# Meeting Minutes for CS2103-T16-3

This document is a comprehensive overview of meeting minutes for CS2103-T16-3. Refer to the document outline for a table of contents. Team Members are:

- Bec Kyung Huhn (Nationality: 🇸🇬, MT: 한국어)
- Chloe Lee (Nationality: 🇲🇾, MT: 中文)
- Loh Jing Yen (Nationality: 🇸🇬, MT: 中文）
- New Jun Jie (Nationality: 🇸🇬, MT: 中文)
- Mayank Keoliya (Nationality: 🇮🇳, MT: English)

## I. Minutes for 2020-08-29

### Agenda

- Start collaborative doc
- Decide project direction
    - User profile
    - Problem Addressed + Value Proposition

### Product Name

ResiReg

### Target User Profile

- Target users
    - OHS* admin at Residential Colleges (RCs)* in NUS

### Value Proposition

- Manage students, room allocations and billing faster than a typical GUI app.

### Product Optimization

#### Potential Features

- CRUD features for students
    - Matric Number
    - Name
    - Faculty
    - Major
    - Modules this semester
    - Residence

    - isRA (if true, then 0 fees)

- CRUD for room type
    - Room with air-con
    - Corridor with air-con
    - Type: 'Quarantine Suite' (0 vacancy, only for SHN students)

- CRUD features for room allocation
    - Assigning a student to a room

- CRUD for staff
    - Authentication with password

- Register electrical appliance/portable aircon
    - Student can install appliance in room, with appropriate billing
    - Amount to student bill

- Billing
    - Generate student bill based on room type
    - Set due date
    - Pay Bill
    - Late fee (automatically $50, 10 days after due date)
    - Award 10 demerit points if fees not paid in 30 days

**Summary**

The team eventually settled on a Residence management solution named ResiReg, in spirit of the recent housing challenges. The meeting concluded at 1155.

## II. Minutes for 2020-09-05

**User Stories**

Refer here.

**Rough Plan for Project**

1. v1.2 (MVP) : Housing Management (CRUD Student + Room)
2. v.1.3 : Finance Management (Bill Student)
3. v.1.4  UX ("Help", "Man", Import CSV, Export Full Project, Auto-Backup)

## III. Minutes for 2020-09-12

In this meeting, we decided on the features for v1.2 of our product, and drafted a User Guide for it. We updated the table in 2020-09-05's minutes, to display the commands (syntax) and their behavior, as well as assigned each team member their features to work on for 1.2.

Note: Most of our changes have been absorbed into the previous meeting's notes, as well as the GitHub Kanban.

## IV. Minutes for 2020-09-19

**Agenda**

1. Deciding work distribution
    a. Issue Tracker
    b. Activity
    c. Tasks

**Tasks assigned (tentative deadline: Sunday afternoon)**

1. AboutUs: All of Us
    a. Add role + images + email
    b. Make PRs from your fork to the ream repo for each
2. Refine user stories in UG: All of us
    a. Make PRs from your fork to the team repo for each.
3. Update UG + README: Jet and Kevin
4. Update DG: Jing Yen and Mayank

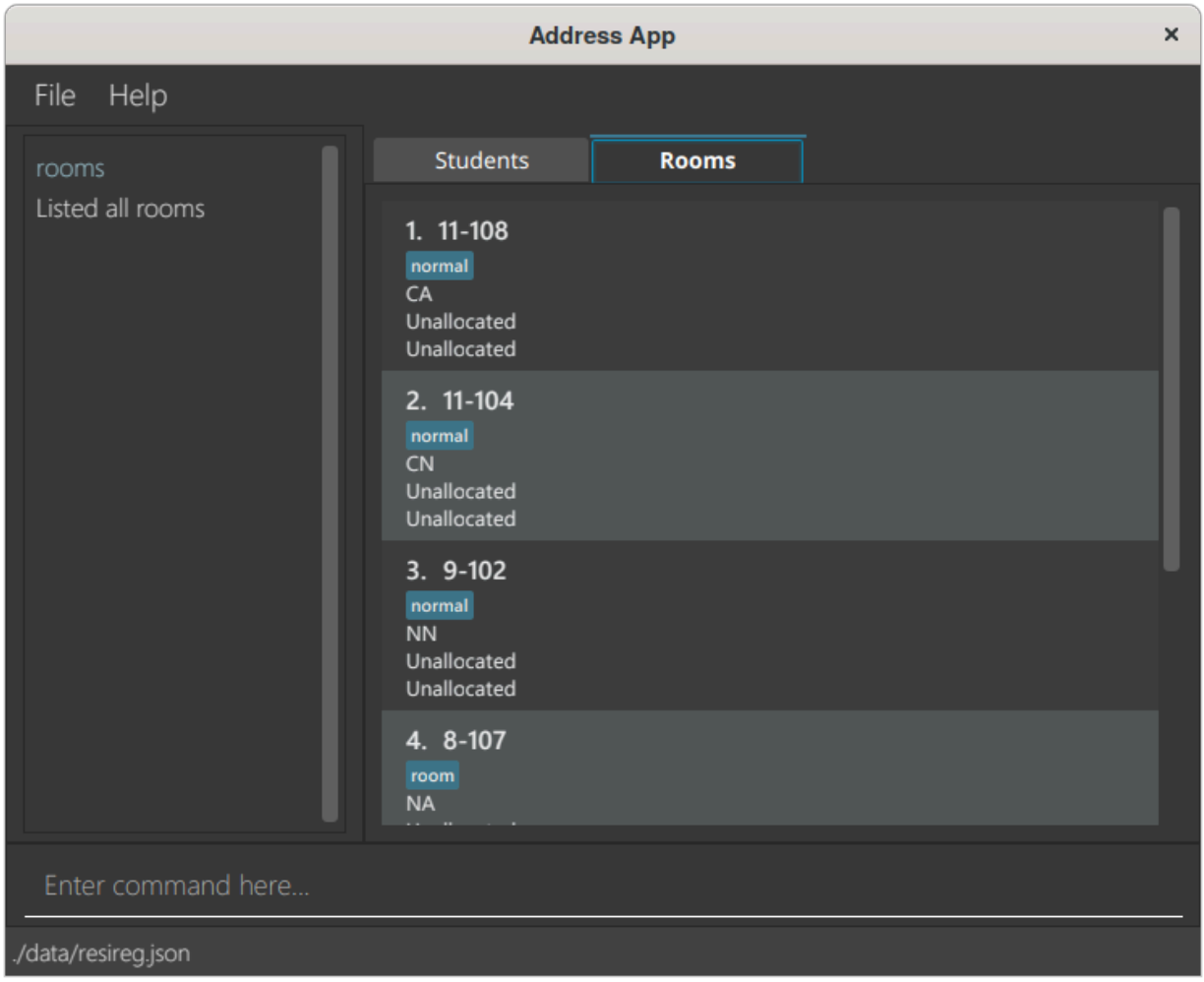
**Roles Assigned**

1. Team Lead: Jing Yen
2. Documentation: Jet New
3. Testing: Chloe
4. Code Quality: Mayank
5. Deliverables and deadlines: Kevin Bec Kyung Huhn
6. Integration: Jing Yen
7. Scheduling and tracking: Mayank
8. Git expert: Chloe
9. IntelliJ expert: Kevin Bec and Jet

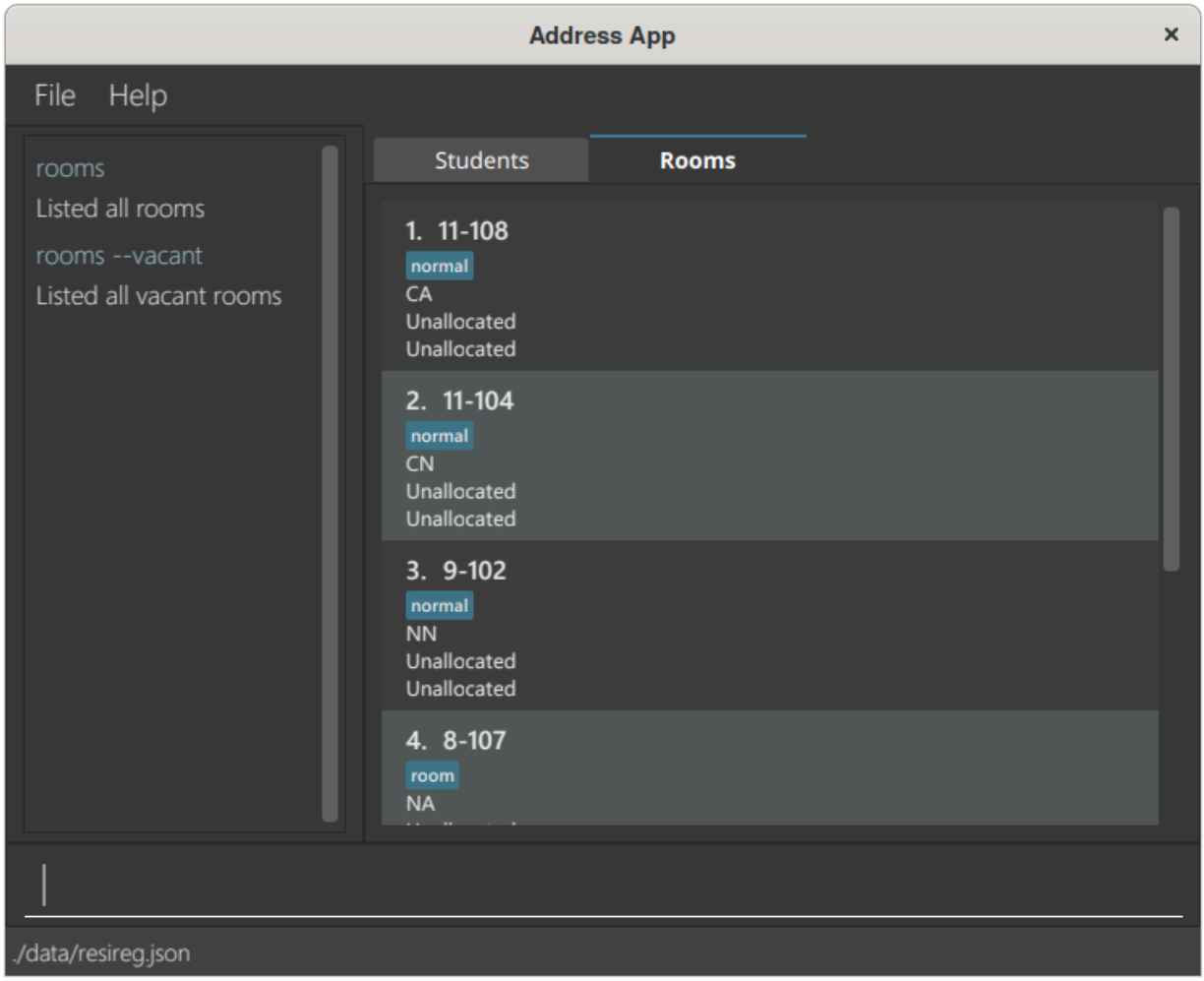
In this meeting, the team agreed to drop the COVID-related stories, since it would be adding quantity (not quality) to the user experience. Instead, we chose to add new features that provide a faster CLI experience to the target user, and thus improve productivity. For example, we added user stories for tasks such as undo/redo, autocomplete, viewing command history, and more.

## V. v1.2 Milestone

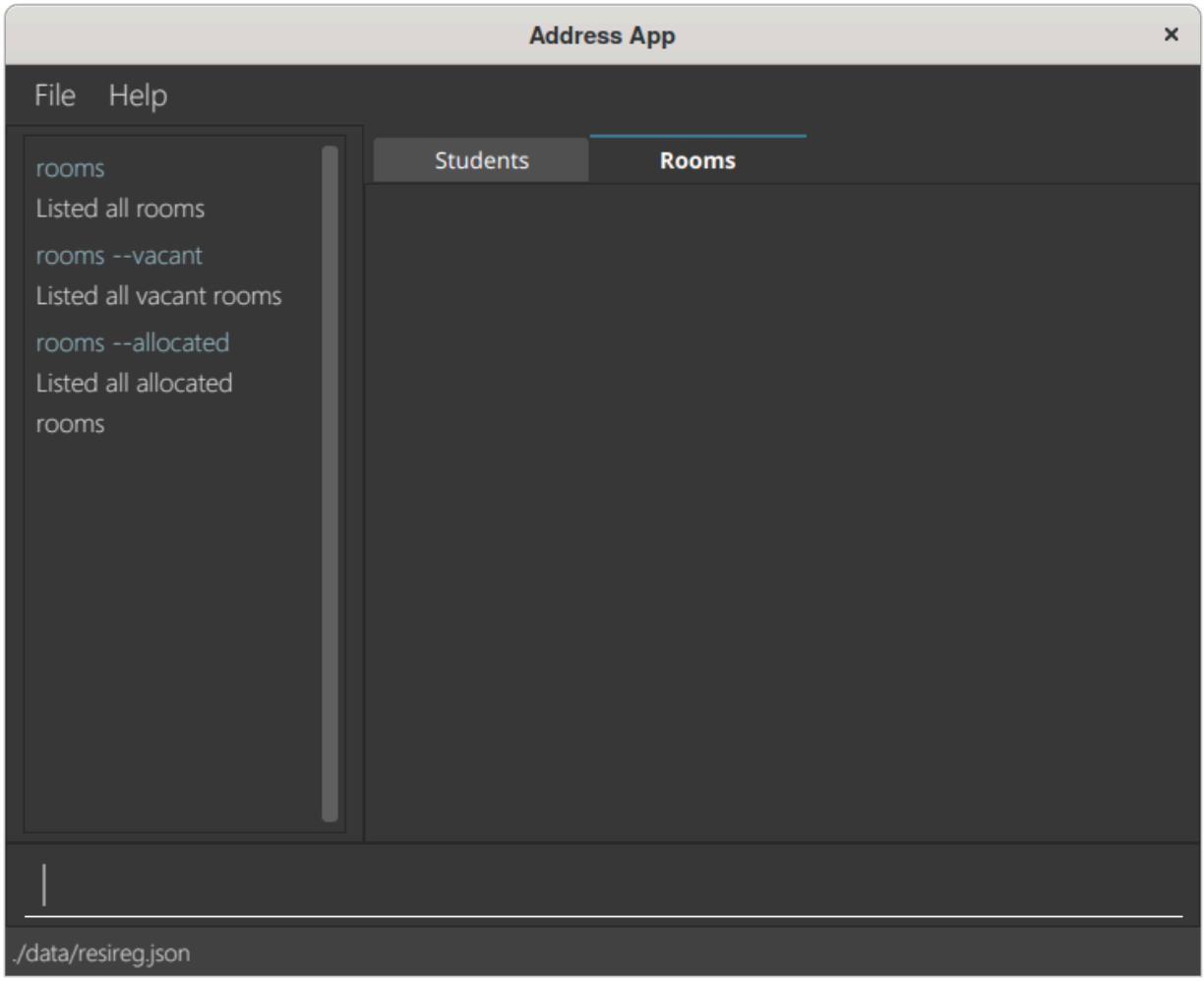
### Listing all Rooms



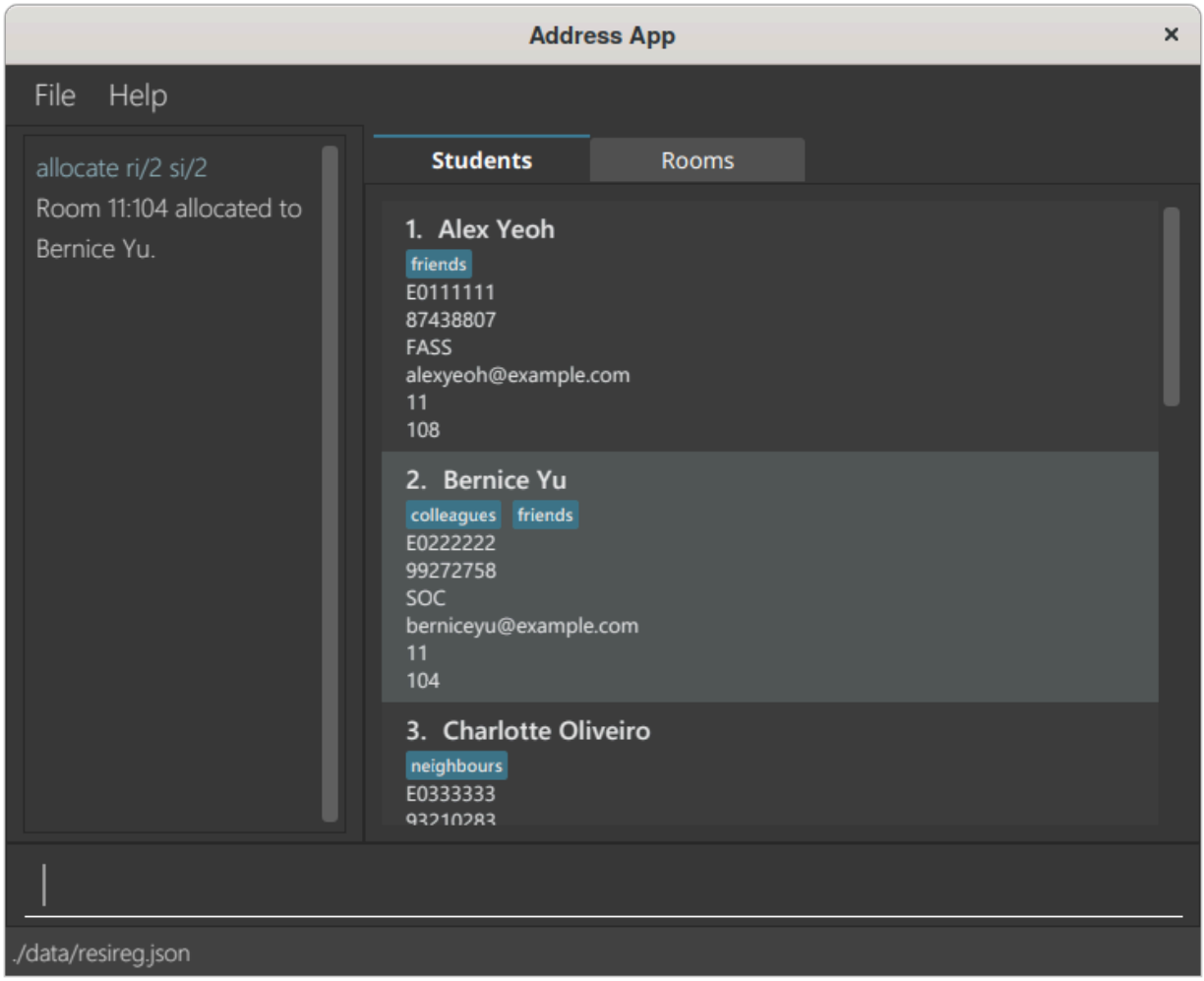
Listing only vacant rooms



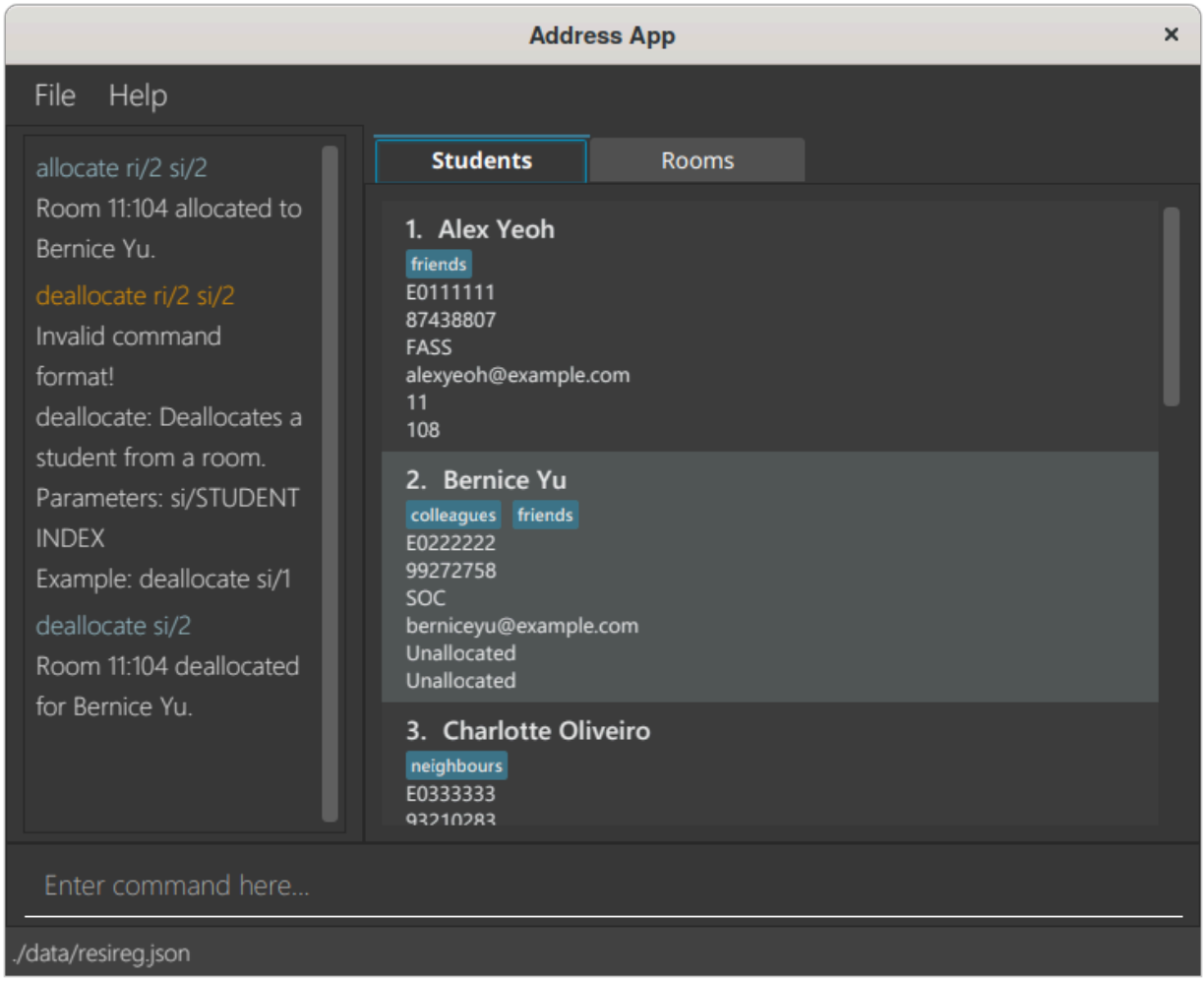
Listing only allocated rooms



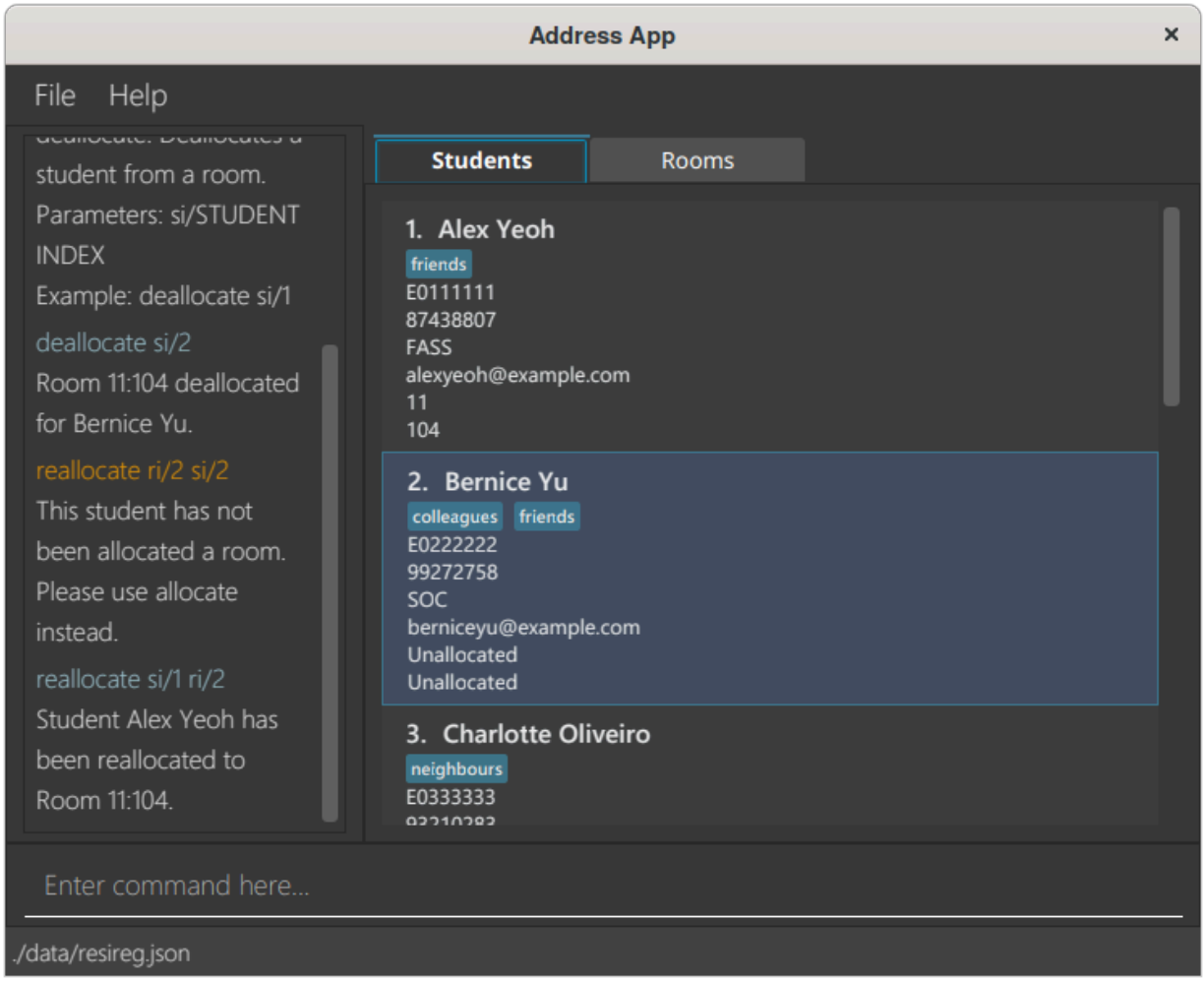
Allocating a room to a student



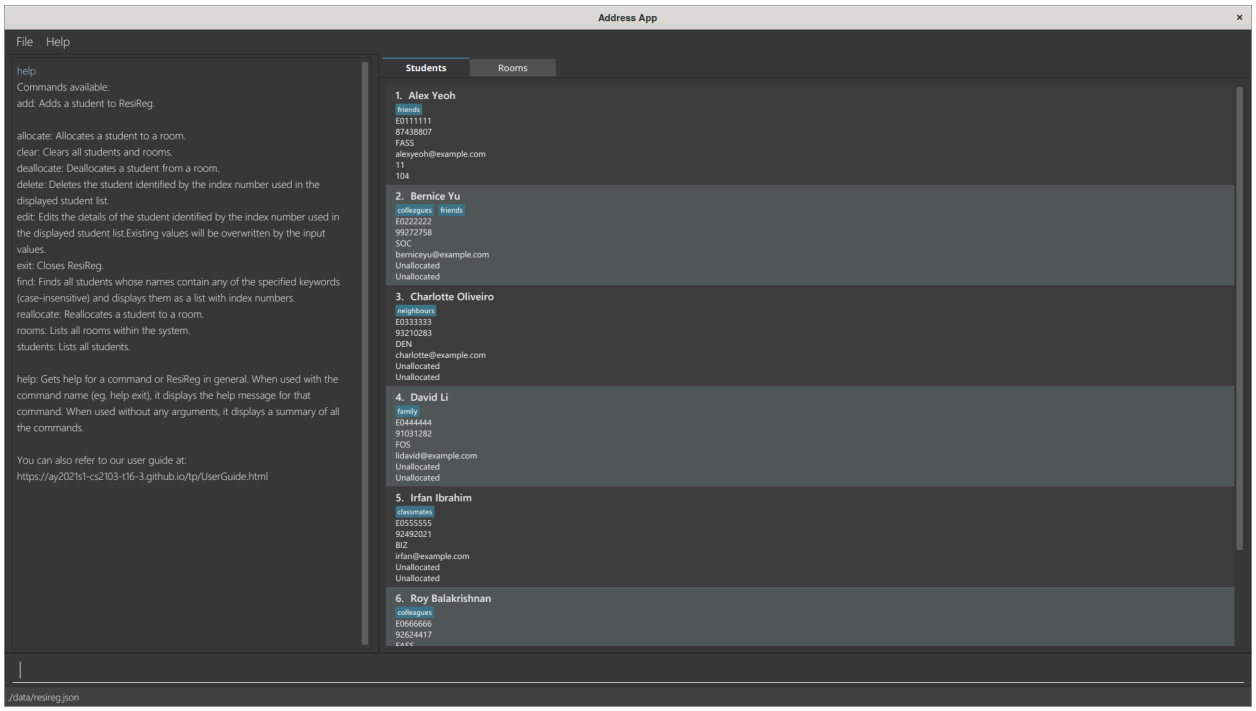
Deallocating a room for a student



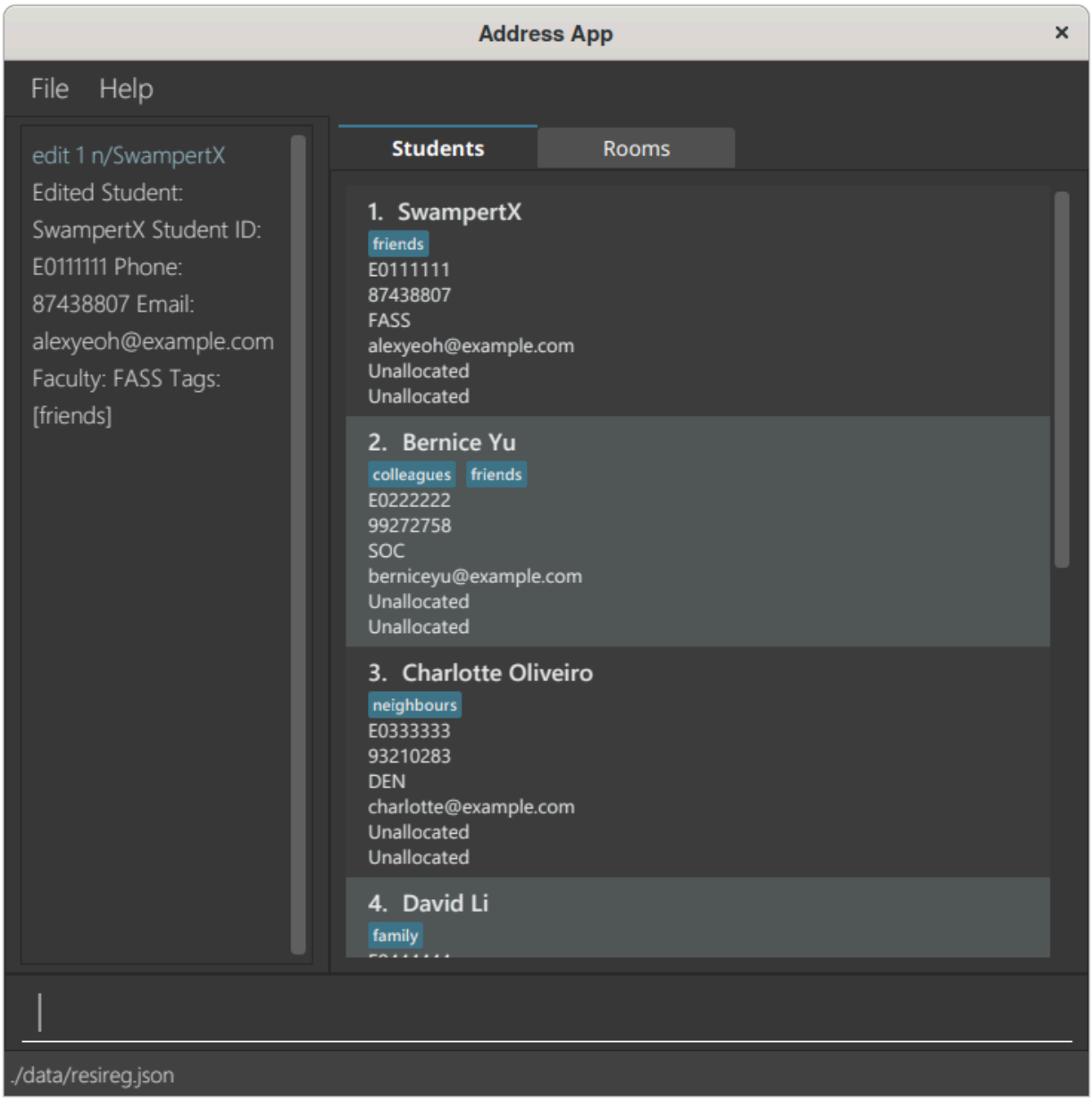
Reallocating a room for a student



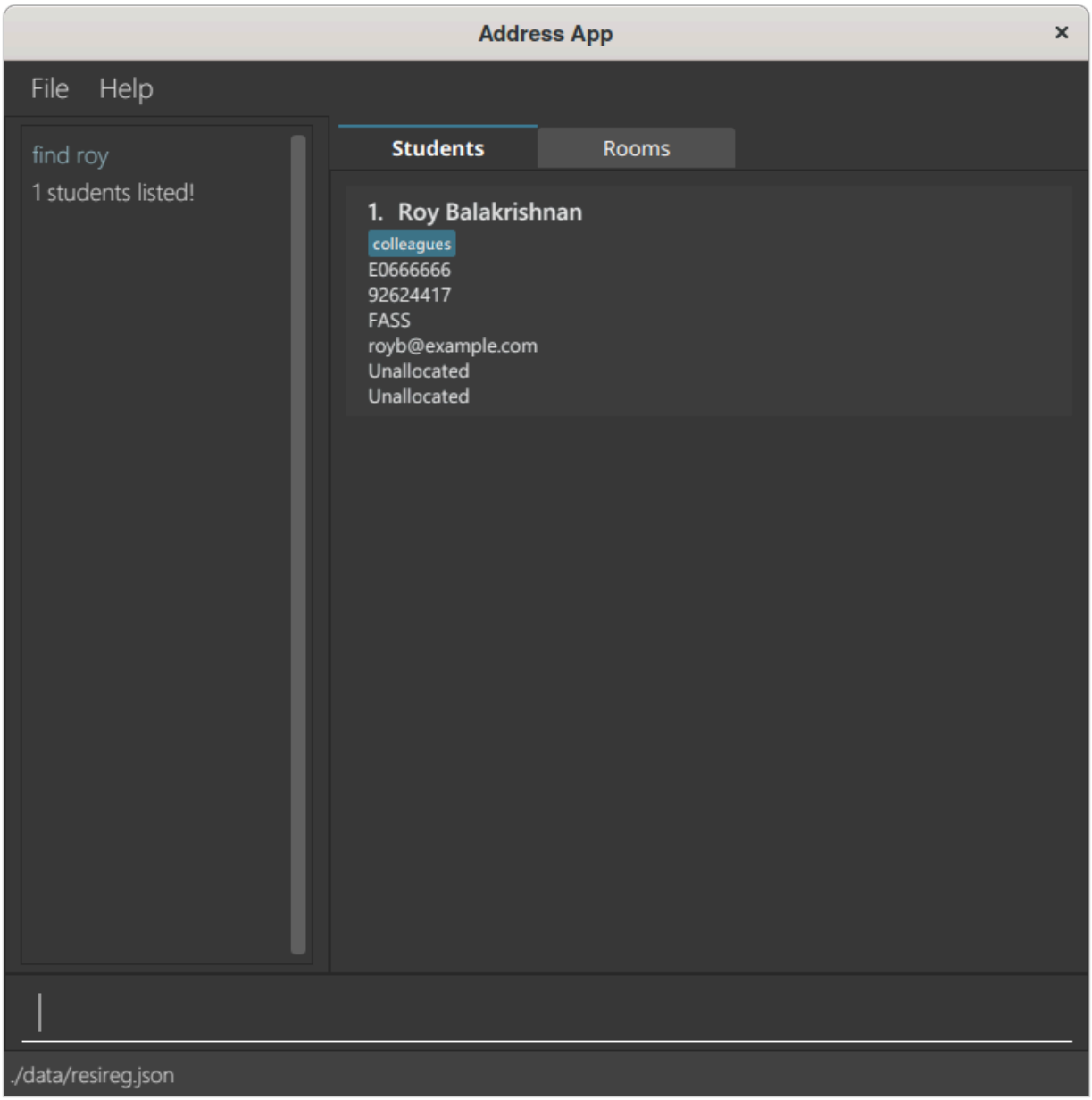
# Help


Address App
File Help
help
Commands available:
add: Adds a student to ResiReg.
allocate: Allocates a student to a room.
clear: Clears all students and rooms.
deallocate: Deallocates a student from a room.
delete: Deletes the student identified by the index number used in the displayed student list.
edit: Edits the details of the student identified by the index number used in the displayed student list.Existing values will be overwritten by the input values.
exit: Closes ResiReg.
find: Finds all students whose names contain any of the specified keywords (case-insensitive) and displays them as a list with index numbers.
reallocate: Reallocates a student to a room.
rooms: Lists all rooms within the system.
students: Lists all students.
help: Gets help for a command or ResiReg in general. When used with the command name (eg. help exit), it displays the help message for that command. When used without any arguments, it displays a summary of all the commands.
You can also refer to our user guide at:
https://ay2021s1-cs2103-t16-3.github.io/tp/UserGuide.html
Students
Rooms
1. Alex Yeoh
friends
E0111111
87438807
FASS
alexyeoh@example.com
11
104
2. Bernice Yu
colleagues friends
E0222222
99272758
SOC
berniceyu@example.com
Unallocated
Unallocated
3. Charlotte Oliveiro
neighbours
E0333333
91210283
DEN
charlotte@example.com
Unallocated
Unallocated
4. David Li
family
E0444444
91031282
FOS
lidavid@example.com
Unallocated
Unallocated
5. Irfan Ibrahim
classmates
E0555555
92492021
BIZ
irfan@example.com
Unallocated
Unallocated
6. Roy Balakrishnan
colleagues
E0666666
92624417
./data/resireg.json

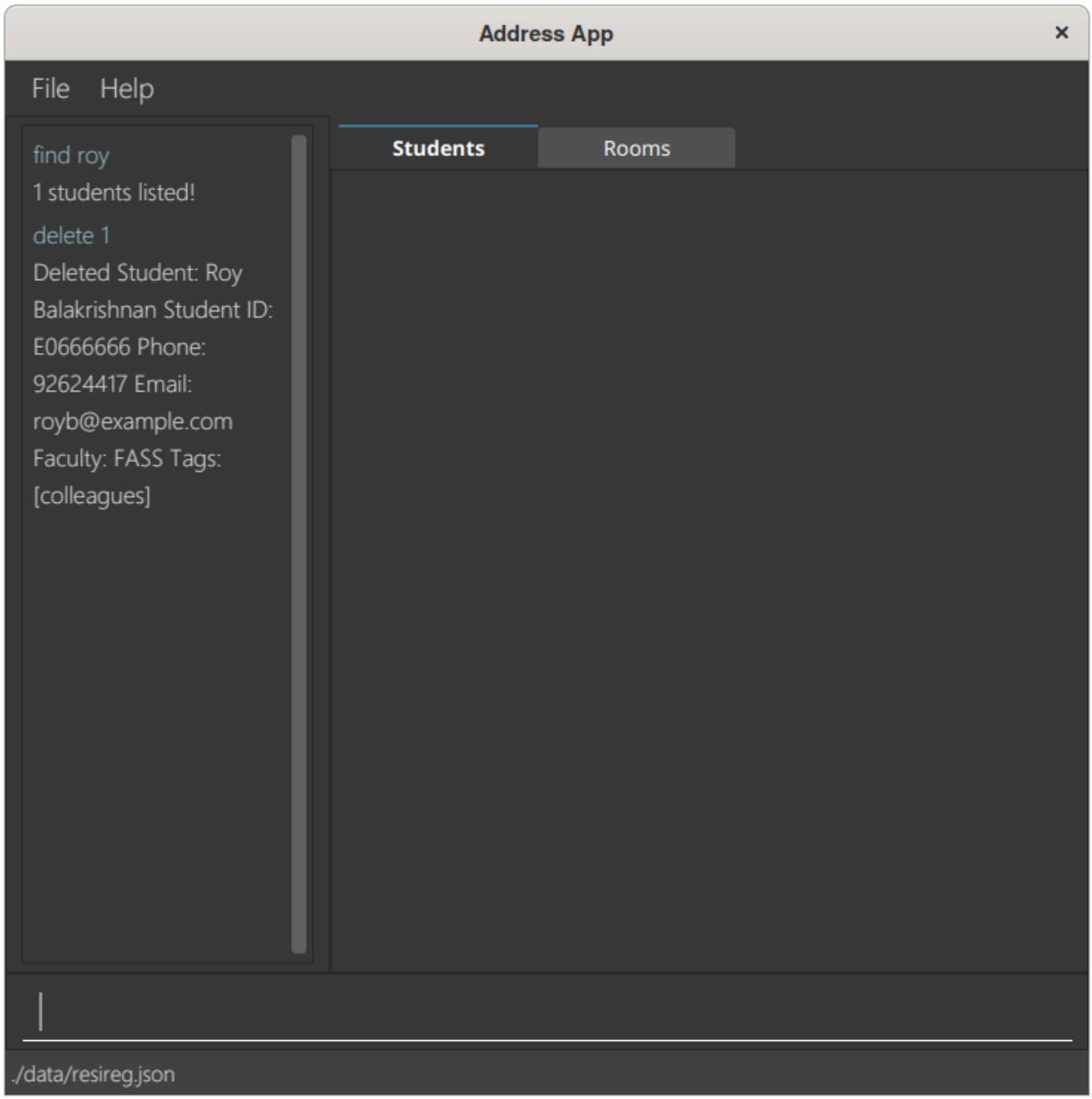
Edit Student Details



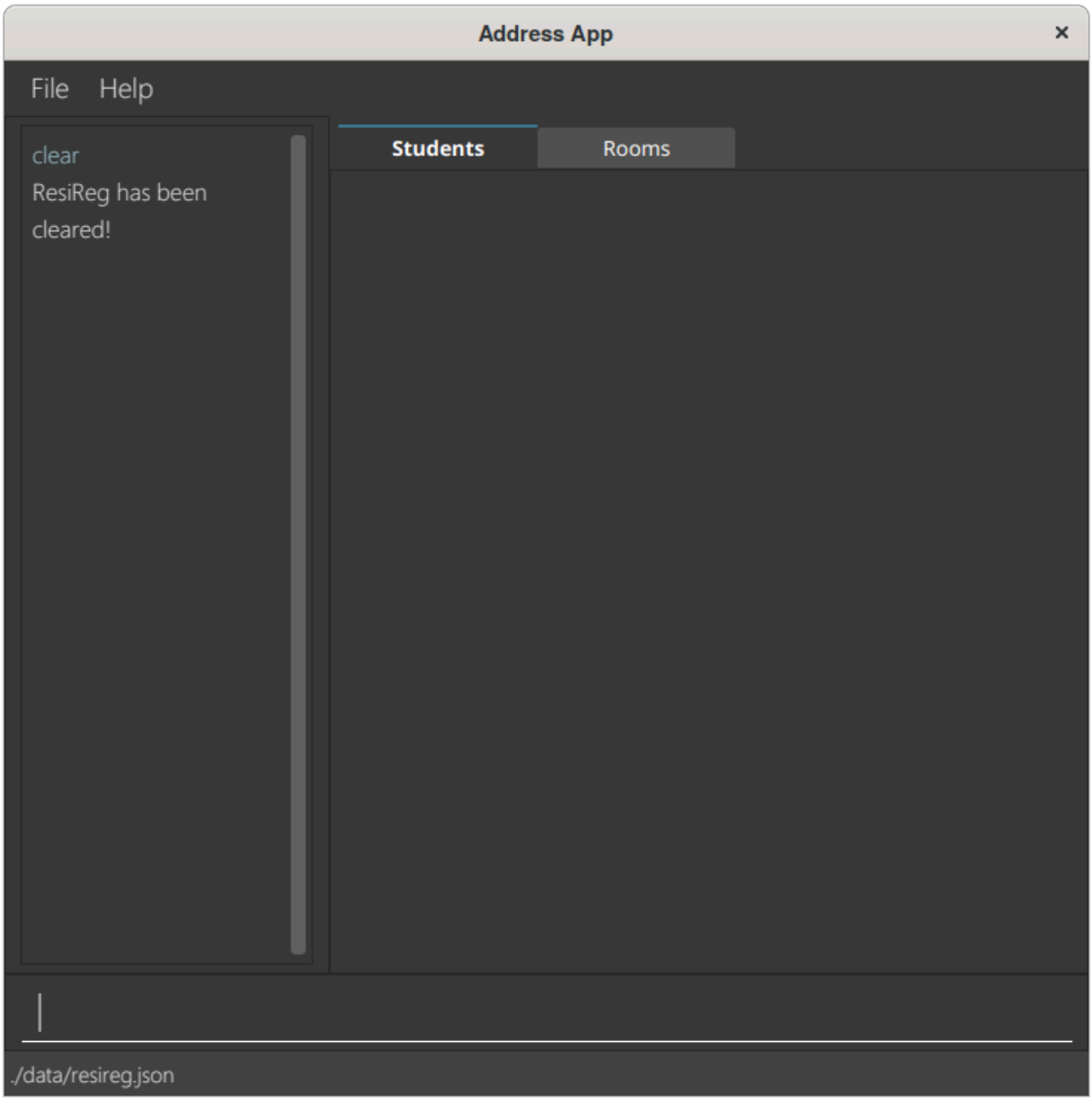
Find Student



Delete a Student



## Clear all Data




## Exit

It exits the app.

## Problems Identified

- **Lack of "Big Features" With a Lot of Code**
  We're optimizing for grade - which means each of us should end with atleast:
  - **Ownership of 2 big features** (e.g. refactoring Allocation, implementing undo/redo command, support for archiving semesters).
  - Atleast 5 commands added/modified.
  - **5k lines of code** contributed. This means that we need to add multiple features that require a lot of lines of code written.
  - 7 use cases in documentation.
  - 1 major write-up about design considerations.

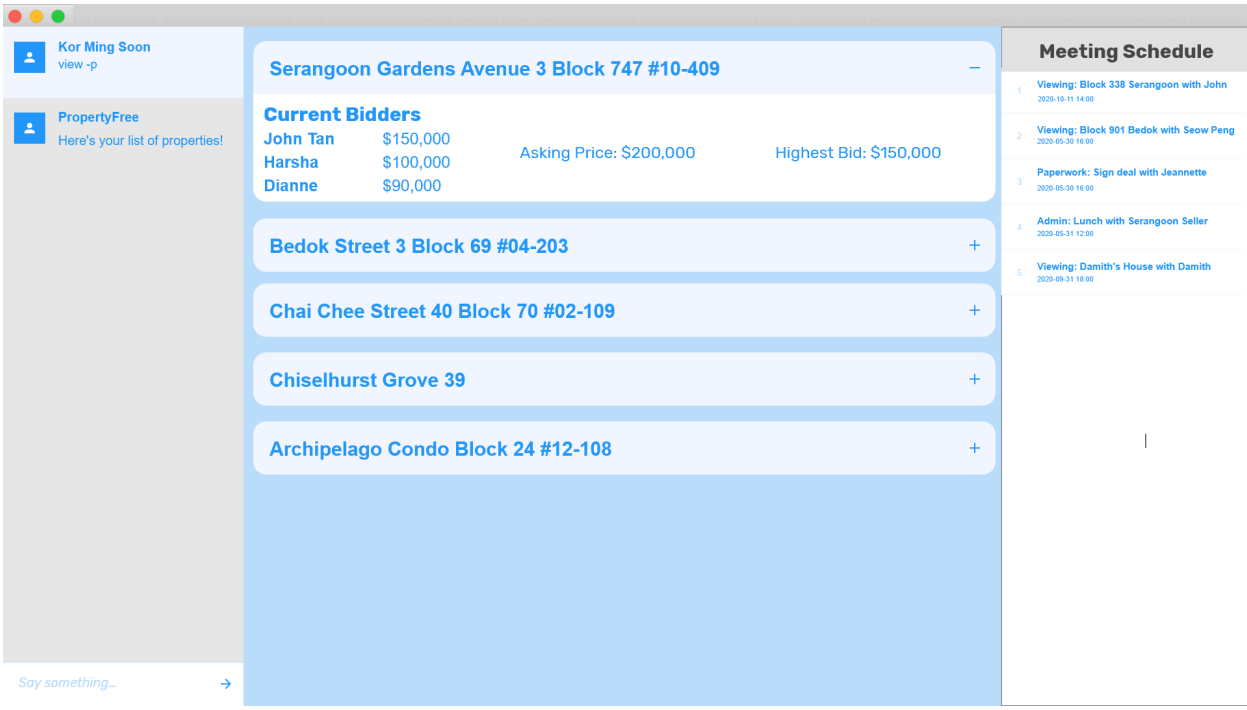
- **Lack of effort in UI**
  Currently, UI looks very similar to AB3. A peer reviewer from another team would NOT consider our effort spent in working on the UI to be atleast the same as AB3.

- **Testing, Logging and Assertions**
  Counts for 10 marks (individual). Our code coverage as a team has dropped to 55%, and logging/assertions are non-existent. It is now incumbent on every team member to add tests, logging and assertions along with their code **for every PR made after Thursday's meeting**.

## Task List due by Saturday

- **UI**:
  Assigned to: Jing Yen or Chloe
  Due by: Sunday night.
  - Currently, our UI is strikingly similar to AB3 (same theme, same font).
  - The Command History on the left pane doesn't serve much purpose.

- Ideal:




  - Change to a UI that looks like this (but with 2 tabs: rooms and students).
  - Remove “History” Pane, show feedback only from the last command.
  - The central pane should show the active tab.
    - Each item on the central pane should be collapsible.
    - Preferably, the central pane should appear like a table with multiple expandable rows.
  - The pane on the left (here, “Kor Ming Soon”, etc) should show cards in the inactive tab.

- Enabling filtering/finding of rooms: Chloe (tonight)
  - Requires only the addition of 2-3 commands.
    - Benefit: Increases LoC and feature list
  - Enhancement: improve naming and syntax of commands (make them more consistent).

- Archiving Semesters: Jing Yen (PRed, tonight)
  - Requires addition of multiple commands (load, archive), and a change in the file storage system.

    - Analogous to HireLah's sessions feature.
- Aliasing: Mayank (PR - merge by afternoon - Chloe)
    - Need to update UserPrefs and Command Map. WIll be documenting decisions made in the DG.
    - CLI-friendly
- Command History: Kevin (Sunday night)
    - Shows previous command on pressing "up" arrow key
    - Benefit: LoC and CLI-friendliness
- Undo/Redo: Kevin
    - Update DG with design decisions
- Allocate, Deallocate and Reallocate Commands: Jet
    - Add tests, assertions and logging (Sat night)
        - Unit testing to be done by Jet (including JsonAdaptedRoom)
        - Integration testing to be done by Kevin with Undo/Redo
    - Update DG with design decisions (Sun night)

- Add Testing and Documentation (UG + DG) for **whatever code you've written so far.** Optionally, update your contributions on the profile page.

**Big Features for next week (to be assigned on Saturday):**

1. Add Bills Model and relevant commands (dependency: UI) : Jing Yen
    - A Student contains a list of bills
        - Exactly like larry's tP
    - A bill has 2 states: paid or unpaid
    - A bill has due date
    - If bill due date has passed, then UI shows an orange colour for the student card
    - Commands: add bill for student, delete bill for student, mark bill as paid.

2. Statistics + 1 small feature : Jet
    - As an admin I want to visually view statistics of the vacancy of rooms so that I can communicate easier to admins of other buildings.
    - Show a chart of vacant, and non-vacant rooms in the building disaggregated by room type.
        - LARRY'S REPO
        - Total # of allocated rooms, Total # of unallocated rooms
    - Small Feature: Allow student ID and room label to be entered as an alternative to si/ and ri/, e.g. sid/E0123456, r/08-110.

        - Enhancement: if a user clicks on a student card, then it automatically copies the student's ID to the clipboard, that way they can speed up the typing of commands

3. Importing and Exporting to CSV + Binning: Mayank
    - Will require creation of an Exportable interface, that can convert a UniqueStudentList or UniqueRoomList into a CSV file.
        - See a tP that already does this -> copy it.
        - Check if external library is needed (ask Damith or check the forum)
    - Import CSV (for student and rooms - 2 commands)
        - Overrides existing ResiReg data

4. Multiple Small Features (dependency on archiving) : Chloe
    - Allow a one-time adding of rooms when ResiReg is used for the first time
        - no addresbook.json (HireLah implementation)
    - Edit Room by Type (1 command)
    - Converting static enums to dynamic storage objects (suggestion)
        - Will require some thinking.
        - Faculties: Enum with 16 values.
        - Room-Type: Enum with 4 values.

5. Demerit points : Kevin (dependency on archiving)
    - Demerit points = 0 on construction
    - Add demerit points
    - Delete demerit points
    - Set demerit point limits and when a student hits the limit show some kind of message to remind the admin to email
    - Prevent a student from being allocated a room if a student's demerit points is above ceiling

V1.4 (optional)

- Same enhancement of copying for Rooms
- Find by Matric Number
- Find by Faculty
- Find by Multiple Faculties
-

November 5: midnight: self-imposed code freeze

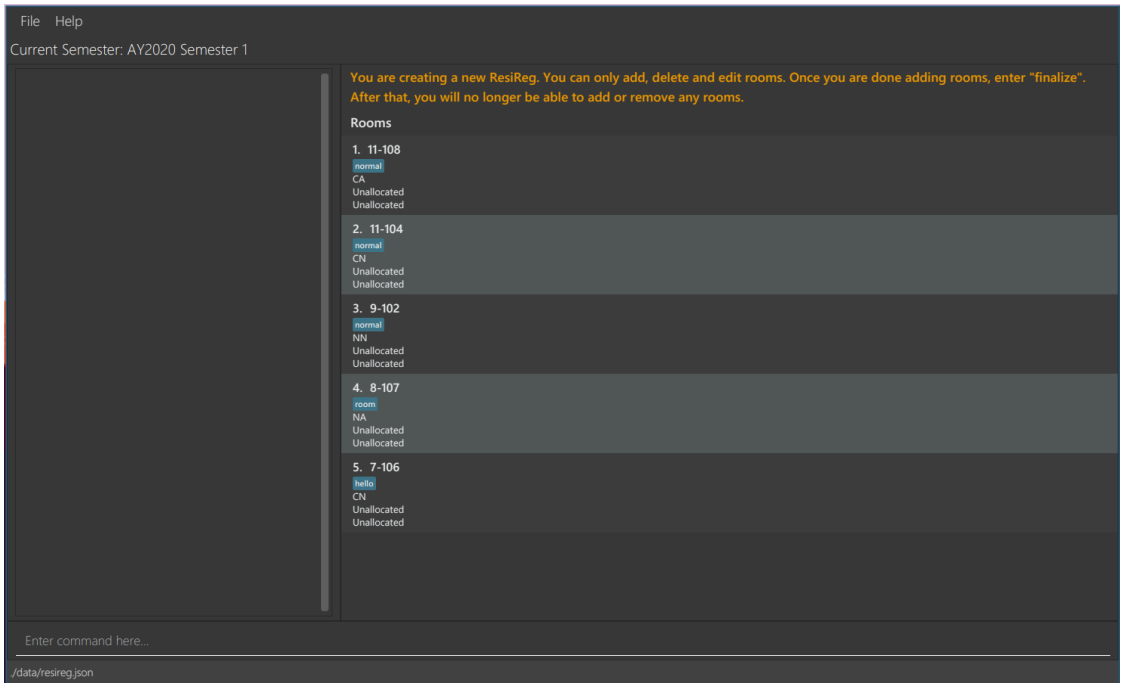
November 9, midnight : deadline for tP

**Final features added for v1.3:**

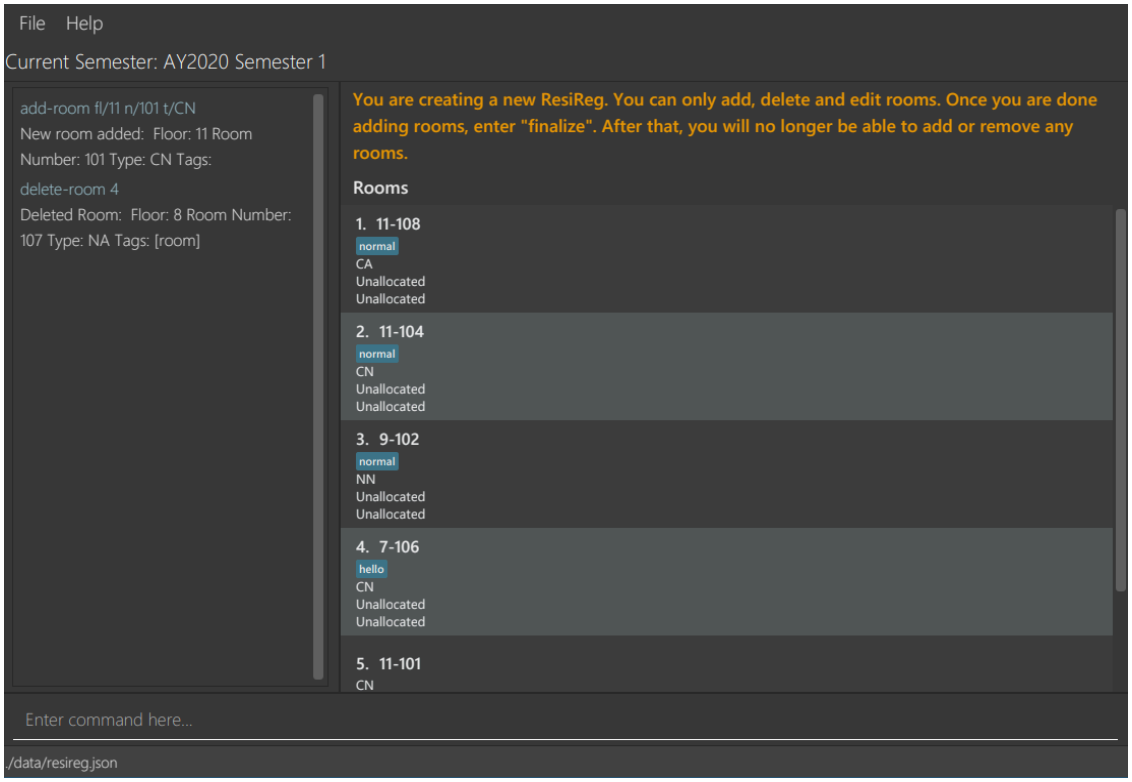
- Undo and redo (using both keyboard shortcuts and commands)
- Command aliasing
- Concept of semester, and ability to archive data at the end of a semester
- Expanded room filtering
- Ability to view students and rooms either in separate tabs or side by side
- Unix shell-style command history
- Trash bin for deleted items
- Statistics

**V1.3 features demo**

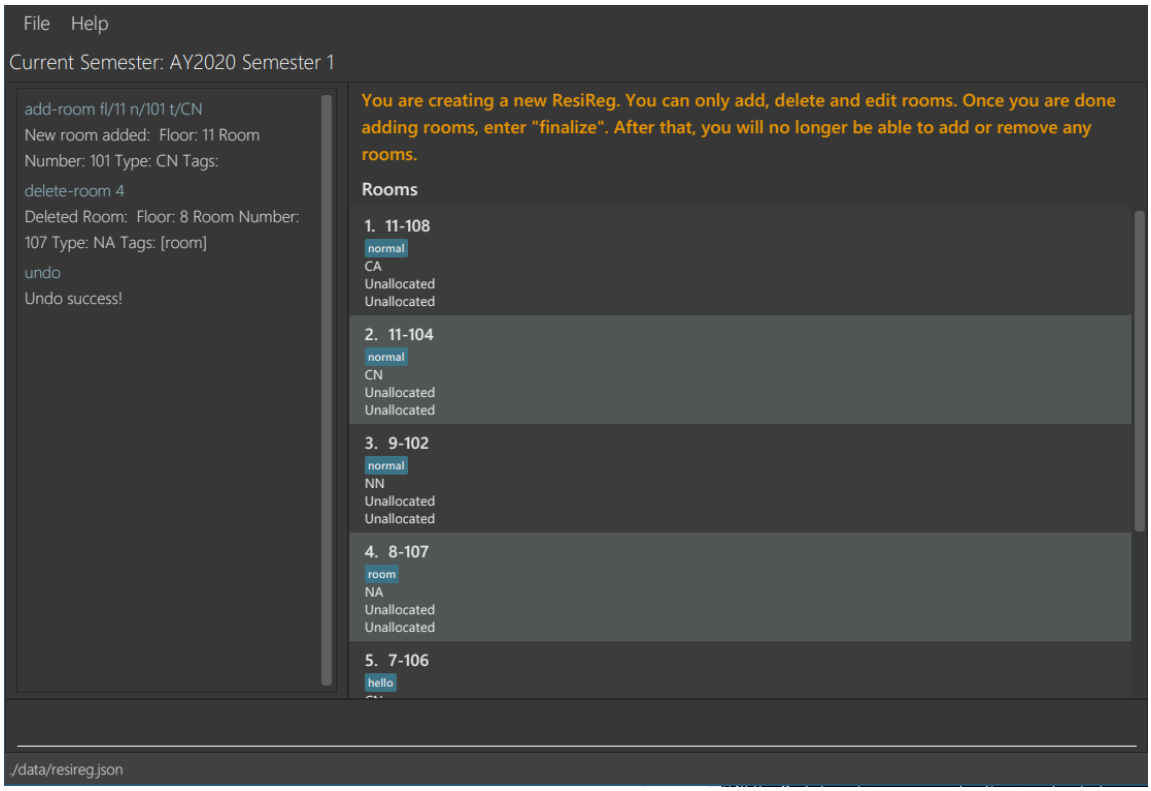
Upon starting up the app without a data file (resireg.json), the user can add and delete rooms:



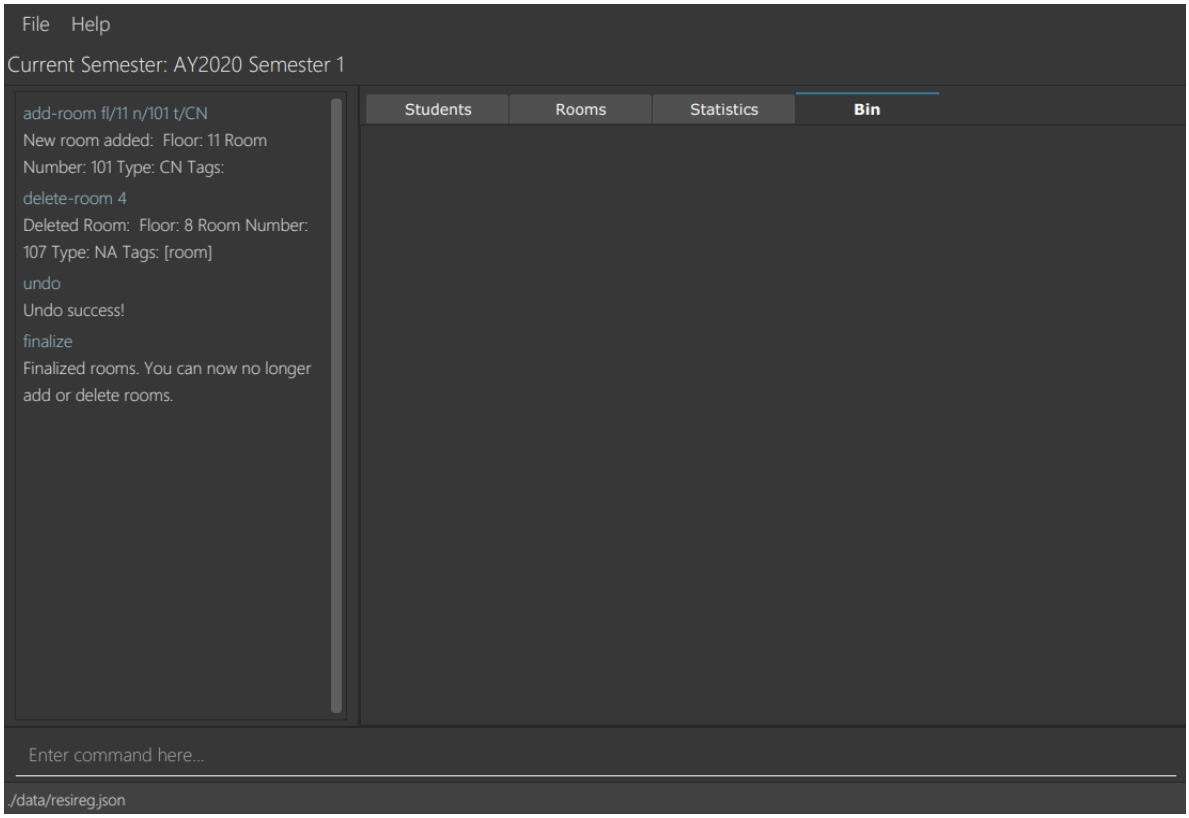

Rooms can be added and deleted (see command history on left for commands entered):



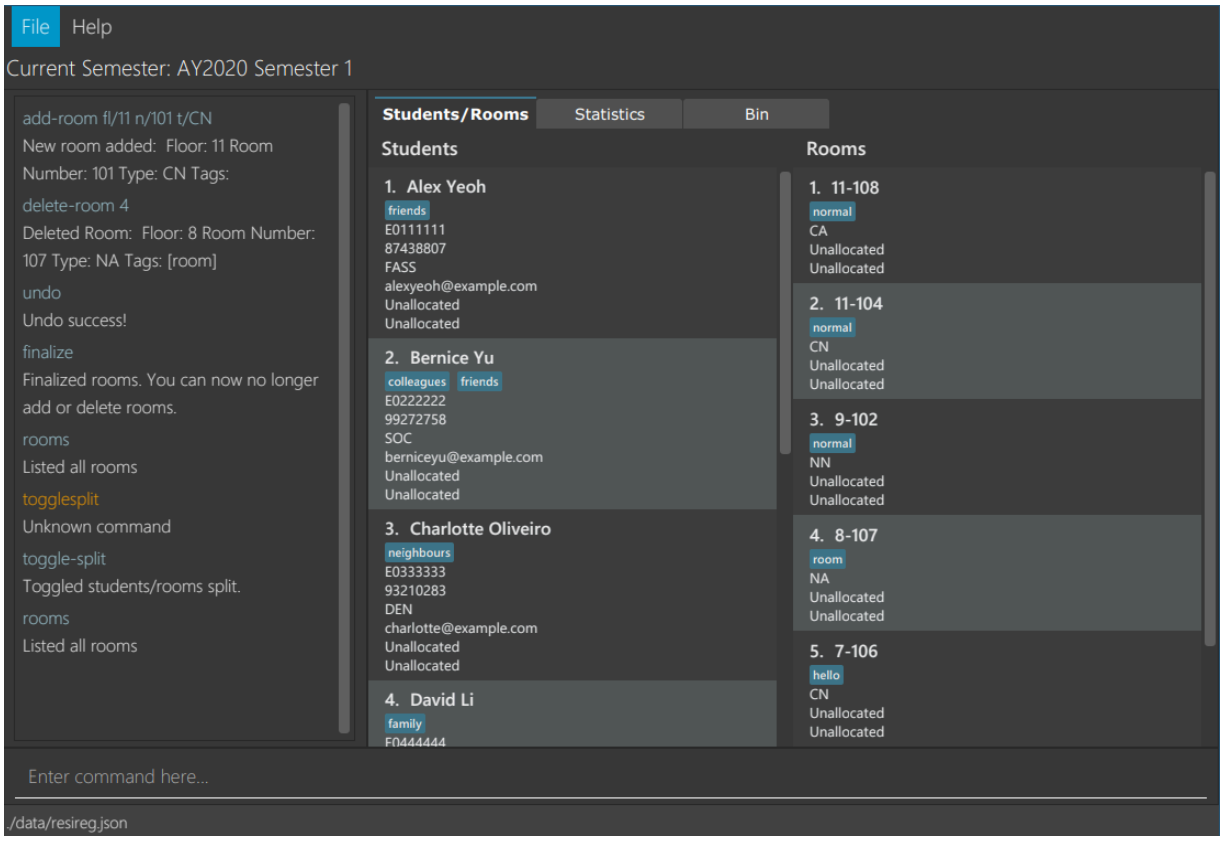

Any action can be undone/redone either using Ctrl-z/Ctrl-y or the undo/redo commands (room deletion undone below):



Finalize the list of rooms:

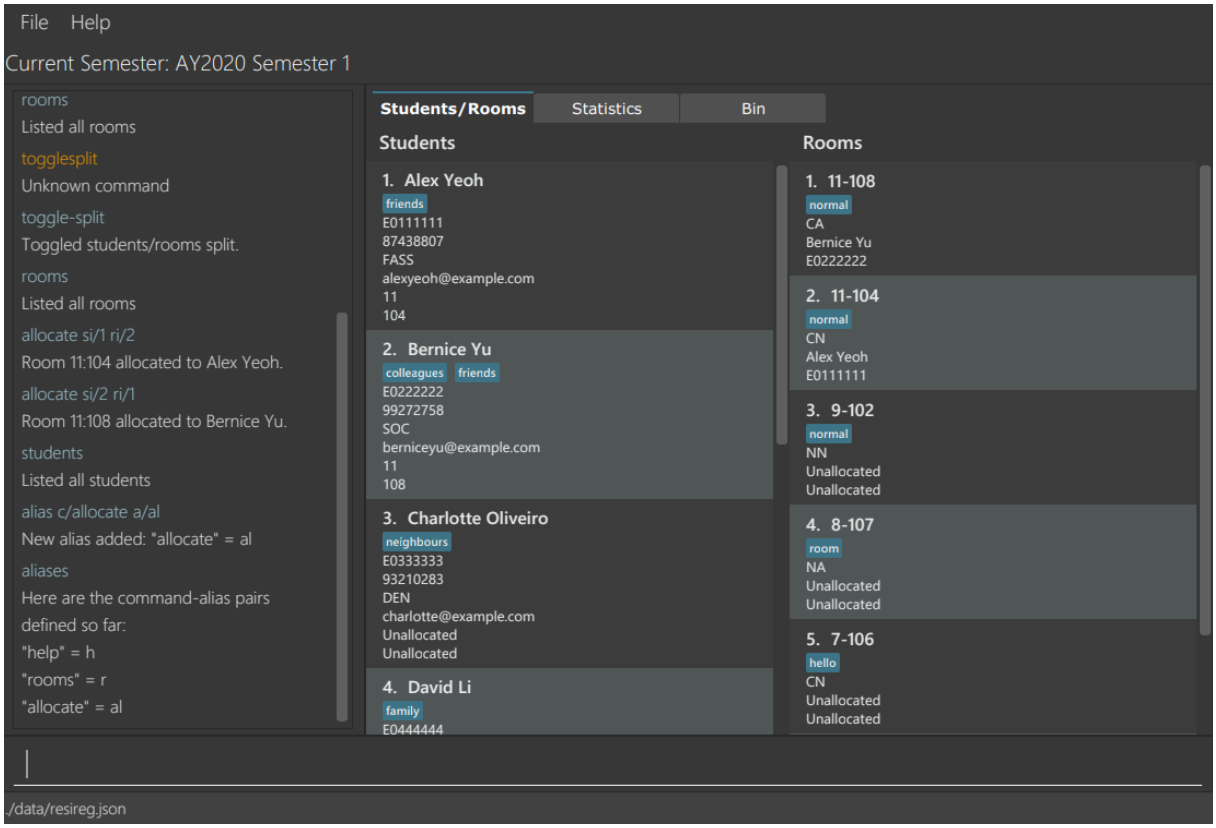



Room allocation works the same way as it does it v1.2. But now you can choose to view the students and rooms side by side to make it easier to do allocation. The toggle-split command merges the rooms and students tab together:

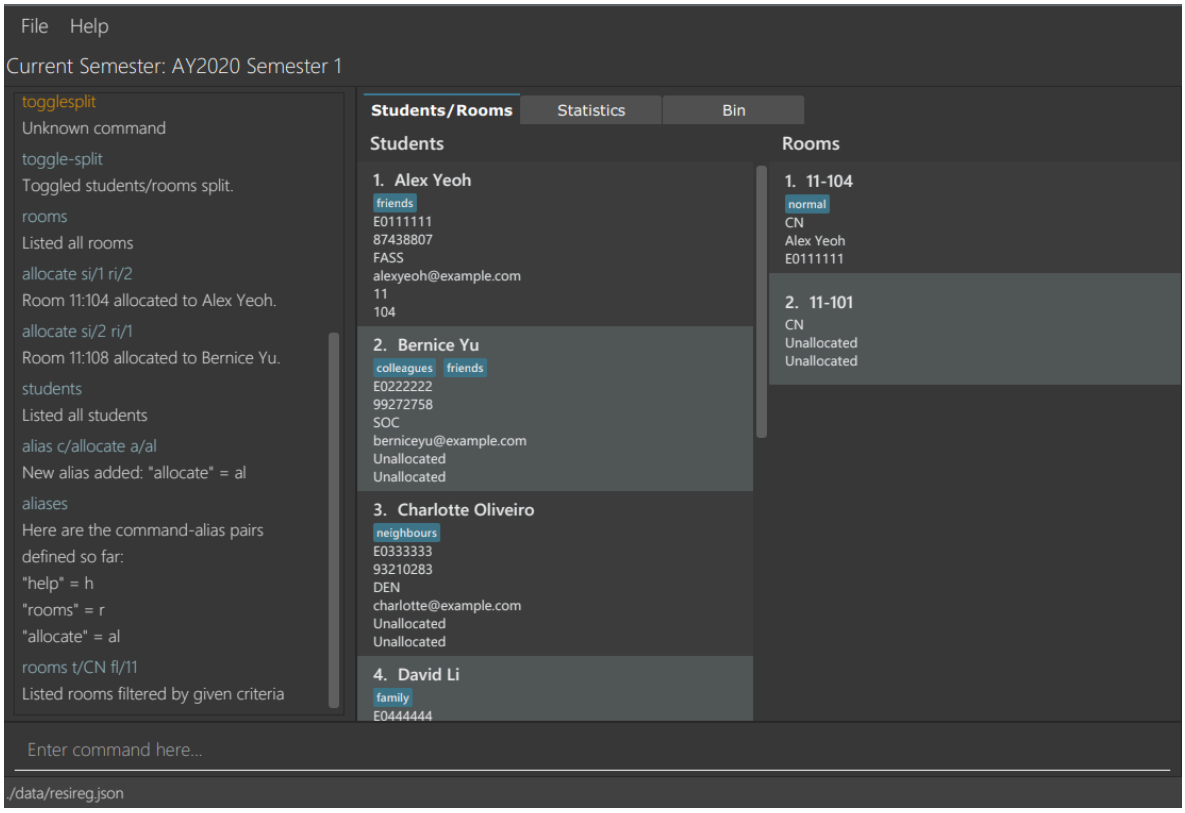



Clicking on the statistics tab lets you see room allocation statistics (command to do this coming soon):

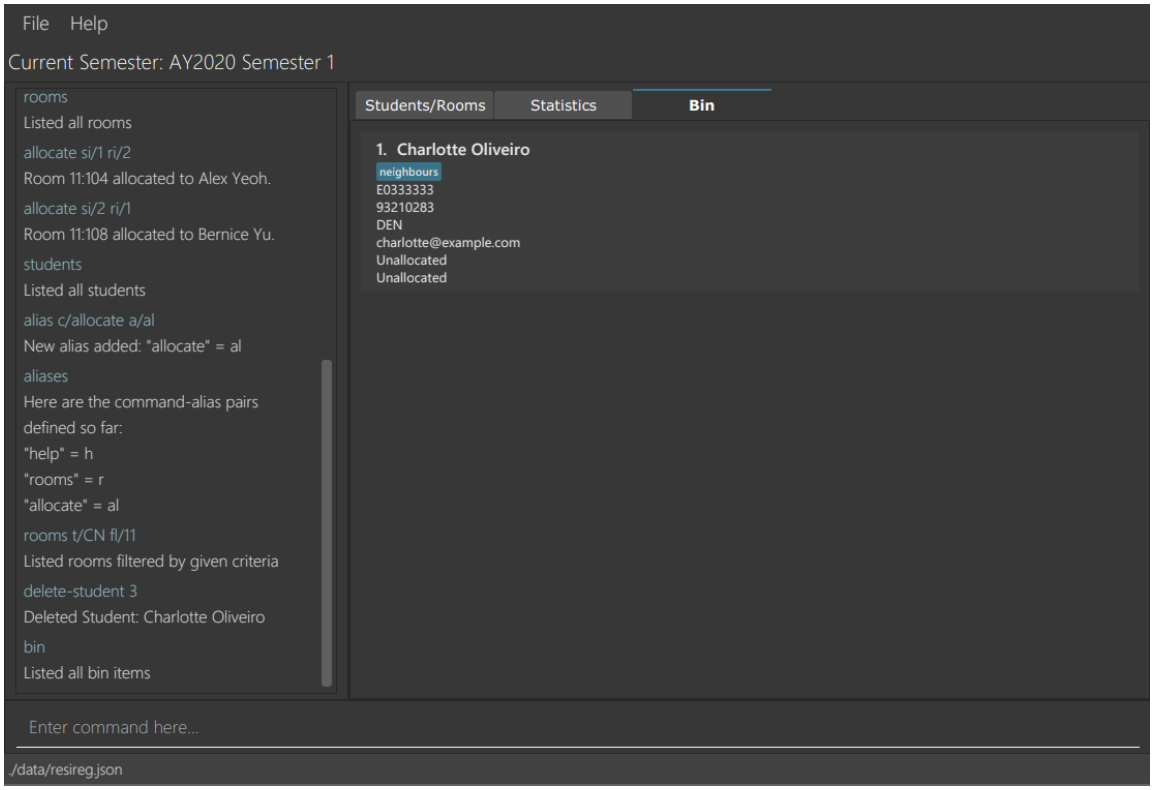

Aliases can be added for frequently used commands. Aliases added can of course be listed and deleted (not shown):



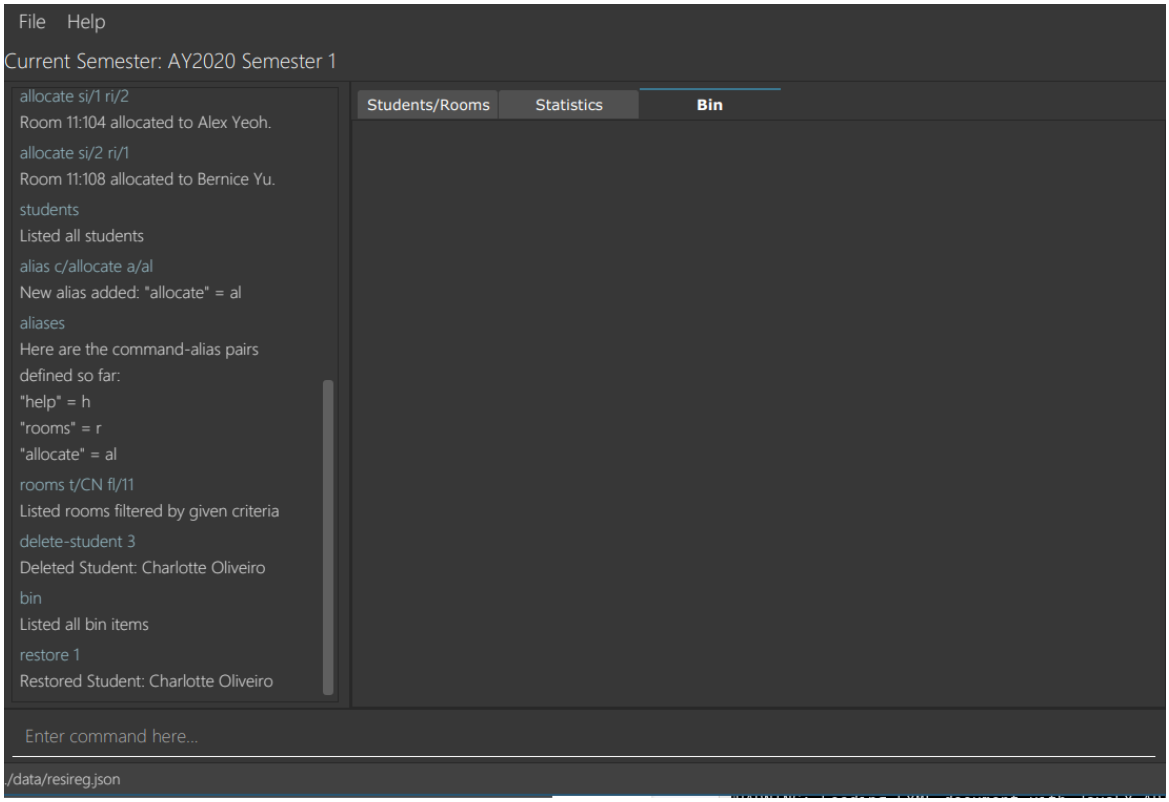

More advanced room filtering (below, we have filtered for all the corridor non-aircon rooms on the 11th floor):




We now have a trash bin for deleted students:

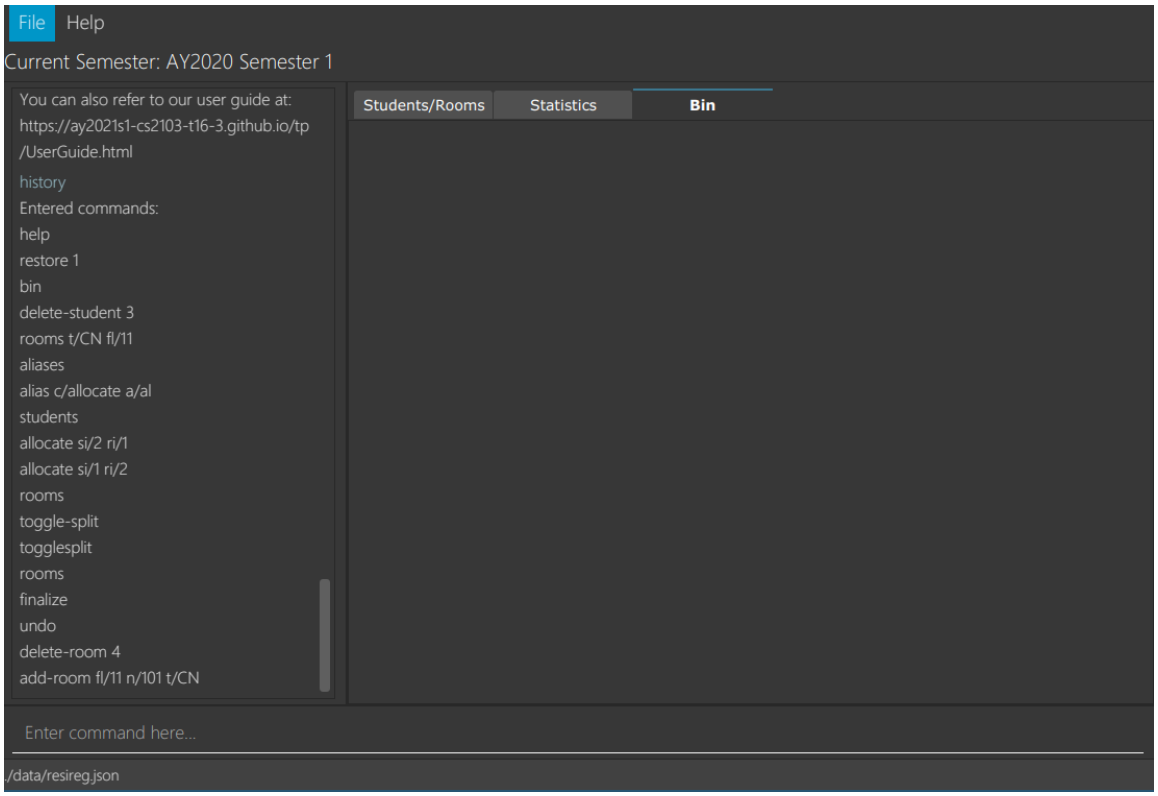


Deleted items can be restored:

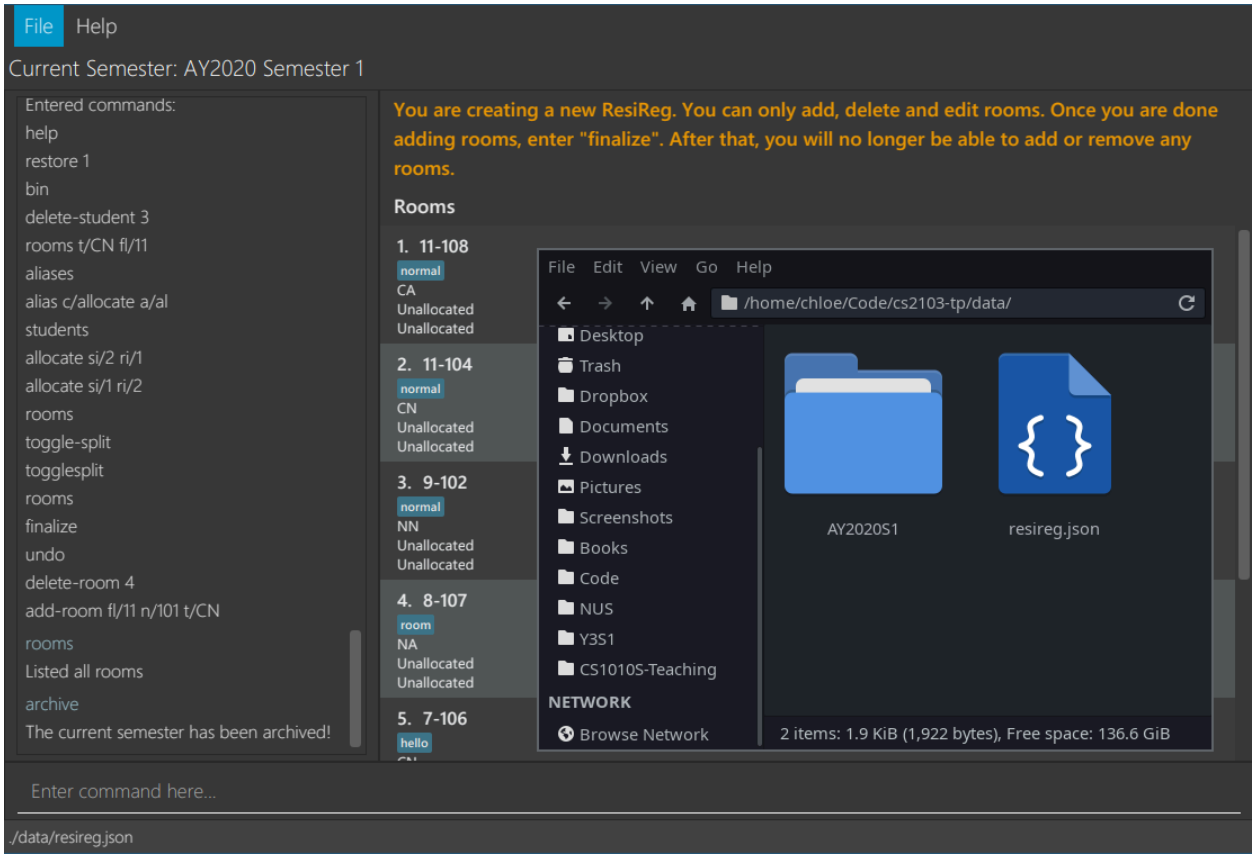



(not shown) items in the trash bin are deleted automatically after a certain number of days. The number of days can be set using the set-bin-expiry command.

History of previously entered commands can listed using the history command. Previously entered commands can also be accessed unix-shell style using the up/down arrow keys to make it easier to enter similar commands repeatedly (not shown):




At the end of the semester, the data can be archived. The archived file in a folder named after the current semester:


File
Help
Current Semester: AY2020 Semester 1
Entered commands:
help
restore 1
bin
delete-student 3
rooms t/CN fl/11
aliases
alias c/allocate a/al
students
allocate si/2 ri/1
allocate si/1 ri/2
rooms
toggle-split
togglesplit
rooms
finalize
undo
delete-room 4
add-room fl/11 n/101 t/CN
rooms
Listed all rooms
archive
The current semester has been archived!
You are creating a new ResiReg. You can only add, delete and edit rooms. Once you are done adding rooms, enter "finalize". After that, you will no longer be able to add or remove any rooms.
Rooms
1. 11-108
normal
CA
Unallocated
Unallocated
2. 11-104
normal
CN
Unallocated
Unallocated
3. 9-102
normal
NN
Unallocated
Unallocated
4. 8-107
room
NA
Unallocated
Unallocated
5. 7-106
hello
File Edit View Go Help
/home/chloe/Code/cs2103-tp/data/
Desktop
Trash
Dropbox
Documents
Downloads
Pictures
Screenshots
Books
Code
NUS
Y3S1
CS1010S-Teaching
NETWORK
Browse Network
AY2020S1
resireg.json
2 items: 1.9 KiB (1,922 bytes), Free space: 136.6 GiB
Enter command here...
./data/resireg.json