



VICTORIA UNIVERSITY OF
WELLINGTON
TE HERENGA WAKA

School of Engineering and Computer Science
Te Kura Mātai Pūkaha, Pūrorohiko

SWEN 303 (2021)

User Experience Engineering

**Assignment 2 - Design
Flat Finances**

Qing (Cecilia) Lu

Student ID: 300363602

Part 1: Introduction

Description - Flat Finances

- **Describe existing solutions**

Some corporations developed some tools and systems to help students with their flat finances. The *Live sorted* is a good example. The advantages of the system include:

- It allows students to manage their flat finance in many aspects, such as paying rent, shopping together, utility bills, etc.
- It is easy to get access to the system, as it is a webpage-based system.
- Also, it provides the *sorted budgeting tool* to help with future cost planning. This budgeting tool enables users to add master categories, which supports customisation for different users.

However, the system also has some disadvantages, e.g.:

- The budgeting tool doesn't support persistent recording. It doesn't properly save the data for later use. In this case, users have to fulfil the forms every time they want to budget.

I didn't find a mobile app that is designed specifically for students' flatting finance. However, some existing apps can also achieve some functions that a flatting student would use. For example:

| Name of App | Advantages | Disadvantages |
|--|--|---|
| <i>Spendee</i> | It is a free budgeting app that records expenses, shows financial habits and analyses expenses. It helps to stick to financial goals. | It is not a flatting-focused financial app, leading to a lack of some specific functions for students' shared flatting costs. |
| <i>Fudget - Budget Planner Tracker</i> | It has a simple user interface without too many buttons (too many buttons may confuse users). It enables the user to record simple lists of expenses, and keep track of the balance. | Because it is simpler than other apps, it doesn't provide the function of adding new categories of expenses. It doesn't show any diagrams either. In addition, it doesn't analyse future costs. |

- **Business Objectives**

As a software development corporation, we intend to develop a student flatting finance system (SFFS) that would help students to manage their fixed and shared flatting costs. (We will pay for the system development.) In this way, we can collect students' flatting costs behaviour data. Then we will gain profit by selling this data to companies or organisations that want to advertise to students.

The more students who use the SFFS, the more data we can collect, the more business clients (e.g., advertisers) we are likely to have. Therefore, we hope as many students as possible could use the SFFS. Thus, the SFFS needs to meet the flatting-finance-management requirement of almost all kinds of flatting students. Also, it must be simple to use — a nice and clean interface without too much confusion. Hence, we can do interviews with some flatting students and explore their goals, requirements, and preferences for system designing. In this way, we hope to increase the number of users.

We can attract advertisers by providing them with useful information. Thus, the SFFS needs to explore students' purchase and consumption habits. In this case, it should categorise the costs, record consumption frequency, and estimate future budget. Therefore, the SFFS needs to enable students to set up their budget, classify their existing costs (add new categories when necessary), show the balance, and analyse future costs based on their previous costs in all categories in a specific time period.

In order to make the SFFS sustainable, we should also enable students to send their feedback to us, which helps us to develop the SFFS better.

In sum, the main business objectives of this flat finance system are:

- Increase the number of users;
- Collect consumption behaviour data from flatting students;
- Sell data to companies that want to advertise to students, and gain profit.

• **Importance to Stakeholders**

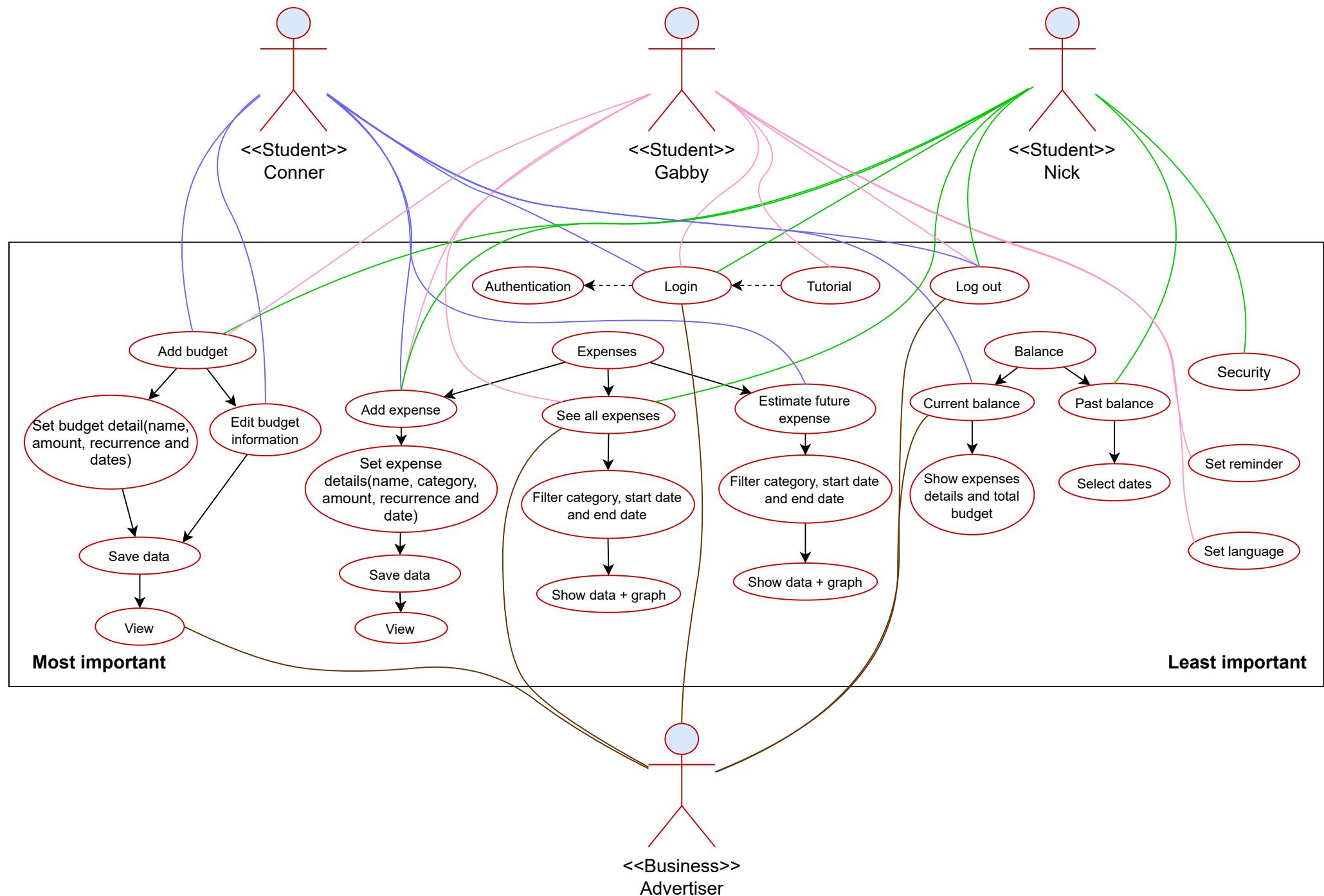
There are two main groups of stakeholders: flatting students, and businesses that want to gain the consumption data of students.

This system is designed specifically for university students who live in a shared flat. It helps students to gain independence in terms of financial management. Managing all of the costs in mind (or on paper) can be daunting. This flat finance system is a great "financial assistant" for those students. It allows students to classify their costs, keeps track of their money, and shows the balance. In addition, it roughly estimates students' future costs in different categories in order not to get caught out by flatting costs. Therefore, the system, as a warning of over-consuming, is vital to flatting students who want to decrease the time spent on managing their flatting expenses efficiently.

The SFFS is a great way of exploring the flatting finance situation of students. Businesses and advertisers can gain students' consumption behaviours, budget, balance, etc. Based on students' different preferences and spending power, those businesses can release different advertisements that students might be interested in, such as grammar software, discount information of clothes shops, new arrivals of books, job hunting platforms, and so on. By pushing proper advertisements, they can increase their awareness and maximise their performance-marketing.

Model Overview

Model graph:



Describe the process of creating the persona model:

- Identify persona generation techniques that I used:**

Qualitative approaches: Conduct user research; Condense the research; Refine; Make them realistic; and Brainstorm.

- Describe the use of the techniques:**

Firstly, I conducted user research. I started with thinking about the targeted users — flatting students who are keen to manage their flatting costs for some reasons (i.e., those who have limited flatting budget, those who want to manage shared costs, or those who are more concerned about over-consume and love to plan their money better). Then I decided to do interviews with some of my friends who are now living in student flats. I select both boys and girls with different backgrounds; and ask them why they are using the SFFS, their behaviours, goals, and expectations.

Secondly, I condensed the research into characteristics to build prototypes of the users that I can talk to. Firstly, I found three photographs (2 boys and 1 girl) as visual representations of the persons that I can attach mental models to. Then I brainstormed three different groups of personas that cover the main types of flatting students; and listed their background, behaviours, personalities, and requirements.

Thirdly, based on the brainstormed information, I refined the characteristics and separated them into three categories with identified traits:

- **Conner Rutherford**, a local student who is doing a part-time job to support his hobbies,
- **Gabby Kim**, a wealthy international student who has to report bills to parents,
- **Conner Angus**, a studious student whose source of flatting expenses is academic scholarships.

By looking at the photographs and main traits of the personas, I tried to imagine to be in their heads to make them realistic. I picked reasonable background, features, and goals, and combine them to make the personas close to real persons.

- Describe the process of creating the tasks:**

Some tasks are designed before the interviews. I downloaded existing budgeting apps and got some ideas about the tasks that the apps support. I listed some of the tasks that students might be attracted by, i.e., record costs, view past costs, show current balance, etc. Then I included them in my interview questions (in order to ask them whether they need those functions).

Some tasks are mentioned by interviewees. For example, one student told me that he expected the system to enable him to select the cost category and time period; another student wished to get an idea about her future flatting costs and balance so that she could plan her extra money; while several students cared about their privacy.

Some tasks are created after the interview. Considering not everyone is good at spatially imagining the cost patterns, a simple way of visualising statistical information is charts and graphs. Therefore, the system needs to show the percentage of costs in different categories in pie charts; and show the trends of costs in line graphs.

- **Describe how to prioritise these tasks:**

1. **Budget:** Budget is fundamental to all functions. No one has an unlimited budget. The budget is the *minuend* of all costs. The SFFS helps students from spending money beyond their budget. For this task, recording budget information is compulsory; while editing budget information is only needed when necessary.
2. **Costs:** Managing costs is the task that all interviewees need. Therefore, it follows budget recording. The priority of sub-tasks is:
 - **Record costs:** After setting the budget, students need to record their costs within budget, because These costs are the original data for all types of data analysis, such as estimating future costs and balance, and analysing consumption behaviours.
 - **View past costs:** Expenses that have been incurred are the most objective data for users to get a picture of their consumptions. This helps to keep track of existing costs.
 - **View future costs:** This task saves users' calculation time and is provided for reference. It has a lower priority than "view past costs" because future costs are just rough estimated values calculated based on existing costs. Therefore, it is less objective than the past cost.
3. **Balance:** After managing the costs, the SFFS needs to show the balance to users:
 - **Show current balance:** All users care about their balance. Showing balance enables users to make sure they have enough money to pay for future bills.
 - **View past balance:** This task shows the balance trend, which only assists in understanding consumption speed.
4. **Password** and **Set reminder:** These tasks are only for personalisation. They are not the core task of the SFFS, so they have the least priority.

Part 2: Personas and Scenarios

Conner Rutherford

Connor is a 21-year-old domestic student. He is a second-year Law student at Victoria University of Wellington, and has been living with two flatmates at Stafford House (one of the accommodations in Wellington) for a year. He does a part-time job at the supermarket. He has to pay his own way for pretty much everything. Because he has a limited living budget, he has been used to recording all of his costs and limiting his expenses depending on his balance. Thus, he will use the student flatting finance system to keep track of his money and plan for future costs. When using this system, his goal is to make sure that he doesn't over-consume so that he has enough money to support his future life.



Activities: Conner likes to study at the flat or in the library most of the time. He plays basketball with his friends at the gym every Sunday afternoon. He also likes to play computer games and console games.

Attitudes: Conner is a planner. He plans his study schedule as well as his student hall finances. Therefore, Conner cares about every payment he made during his University study.

Aptitudes: Conner is athletic and intelligent. He is good at quickly adapting to a new environment.

Weaknesses: Conner is not good at Math.

Domain Knowledge: Conner has no experience with similar flatting finance systems.

System Knowledge: Conner is not familiar with the system, due to the lack of experience with it in his first year of flatting.

Interaction: Conner will interact with the SFFS when there are new payments related to the student flat or money spent on goods or services (At least once a day).

Priorities: Conner wants to have a good picture of his past costs and current balance. He needs the data clearly presented to him.

Motivations: Conner uses the SFFS to carefully plan his spending each month. He and his flatmates take turns to cook for all of them and they share meals. Every Sunday afternoon, they go to the supermarket and prepare materials for the coming whole week. Conner often checks out; and other flatmates will later pay Conner back based on the items they bought. The SFFS helps Conner to determine how much money his flatmates need to transfer to him. Apart from this, the SFFS also calculates his total spending and estimates future costs. This saves a lot of time and effort considering Conner is not very good at Math.

Computer self-efficacy: Conner has used lots of software and he is familiar with different kinds of interfaces. He always gets started with a new software quickly. Therefore, he is quite confident with handling the SFFS. Based on his experience with other software, Conner can solve most of the questions relating to SFFS by himself.

Risk tolerance: Conner has low-risk tolerance. He prefers the system he uses to be safe, simple and reliable.

Information Processing: Conner is good at absorbing information as his course is really information-heavy.

Goal: Conner hopes to finish his degree without running out of money. Therefore, he needs a finance system that shows him current balance and estimated future costs so that he can plan his part-time jobs to earn enough money to meet his requirement in advance.

Gabby Kim

Gabby is a 19-year-old international student (from Korea) studying at Victoria University of Wellington. She is a first-year Bachelor's student majoring in English Literature. This is her first year studying and living in New Zealand. Gabby has a wealthy family and her parents pay for her living expenses. There is no restriction on her spending but her parents want to keep track of her money. Because Gabby has no experience in managing her spendings, the SFFS is good assistance for her to record the expenses. Her goal is to complete the recording process as easily as possible.



Activities: Gabby goes to lectures from Monday to Friday. She usually studies in the library before the deadline of the assignments. In her spare time, she likes to go shopping in clothing stores and makeup stores. She also likes to drink coffee and try all kinds of restaurants in Wellington. She goes to a party every Friday night.

Aptitudes: Gabby is a humble person. She always takes other people's good advice. She is friendly and generous. She likes to share her life stories with people. Thus, she has lots of friends.

Weaknesses: Gabby has a poor memory. She is not good at memorising numbers. In addition, she got relatively poor marks for math and statistics during high school.

Domain Knowledge: Gabby has no experience with similar flatting finance systems.

System Knowledge: She is not familiar with the finance management system because this is her first year living independently. Her parents used to take care of her daily life and prepare everything she needed before she came to New Zealand.

Interaction: Gabby tends to record her expenses every time she makes the payment.

Priorities: Gabby wants the system to be easy to use and can remind her of recording the costs each day.

Motivations: Gabby is happy to record her costs and let her parents know the detailed information, because, in this way, her parents would know how much they need to transfer to Gabby in advance.

Computer self-efficacy: Gabby is not very good at electronic devices and software. She is not confident that she can use the system correctly and efficiently.

Risk tolerance: Gabby has low-risk tolerance. She prefers the system she uses to be safe, simple, and reliable.

Information Processing: Gabby is not very comfortable with handling a lot of information. She prefers to only handle the most important information.

Family: Gabby is a lucky girl. She has a happy family. Her parents trust her and always support her decision mentally and financially. She has been used to have her parents keep track of her money and make her budget.

Nick Angus

Nick is a 25-year-old international student from South Africa. He is doing his second year of Master's study majoring in Architecture at Victoria University of Wellington. His main source of revenue is his Master's scholarship. Because he is planning to continue his study and do a Ph.D. degree next year, he spends most of his time doing his academic research. So far, he has published 2 journal papers. In this case, he doesn't have extra time to do a part-time job. Therefore, he is keen to manage his expenses carefully.



Activities: Nick studies in the Master Room at University from Monday to Saturday. He loves his major and enjoys doing academic research. Sometimes he joins academic conferences with his supervisor. He always takes a break every Sunday afternoon. Usually, he goes to the gym with his flatmates every Sunday afternoon. Then they take the bus and go to Pak'nSave to prepare the whole week's foods and drinks.

Aptitudes: Nick is very studious. He never hesitates to do research on things he is interested in or things that can help him improve his life.

Weaknesses: Nick is not very good at socialisation. His friends help him to overcome it by taking him to parties sometimes.

Domain Knowledge: Nick has experience with similar finance management systems. He has been using a free budgeting app named *Spendee* for more than four years.

System Knowledge: Nick is familiar with finance management applications, because he started to manage his money in the first year of his Bachelor's degree. He wishes to have a finance management application that is designed specifically for flatting, as most of the existing applications have lots of functions that he doesn't need, and lack the task of estimate future costs.

Interaction: Nick has the habit of recording every payment and checking his financial statement and balance everyday before going to sleep.

Priorities: Nick hopes the system could have a user-friendly interface. Also, he expects the SFFS could enable him to set a password, as he cares about personal privacy.

Motivations: Nick's scholarship provided a limited amount of money. It is only for basic daily living. Therefore, Nick is happy to use this system because it helps him to keep track of his money and plan his future costs within his limited budget.

Computer self-efficacy: Nick is good at electronic devices and software.

Risk tolerance: Nick has low-risk tolerance. He prefers the system he uses to be safe, simple and reliable.

Information Processing: Nick is good at processing information. He likes to show data in tables, charts, or graphs. Therefore, he prefers the information is shown in tables, charts and graphs as well.

Scenarios, Use cases and Journey

1

| All users | Set a budget |
|---------------------------|---|
| User Intention | System Requirements |
| Click 'Create new budget' | |
| | Ask for budget information: name, amount, recurrence and start date |
| Fill in information | |
| Click 'Save' | |
| | Confirm and show budget |

2

| All users | Record a cost |
|-------------------------|--|
| User Intention | System Requirements |
| Click 'Add transaction' | |
| | Ask users to fulfill the form: transaction category, name, date, amount of money, and recurrence |
| Fill transaction detail | |
| Save data | |
| | Confirm transaction |
| | Cut money from budget |
| | Show current balance on the bottom |

3

| Gabby, Nick | View past costs |
|-----------------------------|---|
| User Intention | System Requirements |
| Click 'View past costs' | |
| | Display past costs sorted by time (default: from the newest) |
| Click 'Filter by category' | |
| | Display the drop-down menu of categories |
| Select category 'Foods' | |
| | Display all spendings only in the category 'Foods' |
| Click 'Filter by time' | |
| | Display a calendar menu |
| Select 'From' and 'To' date | |
| | Display all spendings only in the target category 'Foods' and selected timeline |

4

| Conner | Show current balance |
|----------------------|---|
| User Intention | System Requirements |
| Click 'Show balance' | |
| | Calculate balance |
| | Display balance |
| Click 'Show details' | |
| | Show total budget, spending details and generate a pie chart to demonstrate the proportion of all spendings and current balance |

5

| Conner | Estimate future costs |
|----------------------------------|--|
| User Intention | System Requirements |
| Click 'Estimate next month cost' | |
| | Check past monthly costs |
| | Past monthly costs found |
| | Calculate future monthly costs based on past costs |
| | Display future monthly costs and a pie chart showing the estimated proportions in all categories |

6

| Nick | Show past balance |
|-----------------------------|--|
| User Intention | System Requirements |
| Click 'Show balance' | |
| | Calculate balance |
| | Display balance |
| Click 'Show past balance' | |
| | Display calendar menu |
| Select 'From' and 'To' date | |
| | Display balances during the selected period and generate a line graph to demonstrate balance changes |

7

| Conner | Edit budget information |
|-----------------------|--|
| User Intention | System Requirements |
| Click 'Target budget' | |
| | Display budget |
| Click 'Edit budget' | |
| | Show current budget information(name, amount, recurrence and start date) |
| Change 'Amount' | |
| Click 'Save' | |
| | Confirm and show new budget |

8

| Gabby | Set a reminder |
|----------------------|--|
| User Intention | System Requirements |
| Click 'Setting' | |
| | Display Setting menu |
| Click 'Set reminder' | |
| | Ask for reminder time and recurrence |
| Fill in information | |
| Click 'Save' | |
| | Save the reminder time and recurrence. |
| | Activate the alarm |

| Nick | Security |
|---------------------------------|------------------------------------|
| User Intention | System Requirements |
| Click 'Setting' | |
| | Display Setting menu |
| Click 'Security' | |
| | Display Security menu |
| Click 'Always ask for password' | |
| | Change system setting |
| | Require password in the next login |

User Journey:

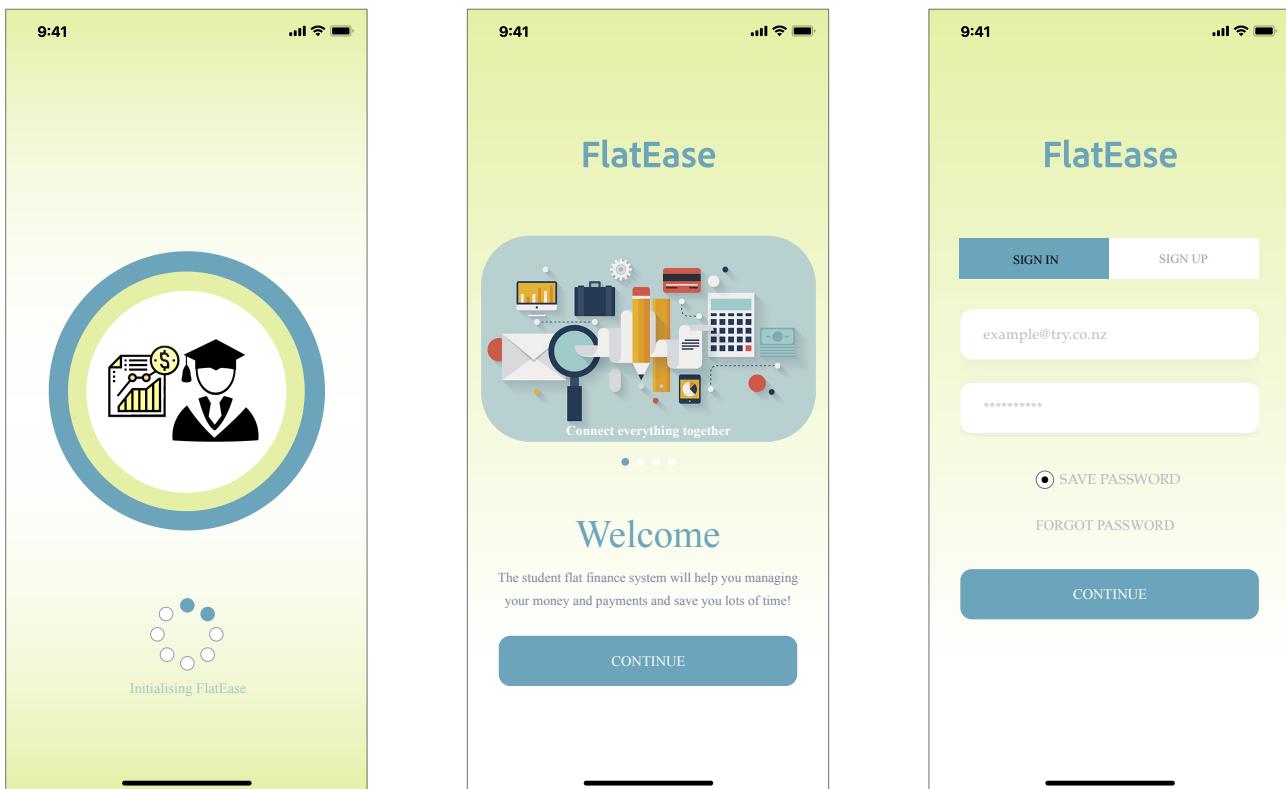
| Conner - Edit budget - Set new budget successfully | | | |
|--|--|---|---|
| | Access | Find budget | Change budget |
| Saying | What a great day | First time changing my budget! | Got it done! |
| Thinking | Just got promoted at work, I have more budget from now on! | How do I change the budget? Where should I click? | Wish this can be more straightforward so I don't have to spend time figuring this out |
| Doing | Launch SFFS | Click 'Target budget', then click 'Edit budget' | Type the new amount and press 'Save' |
| Insight: A tutorial to demonstrate where each function is located can help users get familiar with the UI. | | Ownership: Front-end developers. | |

| Gabby - Set a reminder - Make sure the system reminds her everyday | | | |
|---|--|--|--|
| | Access | Find section | Set reminder |
| Saying | I keep forgetting to record spendings | 9 o'clock at night is a good time | All set! |
| Thinking | Need to set a reminder | Reminder at every evening should solve the problem | I wish the app could give me a pop-up window for setting the reminder. |
| Doing | Launch SFFS | Click on 'Settings' then 'Set reminder' | Select '9pm' and press 'Save' |
| Insight: | Some users are too lazy to find functions or they may forget to record their costs. For those users, a pop-up recommendation window is just the type of thing they need. | Ownership: Front end developers | |

Part 3: Design

The SFFS, named **FlatEase**, is a mobile app. Mobile is more portable than a laptop; it allows users to record their costs and check their financial information *anytime* and *anywhere* as long as they bring their mobiles. The design inspiration of the app's colours comes from **10-dollar** and **20-dollar** New Zealand notes. The design details are shown in the images below:

- **Launch**

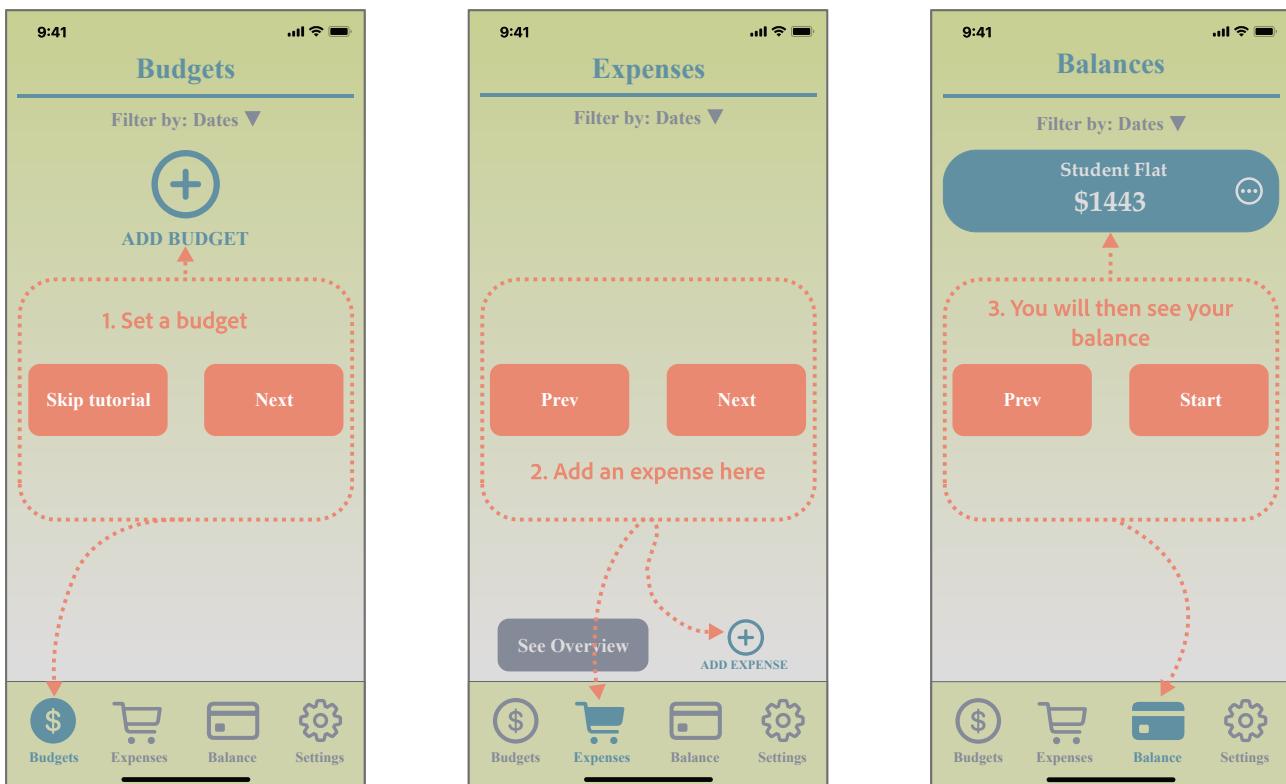


| | |
|-------------|---|
| Persona | All personas |
| Task | Launch app |
| Description | When launching the app, the user will see this initialisation page. The icon demonstrates that this app is designed specifically for university students and their money. Also, it has a loading circle to show the loading progress. |

| | |
|-------------|---|
| Persona | All personas |
| Task | Welcome page |
| Description | On this page, all the elements are commonly related to university students. This aims to shorten the distance between the app and students. Also, this page includes a brief introduction to the app and offers a warm welcome to students. |

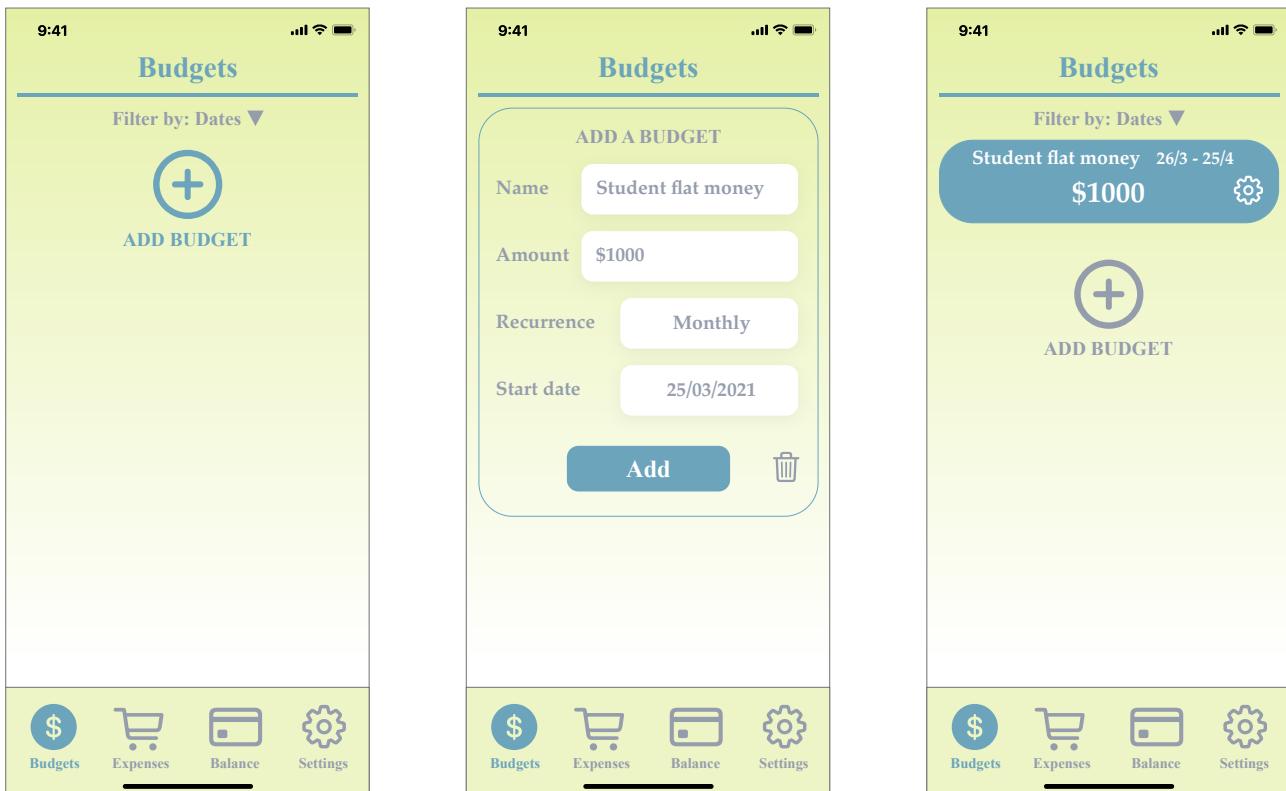
| | |
|-------------|--|
| Persona | All personas |
| Task | Login+Authenticate |
| Description | Data is stored in the cloud. Users can get access to their information even when they change their mobile devices. The "Sign-in" & "Sign-up" password ensures the user's privacy. Also, it allows the user to deal with forgotten passwords. |

- **Tutorial for new users**



| | |
|-------------|--|
| Personas | Gabby |
| Tasks | Play a tutorial for those who don't have experience with similar apps. |
| Description | <p>The first time when a user launches the app, it will play a tutorial to show the main menu and hidden functions. The background is darkened in order to make the orange tutorial more conspicuous.</p> <p>The three images shown above are only a simple example of the tutorial. The actual tutorial will contain more details about the hidden functions. Users can skip the tutorial if they are confident with handling this app.</p> <p>*Note: The button that is clicked is coloured blue, while other buttons are grey.</p> |

- **Setting A Budget**



| | |
|-------------|---|
| Persona | All personas |
| Task | Add budget |
| Description | Click the button to add a budget. The user can add more than one budget. |

| | |
|-------------|---|
| Persona | All personas |
| Task | Set budget details |
| Description | The user can set the <i>Name</i> , <i>Amount</i> , <i>Recurrence</i> , and <i>Start date</i> for a specific budget. |

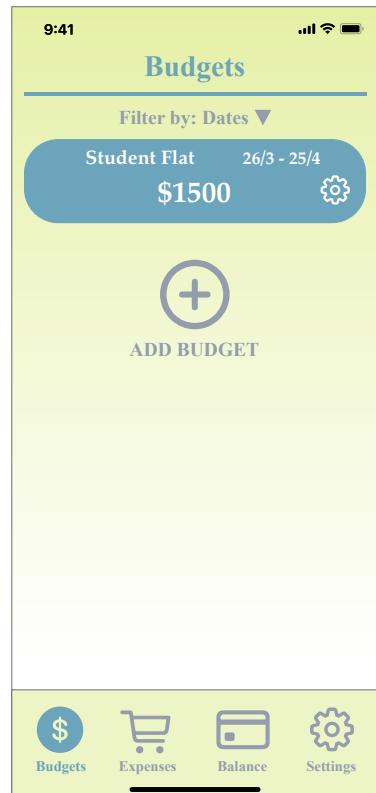
| | |
|-------------|---|
| Persona | All personas |
| Task | View budget |
| Description | After adding a budget, the user can see all existing budgets (name, dates, and amount). |

More details about budget tasks:

As this app aims to help students managing their money, the user must set their budget first so that the app can give them a warning when running out of their budget. Also, the budget will be used to estimate future balance (future balance = budget - estimated future costs).

The app allows users to delete a budget on the editing page by clicking (please see the second image on the next page for more details).

This app accepts different categories of budgets. In the budget list, we can see the budget categories, each of which with name, start and end date, and amount of money. In addition, the setting icon for each budget allows users to change the budget information when necessary.

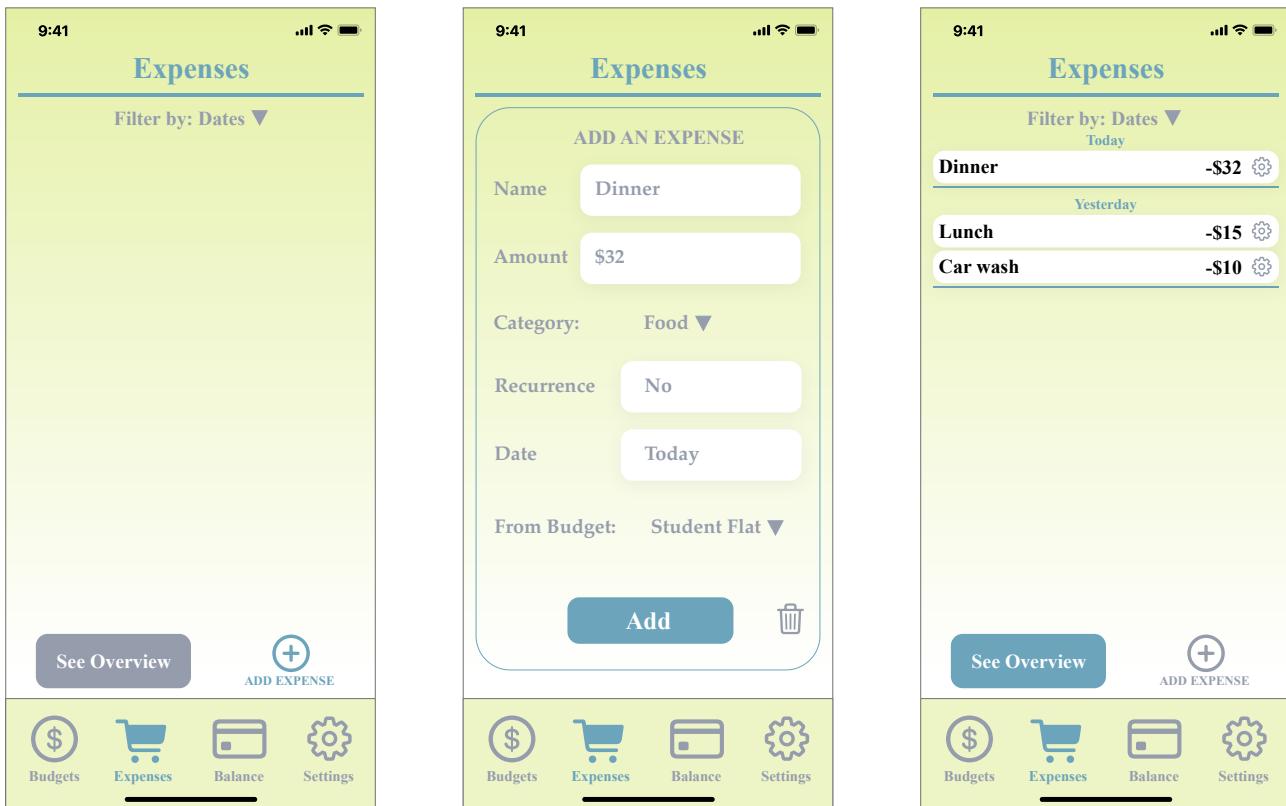


| | |
|-------------|---|
| Persona | All personas |
| Task | Filter budget |
| Description | If there is more than one budget, the user can filter their budget list by date, name, or amount. (Click icon Dates ▼ , then you will get the drop-down menu shown in the above image. Pick one from the three items for filtering.) |

| | |
|-------------|---|
| Persona | All personas |
| Task | Edit budget |
| Description | It will ask the user to confirm whether to remove a budget. It is safer to delete data when the user can see the details, in case they mistakenly delete one. Therefore, I put the delete function on the "Edit Budget" page. |

| | |
|-------------|---|
| Persona | All personas |
| Task | Edit budget |
| Description | If you modify an existing budget, e.g., change the amount from \$1000 to \$1500, you can change the amount to \$1500 on the "Edit Budget" page. Click "Save", and the budget will be updated (see the image above). |

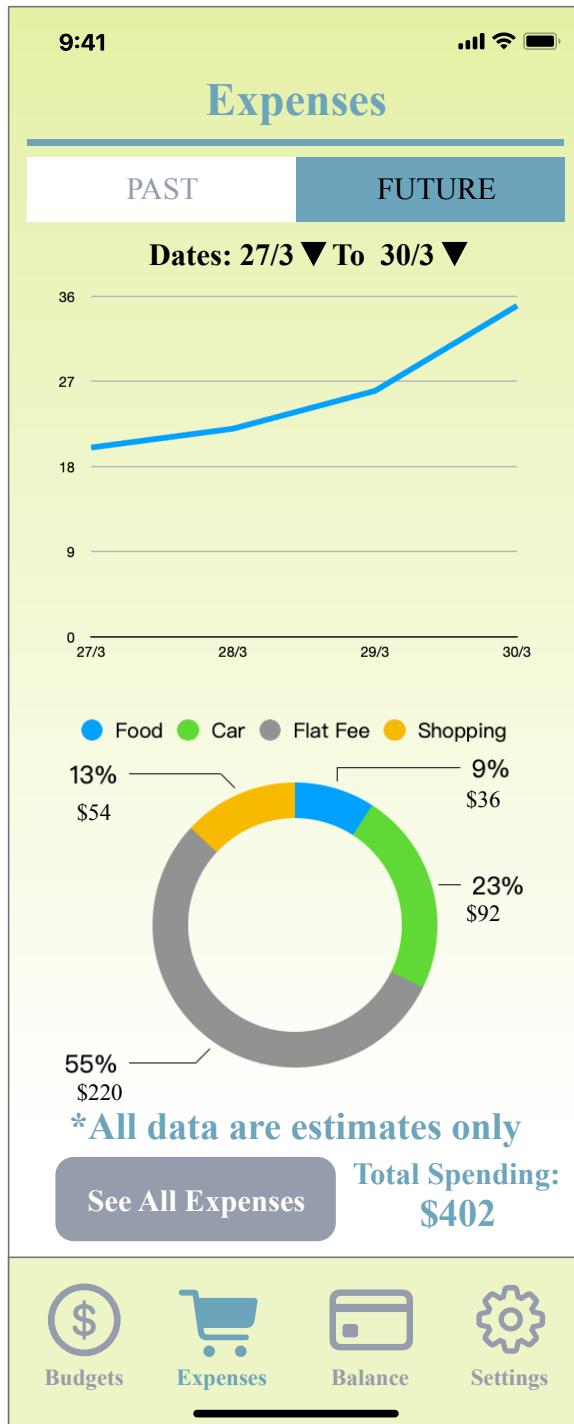
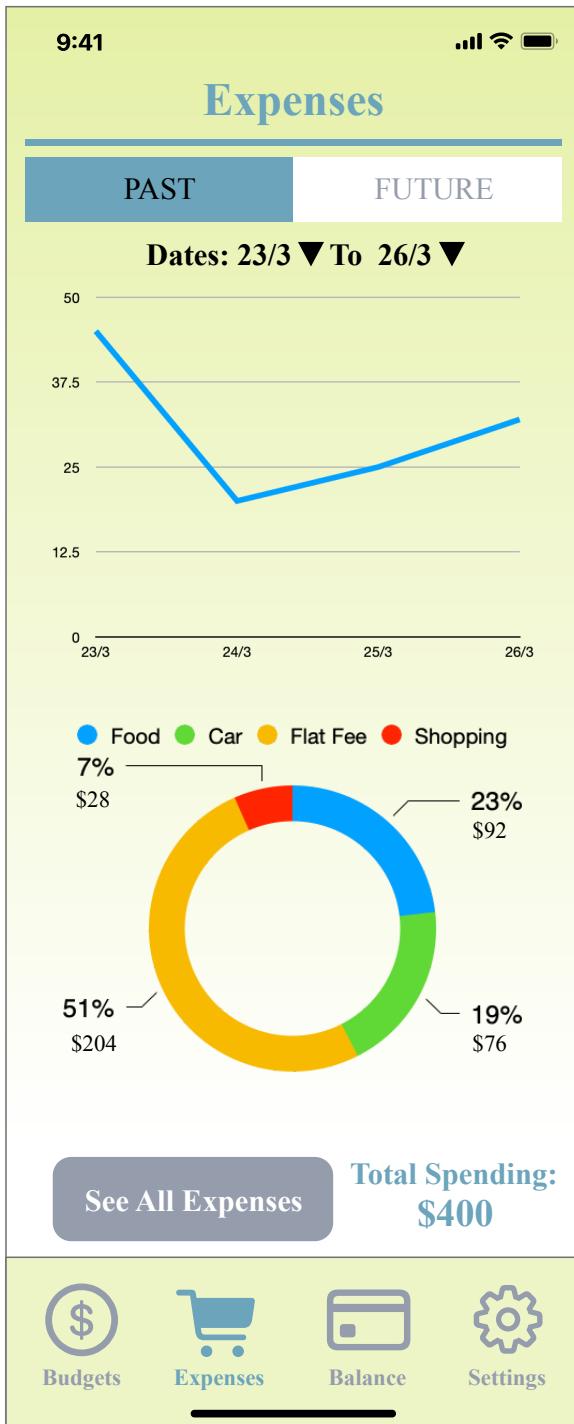
- **Expense**



| | |
|-------------|--|
| Persona | All personas |
| Task | Expense |
| Description | Select the icon “Expenses”  on the bottom, and you will notice that this icon becomes blue. Then you can see “Add Expense” & “See Overview” buttons. In addition, there is a filter for expenses. |

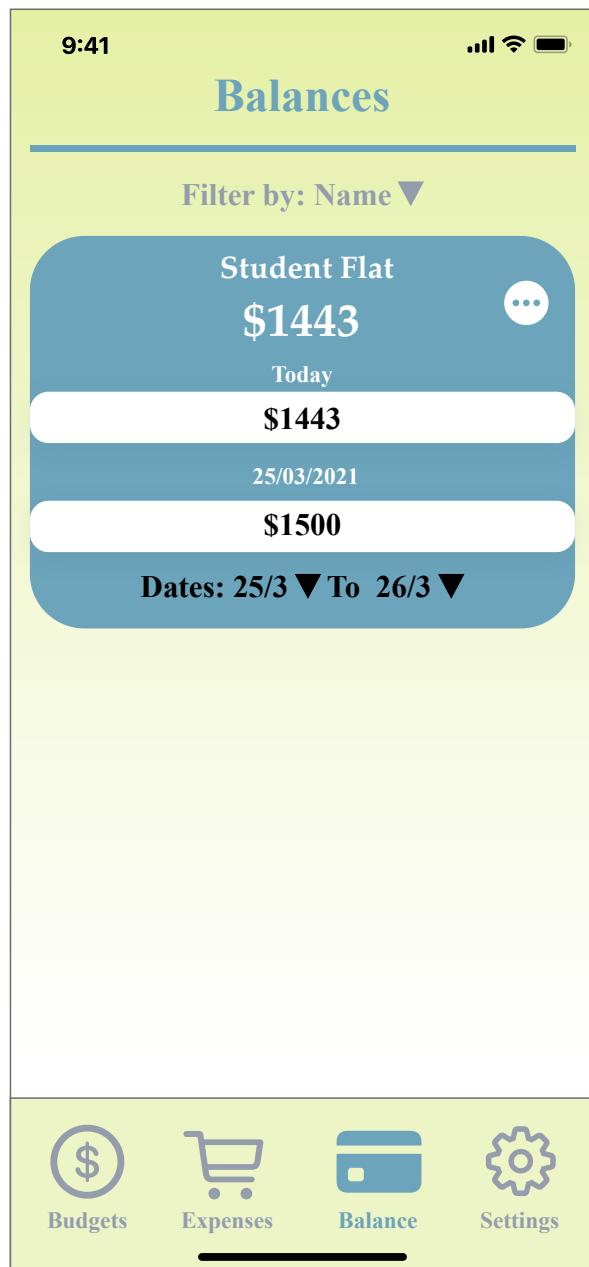
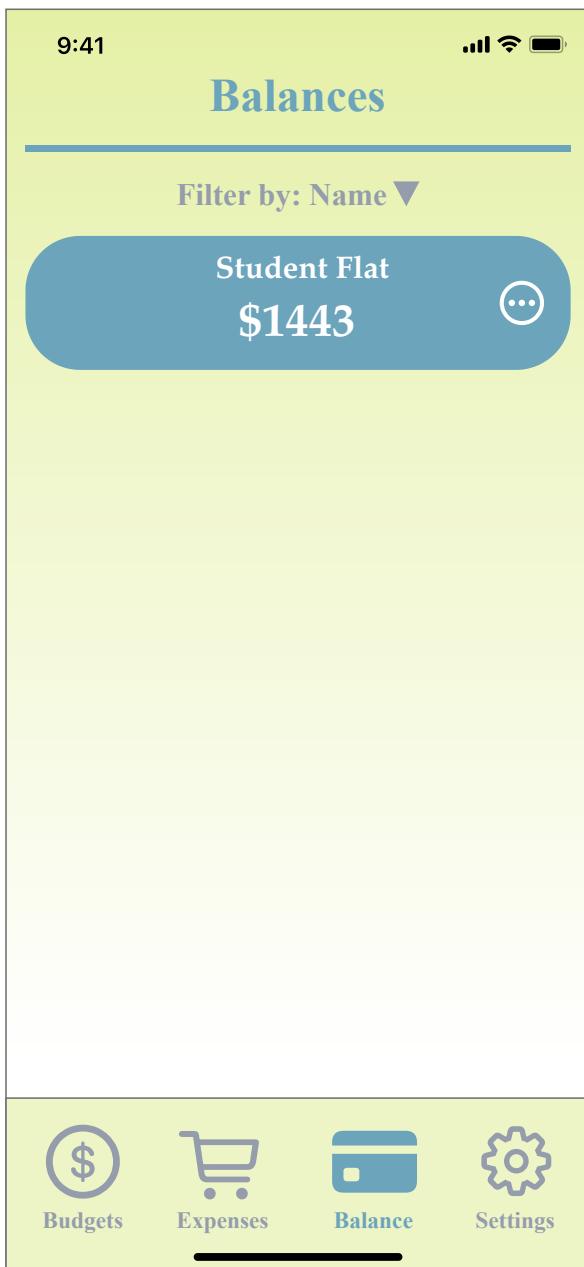
| | |
|-------------|---|
| Persona | All personas |
| Task | Add expense |
| Description | All users need to add their expenses. The required information is shown in the image above. They are required to set the category for the expense because they can filter all expenses by categories later. |

| | |
|-------------|---|
| Persona | All personas |
| Task | See all expenses |
| Description | This page provides a list of all recorded costs. You can filter expenses by “date” or “category”. Also, the cost information could be edited again. Click “See overview” button to see graphs of costs (see next page). |



| | |
|-------------|--|
| Personas | Gabby, Nick |
| Tasks | Show data and graph |
| Description | By clicking the “See Overview” button, you are led to this page. The default tab is “PAST” costs. You can select the start and end date of the expenses that you are interested in. Then it shows a line chart and a doughnut chart. Also, it provides the total costs during that time. |

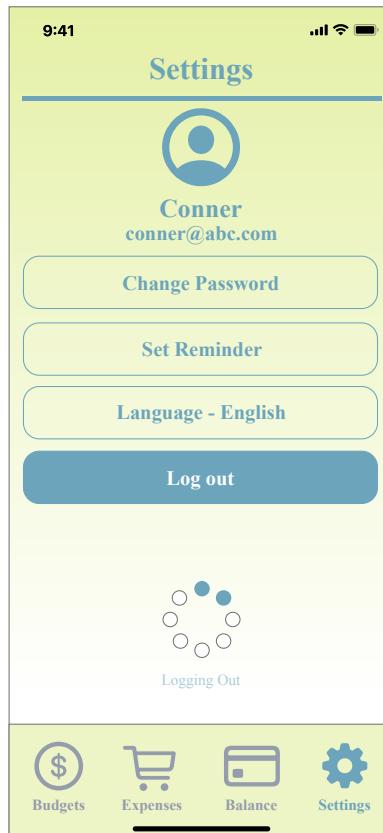
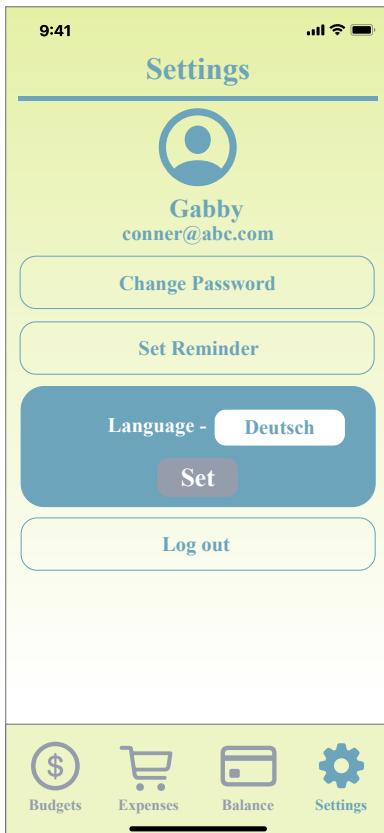
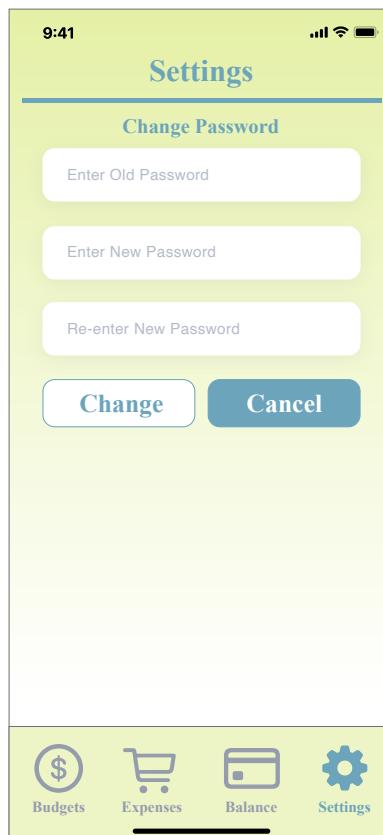
| | |
|-------------|---|
| Personas | Conner |
| Tasks | Estimate future expenses |
| Description | The app helps the user to get an idea about their future costs. This function allows you to select dates, and shows graphs and total costs. Because the future costs are estimated statistically, a <i>Note</i> is provided to remind the user that they can estimate future costs in other ways. |



| | |
|-------------|---|
| Personas | Conner |
| Tasks | View current balance |
| Description | This page shows the balance for all categories (the example here contains only one category — Student flat). You can filter the balance list if needed. |

| | |
|-------------|--|
| Personas | Nick |
| Tasks | View past balance |
| Description | By clicking the blue bar of a specific balance, you can then pick the dates to check your balance sometime in the past. (Will consider line and pie graphs if required.) |

- **Settings**



Functions in the Settings include:

| Task | Persona |
|-----------------|--------------|
| Change password | Nick |
| Set reminder | Gabby |
| Set Language | Gabby |
| Log out | All personas |

Tasks:
security (change password),
Set reminder,
Set language,
log out.

Therefore, the tasks in the “Settings” section cover all requirements of personas.

Part 4: Reflection

• Pros

This section discusses the reasons why some design choices were made, and why they are good for the user experience.

1. **Colours:** There are three main colours on the interface -- blue, green, and grey. The inspiration for blue and green is from 10-dollar and 20-dollar notes, which are the two most popular notes in New Zealand. People are very familiar with them. As this is a finance management app, the note colours remind users of money.
2. **User-friendly interface:** The whole interface is very simple and concise with visible objects (noun) and commands (verb). It clearly displays all functions to achieve their goals. Text and layout are with strong visual alignment. All the button labels follow the rules for choosing the right words, e.g., use action words, use precise diction, etc. In addition, the icons in this design are commonly used for the same function in other apps. This makes it easier for users to get familiar with this app.
3. **Scroll and fit:** This is a universal design, because the interface automatically fits all screens with different resolutions. If the content can not be displayed within the regular page, the app supports scroll pages.
4. **Tutorial:** Tutorial is provided for personas who are not good at mobile app usage. You can also skip the tutorial if you are confident with handling the app.
5. **User forgiveness and error prevention:** The app will pop up a window to ask the user to confirm when they click “delete” buttons or icons. This design prevents them from deleting useful data by accidentally clicking the wrong button.
6. **High task coverage:** This design covers all personas and their requirements (all functions from the model). (see appendix for design coverage below).
7. **Visible system status:** It shows the loading progress on a loading circle. This improves the user experience by showing them the task completion progress.

• Cons

This section discusses the cons of my design and their corresponding improvements. If I would do the assignment again, I would like to keep all those pros (mentioned above) and improve the cons:

1. **Balance:** The “Balance” part may confuse the users. At the moment, the “Balance” is considered a result of budget minus expenses. Some people might not understand the purpose of “Balance”. Therefore, I have been thinking about merging “Balance” into the “Budget” section in the future. For example, Balance could be a function as the “Wallet” in the Budget. Without splitting Budget and Balance, I don’t have to explain “balance = budget - expenses”.
2. **Font:** I used “Times new roman”, “Adobe Clean” and “Palatino” in the design. This can be improved if I get more time in the future to experiment with other fancy fonts. For example, it’s a good idea to use fonts similar to children’s writing for the tutorial.

3. **Voice input:** The current design doesn't involve the use of speech recognition progress. This is not so good for the accessibility of the app. To improve this aspect, I will try to add some voice input buttons at certain menus to achieve the tasks by voice control.

- **What I have learnt about the problem by proposing a solution**

When I did the assignment, I learned that designing a UI is harder than it looks. I use many software every day and interact with all kinds of UI, but when it comes to designing one, a lot of factors are involved. Here are some examples of what I have learnt:

The first problem that I encountered is “how to choose your colour wisely”. I don’t want a colour that’s too boring, and the colours have to be user-friendly and eye-catching as well. Therefore, I checked lots of beautiful colour combinations online, and followed the suggestion of using colours that are commonly found in life.

The interface then needs to show the user a proper path to their desired functions. So I mapped out each step a user will take (user flow) by wireframe. Also, I need to use some simple but understandable icons and texts to achieve this task. It’s a good idea to take inspiration from icons in similar apps/functions.

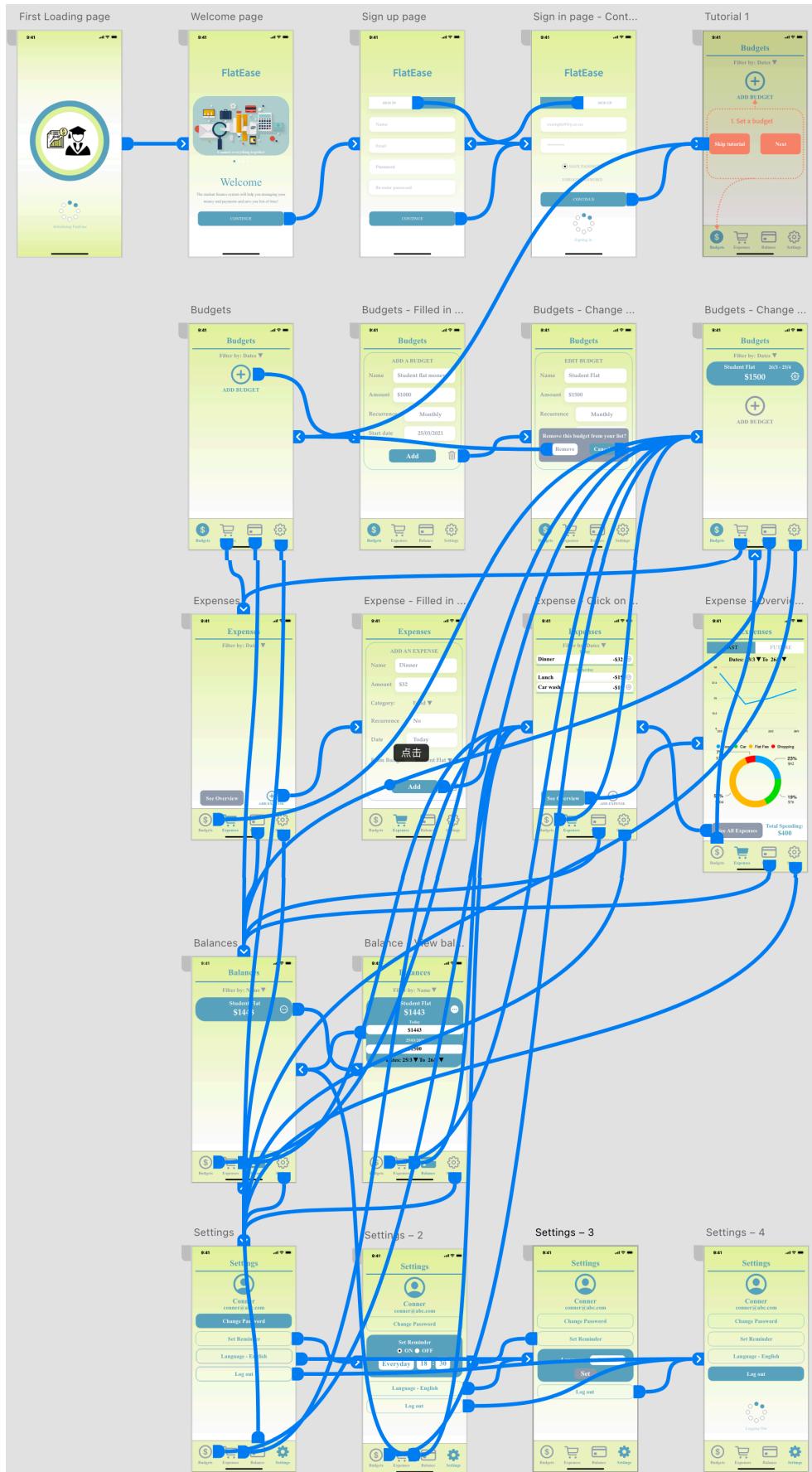
I also learned that the app needs to give the user room to make mistakes. Especially when removing or deleting stuff, the program should ask for confirmation instead of executing the removal directly.

- **Reflect on any changes you made to the personas or tasks based on creating these initial design documents**

I didn’t update the personas or the target tasks. The only function that I added is “Set Language” in the “Setting” section. Some students may have their language preferences, e.g., they may be used to their mother tongues, or they may be learning a second language (setting apps to that language is a good way of practicing a new language).

Appendix

1. This graph shows the wireframe:



2. This graph shows the design coverage (design covers 100% personas and tasks/requirements). The page numbers in blue boxes demonstrate that the task is achieved on that page in Part 3. For example, P.15 means the function "Login" is shown on the 15th page in this PDF document.

