

v1.2b Features Demo can be found in Week 9 meeting minutes ([v1.2b Features Demo](#))

Week 4 Meeting 29 Aug 2022 10:00 am - 11:10 am

Attendees: Sean, Caden, Joan, Xinyu, Zhang Yue

Location: Zoom + Google Meet

tP Direction and Idea

Product Name

Teacher's Pet
eduCATor

Decide project direction

Evolve

Target user profile

Small-scale Coding Bootcamp instructors with prior experience in Linux who teach P1-3 students

Problem addressed

Attendance taking and reporting health conditions

[Contacting](#) parents easily through contact management software

Organise students contact into classes, know which student belongs in which class

CRUD the class and schedule

Keep track of the syllabus that will be taught (bullet points), if the syllabus cannot be all covered in the lesson, pushed to the next lesson

Week 5 Meeting 5 Sept 2022 10:00 am - 10:35 am

Attendees: Sean, Caden, Joan, Xinyu, Zhang Yue

Location: Google Meet

First stage:

- **Divide the user stories among team members.**
- **Each member will go through their user stories to discard (e.g., cross out, or move to a different location, but not delete) which are *definitely not needed* for v1.2.**

Aim to create at least 30 user stories. Include all 'obvious' ones you can think of but also look for 'non-obvious' ones that you think are likely to be missed by other competing products.

Second stage:

- **All members discuss the remaining user stories (i.e., the ones not discarded in the first stage), and try to trim the list further.**

BCD Extensions (by Wednesday Night) -> try to be something that is related to our team project

Indicate which extension you are doing, has to be unique

Name	Extension
Joan	B-ViewSchedules C-Statistics
Xinyu	C-Help
Caden	C-BetterSearch
Zhang Yue	C-Undo
Sean	C-Update

Everyone attempts for CI

Each person comes up with 10 user stories through this [link](#) (*raw user stories, refer to below for updated version*)

The user stories after refinement are listed below.

(Priority Legend: H for **High**, M for **Medium**, L for **Low**)

S/N	As a/an ...	I can ...	So that...	Priority
1	tutor who cares about my students' performance	view the scores and grades of each student	I know who to focus on and provide more help	H
2	tutor who doesn't like to use a mouse	have access to all commands using the keyboard	I can stay away from the mouse and type faster	H - basic functionality
3	brand new bootcamp instructor	view all the students under me	i know who is in my class	H - Basic functionality, marking attendance
4	tutor who prefers to mark attendance in alphabetical order	sort student names chronologically	I can view my students in order	H - Can be done together with List. Also, super core to the application since a primary goal is to mark attendance
5	Careless tutor	find duplicate students	I can detect and deal with the same student that is added multiple times	H - Can be implemented together with Add
6	forgetful tutor	keep track of students' contacts	I know which class the student is from when he or she contacts me.	H - core functionality
7	tutor who has to collate attendance	quickly know the number of absentees in a class	i can complete admin work /paperwork faster	H - core functionality
8	Bootcamp instructor	View my schedule	So I know what classes I have that day	H - core functionality, being able to view what classes you have
9	forgetful tutor	view a list of students with attendance marked and unmarked	I can be clear of the status of my students	H - data display (core feature)
10	careless tutor	delete students	I can delete mistakenly added students	H - delete functionality is a core feature
11	lazy tutor	easily import students' contacts from csv file	I do not need to add them manually	H - easy to set up
12	busy bootcamp instructor	mark the attendance by passing all the students names as inputs	i can mark attendance quickly	H - marking attendance is core feature

13	easy-going tutor	view errors messages in a calm and relaxing way (show user what the error is and what he should input instead)	I can be reassured that I can fix the error	H - Try and resolve all known bugs from the start
14	tutor prone to making mistakes	edit the information (student name, class information etc)	make changes	H - update function
15	tutor with a lot of students	use search function	I can find items even if the keyword matches the item only partially	H if involves full search, M if involves partial search
16	busy tutor	retrieve settings and information quickly from my hard drive	I do not need to reconfigure my application and student	M/H - To be done after aliasing commands

			records every time I start up	
17	responsible tutor	quickly check students' status	I can ensure no one is missing out on any lesson.	M - attendance tracking is a core feature, but this is an add on feature
18	tutor with lots of classes	view tasks in the form of a schedule	I can view the schedule for a specific date	M - core to our application but we can't do this so early
19	tutor who wants to evaluate a student's conduct	quickly check how many days of school has a student missed		M - good to have functionality
20	tutor who teaches different year batch eg. p1 and p2	quickly check the syllabus/content of the class I have to teach	i will not get confused and teach the wrong content	M - good to have, so that you can see what you are teaching, relatively important
21	forgetful tutor	use a command to access a help page.	I can easily refer to the commands when i forget them	M - help functionality
22	diligent tutor	extract contacts of the students from the same class	I can send them slides after class and relevant clarifications.	M - not key feature
23	brand new bootcamp instructor	view all the classes that i am assigned to	i know which classes i will be taking	M - not something super important
24	tutor who's prone to making mistakes	undo commands	I can quickly amend my mistakes	M - Sounds rather complex to do
25	programmer who likes to DIY things	Coming up with alias commands in this app	I do not need to memorise the new commands.	L/M - quite good to have since some users can find it to be very useful
26	first-time user	see a help message explaining what commands I could try first	I can start by trying features that are more suited for new users	L - features not finalised and toy data needs to be generated
27	forgetful tutor starting the class	see pictures of students to names	I can easily refer to it when I forget my students name	L - Good to have
28	bootcamp instructor teaching many different kinds of classes (java, C)	view the different classes in my calendar (exporting to .ics)	easily keep track of my classes	L - good to have
29	lazy tutor	use more natural date formats e.g., Mon in a user command can be interpreted as the date	to type faster/more easily	L - good to have

		of the next Monday in the calendar.		
30	bootcamp instructor	add notes for the next lesson (automatically)	i know where i left off my teaching and what i should cover again	L - not priority, good to have
31	bootcamp instructor with many classes	be notified of which is my next class and at what location (zoom or in person)	i do not need to remember everything exactly	L (since 3rd party, need prior approval from CS2103T forum) Notification system is not key

Week 6 Meeting 12 Sept 2022 10:00 am - 11:10 am

Agenda:

- Discuss the project timeline and the division of work over the next few weeks, it should end on week 13(?)
- Do individual TP task (etc website, etc)
- Write distributed part of the User Guide by Wednesday 3-4 meeting

Feature List

V1.2

1. (Joan) Display each student's details -> list [studentName] /details
 - Student ID - int
 - Scores - array of int. Fixed size of 4 for now.
 - Attendance - array of int/String? index 0 is attendance for class with id = 0?
 - Contact - String
 - Classes - array of Classes
2. (Zhang) Display all students in sorted order (names) -> list [className] /students
3. (Caden) Display class details -> list [className] /details
 - Attendance list (same as 2?)
 - Number of absentees
 - Class curriculum
 - Class location
 - Class ID
4. (Xinyu) Manage student's details -
 - Add student record individually
 - Import students' contact in csv format
 - Delete (duplicated) students
 - Include students' exam score
5. (Sean) View schedule
6. (Zhang) Mark/Unmark attendance
 - Mark/Unmark by class (mark all)?
 - Mark/Unmark by student
7. (Sean) Search by: -> search
 - Student ID
 - Student name
 - Class ID

1	tutor who cares about my students' performance	view the scores and grades of each student	I know who to focus on and provide more help	H	Implemented under point 1
2	tutor who doesn't like to use a mouse	have access to all commands using the	I can stay away from the mouse and type faster	H - basic functionality	

		keyboard			
3	brand new bootcamp instructor	view all the students under me	i know who is in my class	H - Basic functionality, marking attendance	Implemented under point 2
4	tutor who prefers to mark attendance in alphabetical order	sort student names chronologically	I can view my students in order	H - Can be done together with List. Also, super core to the application since a primary goal is to mark attendance	Implemented under point 2
5	Careless tutor	find duplicate students	I can detect and deal with the same student that is added multiple times	H - Can be implemented together with Add	
6	forgetful tutor	keep track of students' contacts	I know which class the student is from when he or she contacts me.	H - core functionality	Implemented under point 4
7	tutor who has to collate attendance	quickly know the number of absentees in a class	i can complete admin work /paperwork faster	H - core functionality	Implemented under point 3
8	Bootcamp instructor	View my schedule	So I know what classes I have that day	H - core functionality, being able to view what classes you have	Implemented under point 5
9	forgetful tutor	view a list of students with attendance marked and unmarked	I can be clear of the status of my students	H - data display (core feature)	
10	careless tutor	delete students	I can delete mistakenly added students	H - delete functionality is a core feature	Implemented under point 4
11	lazy tutor	easily import students' contacts from csv file	I do not need to add them manually	H - easy to set up	Implemented under point 4
12	busy bootcamp instructor	mark the attendance by passing all the students names as inputs	i can mark attendance quickly	H - marking attendance is core feature	Implemented under point 6
13	easy-going tutor	view errors messages in a calm and relaxing	I can be reassured that I can fix the error	H - Try and resolve all known bugs from the start	

		way (show user what the error is and what he should input instead)			
14	tutor prone to making mistakes	edit the information (student name, class information etc)	make changes	H - update function	Implemented under 4
15	tutor with a lot of students	use search function	I can find items even if the keyword matches the item only partially	H if involves full search, M if involves partial search	

Week 6 Meeting 14 Sept 2022 3:00 pm - 4:30 pm

Minutes:

- Decided that there is a need to have a module with a class (a session) would inherit from
- Shift the sorted order stuff to a lower priority
- Python basics

Week 7 Meeting 18 Sept 2022 2:00 pm - 3:00 pm

Minute:

- refine existing user stories to simply structure of the application
- improve the user guide according to the updated user stories.

Feature List:

V1.2

- Add (Joan)
 - Student record (be warned i've added duplicates)
- Update (Xinyu)
 - Reschedule my class
 - Student contact details/money paid and owed
 - Teaching notes
- View
 - Which students I have in the day (right side of the view) - Caden
 - Check how much I earn (statistics, top right corner view) - Caden
 - Student's phone number - Zhang
 - Student's home address - Zhang
 - Check students next of kin contact - Sean
 - All students under me - Sean
- Help (Joan)
 - View a list of commands available
- Bye (Joan)
 - Quit and save the application

Overview (stuff we need) -> Fields of the Person class

- Int totalMoneyNot paid;
- Int totalMoneyPaid;
- String notes;
- String nextClassDate: 03-10-2022;
- String startTime: 3pm;
- String endTime: 4pm;
- Contact: email, phone, next of kin's phone;
- Address

Ben Tan

Phone number: 12345678

Next of Kin's number: 23456789

Address: Avenue ABC

Email: ben_tan@gmail.com

Class Date: 3 Dec 2022, 3pm - 4pm

Amount paid: \$150

Amount owed: \$50

Additional notes: abcdef..

// Generate next LocalDateTime

<https://stackoverflow.com/questions/35365571/get-the-next-localdatetime-for-a-given-day-of-week>

New Discussion

- Pivoting to working on a 1-1 coding private instructor, so the Person would be the Class
- If there are recurring classes, there is a use case

S/N	As a/an ...	I can ...	So that...	Priority
1	Tutor who prefers flexibility	Reschedule my class	I can allow the students to more flexibility when arranging for a class	HIGH
2	Meticulous tutor	Add teaching notes	so that I can record any additional information about the student.	HIGH
3	Organised tutor	View which students I have in the day	I know what to expect in the day	HIGH
4	Tutor	update the students contact details	I am able to contact the student or next-of-kin whenever necessary	HIGH
5	Tutor with forgetful students	Check how much a single student owes me	I can remind the student to pay me	HIGH
6	New user	See which commands are available for me to use	I know how to use the application	HIGH
7	Tutor who has many students	Check the students' phone number	I can contact them	HIGH
8	Forgetful tutor	Know the address of the students	Know where I should go to for class	HIGH
9	Tutor who likes a clear overview	View a list of all students	Have a clear view of all students	HIGH
10	Tutor who likes to minimise the number of applications opened on his desktop	Have a way to close the application	Exit the application	HIGH
11	Tutor who has	I can check the	Contact the parents or	HIGH

	many students	students' next of kin's phone number	guardian under certain circumstances	
12	As a tutor who likes to have a clear picture	Have a better UI to view the schedule and the tasks	So that I can view everything in one page	HIGH
13	careless tutor	be warned if I've added an existing student to the student list	I do not have duplicated students in my contact book	MEDIUM
14	Inexperienced tutor at coding	Receive a echo of my command	I know that I've typed the information correctly	MEDIUM
15	forgetful tutor	be warned if the addition of a new session will clash with a previously set session	I can have a peace of mind that my students' sessions do not clash	MEDIUM
16	busy tutor	Sort my students based on the class timing	I know when is my next class	MEDIUM
17	Tutor	search up the students by the class date	I know which students I have on a specific day	MEDIUM
18	Tutor	update the students unpaid session	I can calculate how much money a student owes	MEDIUM
19	Lazy tutor	Use a single command to find multiple students	find multiple students quickly	MEDIUM
20	Lazy tutor	Use a single command to delete multiple students	delete multiple students quickly	MEDIUM
21	Busy tutor	Use shorter commands	I can simplify the commands by using aliases	MEDIUM
22	Tutor who needs to manage many students	view the students he has after or before a certain date	Filter students before/after a certain datetime	MEDIUM/LOW
23	Tutor who wants to do things quick	Delete all students at once	Clear my entire schedule	MEDIUM/LOW
24	Tutor	Find the next available time slot	I do not have to check for clashes each time I schedule the next class	LOW
25	busy tutor	Sort my students based on alphabetical order	I can find the details of my students quicker	LOW
26	Poor tutor	Sort my students based on the amount of money owed	I know who owes the most amount of money so that I can ask for	LOW

			payment.	
27	busy tutor	Add the date of the next class in a more natural format (eg: Next Tuesday)	I do not need to search for the date manually	LOW
28	Forgetful tutor	See tags by students	See necessary info such as #python.	LOW
29	Tutor who loves money	do a summation of how much money earned	Know which students I should follow up with regarding the fees.	LOW
30	Tutor who loves money	Know the total money unpaid by all students	Know the total money unpaid by students	LOW
31	Tutor who wants to keep track of expenses	Check the total amount of money paid by all students	I can check the total amount I have earned	LOW

Week 7 Meeting 19 Sept 2022 5:30 pm - ??? pm

- Generate the tasks for the high-priority user stories

Legend

- Add
- Update
- List
- Delete
- Help
- Bye
- Statistics
- UI

S/N	As a	I can	So that...	Title	Description(optional)
1	Tutor who prefers flexibility	Reschedule my class	I can allow the students to more flexibility when arranging for a class	Add ability to update class date field under Student	Date will be a Datetime in the format YYYY-MM-DD HHmm-HHmm
2	Meticulous tutor	Add teaching notes	so that I can record any additional information about the student.	Add additional notes field Add the ability to update the teaching notes field	Notes are to record any information the tutor wants to tag to the student
3	Organised tutor	View which students I have in the day	I know what to expect in the day	Add the UI to display the schedule for the day Add the Parser to convert data stored into usable format	Day schedule will be displayed permanently on the right side
4	Tutor	update the students contact details	I am able to contact the student or next-of-kin whenever necessary	Add student and NOK phone number fields Add the ability to update phone numbers	Phone number will be of 8 digits, starting with 8 or 9.
5	Tutor with forgetful students	Check how much a single student owes me	I can remind the student to pay me	Add the statistics command	Statistics will appear in the command feedback window
6	New user	See which commands are available for me to use	I know how to use the application	Add pop-out window UI Add prepared link for display	

7	Tutor who has many students	Check the students' phone number	I can contact them	Add the list command Add the phone number field	Phone number will be of 8 digits, starting with 8 or 9.
8	Forgetful tutor	Know the address of the students	Know where I should go to for class	Add the address field Add the method to modify the address field	
9	Tutor who likes a clear overview	View a list of all students	Have a clear view of all students	Add the list command Add the UI to view the list of all students	
10	Tutor who likes to minimise the number of applications opened on his desktop	Have a way to close the application	Exit the application	Add the exit command Add the ability for the application to close	"exit" as input
11	Tutor who has many students	I can check the students' next of kin's phone number	Contact the parents or guardian under certain circumstances	Add the list command Add the next of kin's phone number field	
12	Tutor	have a better UI to view the schedule and the tasks	I can view everything easily in one page	Modify the existing UI to fit our plan	More details in the Figma design
13	-	-	-	Add the ability to delete students from my class list	User must be viewing a list beforehand in order to delete by index

Work distribution:

number refers to the issue number

- Sean (46, 42, 35, 32)
- Xinyu (38, 49, 50, 51, 52)
- Caden (45, 47, 48)
- Joan (34, 40, 44, 53)
- Zhang (43, 41, 36, 33)

Update schedule view to contain these data:

- Name
- Phone
- Address

Week 8 Meeting 3 Oct 2022 10:00 am - 11:00 am

Agenda

- Using consistent callouts
- Having a glossary of common terms, like shortcuts
- How to structure the format
- **Add mark command to issues and user stories**
 - mark [INDEX]
 - Mark under the schedule list
 - UI to show [X] for students who have attended the class today
 - Add attendance field?
 - Implement getID command for schedule list (joan)
 - Checkbox for right panel UI
- Potential to search up the common ways of breaking cs2103t projects? Since likely the previous batches did similar if not the same tp?
- **Deciding which component each person is in charge of? Aka roles**

▼ 📦 Recommended roles and responsibilities

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. Note that being *in charge* of a component doesn't mean you are the only one who should be modifying that component. Rather, you are the one who's expected to, know that component best, review changes done to that component in v1.3-v1.4, act as the gate keeper of its quality, help others when they face difficulties while modifying that component etc.

- **Standardising writing tests? “The weaker the tests are, the higher the risk of bugs”. Write tests immediately when adding a new component?**

Things to note for user guide

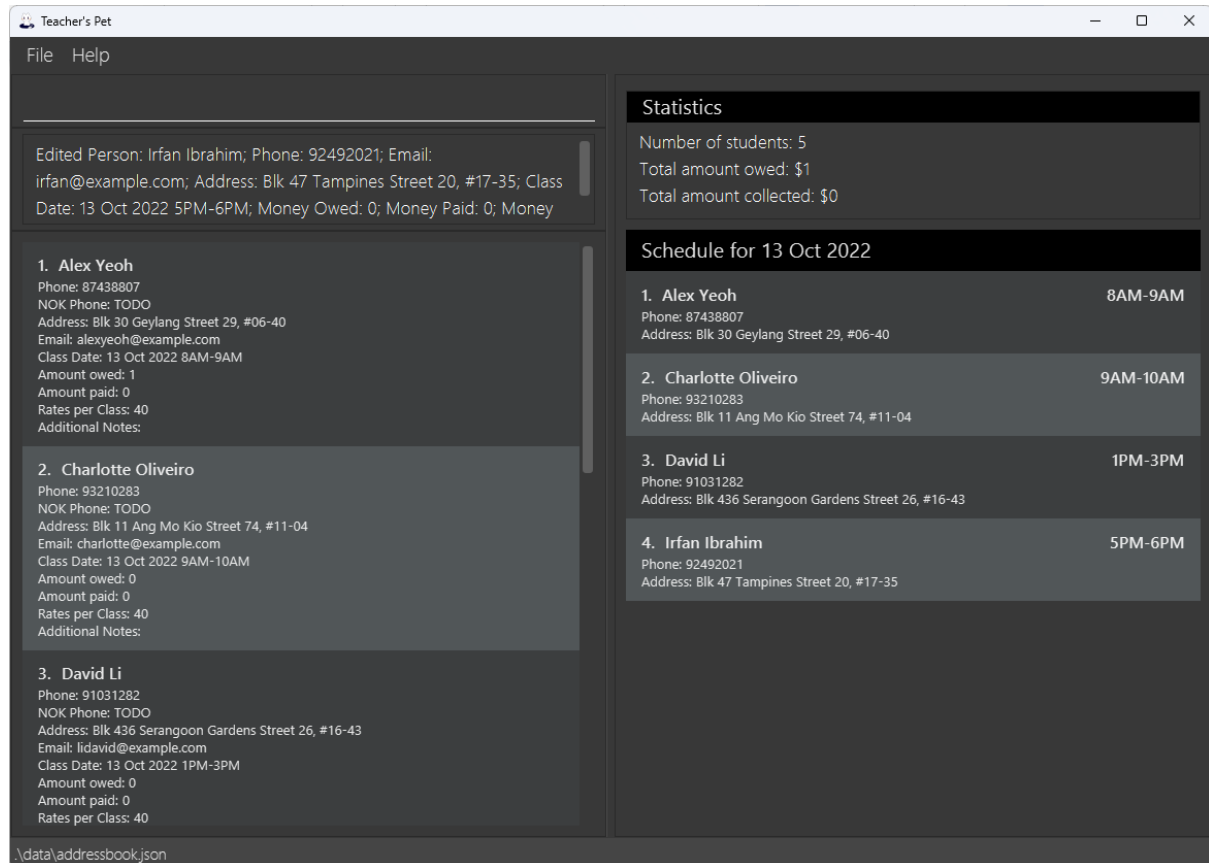
- Add warnings, tips to allow users to better use your user guide
- Adding a sidebar should help with readability

- AB3 is only enough for a pass
- Perhaps look at the GitHub and apple user guide as reference
- For the UI Component for the Developer Guide, it will be good to have code snippets and then have a screenshot of the UI in order to demonstrate what it corresponds to and the structure?
- Adding tutorials to the developer guide?
- Broad philosophy behind it

Week 9 Meeting 10 Oct 2022 10:00 am - 11:00 am

- Review current progress
- Sort out the flow for “mark” command
- Discuss work allocation in the v1.2 and in the future

v1.2 Features Demo



In this v1.2 iteration, we introduced new core features to meet the minimum viable product (MVP) for Teacher's Pet.

The features, on top of the existing AB-3 infrastructure, include:

- A dedicated Schedule list that shows the classes the tutor has for the day.
- An additional Statistics window that calculates financial statistics for the tutor.
- Augmented student's details:
 - Storing the next of kin's phone number for each student
 - Assigning the date and time of the next class for each student.
 - Tracking the amount owed by the student.
 - Tracking the amount paid by the student.
 - Saving the tuition rates the tutor should charge for the individual student.
 - Storing additional notes to keep track of the student's progress.

Schedule List

The screenshot shows the 'Teacher's Pet' application window. It has a menu bar with 'File' and 'Help'. The main area is divided into several sections:

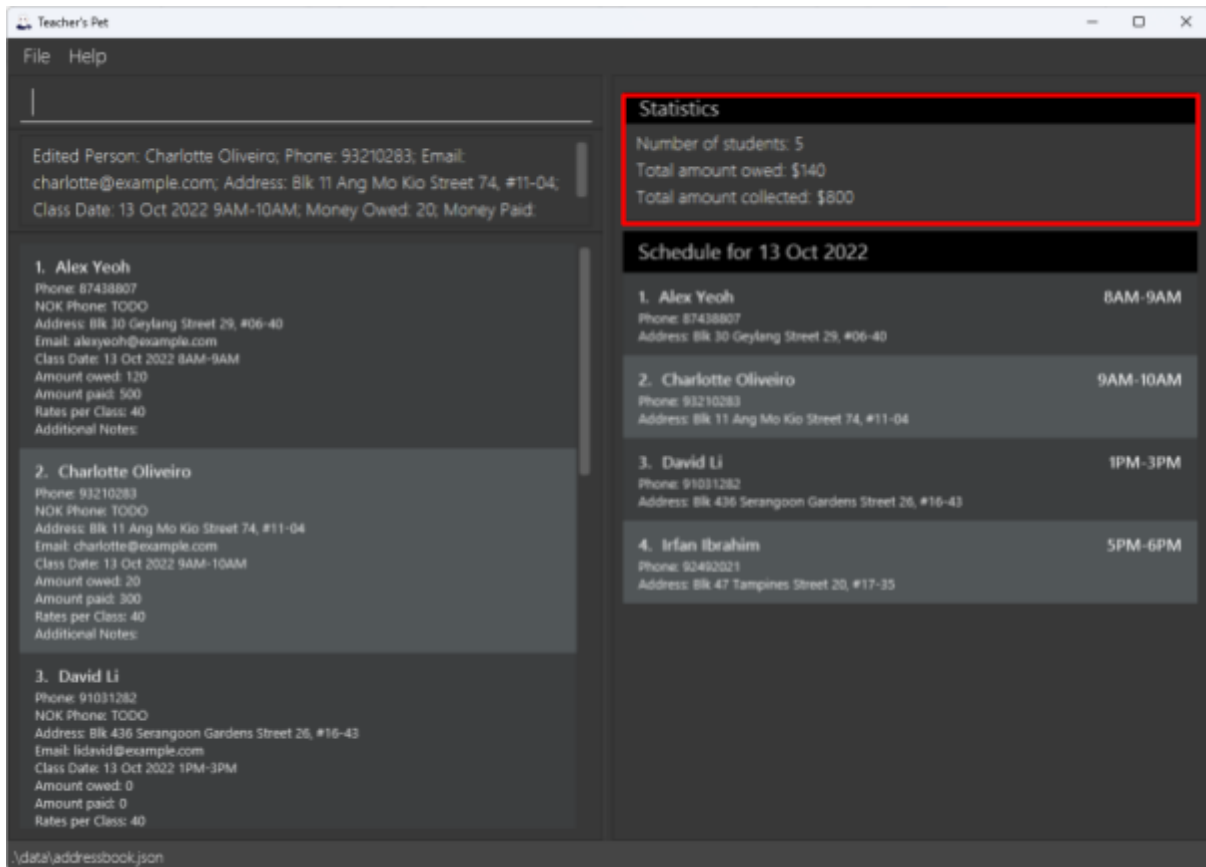
- Statistics:** Number of students: 5, Total amount owed: \$140, Total amount collected: \$800.
- Schedule for 13 Oct 2022:** A list of classes for the day, sorted by time. It is highlighted with a red border.

Student Name	Class Time
1. Alex Yeoh Phone: 87438807 Address: Blk 30 Geylang Street 29, #06-40	8AM-9AM
2. Charlotte Oliveira Phone: 93210283 Address: Blk 11 Ang Mo Kio Street 74, #11-04	9AM-10AM
3. David Li Phone: 91031282 Address: Blk 436 Serangoon Gardens Street 26, #16-43	1PM-3PM
4. Irfan Ibrahim Phone: 82492021 Address: Blk 47 Tampines Street 20, #17-35	5PM-6PM
- Edited Person:** Charlotte Oliveira; Phone: 93210283; Email: charlotte@example.com; Address: Blk 11 Ang Mo Kio Street 74, #11-04; Class Date: 13 Oct 2022 9AM-10AM; Money Owed: 20; Money Paid.
- Student List:** A list of students with their details, including name, phone, NOK phone, address, email, class date, amount owed, amount paid, rates per class, and additional notes.

The status bar at the bottom shows the file path: \data\addressbook.json.

- The schedule list can be found at the bottom right section of the application.
- The schedule list only shows students having classes on that day.
- The schedule list is sorted based on the class timing.
- The schedule list will be automatically updated upon any changes to the student list.
- It shows key information the tutor needs to know about the class, such as:
 - Name of student
 - Time of class
 - Phone number of student
 - Address of student

Statistics Window



- The statistics window can be found at the top right corner of the application.
- The statistics window will be automatically updated upon any changes to the student list.
- The statistics window calculates key statistical values for the tutor, such as:
 - Total number of students in the student list
 - Total amount of money owed across all students
 - Total revenue made from all students

Augmented Student's Details

The screenshot shows a web application titled "Teacher's Pet" with a dark theme. It features a sidebar on the left with a "File" and "Help" menu. The main content area is divided into two columns. The left column displays "Edited Person" details for Charlotte Oliveira and a list of three students: Alex Yeoh, Charlotte Oliveira, and David Li. The right column shows "Statistics" (Number of students: 5, Total amount owed: \$140, Total amount collected: \$800) and a "Schedule for 13 Oct 2022" table. The student details for Alex Yeoh, Charlotte Oliveira, and David Li are highlighted with a red border. The bottom of the page shows the file path "\data\addressbook.json".

Teacher's Pet

File Help

Edited Person: Charlotte Oliveira; Phone: 93210283; Email: charlotte@example.com; Address: Blk 11 Ang Mo Kio Street 74, #11-04; Class Date: 13 Oct 2022 9AM-10AM; Money Owed: 20; Money Paid:

1. Alex Yeoh
Phone: 87438807
NOK Phone: TODO
Address: Blk 30 Geylang Street 29, #06-40
Email: alexyeoh@example.com
Class Date: 13 Oct 2022 8AM-9AM
Amount owed: 120
Amount paid: 500
Rates per Class: 40
Additional Notes:

2. Charlotte Oliveira
Phone: 93210283
NOK Phone: TODO
Address: Blk 11 Ang Mo Kio Street 74, #11-04
Email: charlotte@example.com
Class Date: 13 Oct 2022 9AM-10AM
Amount owed: 20
Amount paid: 300
Rates per Class: 40
Additional Notes:

3. David Li
Phone: 91031282
NOK Phone: TODO
Address: Blk 436 Serangoon Gardens Street 26, #16-43
Email: lidavid@example.com
Class Date: 13 Oct 2022 1PM-3PM
Amount owed: 0
Amount paid: 0
Rates per Class: 40

Statistics

Number of students: 5
Total amount owed: \$140
Total amount collected: \$800

Schedule for 13 Oct 2022

1. Alex Yeoh Phone: 87438807 Address: Blk 30 Geylang Street 29, #06-40	8AM-9AM
2. Charlotte Oliveira Phone: 93210283 Address: Blk 11 Ang Mo Kio Street 74, #11-04	9AM-10AM
3. David Li Phone: 91031282 Address: Blk 436 Serangoon Gardens Street 26, #16-43	1PM-3PM
4. Irfan Ibrahim Phone: 92492021 Address: Blk 47 Tampines Street 20, #17-35	5PM-6PM

\data\addressbook.json

- Each student takes in vital data for the tutor to know, they include:
 - Name of student
 - Phone number
 - **Next of kin phone number**
 - Address of student
 - Email of student
 - **Date and time of next class**
 - **Amount owed**
 - **Amount paid**
 - **Rates per class**
 - **Additional notes**
- **Next of kin phone number**
 - This stores the phone number of the next of kin of the student.
 - This field must be a number containing at least 3 digits.
- **Date and time of next class**
 - The date and time of the next class can be stored in each student, reminding the tutor about the appointment when the day comes.

- The date and time assigned cannot overlap with another class timing. If done so, our application will alert the tutor.

edit 2 dt/2022-10-13 0800-0900

There is a conflict between the class timings.

- The schedule list pulls the date and time of the class from here.

- **Amount owed**

- This stores the amount of money owed by the individual student.
- Amount owed must be a non-negative integer.
- The statistics window uses this amount to sum the total amount owed across all students.
- Default amount owed per student is \$0.

- **Amount paid**

- This stores the amount of money paid by the individual student.
- Amount paid must be a non-negative integer.
- The statistics window uses this amount to sum the total amount paid across all students.
- Default amount paid per student is \$0.

- **Rates per class**

- As each student belongs to a different age group, they may have different tuition rates.
- Rates per class must be a non-negative integer.
- Default rates per class is set to \$40.
- More functions will be added to utilise this attribute in future milestones.

- **Additional notes**

- To keep track of a student's progress, it is wise for the tutor to note down important notes for future references.
- There is no limit in the length of the notes.

1. Alex Yeoh

Phone: 87438807

NOK Phone: TODO

Address: Blk 30 Geylang Street 29, #06-40

Email: alexyeoh@example.com

Class Date: 13 Oct 2022 8AM-9AM

Amount owed: 120

Amount paid: 500

Rates per Class: 40

Additional Notes: Gone through arrays and tree search algorithms with Alex. He is a fast learner and can manage with the contents taught. To focus more on time complexity in the next lesson.

Thoughts from Ms Amy:

- Show available time slots (How do I know what time slots I have available?)
- What class are you teaching? One way is to narrow our target user to one specific topic e.g. Python Programming; other is to add a field “subject” for each student.
- Filter: I

Week 9 Meeting 13 Oct 2022 10:00 pm - 1:10 am

- refine and update the user stories for v1.3 and v1.4
- discuss the procedures of the implementation of user stories
- allocate the work for v1.3 for fair share across each member

1	careless tutor	be warned if I've added an existing student to the student list	I do not have duplicated students in my contact book	
2	Inexperienced tutor at coding	Receive a echo of my command	I know that I've typed the information correctly	
3	forgetful tutor	be warned if the addition of a new session will clash with a previously set session	I can have a peace of mind that my students' sessions do not clash	
4	busy tutor	Sort my students based on the class timing	I know when is my next class	
5	Tutor	search up the students by the class date	I know which students I have on a specific day	
6	Tutor	update the students unpaid session	I can calculate how much money a student owes	
7	Lazy tutor	Use a single command to find multiple students	find multiple students quickly	
8	Lazy tutor	Use a single command to delete multiple students	delete multiple students quickly	
9	Busy tutor	Use shorter commands	I can simplify the commands by using aliases	
10	Tutor who needs to manage many students	view the students he has after or before a certain date	Filter students before/after a certain datetime	
11	Tutor who wants to do things quick	Delete all students at once	Clear my entire schedule	
12	Tutor	Find the next available time slot	I do not have to check for clashes each time I schedule the next class	Constrained between 8am-6pm? Pass in duration of class. Can consider linking it to the stored user preference? Tho that part is more of the UI setting

				so hmmm
13	busy tutor	Sort my students based on alphabetical order	I can find the details of my students quicker	
14	Poor tutor	Sort my students based on the amount of money owed	I know who owes the most amount of money so that I can ask for payment.	
15	busy tutor	Add the date of the next class in a more natural format (eg: Next Tuesday)	I do not need to search for the date manually	
16	Forgetful tutor	See tags by students	See necessary info such as "python" and "intermediate"	Tags for curriculum and difficulty level
17	Tutor who loves money	do a summation of how much money earned	Know which students I should follow up with regarding the fees.	Have a graph / pie chart
18	Tutor who loves money	Know the total money unpaid by all students	Know the total money unpaid by students	
19	Tutor who wants to keep track of expenses	Check the total amount of money paid by all students	I can check the total amount I have earned	

Week 10 Meeting 17 Oct 2022 2:00 pm - 4:00? pm

Do a postmortem of the previous iteration:

Overall workload is still manageable, more time spent on implementing test cases.

When there is a PR, always update your local master by pulling from upstream repo to avoid the weird merge conflict errors that Xinyu mentioned.

Try and approve and merge PR in 2 days, assign people with relevant experience to approve PR.

~~Clarify in tutorial about the "Overzealous input validation"~~ Clarifying with TA on Teams on 17/10/2022

Better to add git commit messages

Add more assertions in the code

If the branch is related to an issue, use the format, Use dashes

issueNumber-some-keywords-from-issue-title e.g., 1234-ui-freeze-error

Try and finish all the issues by a earlier date then fix the documentation

Part to add to the DG:

<https://se-education.org/addressbook-level3/DeveloperGuide.html#proposed-undo-feature>

- The description can contain things such as,
 - How the feature is implemented (or is going to be implemented).
 - Why it is implemented that way.
 - Alternatives considered.

Caden: Statistics window

Joan: Edit class

Xinyu: money-relevant field specification and Sort command

Sean: find command; avail command; ui panel

Zhang yue: phone relevant? multiple input command

Individual task:

Write DG according to the task above

Smoke catcher

Week 11 Meeting 24 Oct 2022 9:00 pm - 10:00 Pm

User Guide

Things to add:

Add [nok phone] Z

Add [tag] Z

Edit rates - XY

Edit notes (append) - Joan

Edit tag Z

Edit class in more natural format - Joan

Delete multiple students Z

Mark - Caden

Unmark or Undo - XY

Pay command - Caden

Find

- Name - Caden
- Email Z
- Address Z
- Phone number - Caden
- NOK Sean
- Students attending a particular class date - Joan
- Tag (Sean)

Avail (Sean)

Sort by (XinYu)

- Class timings
- Alphabet order
- Money owed

Developer Guide

Mark - Caden

Avail - Sean

Find/Delete multiple students - Z

Use case for

- Edit class
- Find students attending a class date

V1.3 FEATURES DEMO

Pay Command

- Indicates that a student has paid a certain amount.
- Format: pay INDEX-s AMOUNT_PAID, e.g. pay 1 200.
 - This will deduct the amount owed of the student
 - This will increase the amount paid of the student

Example of executing *pay 1 40*

```
1. John Doe
friends owesMoney
Phone: 98765432
Next of Kin Phone: 89705432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Class Date: 26 Oct 2022 10AM-7PM
Amount owed: $280
Amount paid: $20
Rates per Class: $40
Additional Notes:
```

```
1. John Doe
friends owesMoney
Phone: 98765432
Next of Kin Phone: 89705432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Class Date: 26 Oct 2022 10AM-7PM
Amount owed: $240
Amount paid: $60
Rates per Class: $40
Additional Notes:
```

Edit Notes (Append) Command

- Append the notes to the current notes of a student.
- Format: edit INDEX nt-a/
 - This allows the tutor to add extra notes without re-typing the existing notes.

Edit Class (Flexible Format) Command

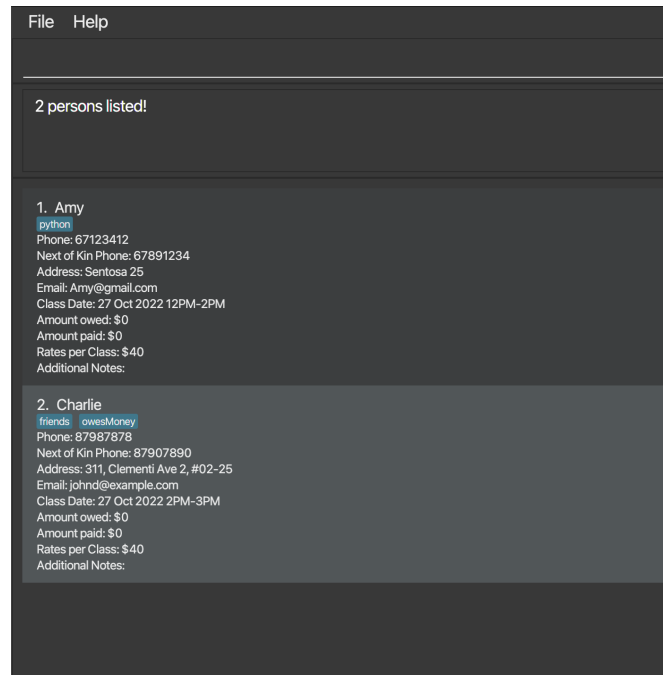
- Edits the class date of a student.
- Format: edit INDEX dt/Day-of-Week 0000-2359
 - This allows the tutor to edit class date with a more flexible and natural format.
 - Day-of-Week is a 3-letter abbreviation and is case-insensitive.
 - e.g. MON, mon, Mon

Find Command

- Find students based on
 - Name
 - Contact Number
 - Next-of-Kin's Number
 - Email
 - Address

- Class Date
- Tags
- Format: find [n/NAME] [p/CONTACT_NUMBER]
[np/NEXT_OF_KIN_CONTACT_NUMBER] [e/EMAIL] [a/ADDRESS]
[dt/CLASS_DATE] [t/TAGS]
- Only 1 parameter can be searched.

Executing *find dt/2022-10-27*



Mark Command

- Marks a student present
- Format: mark INDEX-s
 - This will increase the amount owed of the student by the class's rate.
 - If the student owes any money, he/she will be highlighted in red under the schedule list.
 - This will automatically set the class date to 7 days later at the same timing if and only if there will not be any class conflicts (no other students occupy the class slot).

Example of executing *mark 1*

File
Help

Student has been marked as present

1. John Doe

friends

owesMoney

Phone: 98765432
Next of Kin Phone: 89705432
Address: 311, Clementi Ave 2, #02-25
Email: john@example.com
Class Date: 02 Nov 2022 12PM-2PM
Amount owed: \$40
Amount paid: \$0
Rates per Class: \$40
Additional Notes:

2. Amy

python

Phone: 67123412
Next of Kin Phone: 67891234
Address: Sentosa 25
Email: Amy@gmail.com
Class Date: 27 Oct 2022 12PM-2PM
Amount owed: \$0
Amount paid: \$0
Rates per Class: \$40
Additional Notes:

Statistics

Number of students: 2
Total amount owed: \$40
Total amount collected: \$0

Schedule for 26 Oct 2022

1. John Doe - To collect \$40

12PM-2PM
[X]

Address: 311, Clementi Ave 2, #02-25
Phone: 98765432

Sort Command

- Sort students based on
 - Name
 - Class Date
 - Money Owed
- Order can be
 - Ascending (ASC)
 - Descending (DESC)
 - If order is not specified, by default it will be ascending for name and class date, and descending for money owed.
- Format: sort [NAME] [CLASS] [OWED] followed by [ASC] [DESC]
 - Exactly 1 parameter and at most 1 order must be specified

Executing *sort NAME*

Sorted all students with the given order

1. Amy

[python](#)
Phone: 67123412
Next of Kin Phone: 67891234
Address: Sentosa 25
Email: Amy@gmail.com
Class Date: 27 Oct 2022 12PM-2PM
Amount owed: \$0
Amount paid: \$0
Rates per Class: \$40
Additional Notes:

2. Ben

[friends](#) [owesMoney](#)
Phone: 80018908
Next of Kin Phone: 89705432
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Class Date:
Amount owed: \$0
Amount paid: \$0
Rates per Class: \$40
Additional Notes:

3. Charlie

[friends](#) [owesMoney](#)
Phone: 87987878
Next of Kin Phone: 87907890
Address: 311, Clementi Ave 2, #02-25
Email: johnd@example.com
Class Date: 27 Oct 2022 2PM-3PM
Amount owed: \$0

Week 12 Meeting 31 Oct 2022 10:00 am - 11:00? Am

Meeting minutes:

- Discuss issues specified in GitHub issues