https://nus-cs2103-ay1920s2.github.io/website/admin/projectList.html

Submission: Submit your product name, target user profile, the value proposition, and the public link to your collaborative project notes via TEAMMATES. You'll receive an email from TEAMMATES with the submission link. Only one member needs to submit on behalf of the team. All members can view/update the submission.
Submission link will be sent to you by Mon, Aug 31st (reason: we need a few days to set up the submission system *after* teams have been finalized).

https://nus-cs2103-ay1920s2.github.io/website/admin/projectList.html

https://se-education.org/addressbook-level3/DeveloperGuide.html#product-scope

Product Name:

GradPad

Target user profile

Anybody → Students → University Students → tech savvy University student → Computer Science students
NUS Computer Science undergraduate students, who wish to keep track of their necessary modules (how many are done, how many are left), and also the total MCs tabulation.

Problems addressed

Currently, students have no faster way to check their required modules other than sieving through the NUS website.

Currently, students have no efficient way of tracking their module progress and have to resort to keeping track either through notepads or an excel sheet.

Value proposition

Keep track of your degree progress and modules taken during your time in NUS with ease. Faster and more lightweight than traditional GUI applications, view and update your progress by issuing simple text commands. Modules are conveniently categorized into their respective groupings e.g. Unrestricted Electives, Computer Science Foundations, etc.

User Stories

https://trello.com/invite/b/VXqGflTJ/34653a27e5a2e5eadef9731bbb6e45c9/user-stories

User persona:
John is a Computer Science undergraduate studying at NUS. As a responsible student who takes charge of his own learning, John spends a lot of time and effort tracking the modules he's taken together with his overall degree progress. Since John is a coder, he is adept at using CLIs. He is also a fast typer and likes using text-based interfaces over GUIs that require him to reach for his mouse.

Problem scope:
GradPad helps John keep track of the modules he's taken and his overall degree progress. It allows John to save modules he's taken. John can also see all past modules he's saved in the app. John can specify the module credits for each of the modules he adds. He can then view his total accumulated module credits so far. GradPad allows John to do all this via text commands.

Scenarios:

1. John uses the app for the first time
    a. John wants to know what he can do with the app
    b. John wants to know what commands are available to that he can start using the app
2. It's the end of the semester and John knows he has passed all his modules
    a. John wants to add his modules to the app
    b. John wants to specify how many MCs each module has
    c. John wants to see the modules he's added graphically
    d. John wants to see how many MCs he's done so far in NUS
3. The new semester is starting and John is planning what modules to take
    a. John wants to see all the modules he's taken so far
    b. John wants to see how many MCs he has to go before fulfilling his graduation requirements
    c. John wants to see a list of all SoC modules
        i. John wants to filter out SoC modules he's already taken
    d. John wants to know the prerequisites, corequisites, and preclusions for a module so that he knows how to plan
4. It's 3am but John wants to do last-minute module planning despite being sleepy
    a. John adds the wrong module and needs to delete it
    b. John adds the right module, but specifies the wrong no. of MCs. He needs to edit the module he's added
    c. John knows the name of the module he wants to take, but can't remember the module code, he wants to be able to search the list of modules for it

    d. John remembers a friend telling him to take CS2103T, but he can't remember what the module is titled. He wants to be able to find the module title.
    e. John wants to toggle the app to night-mode so that the white background will stop straining his eyes in the dark.
5. John is now a y4 student and is tired of module planning
    a. John wishes the app could just recommend him some modules to take
    b. John has taken way too many modules since y1 and can't remember all of them, he wants to easily bring up a list containing them
    c. John wants to take a module with prerequisites but can't remember all the past modules he's taken. He wishes the app could automatically tell him if he's eligible.
    d. With so many modules added over the years, John wants a way to categorize the modules he's taken so far

V1.2 Refactorings

Person package

- Name class → ModuleCode class
    - Update validation regex
    - Update NameTest class → ModuleCodeTest
- Phone class → ModularCredits class
    - Update PhoneTest class → ModularCreditsTest class
- Maintain Tag class first
- Remove all other field classes
- Person class → Module class
    - Update to match new fields
    - Update PersonTest class → ModuleTest class
- TypicalPersons class → TypicalModules class
    - Added CS2103 and CS3216 as typical modules
    - All other Persons are still there for now
- Add Module field templates to CommandTestUtil class
    - VALID_CODE, VALID_CREDITS, VALID_TAGS
    - All other templates for Person are still there for now
- Update SampleDataUtil class to create list of modules (not persons)
- UniquePersonList → UniqueModuleList
    - Update UniquePersonList test → UniqueModuleList test
- NameContainsKeywordsPredicate → ModuleCodeContainsKeywordsPredicate
    - Update the test class
- Update exceptions
    - DuplicatePersonException → DuplicateModuleException
    - PersonNotFoundException → ModuleNotFoundException
- Update PersonBuilder class in test package → ModuleBuilder class
- Rename entire Person package → Module package

Model package

- Rename all methods in Model interface
- Refactor ModelManager methods to match interface and new Module class
- AddressBook class → GradPad class
    - UniquePersonList field → UniqueModuleList
- ReadOnlyAddressBook interface → ReadOnlyGradPad interface

Storage package

- Update Storage interface to use GradPad instead of AddressBook
- Update StorageManager to match new interface
- AddressBookStorage interface → GradPadStorage interface
- JsonAddressBookStorage class → JsonGradPadStorage class
- JsonSerializableAddressBook class → JsonSerializableGradPad class
- JsonAdaptedPerson class → JsonAdaptedModule class

Command package

- EditPersonDescriptor class → EditModuleDescriptor class
    - Update all fields and their setters and getters
- Update EditPersonDescriptorBuilder class in test package
- Update PersonUtil test class
- Update EditCommand to work with Modules

Ui Package

- PersonCard → Module Card
    - Update all fields inside also
- PersonListPanel → ModuleListPanel
- PersonListViewCell → ModuleListViewCell

Parser

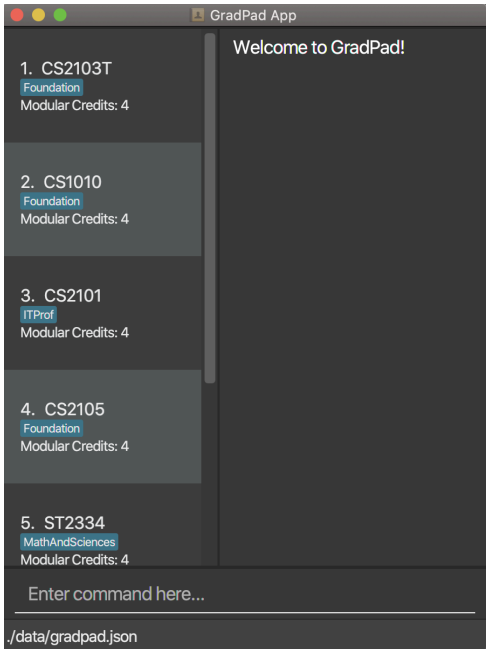
- Update CLISyntax class with new fields
- Update ParserUtil class (remove irrelevant methods)
- Update AddCommand and EditCommand class accordingly
- Update ParserUtilTest class

KIV stuff (can push to v1.3)

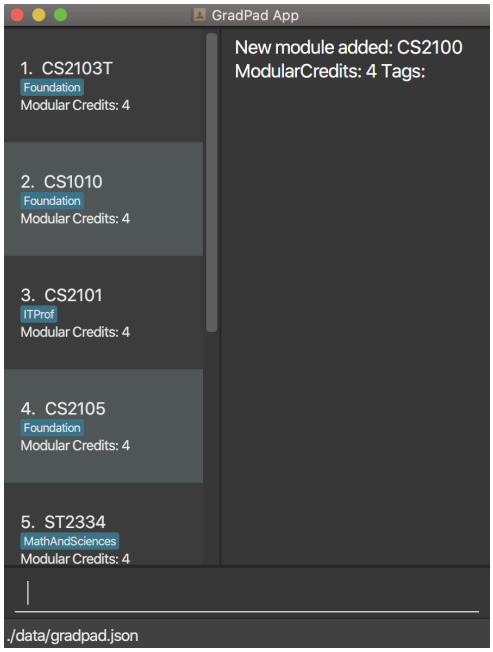
- Find feature
- Support module names

V1.2 features demo

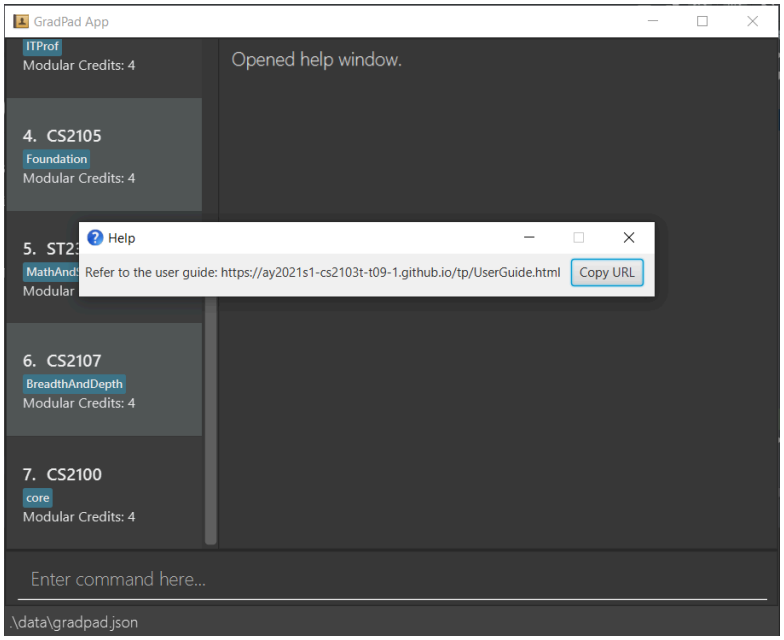
Startup look



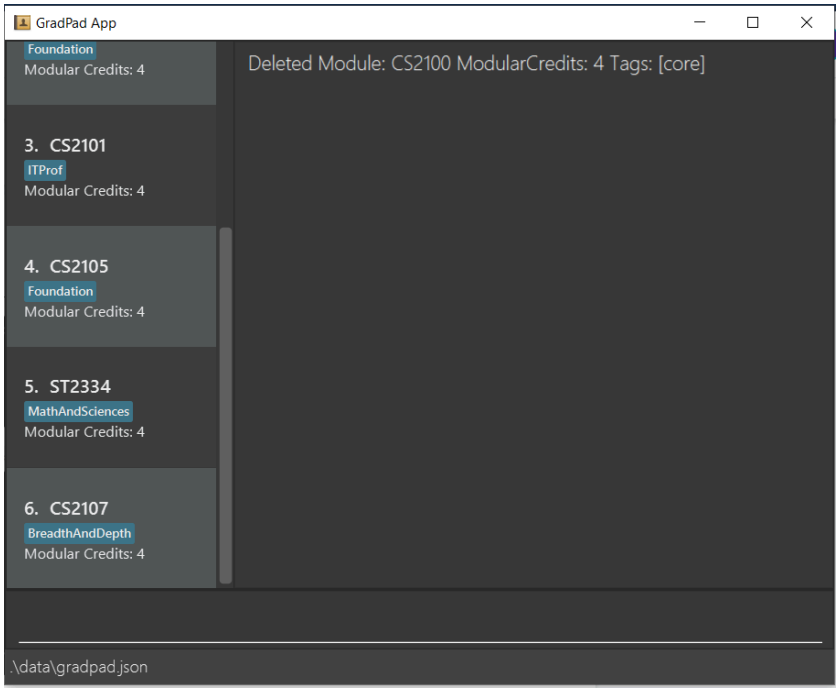
Add a module



Help Command



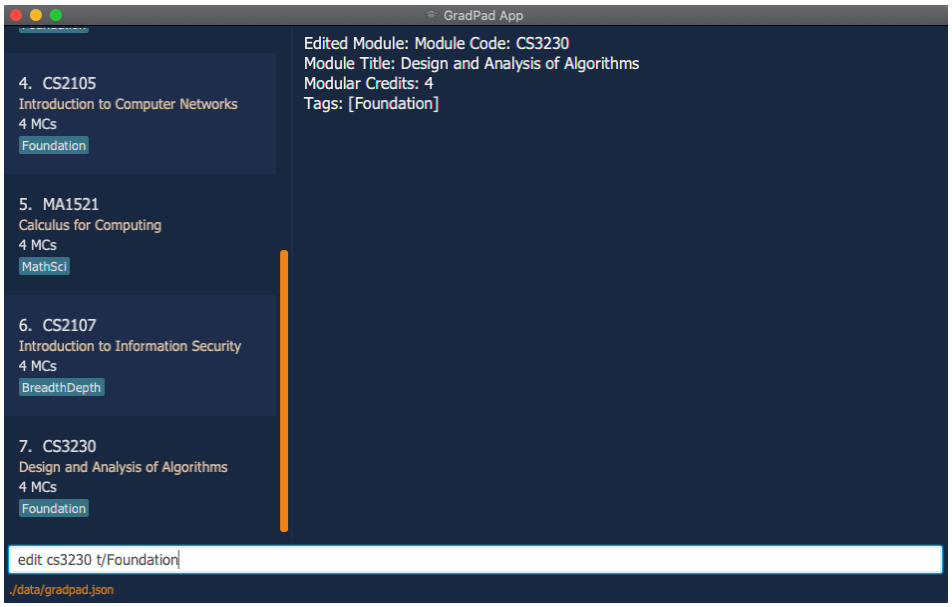
Delete module



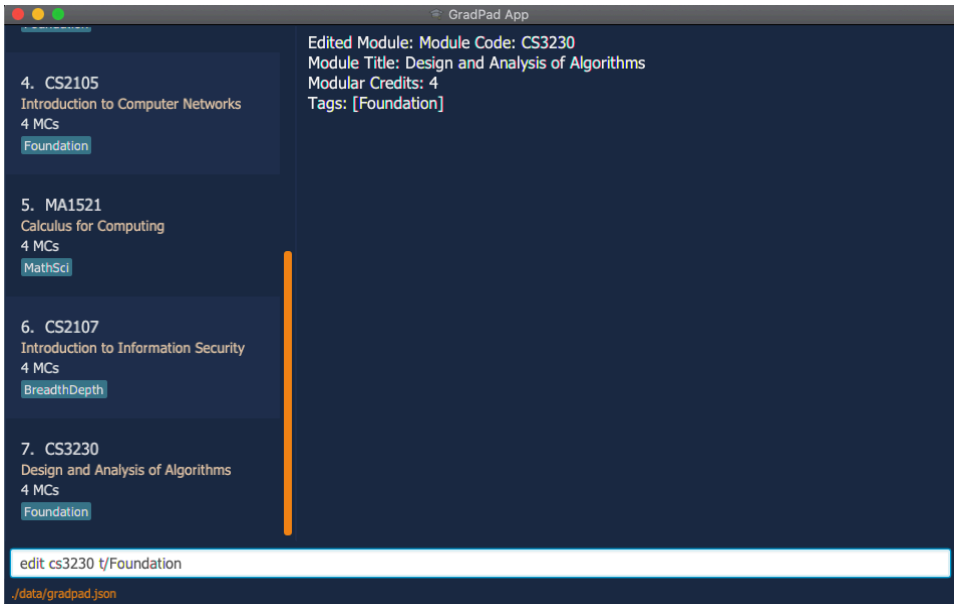
V1.3 Demo

Startup view

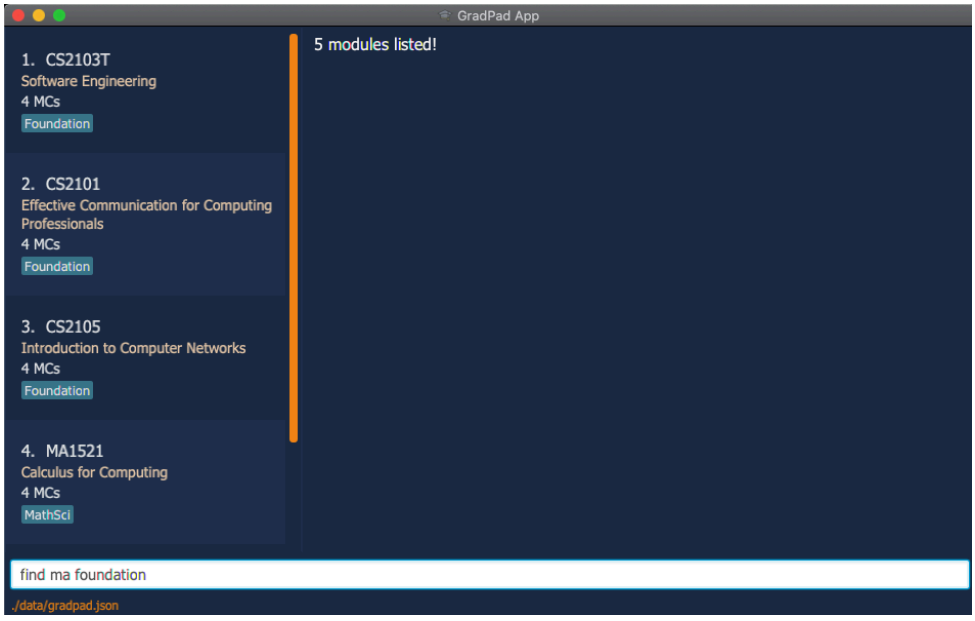
Adding a module (title and MCs auto-retrieved from NUSMods)



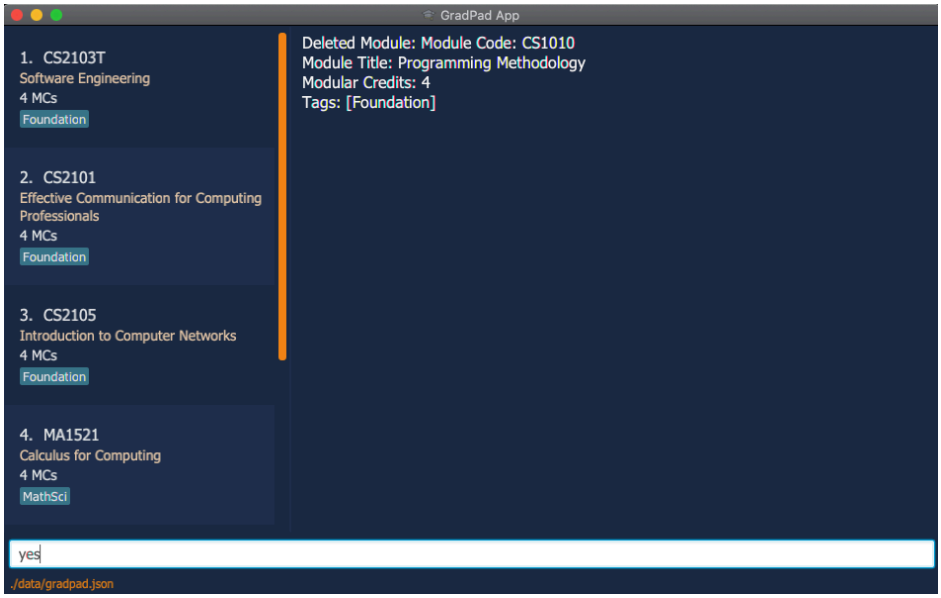
Editing a module



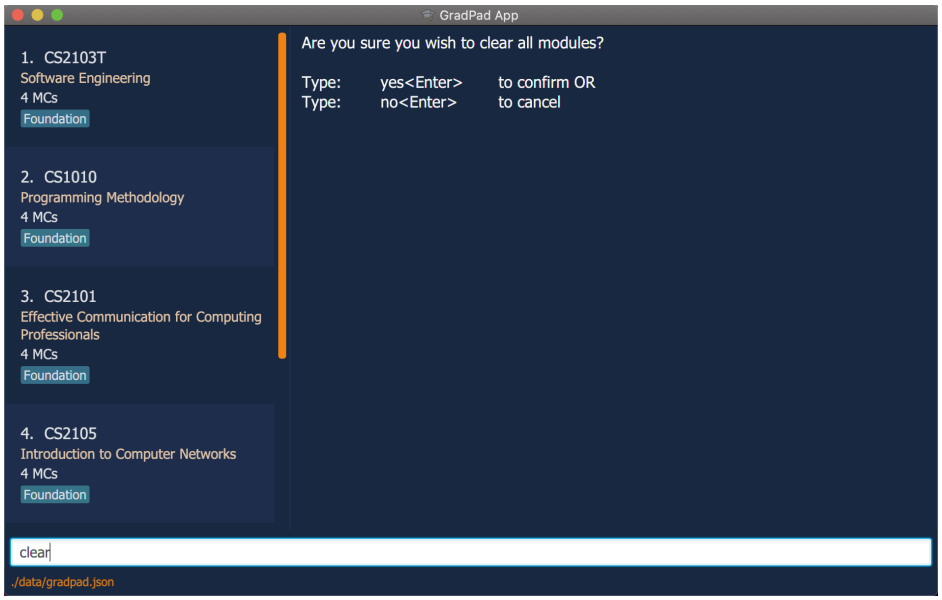
Finding a module in the Completed Modules list by module code or tag



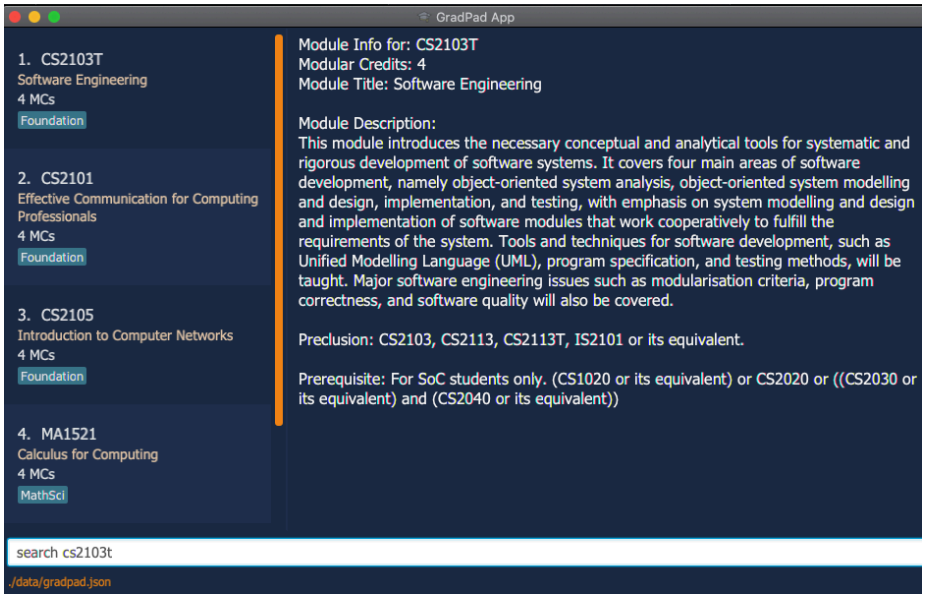
Deleting a module



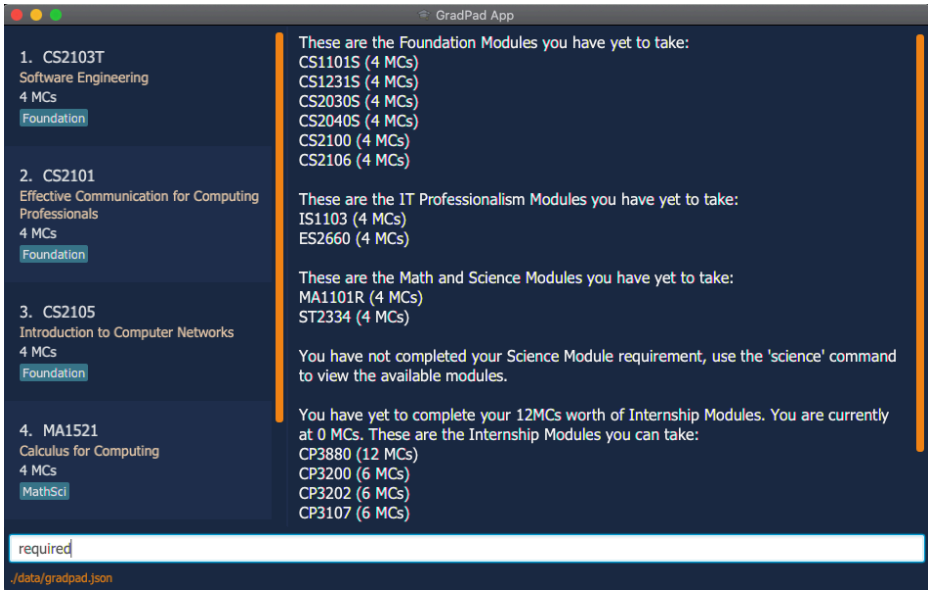
Clearing all modules



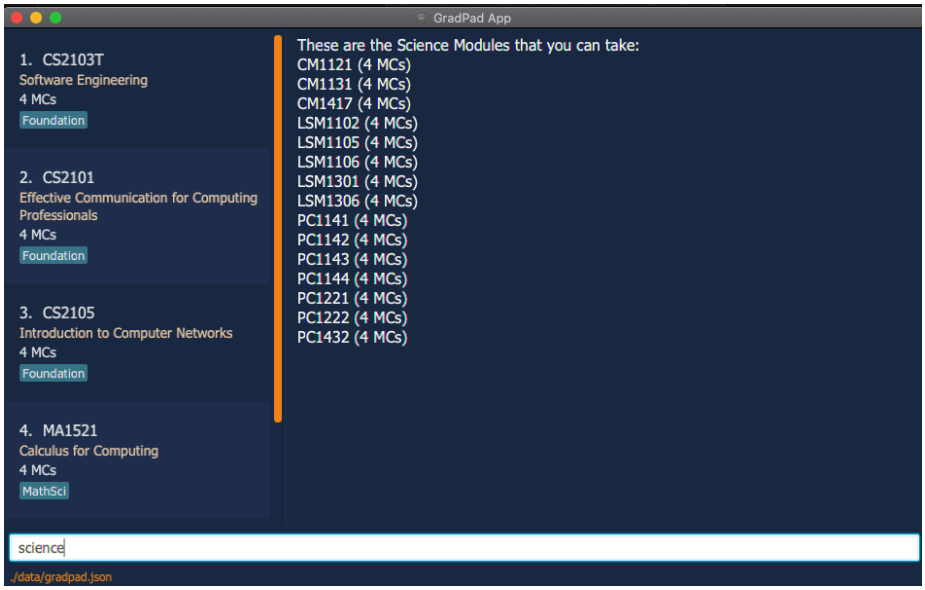
Searching for a module's information (from NUSMods)



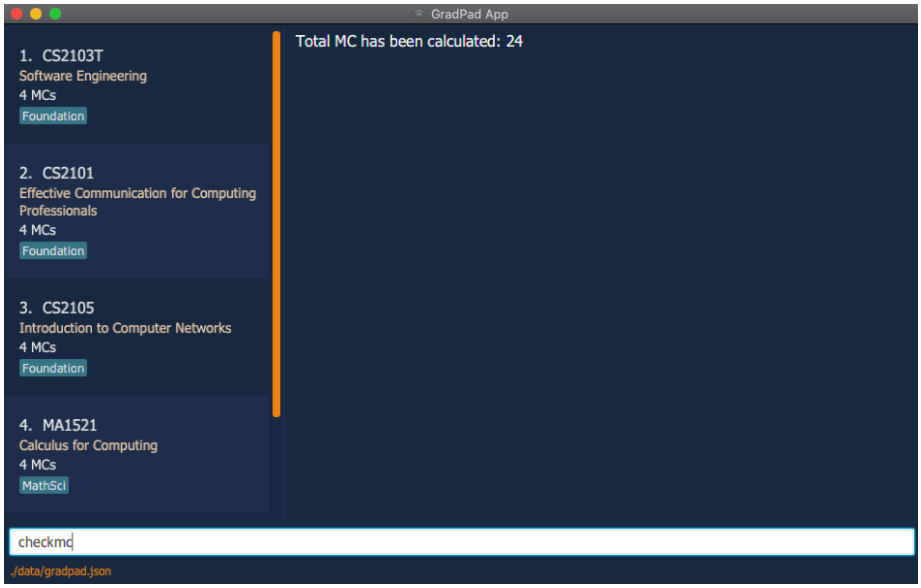
Viewing required CS modules (auto-updates as modules are added to the Completed Modules list)



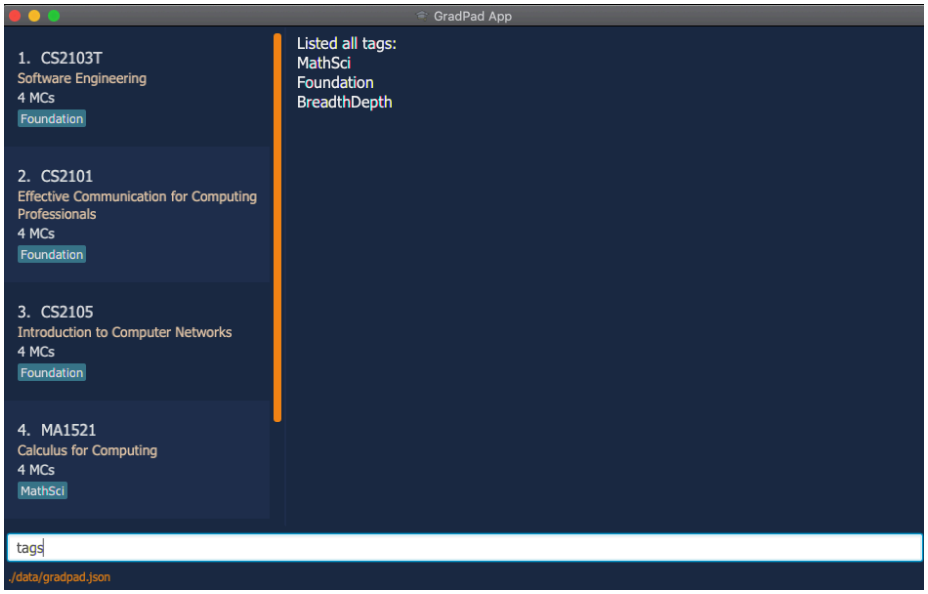
Viewing science modules for CS students



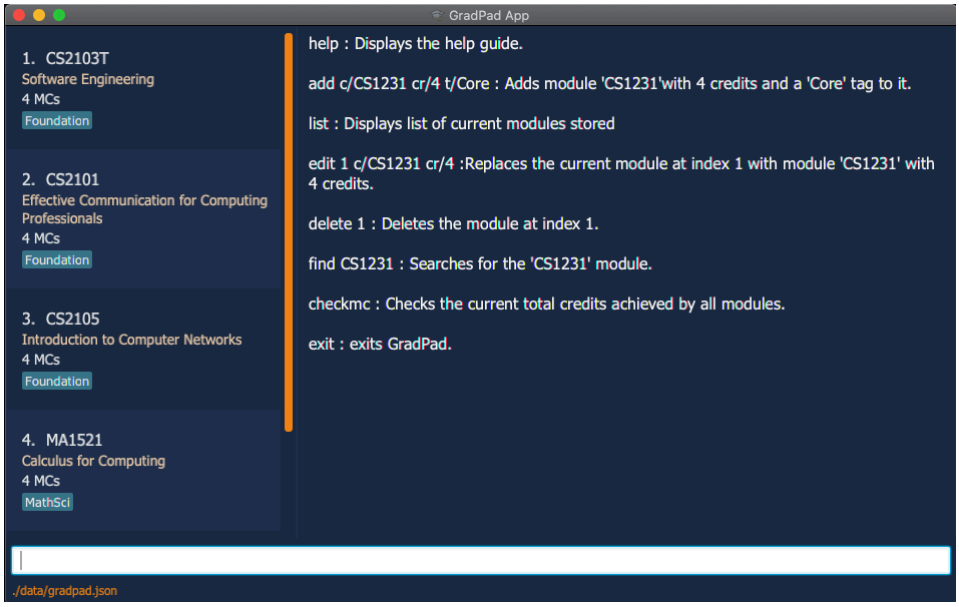
Checking MC tally



Viewing tags that have been added



Viewing help



Hi, welcome to the official demo video of GradPad. GradPad is a module management app which provides NUS Computer Science undergraduates with a more efficient way to plan their modules and track their academic progress. In this video, we will be showing you all the features that GradPad has to offer.

An important point to note before we begin, is that GradPad integrates data from NUSMods to streamline students' module searching process with added convenience. With this, students no longer have to visit NUSMods separately to look up module information. Instead, GradPad offers you all that data in the same app you use to plan your modules.

Now, let's begin.

SYAFIQ

Help feature: `help`

First and foremost, we have the help feature. Just type in `help`, and GradPad will display all the available commands users can use in GradPad, including explanations on how to use them. We have also included examples and notes in the explanation so as to avoid any confusion or doubts.

Add feature: `add cs2106 t/y2s1`

Now, let's officially get started. Once a student has completed a module, he can add it into GradPad. Say he has completed CS2106 in the first semester of your second year in NUS and he would like to add the module into the app. All he has to do is simply type in add cs2106 followed by t slash y2s1 and GradPad does the rest. Note how the app displays the module's title and amount of modular credits even though he did not specify them. This is possible due to the integration of NUSMods in GradPad, which allows the app to automatically retrieve module details based on the module code the user has entered.

Edit feature: `edit cs2101 c/cs2100`, `edit cs2100 t/y2s2`

Next, let's say he has made a mistake while adding certain modules. If he has accidentally added a module that he has not yet completed, instead of deleting the module and adding a new one, he can simply edit it to replace it with another module. For example, he has added cs2101 by mistake when he meant to add cs2100. To do this, he can simply type edit cs2101 followed by c/cs2100. Again, note that the module title and modular credits of the new module is retrieved automatically, thanks to our integration of NUSMods. However, if he wishes to change only the tags of a specific module, he can type in something similar like edit cs2100 t/y2s2 but this time, without a new module code specified.

Tags feature: `tags`

This next feature allows a student to view all the tags he has included in his modules. Simply type in `tags` and all the tags he has added will be displayed.

Find feature: `find y2`, `find cs2101`

Next up, the `find` feature allows a student to filter the list of completed modules to view a group of modules with common keywords. For example, if he wishes to view all the modules he took in his second year, he can type in the find y2. He can also filter the list to view a specific module, say cs2100. For example, he can type `find cs2100` to view that specific module.

SHAOKIAT

List feature: `list`

Now that the user has found his modules, let us revert back to see the entire list of completed modules. Simply type in the `list` command and GradPad will display all the user's completed modules again.

Delete feature: `delete cs2103t`

Next, if the user wishes to delete an existing module `cs2103t` in GradPad, just type in the `delete` command with the module code `CS2103T` and it will bring the user to the confirmation page. This page is to prevent the user from accidental deletion. Just enter yes to confirm the deletion of the module or any other keys to abort. There, CS2103T has been successfully deleted from the list of completed modules.

Force Delete feature: `fdelete cs1010

For more experienced users, there is a command for quicker deletion. Simply use the command fdelete cs1010 to delete the module `cs1010` and bypass the confirmation page.

Required feature: `required`

Now, to view the required modules that the user has still yet to complete, simply type in the `required` command. GradPad will then display all the modules the user has yet to take to complete his CS curriculum. This command even tells the user if he has completed any of his Computer Science requirements as shown.

Gem feature: `gem`

If the user has not completed his General elective modules, simply enter the `gem` command and it will display the whole list of GEMs available for the user to take.

Science feature: `science`

This feature applies for science modules as well! Simply type in `science` command to display the list of available science modules.

yan:

Search feature: LSM1301

After looking at the modules available, the user might want to find out more about a certain module. Simply type search, followed by the module to search for and gradpad will display all the relevant module information, by interacting with the NUSMODS API. Take note that if the user is not connected to the internet, he/she will only be able to search for modules in the CS curriculum. This applies to the edit and add command as well.

Clear feature

If the user wishes to start a clean slate GradPad, simply use the `clear` command to remove all current modules in GradPad, as such. When the confirmation prompt pops up, type y, ye or yes to confirm or any other key to abort.

Add CS1010, CS123

Force Clear feature

For experienced users who wish to bypass this confirmation step, simply use the `fclear` command and all your modules will be deleted, without any confirmation needed!

Exit feature

If the user is done using gradpad, simply type in `exit` to exit our app, as such. A good bye message will be displayed, and the app will shut down in 1.5 secs.

Those are all the features students need to use GradPad. With that, we have come to the end of the demo video. We hope that with GradPad, students no longer dread managing their modules. Students should no longer waste their time planning modules the old fashioned way, go GradPad today.