

<b>Week 4 - Project Direction (Deliverables: TEAMMATES)</b>	<b>3</b>
Targeting Resident Assistant / Hall comm publicity [Evolve]	3
Product Name:	4
Target User Profile:	4
Problem Addressed:	4
Link of Document	4
<b>Week 5 - Project Direction (Deliverables: Tutorial Sharing of User Stories)</b>	<b>5</b>
Deliverables:	5
iP BCD-Extension	5
User Stories	5
<b>Week 6: Feature List</b>	<b>6</b>
<b>Week 7 - tP Briefing and v1.1</b>	<b>8</b>
<b>Week 8 - Finishing 1.2 Iteration</b>	<b>11</b>
Agenda	11
Informal Demo	11
v1.2b things	11
DG/UG Standardisation	11
<b>Week 9: v1.2 Features Demo</b>	<b>13</b>
Exporting of information	13
Adding a new resident	13
Deleting a resident	13
<b>Week 10: Project Post-mortem, UG/DG</b>	<b>15</b>
Do a postmortem of the previous iteration	15
UG/DG:	15
<b>Week 11: v1.3 Features Demo</b>	<b>16</b>
Listing all student groups	16
Viewing of hall events	17
Assigning of residents to events	18
Adding of hall events	20
Editing of hall events	21
Deleting of hall events	22
Clearing attendees list of events	23
Filtering of residents by events	25

### **Admin Info**

Team ID: CS2103T-T11-2

Tutorial Document Link: [Tutorial Link](#)

Tutorial Zoom Link: [Tutorial Zoom](#)

iP Dashboard: [iP Dashboard](#)

tP Dashboard: [TP Dashboard](#)

Project Site: [Project Site](#)

## **Week 4 - Project Direction (Deliverables: TEAMMATES)**

- **Targeting Resident Assistant / Hall comm publicity [Evolve]**
  - 300 contacts
  - Contact automation, need to contact people, manually call?
  - Based on events such as paying rent, electrical bills, dinner and dance. Maybe block event, filter by faculty, filter by cca, filter by demerit points, filter by level.
  - CRUD their contacts:
  - Functionalities:
    - Demerit Points (1 full fledged feature)
      - Student, date, reason (Alcohol, equipment abuse, sexual offences)
    - CCA (1 full fledged feature)
      - CCA Attendance Tracker
    - Maintenance contacting and scheduling (1 full fledged feature)
      - Add on, date, reason, and room
    - Tagging + EZ export + Filter
      - Filter: Blocks / Faculty / CCA / Demerits / Level / Gender / Nationality (1 full feature)
      - Easily export to a human readable format
    - Past records by batch (1 small feature)
      - Past CCA members, past roommates, past batches, past level
    - Events (1 full fledged events)
      - Block Event
      - Hall Event
      - Level events
- **KIV**
  - Booking of facilities (1 full fledged features)
    - KIV
  - Room bidding / Room allocation
    - Too complicated to test in PE
    - KIV: Keep it simple first, if we have nothing left to add, we add this -Yan Cheng
  - Customizing room ranges / blocks
  - 'Settings' kind of concept
- Thumbs up by YanCheng, Kok Siang
- TODO: Evolve it more

### **Deliverables:**

By 04 Sep 2020:

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.

Product Name:

- Hall-y! Your smart hall management contact application.

Target User Profile:

- NUS Hall Administrative Staff (in charge of students staying in the block)

Problem Addressed:

- Difficult to keep track of and contact 100 students especially with day-to-day activities, like CCAs, events.
- Update hall records (such as demerit points and CCA points) for further reference
- Track maintenance information for students' rooms, including their maintenance staff and job done
- Track hall event attendance (some events need to be attended or else there will be consequences)

Link of Document

- [Hall-y CS2103T Project](#)

## Week 5 - Project Direction (Deliverables: Tutorial Sharing of User Stories)

Deliverables:

User Stories

- List a bunch of user stories
- Prioritise

iP BCD-Extension

Thuya: C-FlexibleDataSource

Kok Siang: C-Priority

Jiefeng - C-FriendlierSyntax

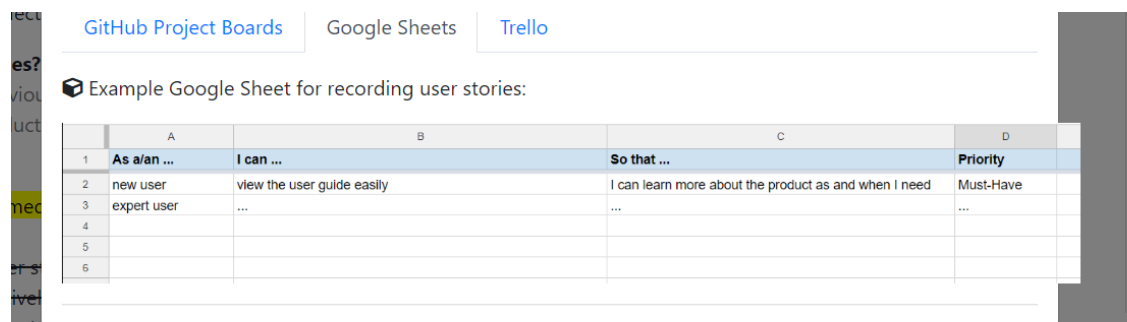
BiaoYi - C-Update

YanCheng - C-Help

User Stories

[User Stories](#)

User Stories



Example Google Sheet for recording user stories:

	A	B	C	D
	As a/an ...	I can ...	So that ...	Priority
1				
2	new user	view the user guide easily	I can learn more about the product as and when I need	Must-Have
3	expert user	...	...	...
4				
5				
6				

*User story format: As a {user type/role} I can {function} so that {benefit}*

Step 1: Define the target user as a persona:

Jeff is a hall administrative staff, he interacts with a lot of hall residents, namely NUS students. And every day is an interesting day, because there are a lot of activities going around, CCA's, events and other interesting stuff.

Step 2: Define the problem scope:

- Hall-y helps Tommy keep track of all the residents' contacts in the hall. It does not cover communicating with contacts.
- As a hall resident officer, basically there are too many residents in hall, and every

Step 3: List scenarios to form a narrative:

- First use:
  - He just started using Hall-y, he downloads it and launches it to check out what it can do.

Step 4: List the user stories to supports the scenarios:

## **Week 6: Feature List**

Residents:

### **Contact info**

- 1) Email <Class>
- 2) Phone numbers
  - a) Home Number
  - b) Mobile Number
- 3) Telegram handles (KIV)
- 4) Residential Address

### **Person info**

- 1) Name <Class>
- 2) Gender
- 3) Nationality

### **Academic info**

- 1) Matric Number <Class>
- 2) Matriculation year <Class> (low prio drop if needed)
- 3) Faculty <Enum> (postponed)

### **Residential info**

- 1) Block
- 2) Room number
- 3) Demerit point (KIV)

### **Exporter Class**

- UI that we can show the list of exported classes
  - Or we can change this to saving a file
- we need the internal classes in charge of molding the data
- export interface
  - Emails, phone numbers
  - interface exportable

### **Delegation:**

- Take 3 bullet points
  - Integrate it into the current user guide draft

### **5 areas:**

- 1) Create (Thuya)
- 2) Read (Biao Yi)
- 3) Edit (Kok Siang)
- 4) Delete (Jiefeng)
- 5) Export (Yan Cheng)

### **TODO by:**

- Tuesday Night (wednesday changes)

## **Saturday**

Feature list of Hall-y 1.2:

- 1) Management of residents' record:
  - a) Create residents' record
  - b) Read residents' record
  - c) Edit residents' record
  - d) Delete residents' record
- 2) Exporting of contact info
  - a) Export in mass email format (any simple format)
  - b) Phone numbers (any simple format)
  - c) Telegram handles (any simple format)

## **Week 7 - tP Briefing and v1.1**

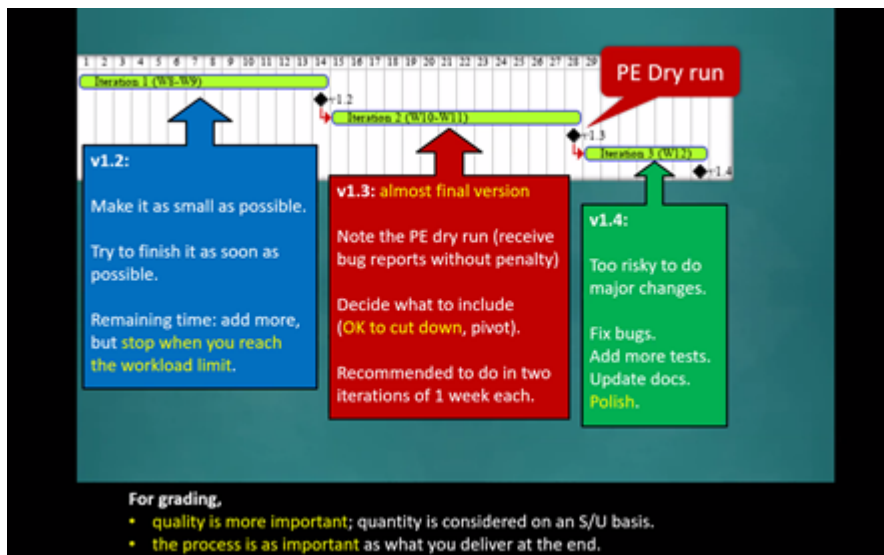
An attempt to TLDR the tP requirements. [Read this for more info](#). No bell-curve btw.

1. Product Design (5m)
  - a. Evaluated by tutor (demo and UG) and peers (peer testing and UG)
  - b. No need for High in all areas – will minus for feature flaws
2. Implementation (10m)
  - a. Code Quality
    - i. Cover logging/exceptions/assertions/defensive coding
    - ii. SLAP, DRY, and other things as in coding standard
  - b. Effort
    - i. Step 1: Effort for entire project i.e. how much from AB3. Graded by tutors and peers, seems like most get max here. May rebut using DG Appendix (IDK what this is, probably will release info in future weeks)
    - ii. Step 2: Peer evaluation. You may rebut if you disagree. Tutor/Damith will determine from evidence too.
3. Project Grading (10m) – in P.E., grading works in 70-30, 70% for your stronger aspect
  - a. Developer Testing
    - i. How many bugs found in your product/section – will be split if the section is from team
    - ii. Test code
    - iii. Minus for not handling real user behaviour, product differs from UG, weird product behaviours
  - b. System/Acceptance Testing
    - i. How many bugs you found in another product
4. Documentation (10m)
  - a. Whole DG and UG, graded by tutor and peers
  - b. How much effort is due to you
  - c. Minus for UG, DG bugs found in during PE
5. Project Management (5 + 5 = 10m)
  - a. Process
    - i. Do iteratively, use and reach milestones, define/assign/track issue on issue tracker
    - ii. Reach 60% of milestone can already
  - b. Team tasks
    - i. General chores like issue tracker, general misc info in UG/DG
    - ii. Merge code in 4 out of 6 weeks(7,8,9,10,11,12)

### **18 Sep – tP Briefing Notes**

**Read Appendix[E]: Project Schedule Tracking and then recommended to do 1 as team activity i.e. in AboutUs page, each person add their own name. So can practice the workflow**





## tp guidance

**Guidance for the project?** E.g., is this a good design?

- No direct answer from the tutor
- Can ask in the forum (or send to prof via tutor)
  - might get some general guidance

**Then how?**

- ☐ Does it work?
- ☐ Does it keep working?
- ☐ Is it easy to change/extend?
- ☐ Is it easy to break?
- ☐ Does it make the code complicated?
- ☐ Do known design patterns/principles apply?
- ☐ If in doubt, go with the easier alternative.

**Also note:**

- tp is mostly graded by peers.
- Look at things from the reader/user POV decide what's best for them.
- See what others are doing
- Tutors are monitoring for major red flags

**Turned you made the wrong decision?** Change it!

This is a non-exhaustive list; you may define additional roles.

- Team lead:** Responsible for overall project coordination.
- Documentation** (short for 'in charge of documentation'): Responsible for the quality of various project documents.
- Testing:** Ensures the testing of the project is done properly and on time.
- Code quality:** Looks after code quality, ensures adherence to coding standards, etc.
- Deliverables and deadlines:** Ensure project deliverables are done on time and in the right format.
- Integration:** In charge of versioning of the code, maintaining the code repository, integrating various parts of the software to create a whole.

- *Scheduling and tracking*: In charge of defining, assigning, and tracking project tasks.
- *[Tool ABC] expert*: e.g. IntelliJ expert, Git expert, etc. Helps other team member with matters related to the specific tool.
- *In charge of [Component XYZ]*: e.g. In charge of **Model**, **UI**, **Storage**, etc. If you are in charge of a component, you are expected to know that component well, and review changes done to that component in v1.3-v1.4.

Biao yi - Documentation

Thuya - [Git] expert / Integration

Jie Feng - Testing / Team Lead

Kok Siang - Deadlines and deliverables

Yan Cheng - Scheduling and tracking / Code quality

Take out tagging feature

- Plan the next iteration. As you know, you should follow the *breadth-first iterative* process. Therefore, first you must decide what functionalities should be in the product if you had only two weeks to implement it. You have done that already when you chose user stories for v1.2, translated that to features, and even drafted the UG based on those features. You can tweak that plan further at this point if you wish, given that you now have some idea of how fast the team can work when using the prescribed workflow.
  - Aim to produce a *working MVP* at the end of this iteration even if the functionalities are not polished (polishing can be done in a later iteration).
  - If possible, break the iteration into two *increments* i.e., aim to produce an even simpler but working version after one week.
  - Avoid depth-first implementations: "I'll do the back-end part of feature X in this iteration" is not acceptable as that is not in the spirit of iterative development. Remember, we are pretending this to be the last iteration; why would you implement the back-end part of a feature in the last iteration?  
It is OK to add simpler versions of bigger features, but not OK to add partial features that can't be used yet.
- Divide the work among the team members i.e., the work required for the current iteration.
- Reflect the above plan in the issue tracker by assigning the corresponding issues (create new issues if necessary) to yourself and to the corresponding milestone. For example, the user story pertaining to the increment **show a place holder for photo, showing a generic default image** should be assigned to Jake and to milestone **v1.2**

Optionally, you can define a **mid-v1.2** milestone to keep track of things to be done within the first half of the milestone

## **Week 8 - Finishing 1.2 Iteration**

### **Agenda**

- 1) Close 1.2 - push to Monday
- 2) Prepare informal demo for tutorial
  - a) As we don't have in-class tutorials this semester, do the following in lieu of a live demo:
  - b) [one member] Run your app using the version tagged v1.2. Take screenshots of each available feature in action. Add those screenshots to your collaborative project notes document with an appropriate heading e.g., v1.2 features demo. Alternatively, **you can screen-record a demo**, upload it to somewhere, and post the link in the project notes document.
- 3) OP Consult - discuss on Wed again. Prep/Brainstorm needed
  - a) Demo format / presentation flow / tagline
  - b) Pitch format / presentation flow / tagline
- 4) Discuss DG UG, 1.2b
  - a) Standardisation of DG UG
  - b) 1.2b tasks
  - c) DG UG Work allocation?

### **Informal Demo**

Upload the videos into the project document.  
Show the add/edit command with new fields

New features to show:

1. Yan Cheng will demo export command
2. The rest will show the add/edit command with new fields
3. Biao Yi will take a screenshot of the add commands(add to colab docs)

### **v1.2b things**

Either test cases or documentation.  
Must do documentation.

Issue #57: v1.2b

Test cases for YanCheng and BiaoYi Issue #70, #71: v1.2b

### **DG/UG Standardisation**

Each person takes one area to standardise.

General points to standardise.

- Title case for titles only. Will be using sentence case in sub-headers everywhere else.

- In general, each DG diagram should begin with a description, followed by the diagram. In the description, it should reference the diagram as well. Of course, each diagram has to be labelled.
- KIV grammar and punctuation mistakes

## Week 9: v1.2 Features Demo

# Week 9 [Fri, Oct 9th] - Tutorial

## 0 Demo v1.2

The tutor to confirm the following has been done.

Admin ↗ tP v1.2: Demo ★★★

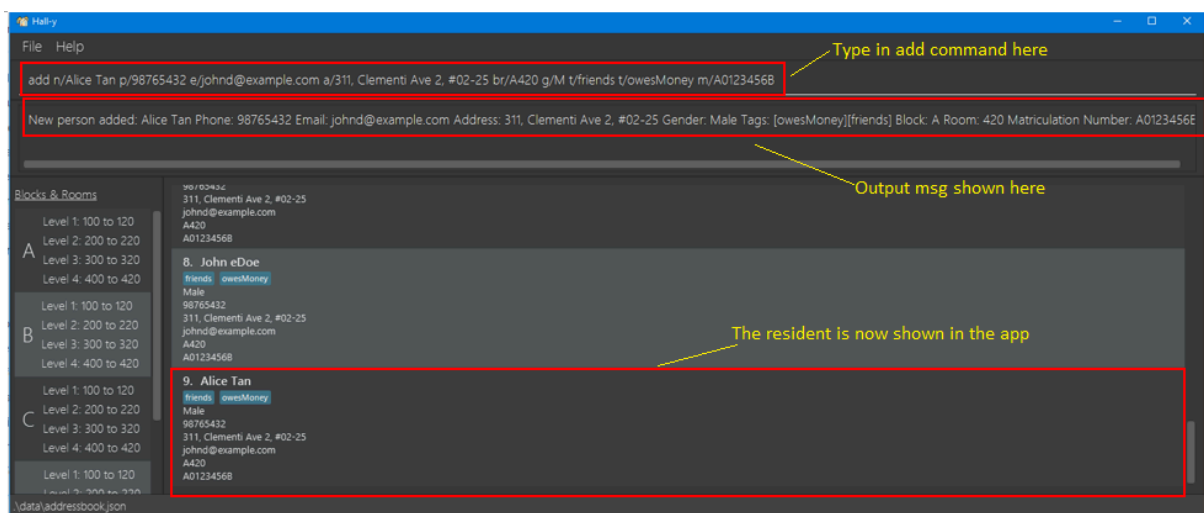
- As we don't have in-class tutorials this semester, do the following in lieu of a live demo:  
[one member] Run your app using the latest released version **v1.2** (or **v1.2b**, if applicable). Take screenshots of each available feature in action. Add those screenshots to your *collaborative project notes* document with an appropriate heading e.g., **v1.2 features demo**. Alternatively, you can screen-record a demo, upload it to somewhere, and post the link in the project notes document.

## Exporting of information

### [Export feature demo video](#)

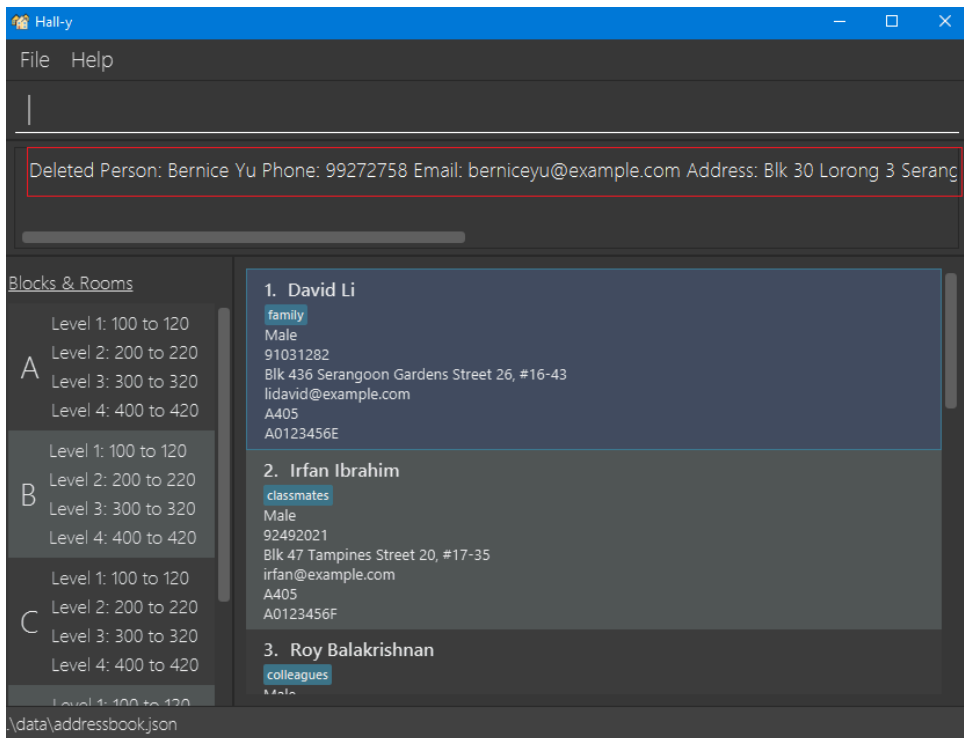
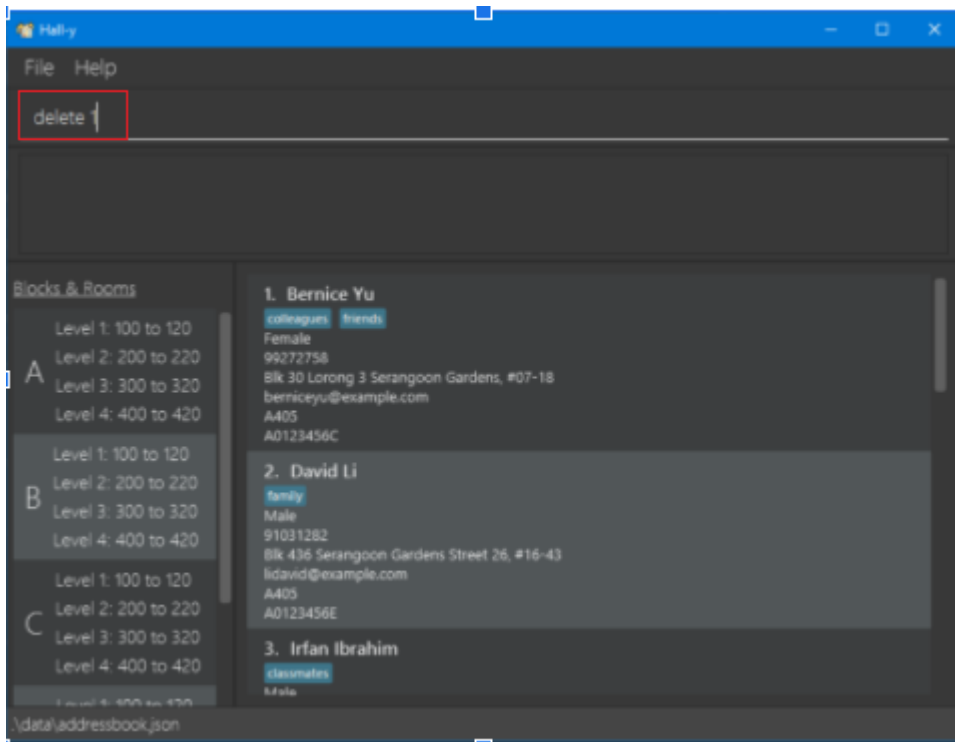
## Adding a new resident

1. Type 'add' followed by the resident details in the specified format in the input box
2. Press enter
3. Output msg is shown below and the new resident is displayed in the list of residents



## Deleting a resident

1. Type 'delete' followed by the index number in the input box
2. Press enter
3. Output msg is shown below and the list of residents will be updated to reflect the deletion



## **Week 10: Project Post-mortem, UG/DG**

### **Do a postmortem of the previous iteration**

- Discuss with the team how the iteration went (i.e., what worked well, what didn't), and your plans to improve the process (not the product) in the next iteration.

### **What went well**

- We accomplished everything we set out to do
- Every feature followed the forking workflow
- Reviewed every code change to the team repo
- Everyone was involved in the issue creation, code reviewing and merging

### **What did not go well**

- Merge conflicts and putting merge request reviews everything to last minute
- We push the deadline too late

### **How to improve, guidelines**

- Directly assign the person, get them to review early as possible
- Guideline: Get the person who has least recently reviewed to do it
- Idea: be fair, get everyone involved. Just make you review something once in awhile
- Finish (commit to master) everything one day before the deadline

### **UG/DG:**

- Update the current UG/DG. Look out for language errors/typos/formatting errors
- Add your v1.3 features into the before Friday.
  - Add to markdown then we'll convert to pdf
  - Lookout for: UG (2.\* + Command summary) + UG Checklist (CS2101 Week 4.2)
  - Lookout for DG:
- Allocation:
  - Biao Yi: Add / edit feature
  - Thuya Oo : Find feature
  - Kok Siang: Create student group and add student group (i.e. converted the previous tag feature)
  - Jiefeng: Hall event
  - Yan Cheng: Assigning command + Export (v1.2) feature
  - Others: help, list, delete
  - Others: clear, exit, saving data BY Pang

## Week 11: v1.3 Features Demo

### Listing all student groups

You can list all student groups by using the list-group command.

The steps for this command are as follows:

1. Enter the list command by typing list-group into the input box
2. Press enter
3. The result box will display all the student groups:

The screenshot shows the Hall-y application window. At the top, there's a menu bar with 'File' and 'Help'. Below it is a command input box. The main area displays the output of the 'list-group' command: 'Listed all student groups' followed by a list: 1. badminton, 2. basketball, 3. choir, 4. dance, 5. hackers, 6. soccer, 7. tabletennis. Below this, there are three panels: 'Blocks', 'Residents', and 'Events'. The 'Blocks' panel lists floors 1 to 4 and rooms 101 to 420 for blocks A, B, C, and D. The 'Residents' panel shows details for Alex Yeoh (badminton) and Bernice Yu (basketball, dance). The 'Events' panel shows '1. Hall Lunch' and '2. Hall Dinner'. The status bar at the bottom indicates the data source is './data/addressbook.json'.

Blocks	Residents	Events
<b>A</b> Floors : 1 to 4 Rooms : 101 to 420	1. Alex Yeoh badminton Male 87438807 Blk 30 Geylang Street 29, #06-40 alexyeoh@example.com A405 A0123456B	1. Hall Lunch 01/11/2020 13:00 @ Dining Hall Hall-wide lunch event.
<b>B</b> Floors : 1 to 4 Rooms : 101 to 420	2. Bernice Yu basketball dance Female 99272758 Blk 30 Lorong 3 Serangoon Gardens, #07-18 berniceyu@example.com A405 A0123456C	2. Hall Dinner 24/11/2020 18:00 @ Dining Hall Hall-wide dinner event.
<b>C</b> Floors : 1 to 4 Rooms : 101 to 420		
<b>D</b> Floors : 1 to 4 Rooms : 101 to 420		

./data/addressbook.json



## Viewing of hall events

You can view the list of hall events on the UI

Hall-y

File Help

Listed all residents

**Blocks**

A Floors : 1 to 4  
Rooms : 101 to 420

B Floors : 1 to 4  
Rooms : 101 to 420

C Floors : 1 to 4  
Rooms : 101 to 420

D Floors : 1 to 4  
Rooms : 101 to 420

**Residents**

1. Alex Yeoh

badminton

Male

87438807

Blk 30 Geylang Street 29, #06-40

alexyeoh@example.com

A405

A0123456B

2. Bernice Yu

basketball dance

Female

99272758

Blk 30 Lorong 3 Serangoon Gardens, #07-18

berniceyu@example.com

A405

**Events**

1. Hall Lunch

01/11/2020 13:00 @ Dining Hall

Hall-wide lunch event.

Charlotte Oliveiro Bernice Yu Alex Yeoh

2. Hall Dinner

24/11/2020 18:00 @ Dining Hall

Hall-wide dinner event.

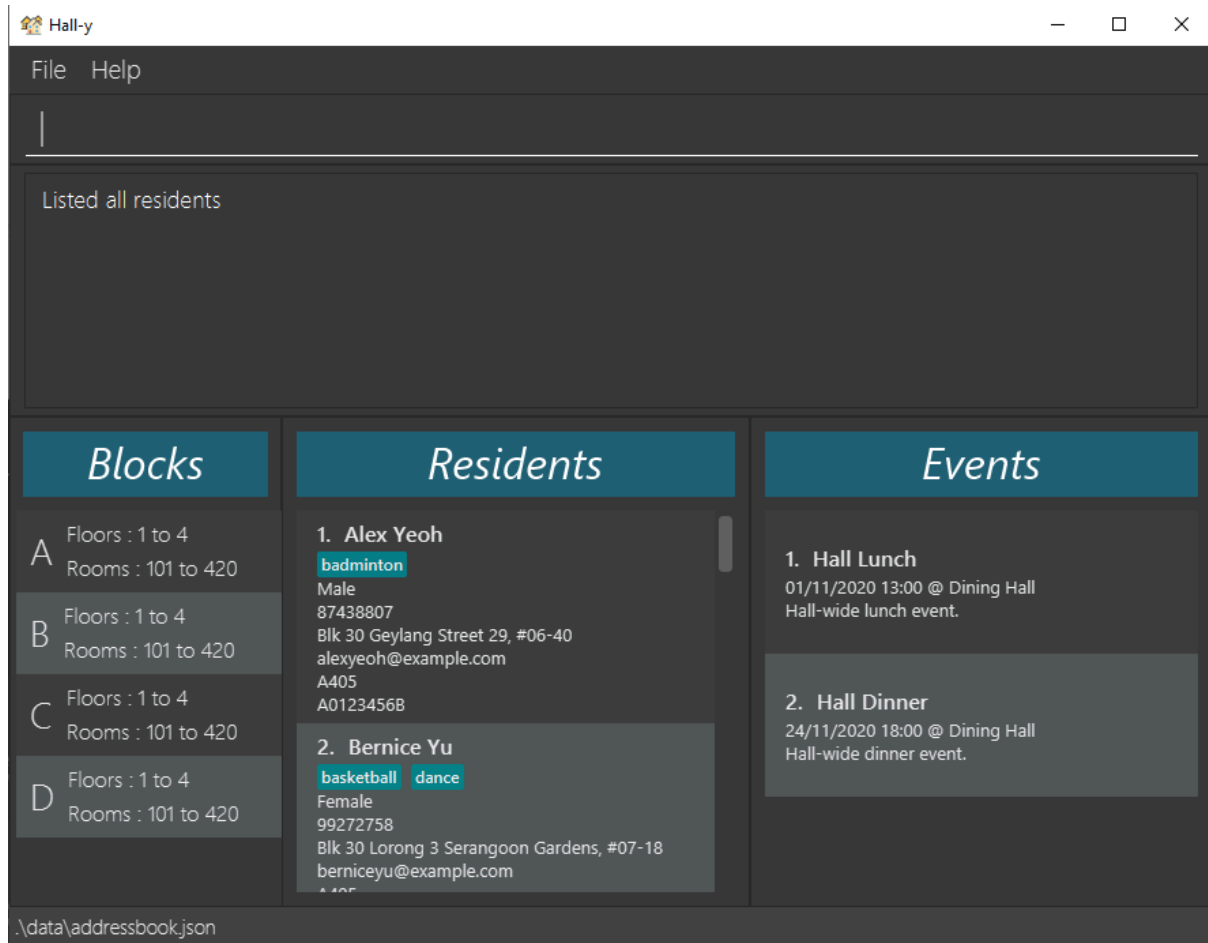
David Li Irfan Ibrahim

.\data\addressbook.json

## Assigning of residents to events

You can assign a resident to an event by using the assign command

Let's say you want to assign the 1st resident "Alex Yeoh" to the 1st event "Hall Lunch"



The steps for this command are as follows:

1. Enter the assign command by typing 'assign 1 1'
2. Press enter
3. The result box will display the details of the resident added to the event
4. The event list will be updated as shown

File Help

Assigned resident Alex Yeoh to Hall Lunch

*Blocks*

A Floors : 1 to 4  
Rooms : 101 to 420

B Floors : 1 to 4  
Rooms : 101 to 420

C Floors : 1 to 4  
Rooms : 101 to 420

D Floors : 1 to 4  
Rooms : 101 to 420

*Residents*

## 1. Alex Yeoh

[badminton](#)

Male

87438807

Blk 30 Geylang Street 29, #06-40

alexyeoh@example.com

A405

A01234568

## 2. Bernice Yu

[basketball](#) [dance](#)

Female

99272758

Blk 30 Lorong 3 Serangoon Gardens, #07-18

berniceyu@example.com

A405

*Events*

## 1. Hall Lunch

01/11/2020 13:00 @ Dining Hall  
Hall-wide lunch event.[Alex Yeoh](#)

## 2. Hall Dinner

24/11/2020 18:00 @ Dining Hall  
Hall-wide dinner event.

## Adding of hall events

You can add a hall event by using the add-event command.

Suppose you want to create an event called Night Cycling, with the following parameters

- Name: Night Cycling
- Location: Meet at Hall Entrance
- Event Date: 30/10/2020 18:00
- Description: A night cycling event held every semester

The steps for this command are as follows:

1. Enter the add-event command by typing `add-event n/Night Cycling l/Meet at Hall Entrance dt/30/10/2020 18:00 d/A night cycling event held every semester.`
2. Press enter
3. The result box will display the details of the event added
4. The event list will be updated as well.

The screenshot shows the Hall-y application window. At the top, there's a menu bar with 'File' and 'Help'. Below it is a command input field. A message box displays the details of a newly added event: 'Night Cycling' on '30/10/2020 18:00' at 'Meet at Hall Entrance' with the description 'A night cycling event held every semester.' The main interface is divided into three panels: 'Blocks', 'Residents', and 'Events'. The 'Blocks' panel lists four blocks (A, B, C, D) with their respective floor and room ranges. The 'Residents' panel shows two residents: Alex Yeoh (male, badminton) and Bernice Yu (female, basketball, dance). The 'Events' panel lists two events: 'Hall Pizza Party' and 'Night Cycling'.

Hall-y

File Help

New event added:  
Night Cycling  
Event Date: 30/10/2020 18:00  
Location: Meet at Hall Entrance  
Description: A night cycling event held every semester.

Blocks	Residents	Events
<b>A</b> Floors : 1 to 4 Rooms : 101 to 420	<b>1. Alex Yeoh</b> badminton Male 87438807 Blk 30 Geylang Street 29, #06-40 alexyeoh@example.com A405 A0123456B	<b>1. Hall Pizza Party</b> 25/12/2020 13:00 @ Dining Hall Just a simple pizza party.
<b>B</b> Floors : 1 to 4 Rooms : 101 to 420	<b>2. Bernice Yu</b> basketball dance Female 99272758 Blk 30 Lorong 3 Serangoon Gardens, #07-18 berniceyu@example.com	<b>2. Night Cycling</b> 30/10/2020 18:00 @ Meet at Hall Entrance A night cycling event held every semester.
<b>C</b> Floors : 1 to 4 Rooms : 101 to 420		
<b>D</b> Floors : 1 to 4 Rooms : 101 to 420		

.\data\addressbook.json

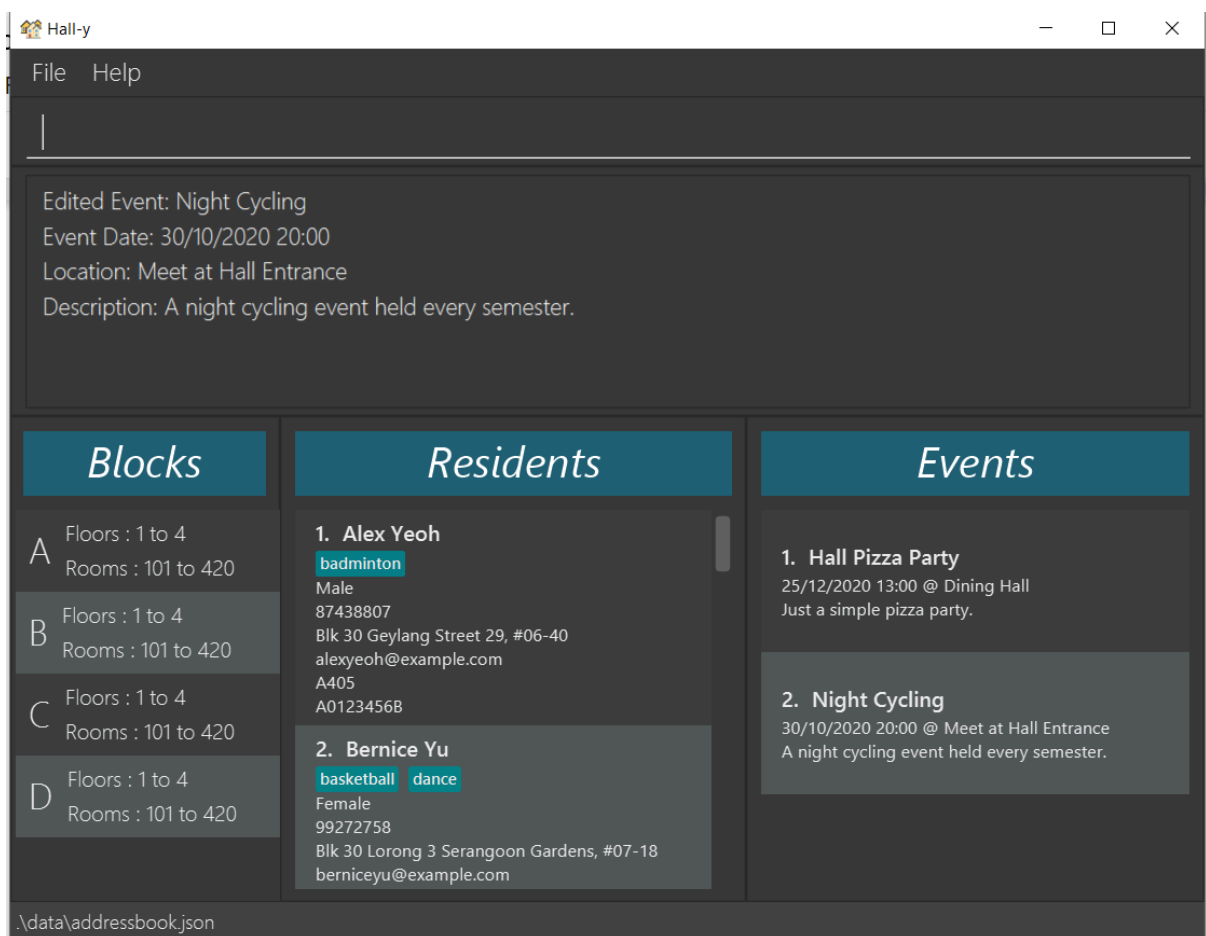
## Editing of hall events

You can edit a hall event by using the edit-event command.

Suppose after you created the Night Cycling event, there was a change for it to start at 20:00 instead.

The steps to edit the hall event are:

1. Assume the event index of Night Cycling, based on the events list, is 2.
2. Enter the edit-event command by typing `edit-event 2 dt/30/10/2020 20:00`
3. Press enter
4. The GUI will update as so:



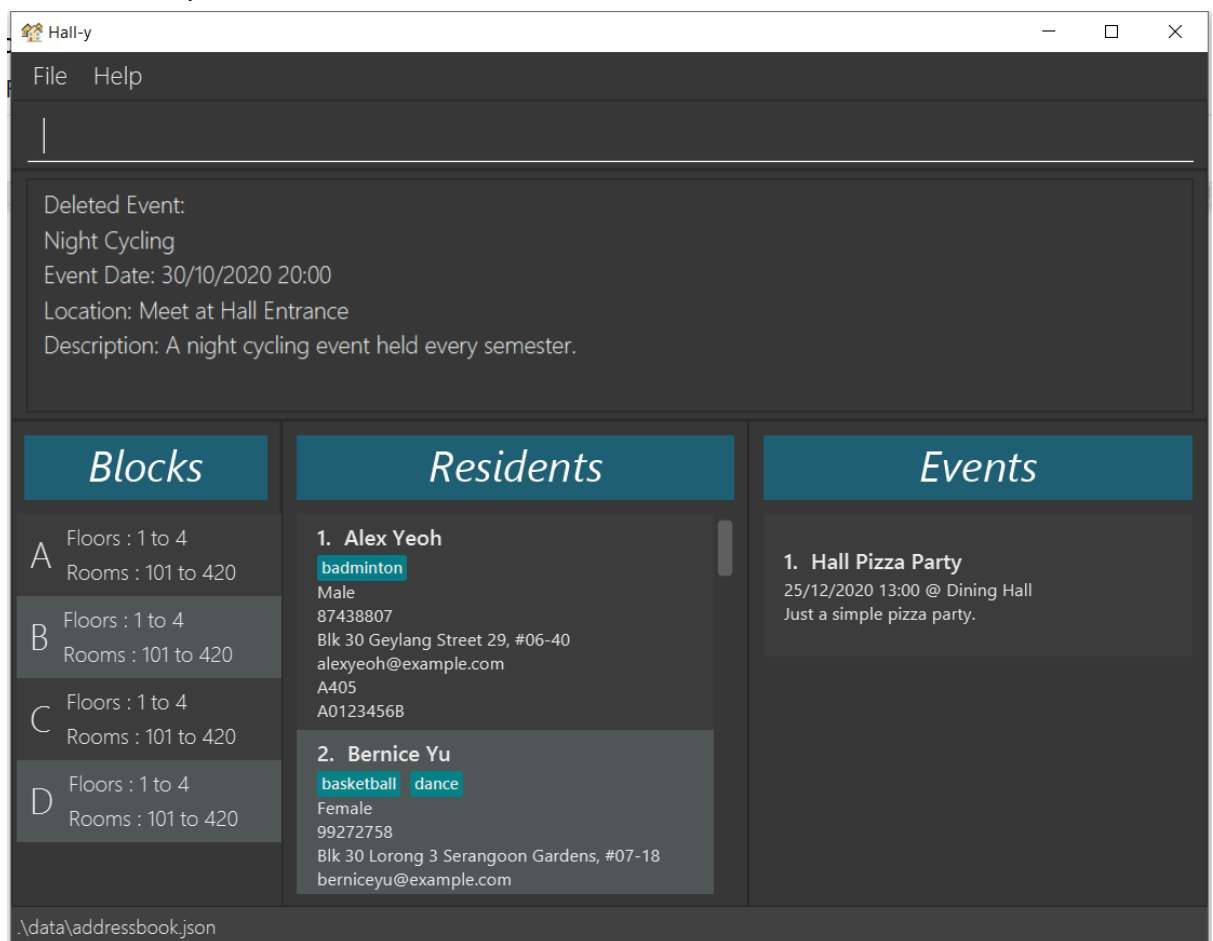
## Deleting of hall events

You can delete a hall event by using the delete-event command.

Suppose you want to delete Night Cycling. Let's assume that based on the events list, it has the index 2.

The steps to delete this event are:

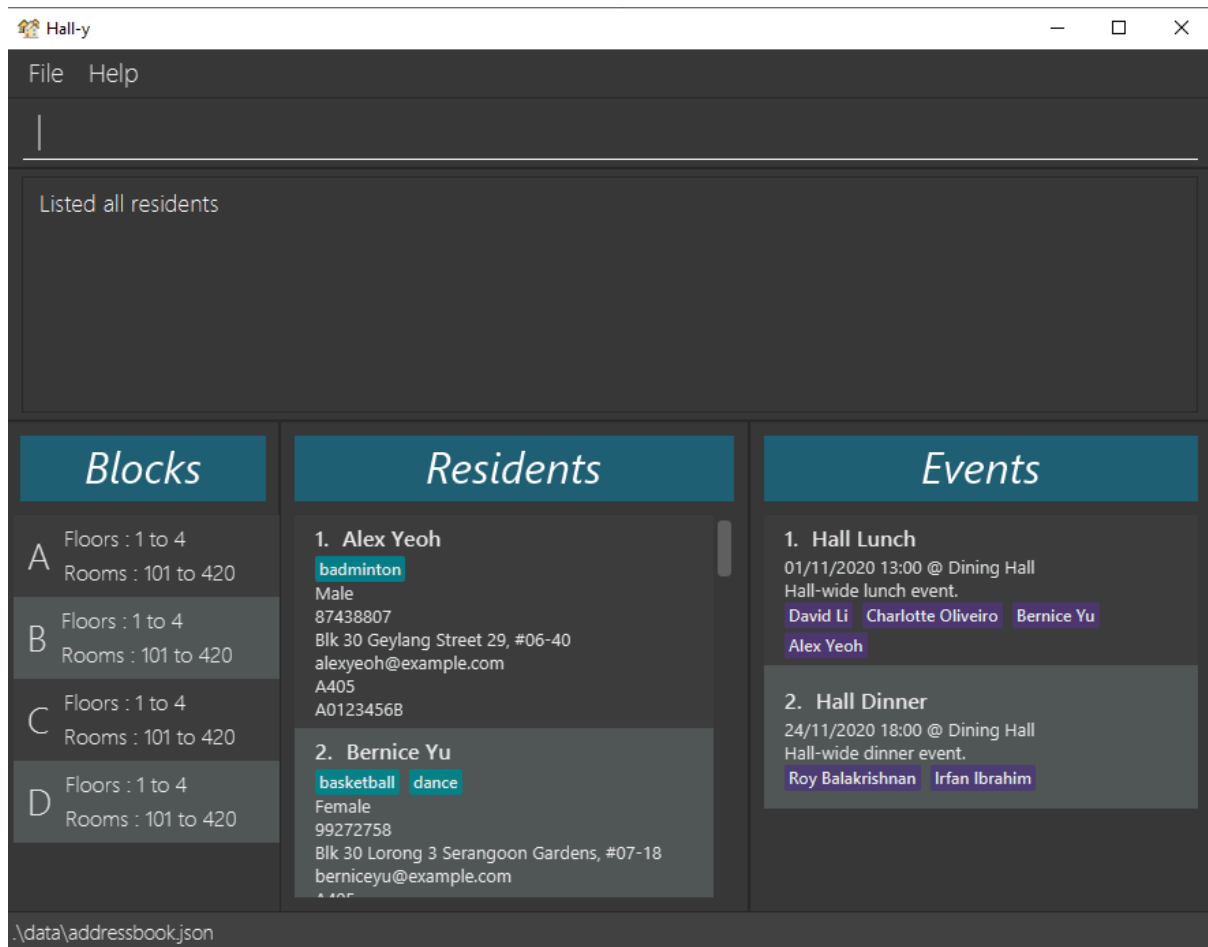
1. Enter the delete-event command by typing `delete-event 2`
2. Press enter
3. The GUI will update like so:



## Clearing attendees list of events

You can clear the attendees list of events by using the clear-event command

Let's say you want to clear the attendee list of the 1st event "Hall Lunch"



The steps for this command are as follows:

1. Enter the clear event command by typing 'clear-event 1'
2. Press enter
3. The result box will display the details of the event
4. The event list will be updated as shown

Hall-y

File Help

Cleared attendee list of Event:  
Hall Lunch  
Event Date: 01/11/2020 13:00  
Location: Dining Hall  
Description: Hall-wide lunch event.

Blocks

A

Floors : 1 to 4  
Rooms : 101 to 420

B

Floors : 1 to 4  
Rooms : 101 to 420

C

Floors : 1 to 4  
Rooms : 101 to 420

D

Floors : 1 to 4  
Rooms : 101 to 420

Residents

1. Alex Yeoh

badminton

Male

87438807

Blk 30 Geylang Street 29, #06-40

alexyeoh@example.com

A405

A0123456B

2. Bernice Yu

basketball dance

Female

99272758

Blk 30 Lorong 3 Serangoon Gardens, #07-18

berniceyu@example.com

A405

Events

1. Hall Lunch

01/11/2020 13:00 @ Dining Hall  
Hall-wide lunch event.

2. Hall Dinner

24/11/2020 18:00 @ Dining Hall  
Hall-wide dinner event.  

Roy Balakrishnan Irfan Ibrahim

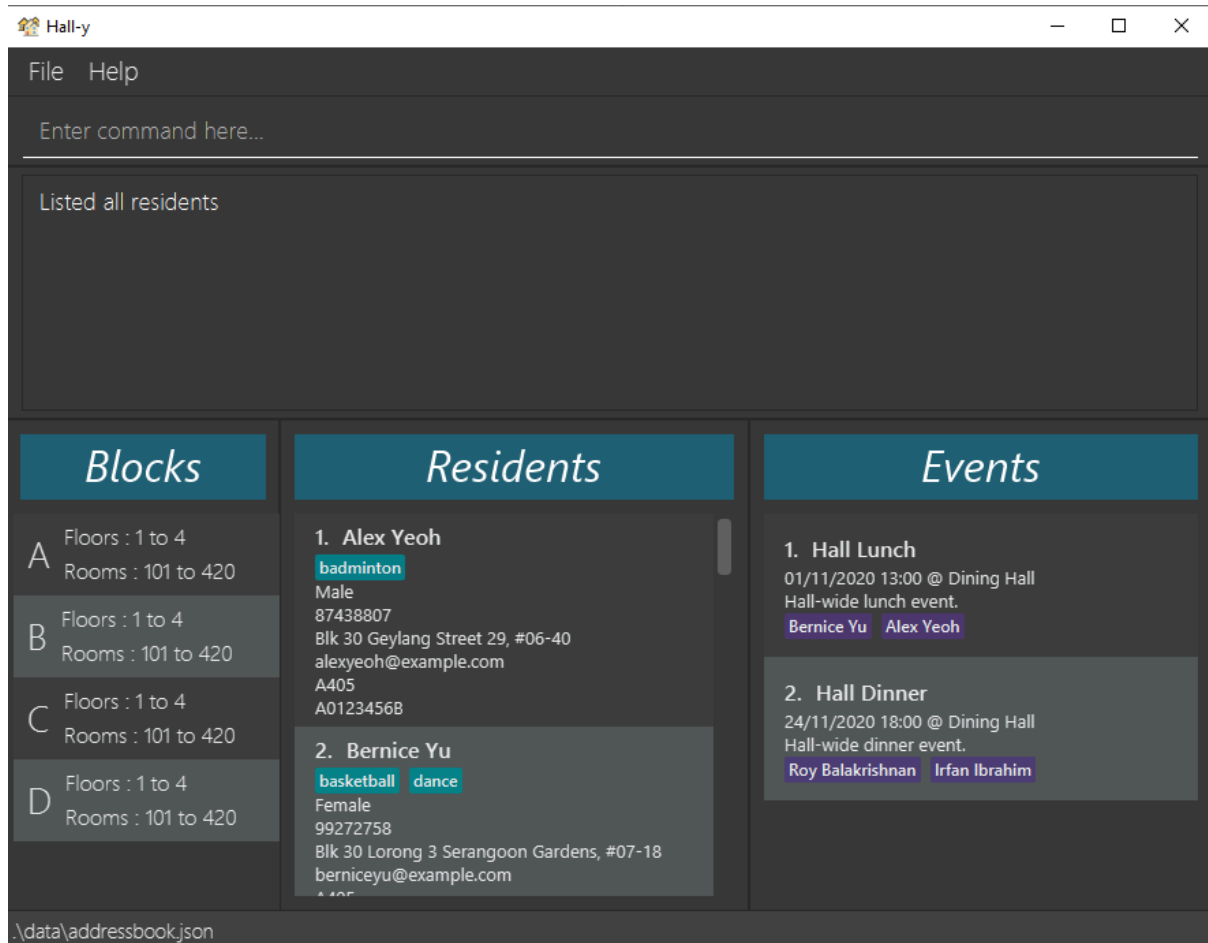
.data\addressbook.json



## Filtering of residents by events

You can filter residents by events by using the filter-event command

Let's say you want to filter the residents attending the 1st event "Hall Lunch"



The steps for this command are as follows:

1. Enter the filter-event command by typing "filter-event 1"
2. Press enter
3. The resident list will be updated as shown

File Help

|

Displaying residents attending event Hall Lunch

## Blocks

A Floors : 1 to 4  
Rooms : 101 to 420

B Floors : 1 to 4  
Rooms : 101 to 420

C Floors : 1 to 4  
Rooms : 101 to 420

D Floors : 1 to 4  
Rooms : 101 to 420

## Residents

### 1. Alex Yeoh

**badminton**

Male

87438807

Blk 30 Geylang Street 29, #06-40

alexyeoh@example.com

A405

A0123456B

### 2. Bernice Yu

**basketball** **dance**

Female

99272758

Blk 30 Lorong 3 Serangoon Gardens, #07-18

berniceyu@example.com

A405

## Events

### 1. Hall Lunch

01/11/2020 13:00 @ Dining Hall  
Hall-wide lunch event.

**Bernice Yu** **Alex Yeoh**

### 2. Hall Dinner

24/11/2020 18:00 @ Dining Hall  
Hall-wide dinner event.

**Roy Balakrishnan** **Irfan Ibrahim**