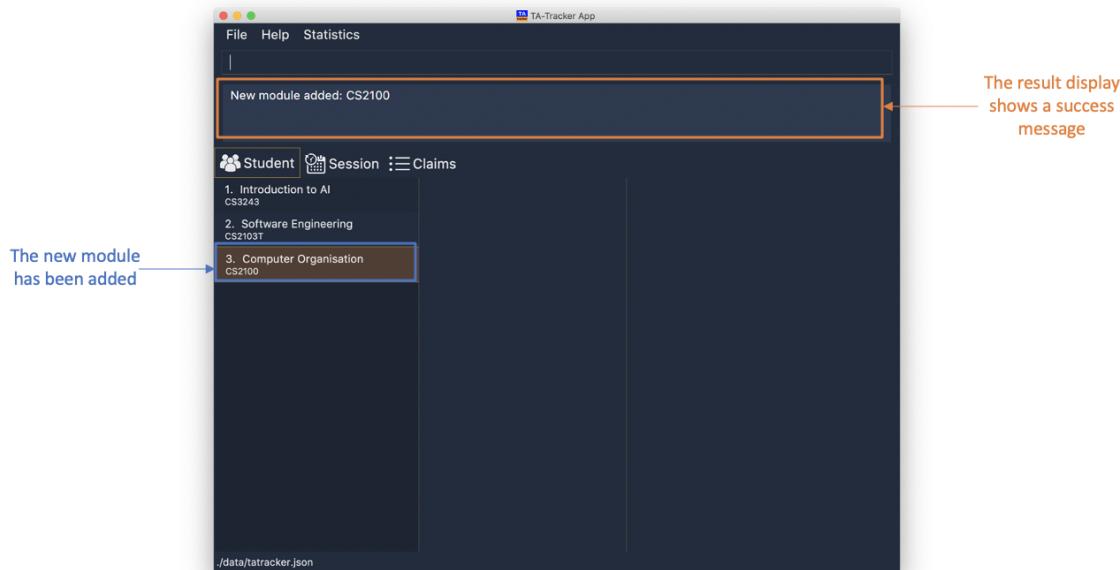
NOTE

TIP

- You can't edit the `MOD_CODE` once the module has been created, so do take care to ensure that the code is correct.

Examples:

- `module add m/CS2100 n/Computer Organisation`



This will add a module with the module code `CS2100` and name `Computer Organisation` to the TA-Tracker.

6.2.2. Deleting a Module : `module delete`

You can use this command to delete a module from the TA-Tracker.

When you delete a module, all groups, students and sessions associated with the module will also be deleted.

When a module is deleted, the **Student View** will go back to its default setting. That is, you will see the details of the first group in the first module of the TA-Tracker. If you were on a different `tab`, you will automatically be switched to the `student tab`.

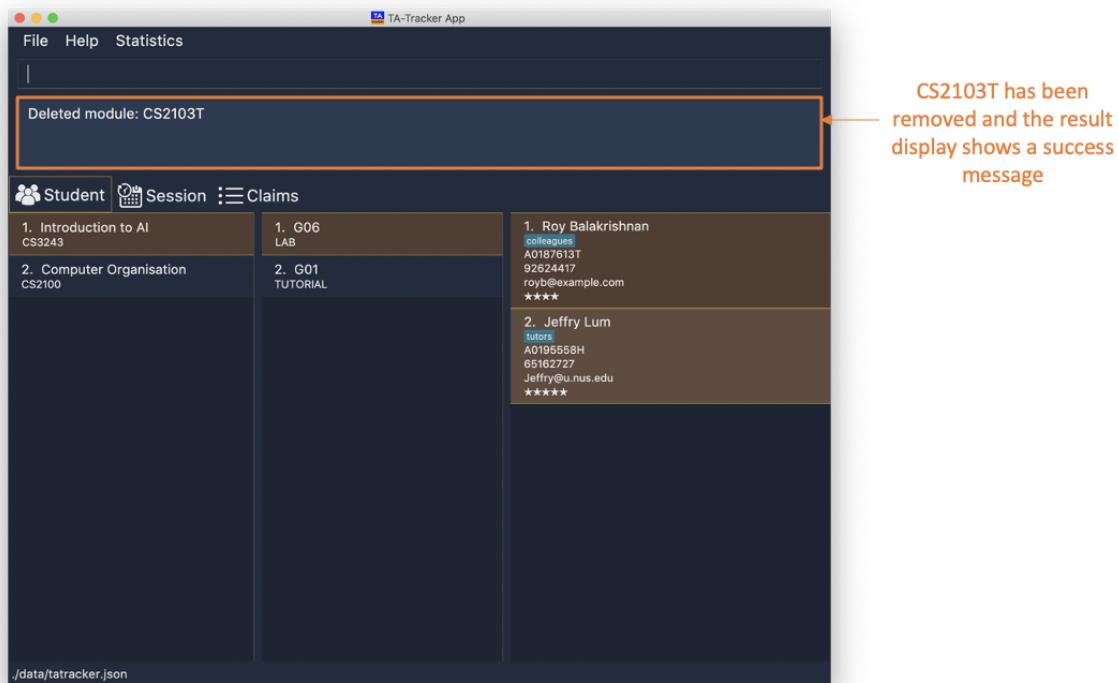
Format: `module delete m/MOD_CODE`

NOTE

- You cannot delete a module that doesn't exist.

Examples:

- `module delete m/CS2103T`



This will delete the module with the module code `CS2103T` from the TA-Tracker.

6.2.3. Editing a Module : `module edit`

You can use this command to edit a module's name in the TA-Tracker.

When a module is edited, the **Student View** will show the groups of the edited module and the students of the first group of the edited module. If you were on a different `tab`, you will automatically be switched to the `student tab`.

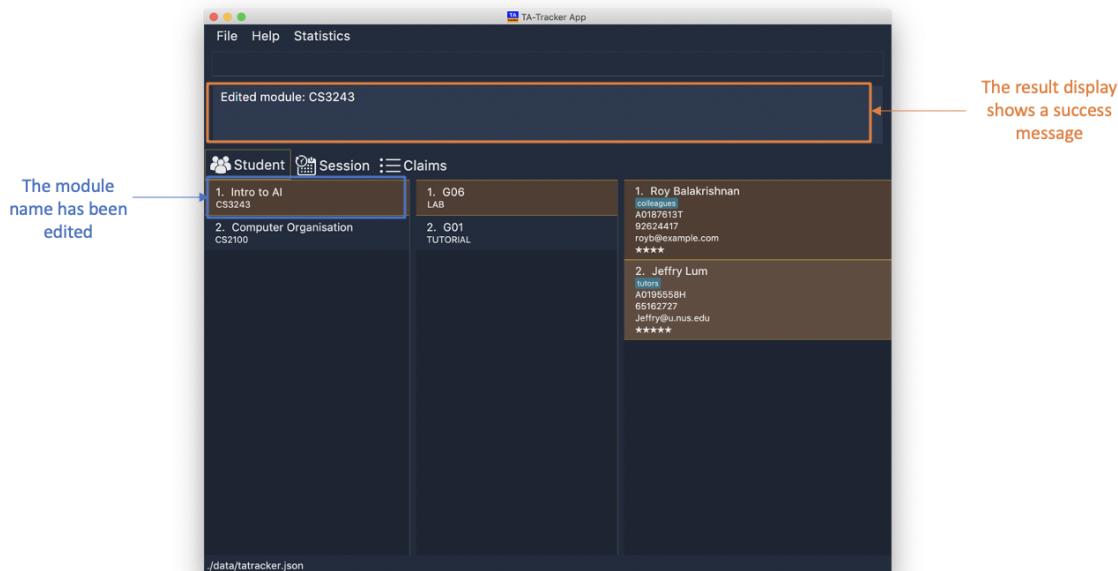
Format: `module edit m/MOD_CODE n/NEW NAME`

- You can only use this command to change the name of the module. The module code can't be changed.
- Editing a module doesn't affect the students and groups inside the module.
- `MOD_NAME` can't be an empty string or a sequence of spaces.
- You cannot edit a module that doesn't exist.

NOTE

Examples:

- `module edit m/CS3243 n/Intro to AI`



This will change the name of the module with module code `CS3243` to `Intro to AI`.

6.2.4. Adding a Group : `group add`

You can use this command to add a group to the TA-Tracker.

When a new group is created, the **Student View** will show the groups of the module this new group belongs to and students of the new group. If you were on a different **tab**, you will automatically be switched to the **student tab**.

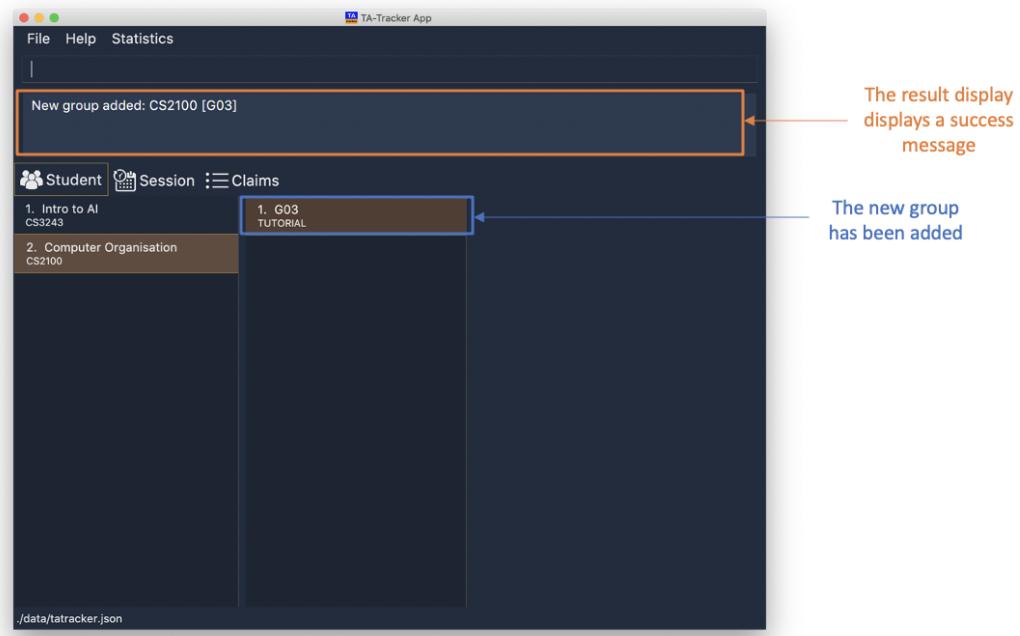
Format: `group add g/GROUP_CODE m/MOD_CODE t/GROUP_TYPE`

- `MOD_CODE` here refers to the module code of the module you want to add the group to.
- You cannot add a group to a module that doesn't exist.
- The `GROUP_CODE` can't be an empty string or a sequences of spaces.
- You cannot add multiple groups with the same group code into the same module.
- You can give a group your own custom `GROUP_CODE` (ex: `10AMGROUP`) if you find it easier to remember. However, we recommend you use the official group code for it.

NOTE

Examples:

- `group add g/G03 m/CS2100 t/tutorial`



This will add a group with the group code `G03`, which is a `tutorial`, inside the module that has module code `CS2100`.

6.2.5. Deleting a Group : `group delete`

You can use this command to delete a group from the TA-Tracker.

When a group is deleted from the TA-Tracker, all students in the group are also deleted.

When a group is deleted, the **Student View** will show the details of the first group of the module the group was deleted from. If you were on a different **tab**, you will automatically be switched to the **student tab**.

Format: `group delete g/GROUP_CODE m/MOD_CODE`

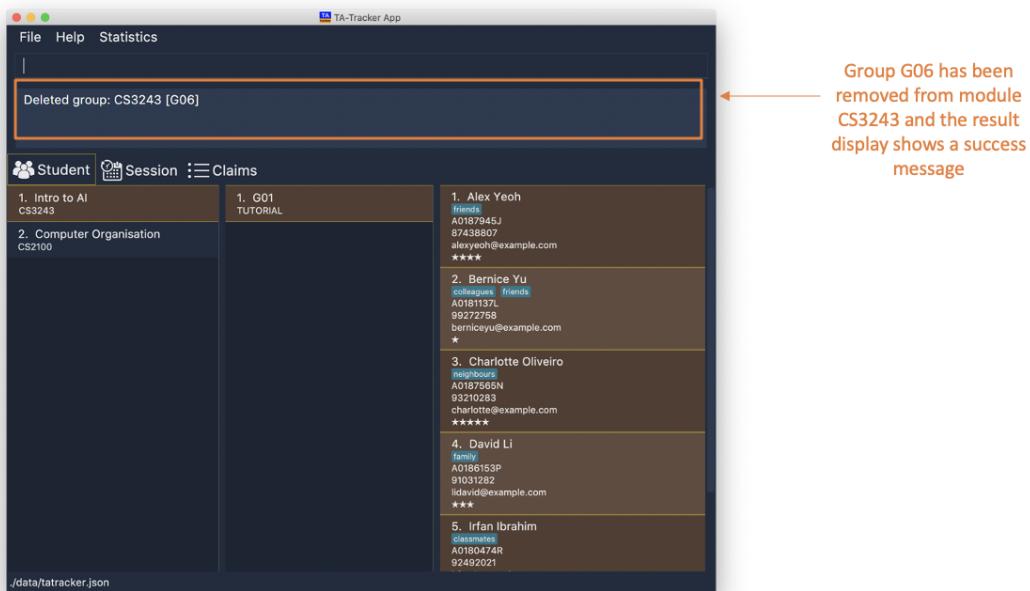
- `MOD_CODE` here refers to the module code of the module that contains the group that you want to delete.

NOTE

- You can't delete a group from a module that doesn't exist.
- A group with the given group code must exist inside the module before you can delete it.

Examples:

- `group delete g/G06 m/CS3243`



This will delete the group with the group code `G06` from the module that has module code `CS3243`.

6.2.6. Editing a Group : `group edit`

You can use this command to edit a group in the TA-Tracker.

This command can be used to change the group code and the group type of the group. The students inside the group will remain intact.

When a group is edited, the **Student View** will show the groups in the module that the edited group belongs to, as well as the students that belong to the edited group. If you were on a different **tab**, you will automatically be switched to the **student tab**.

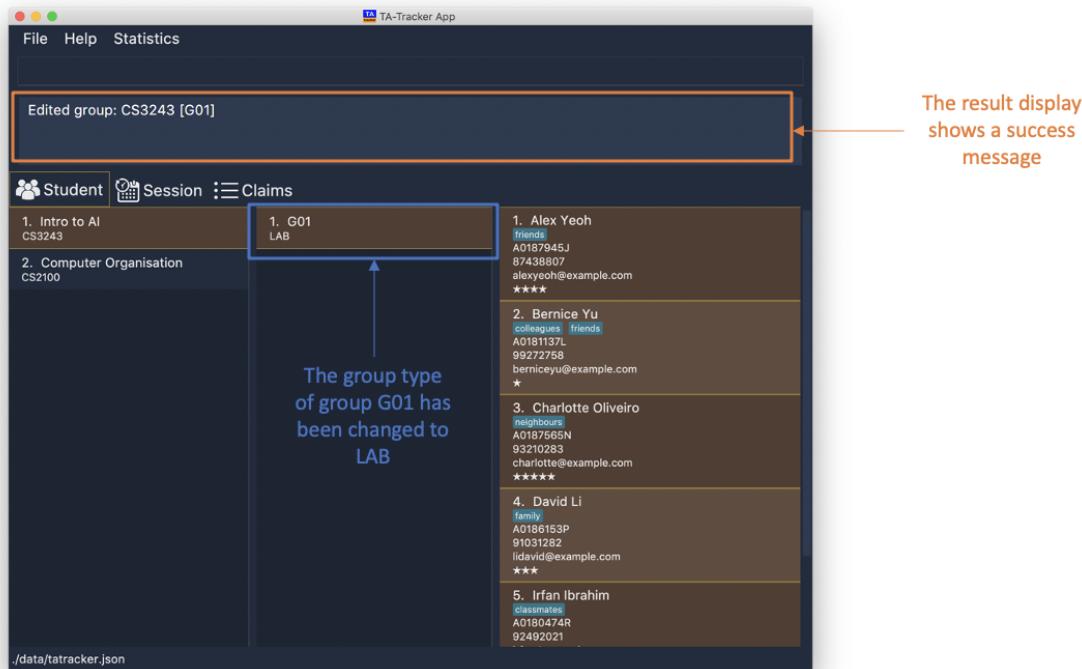
Format: `group edit g/GROUP_CODE m/MOD_CODE [ng/NEW_GROUP_CODE] [nt/NEW_GROUP_TYPE]`

- `MOD_CODE` here refers to the module code of the module that contains the group that you want to edit.
- You can't edit a group inside a module that doesn't exist.
- You can't edit a group that doesn't exist.
- If you are changing the group code, the module shouldn't contain a group that has the same group code as the new group code.
- While the `nt/` and `ng/` prefixes are optional, at least one of them must be mentioned.

NOTE

Examples:

- `group edit g/G01 m/CS3243 nt/lab`



This will change the group type of the group with group code **G01**, inside the module with module code **CS3243**, to be a **lab**.

6.2.7. Adding a Student : `student add`

You can use this command to add a new student to the TA-Tracker.

When a new student is added, the **Student View** will show that the new student is added into the student list of the provided module group. If you were on a different **tab**, you will automatically be switched to the **student tab**.

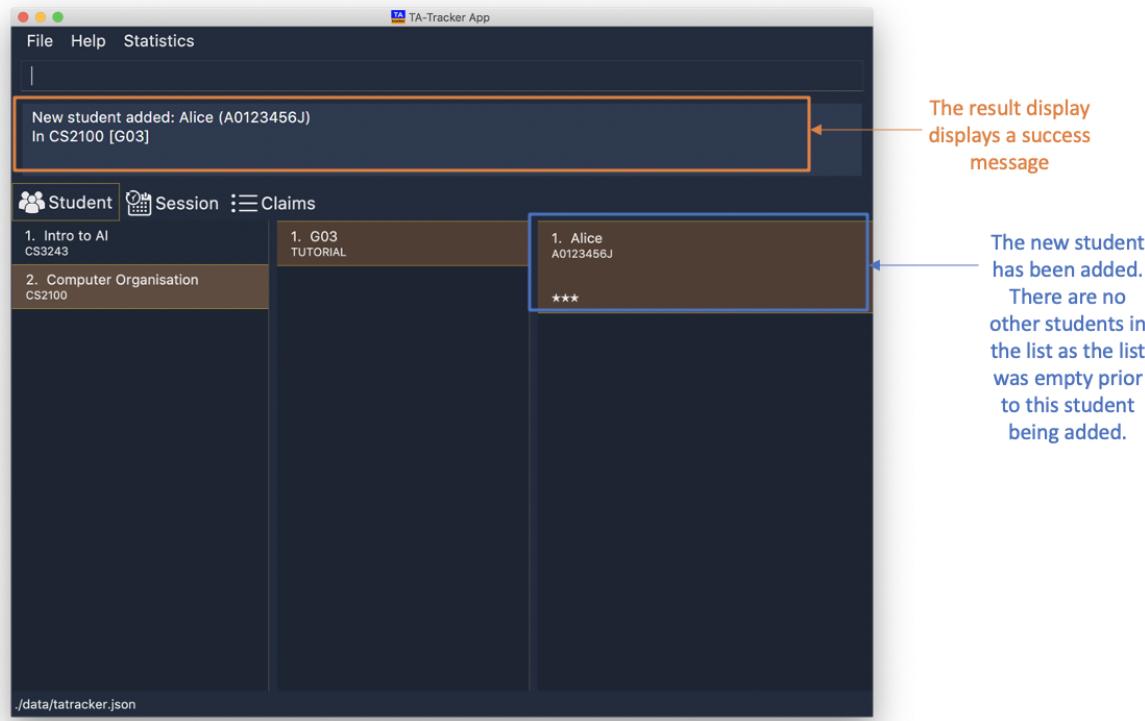
Format: `student add id/MATRIC_NUMBER n/NAME m/MOD_CODE g/GROUP_CODE [e/EMAIL] [r/RATING] [t/TAG]...`

- You cannot add a student to a module that does not exist in the TA-Tracker.
- You cannot add a student to a group that does not exist inside the given module.
- You cannot add multiple students with the same matric number inside the same module group.
- Student names are auto-capitalized. However, if the name includes a hyphen (-) with no space after the hyphen, the second half of the name will not be capitalised.

NOTE

Examples:

- `student add id/A0123456J n/Alice m/CS2100 g/G03`



This will add a student named Alice with the matriculation number `A0123456J` inside group `G03` of the module `CS2100`.

- * You can't edit the `MATRIC_NUMBER` once the student has been created, so do take care to ensure that the number is correct.
- You can use the rating feature to give your students participation marks.

TIP

- A student is given a default rating of 3/5 (average) if you do not specify a rating.
- You can either specify a rating in the add student command or,
- you can edit their rating later by using the `student edit` command (details given below in 5.2.9.)

6.2.8. Deleting a Student : `student delete`

You can use this command to delete a student from the TA-Tracker.

When a student is removed, the **Student View** will show that the student is removed from the student list of the provided module group. You will see an empty list if there are no more students inside the module group. If you were on a different **tab**, you will automatically be switched to the **student tab**.

Format: `student delete id/MATRIC_NUMBER g/GROUP_CODE m/MOD_CODE`

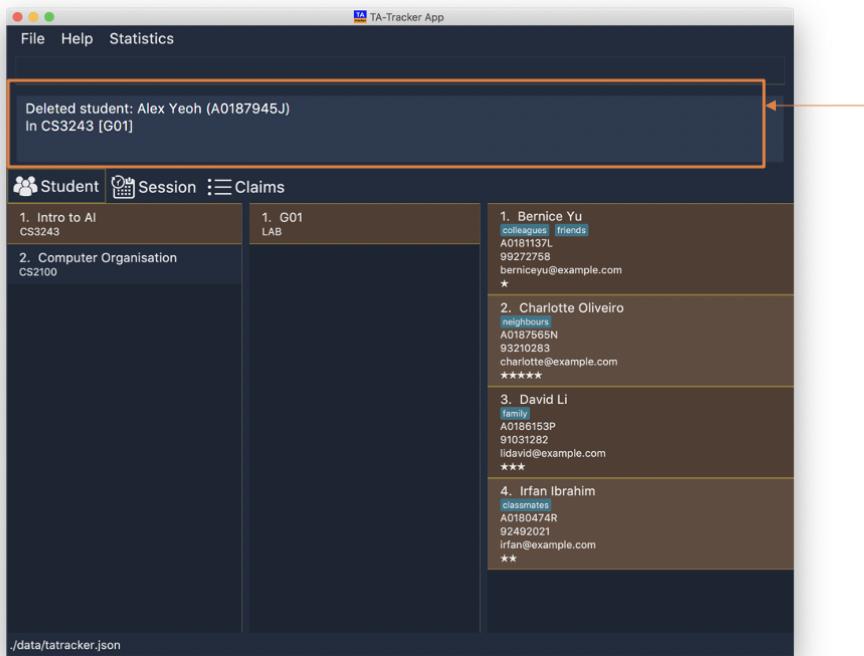
- You cannot remove a student from a module that does not exist in the TA-Tracker.

NOTE

- You cannot remove a student from a group that does not exist inside the given module.
- You cannot remove a student that does not exist inside the given module group.

Examples:

- `student delete m/CS3243 g/G01 id/A0187945J`



Deletes the student with the matriculation number `A0187945J` from group `G01` of the module `CS3243`.

6.2.9. Editing a Student : `student edit`

You can use this command to edit a student in the TA-Tracker.

When a student is edited, the **Student View** will show that the student has been edited in the student list of the provided module group. If you were on a different `tag`, you will automatically be switched to the `student tab`.

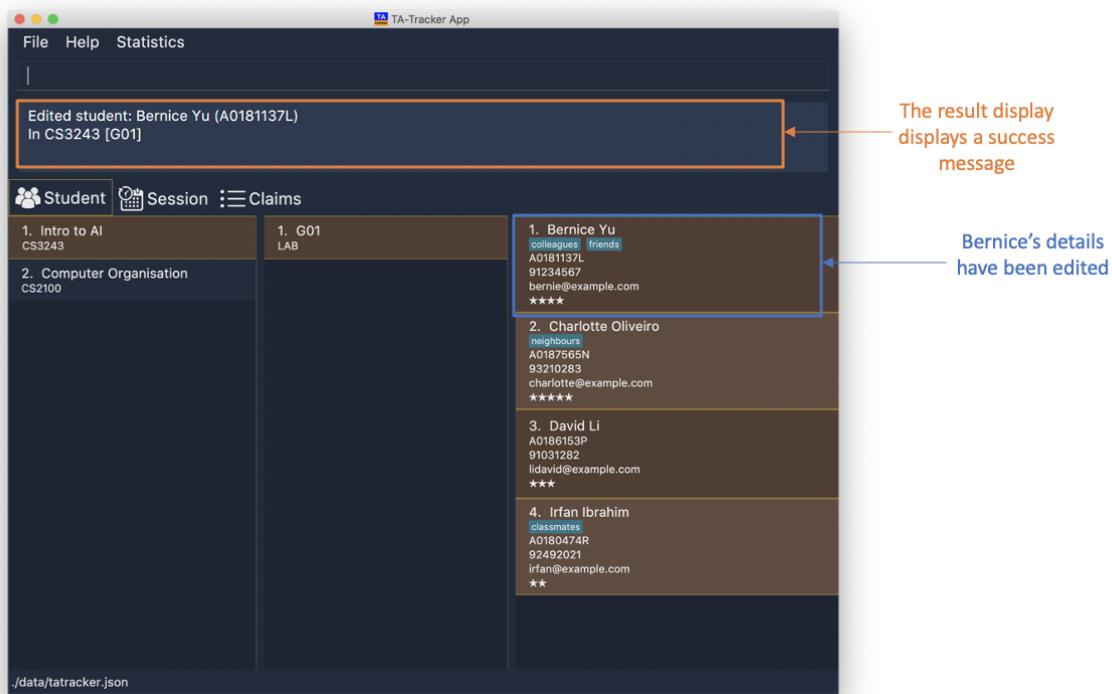
Format: `student edit id/MATRIC_NUMBER m/MOD_CODE g/GROUP_CODE [n/NAME] [e/EMAIL] [r/RATING] [t/TAG]...`

- You cannot edit a student in a module that does not exist in the TA-Tracker.
 - You cannot edit a student in a group that does not exist inside the given module.
- NOTE**
- You cannot edit a student that does not exist inside the given module group.
 - You cannot edit a student's matriculation number
 - You must edit the student with at least one of the optional fields.

- TIP**
- If you edit the tags of a student, the new tags will replace the old tags.
 - You can remove all tags from a student with an empty tag
(i.e. typing **t/** without specifying any tags after it).

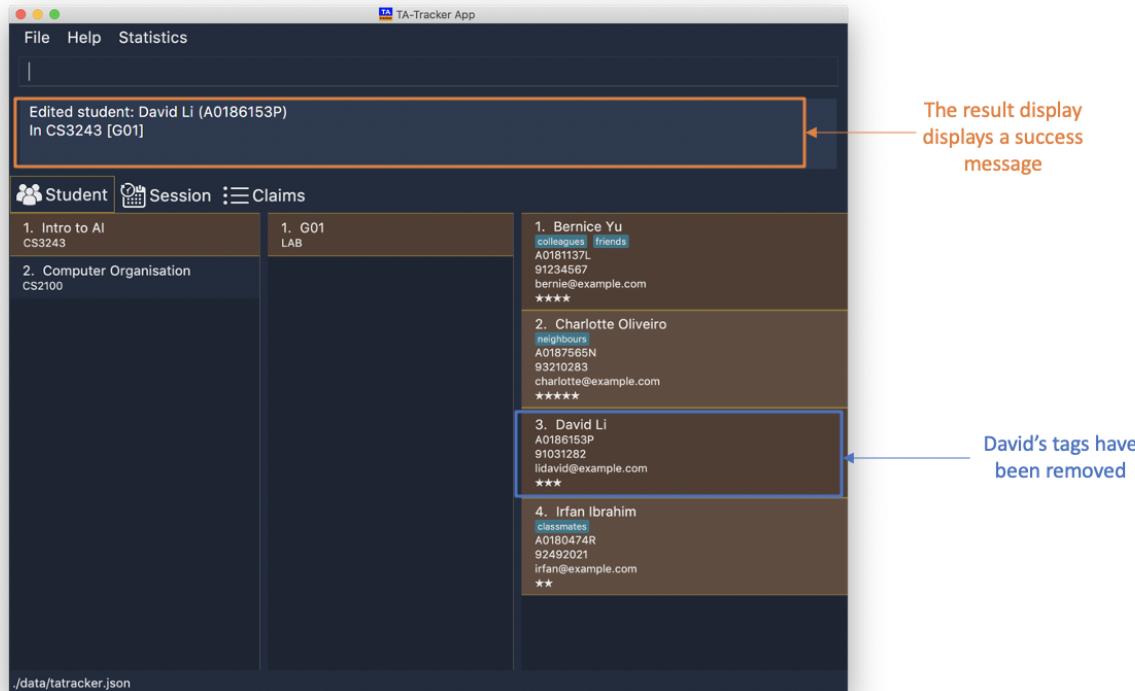
Examples:

- `student edit id/A0181137L g/G01 m/CS3243 p/91234567 e/bernie@example.com r/4`



Edits the student with the matriculation number `A0181137L` from group `G01` of module `CS3243` to have:

- The new rating of `4`
- The new phone number `91234567`
- The new email address `bernie@example.com`
- `student edit m/CS3243 g/G01 id/A0186153P t/`



Edits student with the matriculation number **A0186153P** to have:

- All existing tags removed

6.2.10. Sorting a Group : `sort group`

You can use this command to chose how you want to sort your students in a specific group. The students will be sorted according to your specified **SORT_TYPE**.

The **Student View** will show you the students inside the group that you have sorted. If you were on a different **tab**, you will automatically be switched to the **student tab**.

Format: `sort group g/GROUP_CODE m/MOD_CODE t/SORT_TYPE`

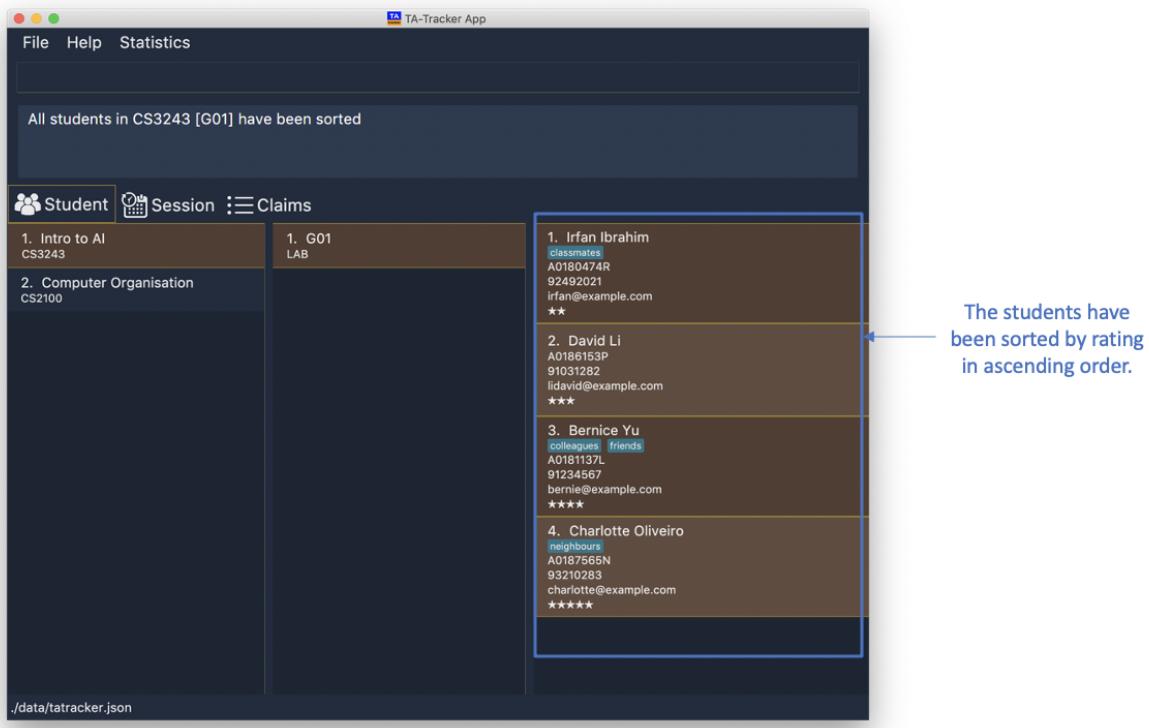
- You cannot sort a group inside a module that doesn't exist.
- You cannot sort a group that doesn't exist.
- To sort alphabetically you can use the following to indicate sort type:
 - **alphabetically**

NOTE

- **alphabetical**
- **alpha**
- To sort by matriculation number, sort type must be **matric**.
- To sort by rating in ascending order, sort type must be **rating asc**.
- To sort by rating in descending order, sort type must be **rating desc**.

Examples:

- `sort group g/G01 m/CS3243 t/rating asc`



Sorts the student inside `G01` of module `CS3243` by `rating asc` (that is, rating in ascending order).

6.2.11. Sorting a Module : `sort module`

You can use this command to chose how you want to sort your students in a specific module. The students will be sorted according to your specified `SORT_TYPE`.

The **Student View** will show you the students inside the first group of the module that you have sorted. If you want to see the other groups, you can use the `student filter` command (details given below in 5.2.13.) If you were on a different `tab`, you will automatically be switched to the `student tab`.

Format: `sort module m/MOD_CODE t/SORT_TYPE`

- You cannot sort a module that doesn't exist.
- To sort alphabetically you can use the following to indicate sort type:
 - **alphabetically**
 - **alphabetical**
 - **alpha**
- To sort by matriculation number, sort type must be **matric**.
- To sort by rating in ascending order, sort type must be **rating asc**.
- To sort by rating in descending order, sort type must be **rating desc**.

NOTE

Examples:

- `sort module m/CS2103 t/alpha`

Sorts all students inside all groups inside the module **CS2103 alphabetically**.

6.2.12. Sorting Everything : `sort all`

You can use this command to chose how you want to sort all of your students. The students will be sorted according to your specified **SORT_TYPE**.

The **Student View** will show you the students inside the first group of the first module in the TA-Tracker. If you want to see the other groups or modules, you can use the **student filter** command (details given below in 5.2.13.) If you were on a different **tab**, you will automatically be switched to the **student tab**.

Format: `sort all t/SORT_TYPE`

- To sort alphabetically you can use the following to indicate sort type:
 - **alphabetically**
 - **alphabetical**
 - **alpha**
- To sort by matriculation number, sort type must be **matric**.
- To sort by rating in ascending order, sort type must be **rating asc**.
- To sort by rating in descending order, sort type must be **rating desc**.

NOTE

Examples:

- `sort all t/alpha`

Sorts all students in all groups inside all modules **alphabetically**.

6.2.13. Filter : filter

You can use this command to see the students in a particular group and module.

You can filter the students in **Student View** in the following ways:

1. module code and group code `m/ g/`
2. module code `m/`

You will see the group with the specified `GROUP_CODE` and the module with the specified `MOD_CODE` highlighted in orange in the list.

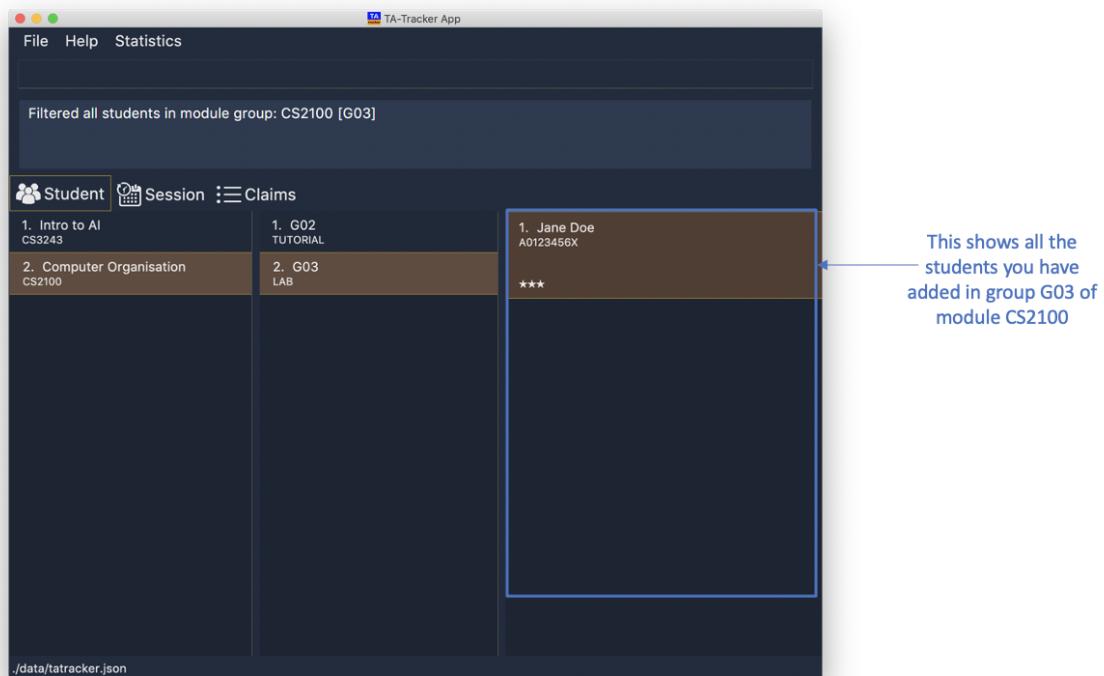
If you were on a different **tab**, you will automatically be switched to the **student tab**.

Format: `student filter m/MOD_CODE [g/GROUP_CODE]`

- NOTE**
- Using both module code and group code will show you the students inside the group with the given group code.
 - The group must belong to the module specified by the module code.

Examples:

- `student filter m/cs2100 g/g03`



This will show you all students in module `CS2100`, under group `G03`.

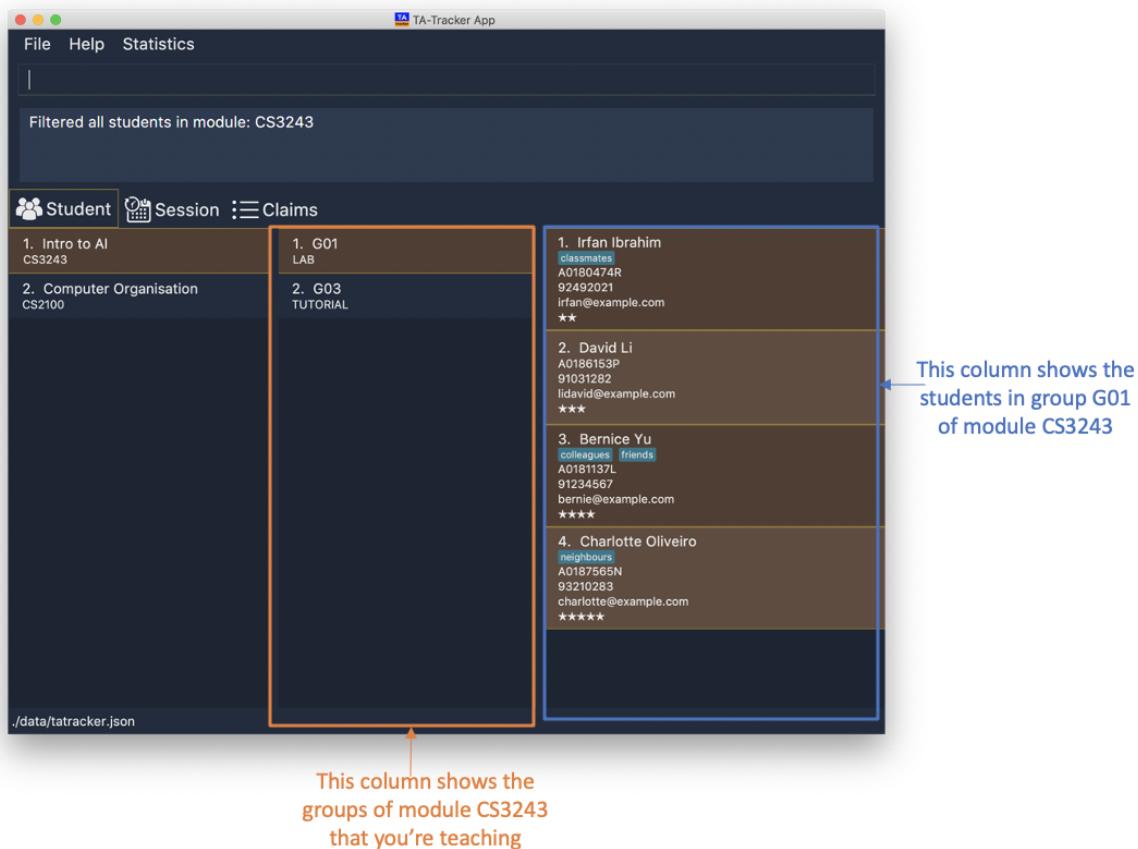
If you use just the module code, you will see:

- The first group of the module specified by the given module code highlighted in orange
- All the students belonging to that group
- You can use this command when you want to see all the groups belonging to a particular module.

Format: `student filter m/MOD_CODE`

Examples:

- `student filter m/cs3243`



This shows you the students in the first group of the module `CS3243`

NOTE **Keywords are case-insensitive. e.g. `cs2103t` is the same as `CS2103T`**

6.3. Session View

This section explains the different commands that can be used in the **Session View**.

6.3.1. Adding a Session : `session add`

You can use this command to add a new session.

The new session will be shown in the **Session View**, and will automatically be marked as `not done`.

If you were on a different **tab**, you will automatically be switched to the **session tab**.

Format: `session add m/MOD_CODE [s/START] [e/END] [d/DATE] [w/WEEKS] [t/SESSION_TYPE] [n/NOTES]`

1. If you want to create a session with the current date, start time and end time, you can omit **s/START**, **e/END** and **d/DATE**.

2. You can edit the end time **e/END** and other details later by using the **session edit** command (details given below in 5.3.3.).

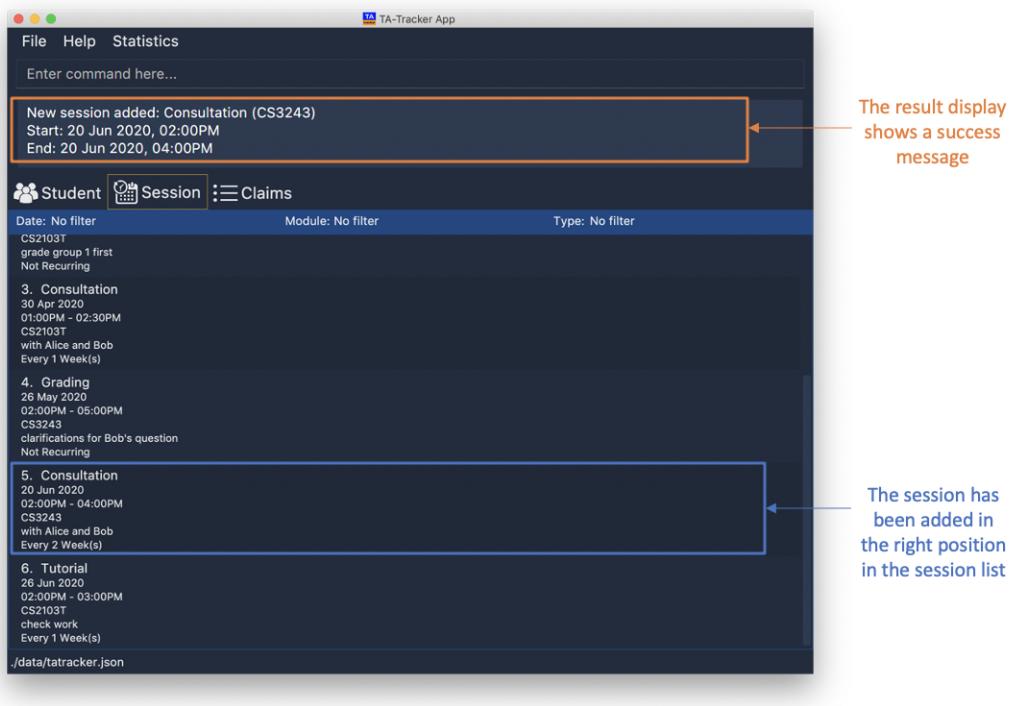
TIP

3. If the session has a recurring period **[w/WEEKS]**, a new session with the updated date and timing will be added to **Session View** when it is marked as done.

4. You will find the recurring period **[w/WEEKS]** field handy for tasks that happen once every few weeks such as your weekly lab sessions or fortnightly assignment grading.

Examples:

- `session add m/CS3243 s/14:00 e/16:00 d/2020-06-20 w/2 t/consultation n/with Alice and Bob`



Adds a new session with:

- `START_TIME` at 14:00 in `24hr` format
- `END_TIME` at 16:00 in `24hr` format
- `DATE` on 2020-06-20 in `yyyy-MM-dd` format
- `WEEK` recurring period of 2 weeks: Once this session has been marked as done, a new session will be created with `DATE d/2020-07-04` (2 weeks later from the date of the original session)
- `MOD_CODE` of CS3243
- `TYPE` consultation
- `NOTE` "with Alice and Bob"

6.3.2. Deleting a Session : `session delete`

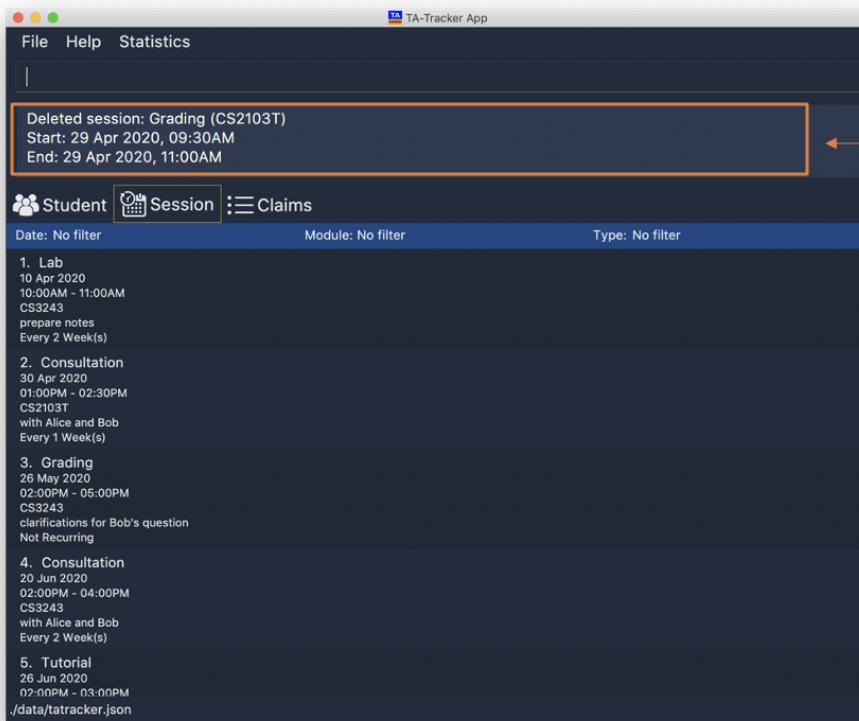
You can use this command to delete a session at a specific `index`.

If you were on a different `tab`, you will automatically be switched to the `session tab`.

Format: `session delete INDEX`

CAUTION | Do not confuse the `session delete` and `session done` commands.

Examples:



The session at index 1 of the session list is deleted and the result display shows a success message

- **session delete 1**

This command deletes the first session in the **Session View**.

6.3.3. Editing a Session : **session edit**

You can use this command to edit a session in the TA-Tracker.

If you were on a different tab, you will automatically be switched to the **session tab**.

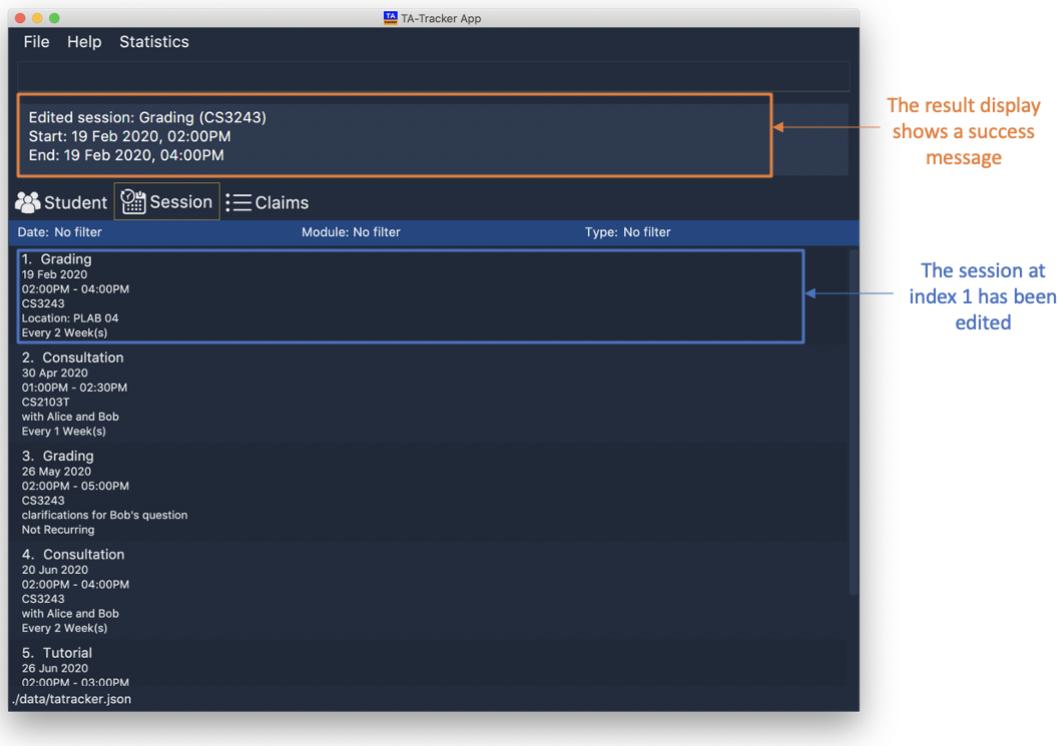
Format: **session edit INDEX [s/START_TIME] [e/END_TIME] [d/DATE] [w/RECUR] [m/MODULE] [t/SESSION_TYPE] [n/NOTES]**

NOTE

- You must edit the session with at least one of the optional fields.

Examples:

- `session edit 1 s/14:00 e/16:00 d/2020-02-19 t/grading n/Location: PLAB 04`



Edits the session at index 1 to have:

- The new `START_TIME` at 14:00
- The new `END_TIME` at 16:00
- The new `DATE` on 2020-02-19 in `yyyy-MM-dd` format
- The new `TYPE` grading
- The new `NOTE` "Location: PLAB04"

6.3.4. Marking a Session as Done : `session done`

You can use this command to mark a session as done.

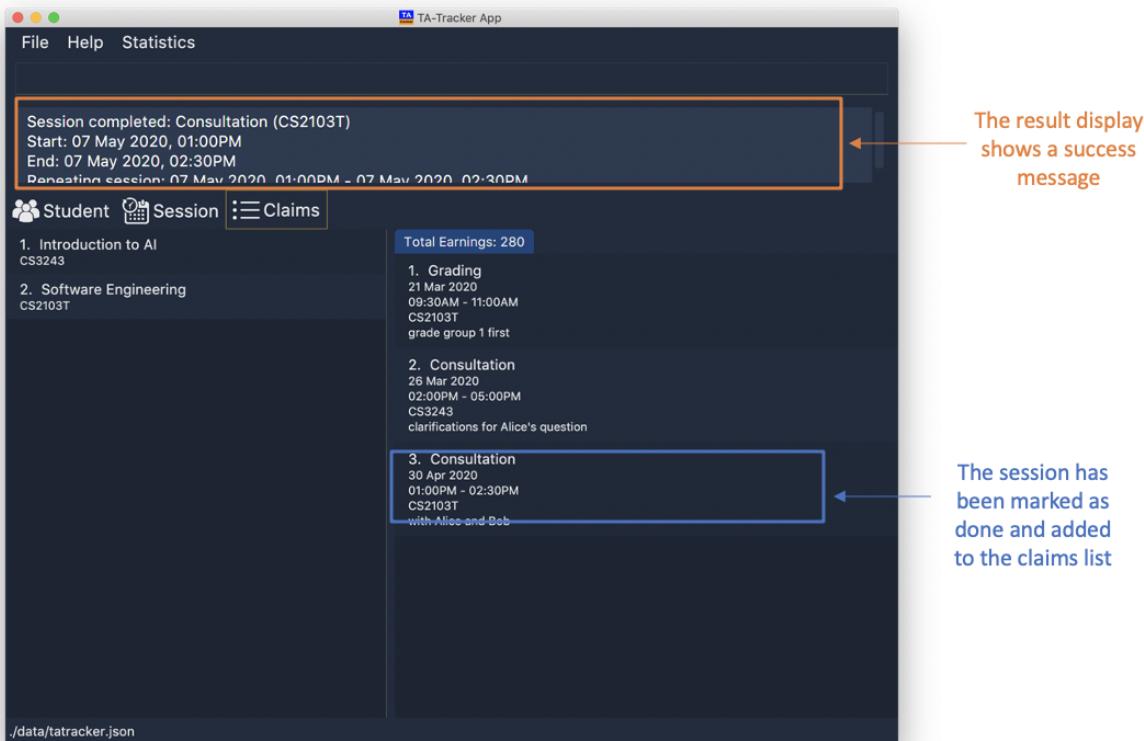
The session marked as done will be removed from the **Session View** and will automatically appear as a new claim in the **Claims View**. If the session marked done has a recurring period, a new session with the updated date and timing will be added to **Session View**.

If you were on a different `tab`, you will automatically be switched to the `claims tab`.

Format: `session done INDEX`

Marks the session with the given unique index as done.

Examples:



- `session done 2`

This will mark the 2nd session in the **Session View** as done.

6.3.5. Filter :filter

You can use this command to filter the sessions in the **Session View**.

You can filter the sessions in the following ways:

- `MOD_CODE` - filtering by module code will show you only the sessions affiliated with that module.
- `SESSION_TYPE` - filtering by session type will show you only the sessions with the specified type
- `DATE` - filtering by a date will show you only the sessions on that date.

The keyword is case-insensitive.

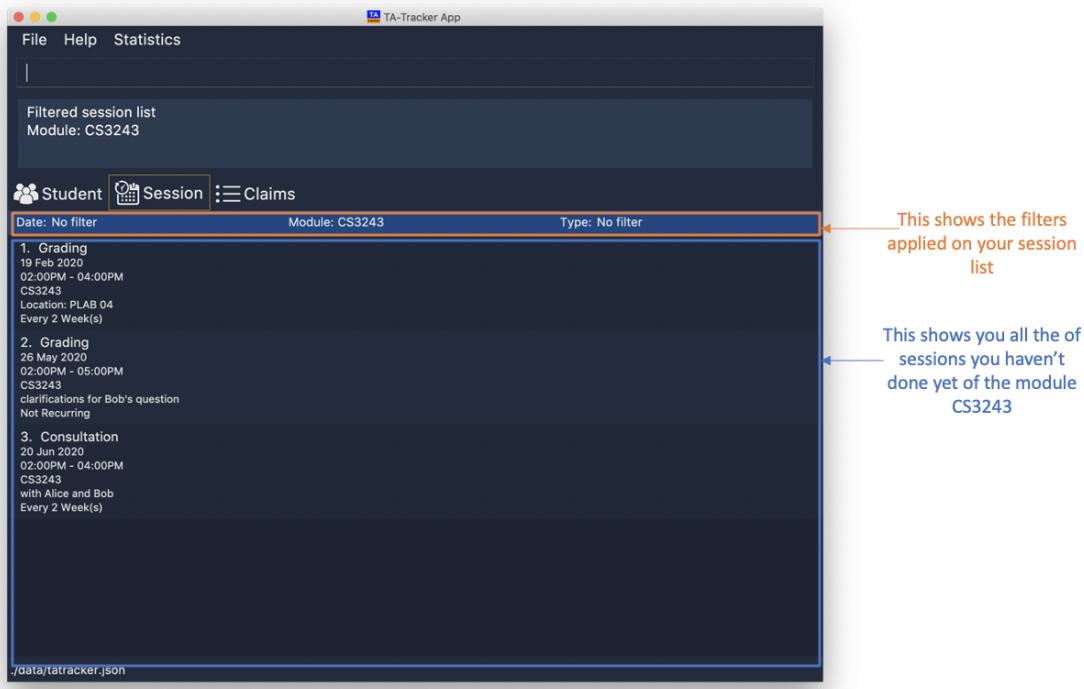
When you execute a new filter command, any previous filters applied on the sessions will be removed and only the filters from the latest command will be applied on the sessions list. The filters that are currently being applied are shown at the top of the **Session View**

If you were on a different `tab`, you will automatically be switched to the `session tab`.

Format: `session filter [m/MOD_CODE] [t/SESSION_TYPE] [d/DATE]`

Example:

- session filter m/CS3243

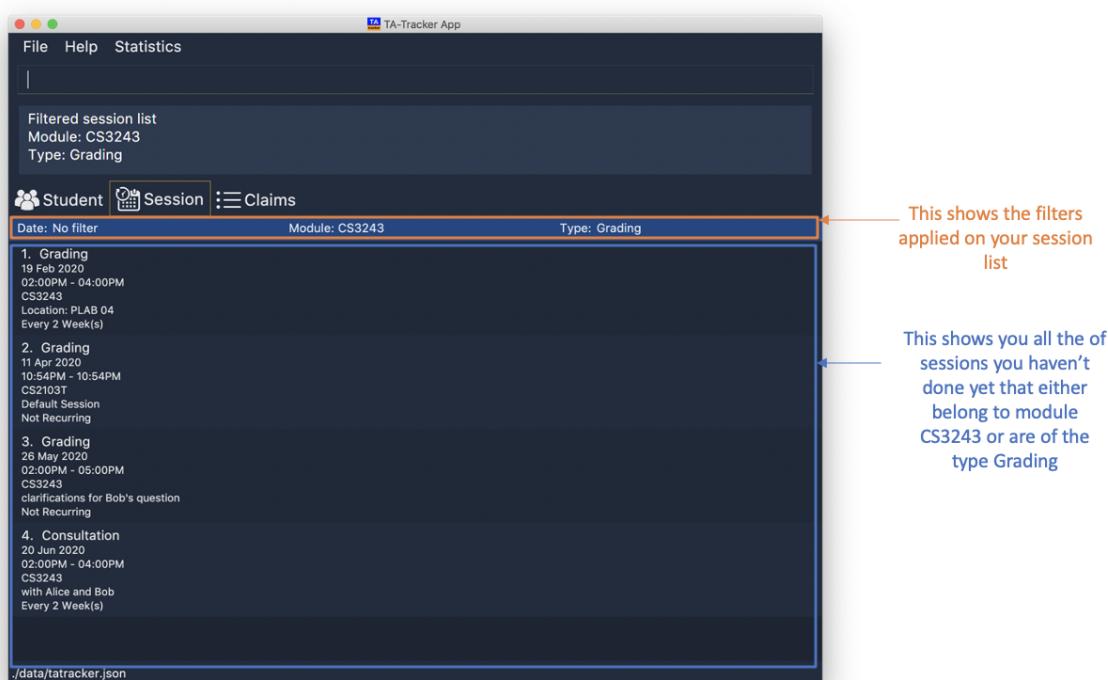


All sessions in the **Session View** with module code **CS3243** are displayed.

- session filter m/CS3243 t/tutorial

All sessions in the **Session View** with module code **CS2103T**, or session type **Tutorial** will be shown.

- session filter m/CS3243



All sessions in the **Session View** with module code **CS2103T** will be shown.

NOTE

As long as a session contains one of the keyword given by the user, the session will be shown to the user.

6.4. Claims View

6.4.1. Filter : **filter**

You can use this command to filters sessions in the **Claims View** by the module code.

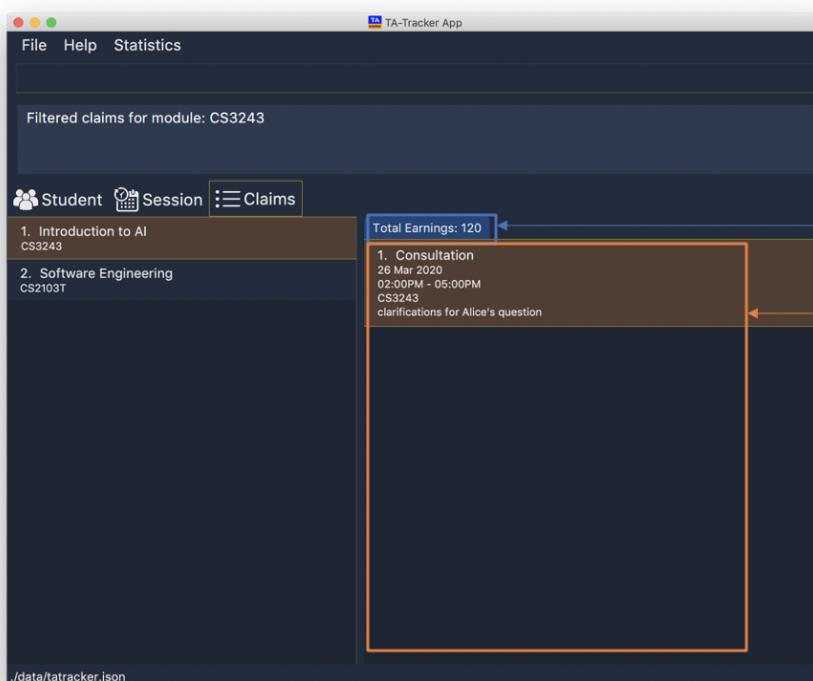
You will see the module with the specified **MOD_CODE** highlighted in orange in the list.

All completed sessions with matching **MOD_CODE** are displayed. Keyword is case-insensitive.

Format: **claims filter m/MOD_CODE**

Example:

- **claims filter m/CS3243**



All sessions in the **Claims View** with module code **CS3243** are displayed.

6.4.2. Changing the hourly rate : **setrate**

Sets the hourly rate for the total income and claim computation.

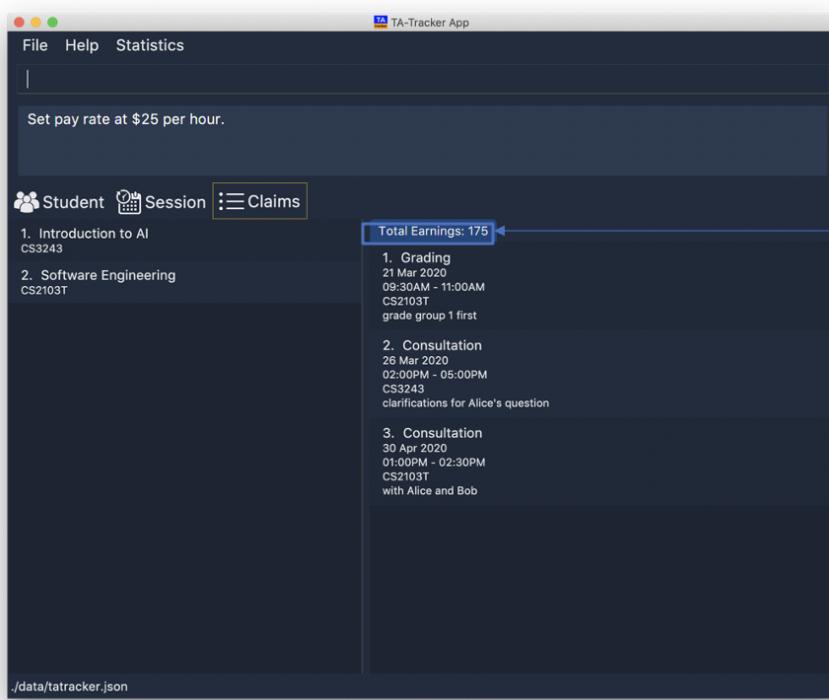
Format: **setrate RATE**

NOTE

- If you don't specify a rate, it is set at \$40 by default (the rate at which most SOC TAs are being paid per hour).
- **RATE** is the amount you want to change the hourly rate to, this value will be used to calculate the **Total Earnings** label in the **Claims Tab** as well as the **Statistics Window**.
- The **RATE** must be a positive integer.

Examples:

- **setrate 25**



Sets the current hourly rate to \$25.

6.5. Statistics Window

6.5.1. Generate Statistic Report : **report**

You can use this command to generate a report to display information such as:

- A breakdown and summary of completed sessions
- The number of hours of each type of completed sessions
- A breakdown of your student's ratings

Optionally, you can specify a module code. If a module code is specified, the report generated will only include data from the specified module.

Pressing the **esc** key on your keyboard will close the statistics window.

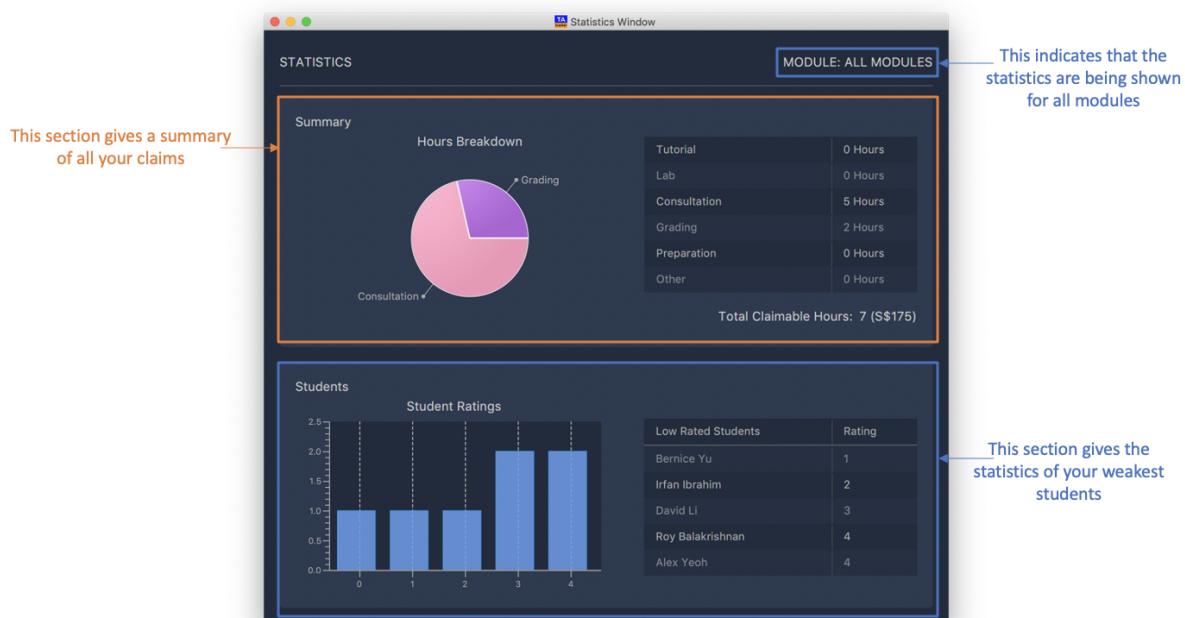
Format: **report [MOD_CODE]**

NOTE

- Similar to the **Claims View**, the report will only display sessions that have been marked as done.
- Total Claimable Hours** is computed using the current specified **rate**. See Section 6.4.2, “**Changing the hourly rate : setrate**”.

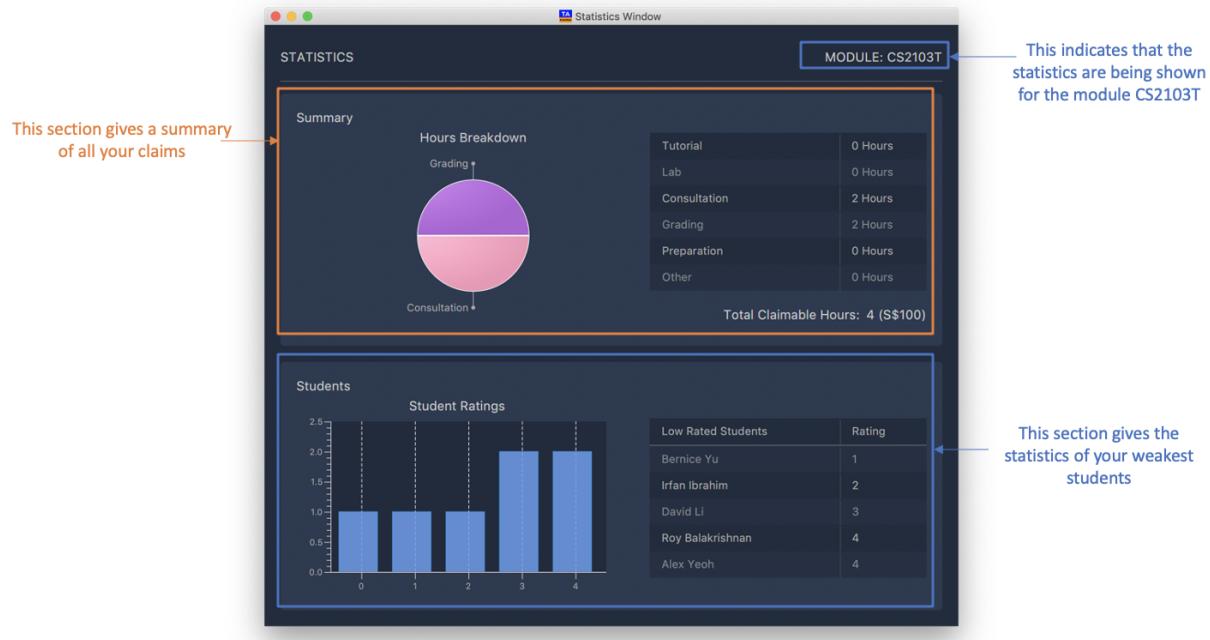
Example:

- **report**



Generate and display a report of sessions and students from all modules.

- **report CS2103T**



Generate and display a report of sessions and students from the module CS3243.

7. Glossary

Term	Explanation	Examples
TSS Claims Form	This refers to the claims form that Teaching Assistants at NUS School of Computing have to fill up at the end of each semester to claim money for the tasks they have completed.	
TA	This is the short form for `Teaching Assistant.	
SOC or SoC	This is the short form for School of Computing.	
CLI	This is the short form for Command Line Interface. It processes commands to TA-Tracker in the form of lines of text	
GUI	This is the short form for Graphical User Interface. It is a form of user interface that allows users to interact with electronic devices through graphical icons	
Index	This refers to the position of an item on a list.	Index of 1 refers to the first item in a list.

Term	Explanation	Examples
Matric Number	This refers to a student's matriculation number.	A0123456X
Group	The is the general term given to a group of students a TA teaches.	lab , tutorial , recitation
TAT	This is the short form of TA-Tracker.	
NUS	This is the short form of National Univeristy of Singapore.	
Module	Refers to one of the academic courses in NUS.	
Tutorial	A tutorial is a regular meeting between a tutor and one or several students, for discussion of a subject that is being studied.	

8. FAQ

Q: How can I transfer my data to another Computer?

A: You can do so by first installing the app in another computer. You can then replace the empty data file it creates with the data file of your previous TA-Tracker folder.

9. Command Summary

9.1. Navigation

- View Help: `help`
- Switch Tabs: `goto TAB_NAME`
- Exit the Program: `exit`

9.2. Student View

9.2.1. Module commands

- Add Module: `module add m/MOD_CODE n/NAME`
- Edit Module: `module edit m/MOD_CODE n/NEW NAME`
- Delete Module: `module delete m/MOD_CODE`

9.2.2. Group commands

- Add Group: `group add g/GROUP_CODE m/MOD_CODE t/GROUP_TYPE`
- Edit Group: `group edit g/GROUP_CODE m/MOD_CODE [ng/NEW_GROUP_CODE] [nt/NEW_GROUP_TYPE]`

- **Delete Group:** `group delete g/GROUP_CODE m/MOD_CODE`

9.2.3. Student commands

- **Add Student:** `student add id/MATRIC_NUMBER n/NAME m/MOD_CODE g/GROUP_CODE [e/EMAIL] [r/RATING] [t/TAG]…`
- **Delete Student:** `student delete id/MATRIC_NUMBER m/MOD_CODE g/GROUP_CODE`
- **Edit Student:** `student edit id/MATRIC_NUMBER m/MOD_CODE g/GROUP_CODE [n/NAME] [e/EMAIL] [r/RATING] [t/TAG]…`

9.2.4. Others

- **Sort Group:** `sort group g/GROUP_CODE m/MOD_CODE t/SORT_TYPE`
- **Sort Module:** `sort module m/MOD_CODE t/SORT_TYPE`
- **Sort All Modules:** `sort all t/SORT_TYPE`
- **Filter Students:** `student filter m/MOD_CODE [g/GROUP_CODE]`

9.3. Session View

9.3.1. Session commands

- **Add Session:** `session add m/MOD_CODE [s/START] [e/END] [d/DATE] [w/WEEK] [t/SESSION_TYPE] [n/NOTES]`
- **Delete Session:** `session delete INDEX`
- **Edit Session:** `session edit INDEX [s/START_TIME] [e/END_TIME] [d/DATE] [w/RECUR] [m/MODULE] [t/SESSION_TYPE] [n/NOTES]`
- **Done Session:** `session done INDEX`

9.3.2. Others

- **Filter Sessions:** `session filter [d/DATE] [m/MOD_CODE] [t/SESSION_TYPE]`

9.4. Claims View

- **Set Rate:** `setrate RATE`
- **Filter Claims:** `claims filter m/MOD_CODE`

9.5. Statistics

- **Generate Statistics Report:** `report [MOD_CODE]`