

SWEN303

Final Project Part Two - User Testing

Jasmine Dong

300486787

Todd Wellwood

300529406

Emilia Greve

300526747

Joel Crampton

300528332

Charles Hughes

300546524

Kamonchanok Suban Na Ayudtaya

300471606

Group Name:

Managing Finances 6

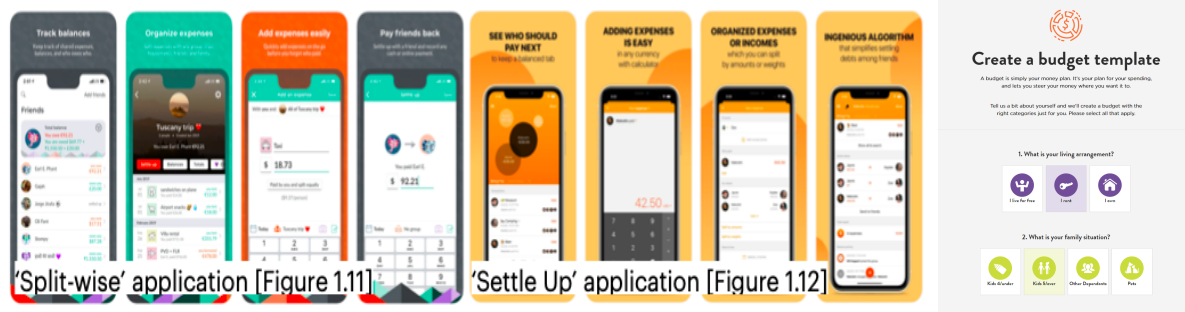
Project Topic:

Managing Finances

1. Background

1.1 Existing Solutions

Various existing solutions act as a manager for shared and individual living costs (fixed and variable). Of those, 'Split-wise' and 'Settle up' are most commonly known to be most effective, according to blogs [Urban Jungle](#), [Wired](#), and [The Balance](#). Additionally, New Zealand based tools such as 'Sorted', and 'Westpac Budget Calculator' are great for providing template-based expense calculating.



Both solutions share similar interfaces and offerings. We can observe that these applications allow users to create groups, allocate expenses (variable and fixed) and monitor outstanding balances. Additionally, these successful solutions implement a minimalistic user interface that can be easily understood and navigated.

Opportunities for improvement in functionality:

- Pay bills directly within the application (Observed in Venmo - Unavailable in New Zealand).
- Use images of receipts to automatically upload expenses (Observed in Divvy (Used for tracking business expenses) and Split-wise premium user interface).
- Set a recurring day in which reminders for payment are automatically sent out.
- Observe relevant user information (e.g. bank account number, payday) from the user's profile.
- Manage personal as well as communal finances.
- 'Westpac Budget Calculator' doesn't offer a login and saving information feature
- 'Sorted' doesn't offer a mobile implementation.

Opportunities for improvement in UI:

- 'Settle Up' offers only a singular button for adding expenses and does not implement a navigation bar.

- Adding an expense is complex compared to 'Split-wise' due to the various container pop-ups.
- All the features displayed on a singular page make the UI unappealing.
- 'Westpac Budget Calculator' has dull colouring, making it boring to use.
- 'Splitwise' application [Figure 1.11] 'Settle Up' application [Figure 1.12]

1.2 Business Objectives

The main objective of a solution that manages shared and individual living costs is to make tracking shared and personal expenses more convenient. In addition, the solution must be easy to use and understand for users with varying degrees of financial knowledge. This objective is supplemented by a UI design that must be attractive, cohesive and easy to navigate.

The primary purpose of a solution that manages shared and individual living costs is to track shared and personal expenses more conveniently and allow for smart budgeting. In addition, the solution must be easy to use and understand for users with varying degrees of financial knowledge. This objective is supplemented by a UI design that must be attractive, cohesive and easy to navigate.

The secondary objective is to accurately and efficiently record and manage incomes and expenses for users. This objective means a solution needs to ensure that inputs from users are simple. The UI for recording transactions must be minimalistic and intuitive to reduce the occurrence of error or user frustration.

The final objective is to observe and edit past expenses and income. This objective allows users to correct inaccurate information and understand how expenses impact budgeting. With the possibility of an extensive list of the recorded transactions, the UI supplementing this objective must be well organised and clear.

These objectives can be more comprehensively met with additional features complemented by a UI that aligns with users' needs and activities. Maintaining user satisfaction and needs ensures the solution maintains and grows its market share which will lead to greater profit.

1.3 Significance to Stakeholders

The finance manager solution is important to various groups and individuals worldwide. Our solution aims to target university students as outlined by the assignment specification.

Managing finances is a necessary aspect of a student's life and it is common to see an influx of students' living situations change yearly as they navigate living by themselves (halls and living at home) to sharing a space (flatting). As a result, they are now responsible for managing finances relating to themselves and others. This can be a complex and hence overwhelming process to understand, with financial disputes potentially causing rifts within personal relationships. This thus demonstrates why a financial plan is vital to students. Moreover, having a reliable financial understanding can encourage good financial decisions and develop stability, consequently reducing stress within a student's life, which arguably is just as important as a financial system.

As this is important to our users, we must ensure that we provide a working and attractive UI to maintain/grow our clientele. Having the user overwhelmed with information and inputs can cause users to turn to different solutions due to frustration, resulting in the loss of market shares. Therefore, we must ensure we provide a user-friendly and intuitive solution for users of varying lifestyles to meet our objective.

Basic functions and significance:

- Add expenses and income. Easily track expenses and income to track spending and available funds (income statement).
- Set saving goals. Assist users in identifying they are meeting their objectives.
- Observe/modify past transactions. Allow for correction of mistakes and ensure our solution can answer queries of expenses through historical transactions. This correction system means users feel confident that mistakes do not result in financial ramifications/inaccuracies.

2. Personas

2.1 Selection Method

Our users are University Students who live a relatively active lifestyle with frequent small expenses (coffee, bus, food) and recurring more considerable expenses (rent, power, wifi). Some are personal, and some are shared. They assume the solution will assist them in tracking these expenses and expect the solution to be easy to use and accurate. They also expect the solution to work on various platforms as various users with different devices will access shared balances and that users have a basic knowledge of how to use devices to access solutions.

In the initial group of personas from past assignments, we found that many personas shared similarities through comprehensive discussion. Each persona was similar in terms of its goals and habits; therefore, we decided to evaluate a persona's quality using comparable attributes unique to a persona to select our final group of personas. It was necessary to focus on differences among our selection as if our personas were too similar, the system would be built around generalised traits, excluding potential users with diverse traits.

To help us make our final persona selection, we created a table of attributes that spanned all of our personas (summary/needs, frequency of use, living situation and additional needs). This table allowed us to observe the duplication of persona characteristics which made the selection process easier. We could select unique personas and combine highly similar ones to generate a more robust persona (encompass a broader range of needs).

The 'Functionality Needs' column allowed us to observe a general overview of what components are considered most valuable to a persona.

The 'Frequency of Use' column allowed us to rate how often each persona interacts with the system (1 = low <-> 5=high). This gave us an indication of the market size they may represent.

The 'Living Situation' column allowed us to observe how users would likely interact with our solution (Alone/Home = personal expense management, Flatting = shared expense management).

The 'Additional Needs' column allowed us to observe what barriers of use a user may face when using our solution (Disabilities, Language barrier).

We decided to choose:

- One persona with a language barrier.
- One persona with low vision.
- One persona with dyslexia.
- One persona who would use the application to manage both personal and shared finances.
- One persona who would use the application to solely manage shared finances.
- One persona who would use the application to solely manage personal finances.

We believe this combination of personas should encompass the needs of our end-users as this persona group should allow us to create a solution which can manage shared and personal finance management conveniently and intuitively while accommodating for disabilities.

Personas were then prioritised based on how their needs/circumstances aligned with the purpose of the solution, their likelihood of use (based on lifestyle and attitude towards finance management) and market potential (size of the market they represent).

2.2 Personas Table of Attributes

Persona	Functionality Needs	Frequency of Use	Living Situation	Additional needs
Frugal Freddy	- Frequent calculations. - Personal finance overview.	5	Alone	N/A
Dave Diffy	- Set spending limits. - Clear concise instructions.	3	Shared Flat	Dyslexic

Spendo Sally	<ul style="list-style-type: none"> - Set spending limits. - Categorise expenses. 	3	Shared Flat	N/A
Takuma Travello	<ul style="list-style-type: none"> - Set spending limits. - Categorise expenses. 	5	Shared Flat	English second language
Oscar Oblivious	<ul style="list-style-type: none"> - Needs a low amount of user inputs (low financial literacy). - Needs a way to track saving goals. 	5	Home	High-functioning Autism
Freya Flata	<ul style="list-style-type: none"> - Needs a visual way to track saving goals. - Shared and personal expenditure. 	5	Shared Flat	N/A
Debbie Detached	<ul style="list-style-type: none"> - Needs an easy to understand interface (low technological literacy). -Wants convenience when adding expenses. 	2	Alone	Low vision
Cameron Careless	<ul style="list-style-type: none"> -Frequent expenses (needs convenience). -Wants to be able to track monthly spending and saving. 	1	Alone	Dyslexic

Samuel Stock	<ul style="list-style-type: none"> - Easily change saving goals. - Customisable setting presets. 	4	Alone	N/A
Paris Pennywise	<ul style="list-style-type: none"> -Shared/personal expenses. -Wants to closely monitor spending (reports). 	1	Home	N/A
Timothy Mcdermott	<ul style="list-style-type: none"> - Shared /Personal expenses - Low financial literacy (low user inputs). - Moderate transaction frequency 	5	Shared Flat	N/A
Amelia Waldorf	<ul style="list-style-type: none"> - Shared expenses only - High financial literacy - High transactional frequency (needs easy expense input). 	3	Shared Flat	N/A
Sarah Yu	<ul style="list-style-type: none"> - Medium financial literacy - Personal expenses only 	3	Alone	English second language

	-Wants to closely monitor spending (reports).			
Jack Apperly	<ul style="list-style-type: none"> - Low financial literacy (low user inputs). - Shared expenses only - High transactional frequency (needs easy expense input). 	1	Alone	N/A

2.3 Persona Summary (Organised based on priority)

Below is the final group of personas selected

Dave Diffy (Main attribute: Dyslexia)

Dave Diffy represents our group of university students looking for a finance solution to accelerate his ability to save conveniently without dyslexia affecting his use of the solution. In addition, he would like a tool to assist in tracking his expenses but has found it hard to identify a suitable solution due to a large amount of text associated with traditional solutions.

Timothy Mcdermott (Main attribute: Shared/Personal expense management)

Timothy McDermott represents our group of university students embarking on their first year of flatting - his expenses are shared and personal. He has a low amount of experience managing his finance as he previously lived in a catered residence hall. He is likely to require assistance in managing his expenses as it is the first time he has to share living expenses and manage his spending.

Samuel Stock (Main attribute: Personal expense management)

Samuel Stock represents our group of university students who only want to manage personal expenses. He has a high amount of experience managing his finances; however, he is involved in the stock market and wants a way to set saving goals and

closely monitor his spending. He would like a tool to assist him but has found it hard to identify a suitable solution because his income and expenditure are dynamically changing.

Freya Waldorf (Main attribute: Shared expense management)

Freya Waldorf represents our group of university students who have had experience flatting - she only wants to manage shared expenses. She has a high amount of experience managing her finances and does not have trouble with personal expenses. However, she finds it hard to keep track of communal expenses during her busy schedule (social lifestyle). Therefore, she would like a tool to assist in tracking her communal expenses.

Traveller Takuma (Main attribute: Language barrier)

Traveller Takuma represents our overseas group of university students living in a shared flat - most of his expenses are personal due to his lifestyle (low social). He has a moderate amount of experience managing his finances. However, he finds it hard to track payments due to his work and uni schedule. He would like a tool to assist in tracking his expenses but has found it hard to identify a suitable solution due to his language barrier.

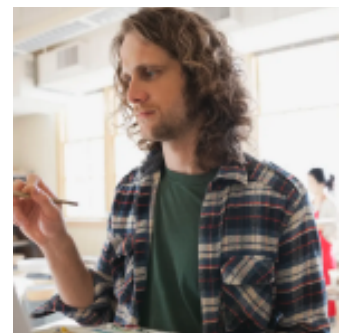
Debbie Détached (Main attribute: Low vision)

Debbie Détached represents our older group of university students - most of her expenses are personal as she is the head of her household. She has moderate experience managing her finances. However, she finds the task strenuous as she has been manually managing her finances. She would like a solution that is easy to understand and intuitive as she has low technological abilities.

2.4 Final Personas

2.4.1 Dave Diffy

Dave is a 25-year-old student studying a Bachelor of Arts in Fine Arts at Victoria University of Wellington. He was drawn to fine art for its heavy focus on shape, form, and colour, which he focuses on as a dyslexic. In his day to day life, Dave misses out on things due to being dyslexic. Everyday activities like using a bus timetable, reading a University assignment brief, and even calculating his own finances. While there are things



Dave can't do because of dyslexia, he is resilient and thinks outside of the box so that he can enjoy life to the fullest. Dave benefits from the use of icons in his day to day life. He likes to keep his mind active through activities that don't involve text, such as puzzles, cooking, and sport.

Due to the heavy workload of his degree, Dave doesn't have time for a job and lives off a student loan, so he wants to better manage his finances to make his loan as small as possible. Dave's flat, although mostly independent, likes to calculate their total spending as a whole. Therefore he and his flatmates would like an option for flat spending. His flatmates are similar to him in the sense that they like to keep it as cheap as possible. They even regularly share meals together in order to keep the cost down, however, calculating the costs each time for food can be pretty cumbersome therefore they would like to be able to calculate their food prices as a flat also.

Activities: Dave is very interested in art, however, he also enjoys tramping and cooking in his free time. If there is an art exhibition occurring, it is likely he will attend.

Attitudes: Dave is a resilient person who can be described as an 'outside-the-box' thinker.

Aptitudes: A very creative thinker, who can visualise problems and ideas easily.

Weaknesses: Due to his dyslexia, Dave struggles with large chunks of text. He additionally has a habit of not taking things seriously due to his carefree nature.

Domain Knowledge: Dave knows about finance tools, but isn't drawn to them due to the heavy use of text.

System Knowledge: Dave hasn't been able to confidently use finance systems, so he doesn't have much system knowledge at all

Interaction: Dave would use the system each week to plan for his next week's finances

Priorities: Dave wants to stay on top of his finances, and not spend too much on student loans.

Computer Self-efficacy: Dave is open and able to use new computer systems, so long as they cater to his dyslexia

Risk Tolerance: Dave has a high tolerance for risk due to the resilience he has learnt through having dyslexia.

2.4.2 Timothy McDermott

Timothy (Tim) grew up in Wellington. He has always been passionate about software development and pursued this academically at VUW. Tim has just entered his second year of study for a Bachelor of Software Engineering. Recently he has moved from living in halls (Capital Hall) to a five-bedroom flat with his friends in Newtown. Tim currently works part-time as a tutor (VUW) to finance his living cost.



Tim expects the financial system solutions to be less time consuming and more accurate than manually tracking expenses. In addition, he wants the UI to be clear and easy to understand as he needs to communicate with a relatively large group (five people) and wants to prevent further conflict.

Tim is the lead tenant and is in charge of paying for rent and utilities. Rent and utilities are automatically charged from his personal account, which he also uses to pay for personal expenses. This makes it hard for him to budget as his flatmates do not always pay their share of expenses on time and there are often disputes on outstanding balances owed to Tim. As a result, Tim sometimes has to use money from savings to ensure bills are paid. Managing flat finances has been a significant source of conflict within his flat group and has affected his ability to save.

Activities: Tim loves programming in his spare time, he also is constantly trying to improve and learn. He spends lots of time reading self-improvement books and autobiographies.

Attitudes: Authoritative and keen, Tim is not afraid to assert himself and rise to the challenge.

Aptitudes: Quick learner, keen eye, detail-oriented individual.

Weaknesses: Needs full control, would like this to be catered to in the solution.

Domain Knowledge: Tim has a good understanding of how financial systems work, he's considered a few. He understands the advantage of a good financial system and wants something that will work for him.

System Knowledge: Tim considered designing his own financial system, ultimately deciding he wants to prioritise his time. He knows what he wants in a system.

Interaction: A solution that is easy to use with little/no financial calculation as Tim does not want to spend time calculating each flatmate's share of an expense. Works on different types of devices (Apple, Samsung Huawei) so he and his flatmates can work collaboratively to track expenses as they occur (Limits loss of information).

Priorities: Tim wants a solution that will be easy to use and allow him to manage his own and his flat expenses. This is so he can use his time to concentrate on his studies and budget more effectively to save up and travel to Australia.

Computer Self-efficacy: Tim is confident when performing a task on a new platform, often exploring technological solutions for his problems, but is less inclined to invest in time-consuming processes as he is a bit impatient. Tim will likely move on to a new solution that better meets his needs if something goes wrong. Currently, Tim uses split-wise but is unhappy with the solution as it does not offer personal finance management.

Risk Tolerance: Tim has little to no risk tolerance, he needs full control over the system and doesn't expect anything to go wrong.

2.4.3 Samuel Stock

Samuel is a young hotshot looking to get ahead and make it big in the world of investing. Whether it is stocks, crypto or even horses, Samuel does it all. While Samuel is studying business, he uses the profit of his investments to pay for his rent and other living expenses. His main goal is to leave university debt-free without needing to pick up any part-time work. To help achieve his goals, Samuel needs a system that lets him quickly adjust his budgeting in time to get in early on potential capital ventures.



Being an experienced risk-taker, Samuel is no stranger to things going wrong. During his high school days, he often would lose all his savings on bad investments and needed to fall back on family for help. Because of these embarrassing experiences, Samuel wants a way to manage his finances in times of hardship easily and have a dedicated system to help him budget in an emergency fund. This safety net will allow him to avoid awkward confrontations with family and make everybody see him as the big-shot capitalist he knows he can be.

Activities: Samuel enjoys gambling and browsing the popular 'r/wallstreetbets' subreddit in his free time. He loves reading business and economic articles whenever possible.

Attitudes: Samuel has a potent superiority complex. Considering himself a leader and extrovert, he cannot help but try to steer the conversation.

Aptitudes: Samuel is knowledgeable and loves to read.

Weaknesses: Overconfidence can often be a slow and insidious foe when Samuel manages his finances.

Domain Knowledge: Samuel has an excellent understanding of making his money work for him but very little skill when budgeting.

System Knowledge: Samuel has had no experience with financial management systems. He has previously felt above them.

Interaction: Samuel is looking for a solution he can use every day at a moment's notice.

Priorities: A cheap system that is easy to configure and rebalance when needed. An auto budgeting feature is a must.

Computer Self-efficacy: Samuel is well-versed with computers, considers himself a superior user, and believes digital solutions will be the future for all things.

Risk Tolerance: All in. Samuel is not afraid to try anything new. He is very comfortable with taking risks.

2.4.4 Freya Waldorf

Freya grew up in Auckland and moved to Wellington in 2018 to study architecture at Victoria University of Wellington. She has recently finished her studies and has found a graduate role at a Wellington Architecture firm. Growing up, Freya was financially stable, however, when she moved away to university, her parents told her that it is up to her to try to look after herself and become financially stable on her own. When Freya began university she was awarded a \$20,000 scholarship which covered her Halls of Residence expenses for the first year. However, she now has a student loan which supported her hall fees in her second and third years. Now as a graduate, she lives with a couple of friends in Thorndon. This is Freya's first time flatting and she thus doesn't quite know the logistics of how to budget.



Freya has a busy schedule and is often found at a local restaurant or bar after a long workday. Her goals are to maintain a sustainable income and prioritise spending on what is most important such as flat expenses, rather than her wants. As a visual learner, she loves looking at a layout that is easy to read/follow, and colours help her differentiate subjects apart. She expects that the solution will keep her up to date with expenses and allow her to observe past expenses.

Activities: She loves to go to art museums and explore Wellington in her spare time. She also enjoys hanging out with her friends and treating herself after submitting an assignment.

Attitudes: Freya has an active lifestyle and considers herself an extrovert, but enjoys time with herself. She loves to get involved, try new things, and has a can-do attitude.

Aptitudes: Freya is very creative and likes to think outside the box. She is very organised, and when she sets her mind to something, she will try to complete it.

Weaknesses: Freya sometimes gets quite stressed when things do not work out the way she wants them to. She has sometimes found it hard to keep track of what is going on in her flat, resulting in her missing payments and being charged for communal expenses she had not agreed to/is aware of.

Domain Knowledge: She knows how to save, and she is not a big shopaholic. However, she has not had the opportunity to organise her own money and wants a solution to help her budget and organise what her money should be spent on.

System Knowledge: Freya is more than capable of working with an online system.

Interaction: Freya will most likely use the app every single day.

Priorities: Freya should prioritise spending her money sustainably to fulfil her needs and wants while maintaining a degree of financial stability.

Computer Self-efficacy: Freya is more comfortable using her mobile phone to access solutions either on a website or through an application but is open to using her laptop. She does not actively seek solutions and has not tried to manage her finances through a solution; instead, she pays for amounts requested by her flatmates on their group chat.

Risk Tolerance: Freya is a perfectionist and prefers not to take risks if she does not have to. She doesn't want to worry about tracking expenses related to her flat.

2.4.5 Takuma Travello

Takuma is a 21-year-old International Student from Japan. He is majoring in Data Science at Victoria University of Wellington. Takuma is residing in a three-person flat. They all are independent and don't like to socialise with each other. Due to being a non-domestic student, he is ineligible to get a student loan. Therefore, his income has to come from his part-time job working at a local sushi shop as well as some parental support. Takuma will need to be on a very strict financial plan however has been 'winging' his situation for the time being. With his university fees and living, he can't afford to be too extravagant with his lifestyle.



Takuma has a basic understanding of English and thus will require an "easy to understand" system that doesn't require too much reading. Although being very technologically smart, he will prefer a system that solves his financial problems quickly and methodically due to his busy schedule. He is not interested in calculating spending as a flat, meaning he would like a version of the system catered for

personal use. Takuma sometimes travels back to Japan and he consequently needs a way to calculate a savings goal.

Activities: Takuma spends most of his time studying for his degree, however on the side he works at a local sushi shop. In his free time, he can be found playing squash.

Attitudes: He has a very switched on approach to life. He values his studies and spends a lot of his time on them.

Aptitudes: He is very analytical and is a very good problem solver. He has very good time management and keeps himself very organised.

Weaknesses: Due to his full-on attitude, he finds it very hard to switch off. Therefore he gets prone to stress. This stress tends to make him forget about sorting out his finances and has a tendency to not keep on top of bills and subscriptions.

Domain Knowledge: Although very technologically smart. Takuma hasn't had much experience with financial management systems due to them not having a very foreign-friendly user interface.

System Knowledge: Takuma hasn't had much system knowledge using financial management software for New Zealand Dollars.

Interaction: Takuma will use the financial management system to help keep him on track with his financial problems on a regular basis.

Priorities: He focuses on his studies as his main priority. His work is also important due to being his main source of income. He thinks about his financial problems for the most part however has a tendency to forget.

Computer Self-efficacy: He has an excellent understanding of computers and doesn't require much technical support with the system.

Risk Tolerance: Takuma will expect an effective system that doesn't require too much experimentation to fit his needs. However, he doesn't give up easily and will take some leeway if the system has areas he can't understand (from a language perspective) to an extent.

2.4.6 Debbie Détached

Debbie is a forty-five-year-old single mother who previously worked as a part-time court stenographer for the last twenty years and is now seeking a change of profession to better support her one-year-old son. She is just starting her journey of studying English literature, her goal being to become a successful writer. Debbie describes herself as a woman who can do it all; she is not worried about her age difference compared to



other students. She believes her experience is a far more valuable asset than education.

However, Debbie is also somewhat cautious of technology and is looking for a pen and paper solution to manage her finances. This is mainly due to Debbie's unfortunate visual impairment. While not totally blind, Debbie struggles with small characters and low contrast visuals. She wants all her information concise and in one place. She is looking for a solution that will accommodate this disability.

Being aware of the student debt crisis in America, Debbie is strongly against getting a student loan. Instead, she works part-time, does odd jobs and babysits/nannies. She figures that she will be able to manage her finances efficiently and successfully graduate from university without compromising her financial stability with the right budgeting tool.

Activities: Debbie enjoys teaching her son and playing games with him. She also is a fan of the occasional drink with the girls from her old work but is conservative regarding the cost of babysitters.

Attitudes: Debbie is a strong independent woman who will rise to meet any challenge; there is nothing she cannot handle.

Aptitudes: Debbie may not know everything, but she is willing to give things a go.

Weaknesses: Little patience for technology. If it does not work the first time, she does not want anything to do with it. Her poor eyesight quickly frustrates her when she can't read something.

Domain Knowledge: Debbie has had experience managing finances but never needed to budget to this extreme and is unwilling to let it go to chance.

System Knowledge: Debbie is used to mapping out her finances on paper but wants a solution with more in-depth knowledge.

Interaction: Debbie likes to balance her accounts at the end of the week. She wants a system to put in all her data at once.

Priorities: Her son comes first. She also is not worried about saving money. Instead, she just wants to get by, week by week. Her goal is not to need a student loan.

Computer Self-efficacy: Debbie is somewhat distrusting of technology. Working in court, she used a specialised typewriter that was quite old and had an extremely barebone graphical UI. Debbie liked this. She also is only able to use computer applications in big text mode.

Risk Tolerance: Debbie will take no chances with her son's future and her financial stability at stake. She has one shot at university, and she will do it right.

3.Scenarios

3.1 Scenario Selection and Merge Process

Our scenarios were chosen and merged based on a selection of criteria that we deemed relevant and important within our system. Once this criterion was decided, we individually ranked the scenarios from 1 (least important) to 5 (most important), then averaged them, with the highest scenarios forming our final selection. The criteria in which we based the ranking of our scenarios are as follows:

1) Similarity

The initial evaluation revealed that we had a few scenarios which were very similar, if not identical to each other. These included expense overview, and adding expenses. We discovered that the scenarios which were most essential to our system were also the scenarios which had the most similarities. This meant we were easily able to recognise the importance of these scenarios and thus they were merged and refined.

2) Relevance

A vital part of the scenarios is their relevance and interaction to our personas. As we are developing our system based on these scenarios it is crucial that our chosen scenarios meet our persona's goals given that their needs are achievable. We had a selection of scenarios that individually sounded like a good idea, but when compared to other scenarios, they were noticeably not as relevant and thus important to a wide selection of our personas.

3) Scope and System Functionality

When determining our scenarios, we also had to keep in mind the scope of the project. While we would like to make our system have additional features and be able to do everything, we also have to remember what is achievable in the timeframe. Furthermore, a necessity of our system is that it must be easy to use and by including more elements we were at risk of overcomplicating the system. This meant that in the ranking process we needed to question if the time and energy put into developing that scenario would ultimately give an equal benefit within the system.

M: Set payment reminders		3	3	3	1	2	1	$13/6 = 2.17$
M: Create/access group account		3	4	5	2	4	5	$23/6 = 3.83$
M: Create/access personal account		4	4	4	3	4	4	$23/6 = 3.83$
JET: Expense overview	M: View spending summary JC: Struggling to pay rent	5	5	5	5	5	5	$30/6 = 5$
JET: Adding an expense/income	M: Creating an expense	5	5	5	5	5	5	$30/6 = 5$
JET: Pen and paper solution		3	3	3	3	3	3	$18/6 = 3$
JET: Need for quick finance		4	4	5	5	4	5	$27/6 = 4.5$

JET: Enabling/disabling accessibility options		5	5	5	5	4	5	$29/6 = 4.83$
JC: Struggling to pay rent	M: View spending summary JET: Expense overview	4	4	5	4	4	5	$26/6 = 4.33$
JC: Calculate food prices as a flat		5	4	5	5	4	4	$27/6 = 4.5$
JC: Improve social life		2	1	1	1	1	1	$7/6 = 1.17$
JC: Save up to travel back to Japan for Christmas holidays		2	1	3	3	3	3	$15/6 = 2.5$

3.3 Final Scenarios

3.3.1 Adding an expense/income

Personas involved: Dave Diffy, Timothy Mcdermott, Samuel Stock, Freya Waldorf, Traveller Takuma, Debbie Détached.

To ensure that all personas can keep track of their money flow within their account, it is important to track expenses and income so the app can determine how much the user can spend and save. This scenario consists of tracking both personal and communal expenditure and income. This includes income transactions such as student loans, investment money, and wages. Additionally, we must track expenditure, e.g. rent, clothes, food etc. This information will be used to build the “Summary overview”.

User intention	System
Registers expenses/income (manually by inputting in the information).	
	Enter a detailed overview of every transaction, including the date, group, category, description, value and whether it is recurring.
	Displays new transactions with the past ones that have been registered previously.
Users want notification when transactions are submitted, especially when in a group.	
	Once a transaction is registered, all shared account users are notified that a transaction has been submitted. The remaining balance is shown.

3.3.2 Summary overview

Personas involved: Dave Diffy, Timothy Mcdermott, Samuel Stock, Freya Waldorf, Traveller Takuma, Debbie Détached.

The summary overview scenario will involve all of our personas. However, particular persona’s tracking their spending to budget their expenditure will use this. When users want to observe the expenditure of a group/personal account during a time period in an organised format, they will view it in the solution. Users will also want to have an income, expense and savings summary to view how much they have spent and how they want to save to meet a goal.

User intention	System
----------------	--------

User needs to view their flat's financial situation.	
	Shows each flat member's recent inputs and outputs.
	Displays joint expenses which need to be paid soon, who has paid and who still needs to pay.
User wants to see an overview summary to see how much they have spent.	
	Displays income statement (user chooses how regular weekly, monthly etc) in a graph/report. Different categories of expenses are grouped which users can access.
User wants to see a savings summary to see how much they have saved and what their goal is.	
	Displays savings goals, as well as how much they have saved (user chooses how regular weekly, monthly etc) in a graph. The user can also add a new goal that they want to try to achieve.

3.3.3 Enabling/Disabling accessibility options:

Personas involved: Dave Diffy, Traveller Takuma, Debbie Détached.

It is important that everybody has equal opportunity and access to our software. Some users require slight modifications in order to use, this scenario describes how to go about enabling these settings for our users. We want to create functional, accessible content for people with disabilities and we want these options to be as easy and non-intrusive as possible.

User intention	System
User wants to change the size mode of the text.	

	System will have an accessibility section that contains an option to toggle big text.
User wants to change the colour mode of the app to make it easier.	
	System will have an accessibility section that contains an option to toggle the colour contrast of the app.
User wants to change the simplification level of the app.	
	System will have an accessibility section that contains an option to toggle the simplification level of the app.
User wants to change the colour to greyscale on the app.	
	System will have an accessibility section that contains an option to toggle the colour of the app to grayscale.
User wants to change the text on the app to icons.	
	System will have an accessibility section that contains an option to toggle the text of the app to icons.

3.3.4 Need for efficient financing

Personas involved: Dave Diffy, Timothy Mcdermott, Samuel Stock, Freya Waldorf, Traveller Takuma, Debbie Détached.

This scenario is applicable to all personas as the need to finance quickly is something that all of our personas may experience. We have implemented this so that a user is able to select between saved transactions and recurring transactions. Saved transactions are previous one-off transactions that the user has saved. Recurring transitions are transitions that occur on a time schedule (e.g, every week). This implementation means that a user does not need to restart their transactions every time they use a system, therefore allowing a quicker finance ability.

User intention	System
Navigate to the add expense page.	
	System prompts the user with a saved and recurring button.
User wants to manage singular expenses/income.	
	Allows the user to select income or expenses.
	System allows for transactions to be edited and applied.
User wants to manage recurring expenses/income.	
	Allows user to select income or expenses.
	System allows for transactions to be edited and applied.

3.3.5 Create access-group account

Personas involved: Dave Diffy, Timothy Mcdermott, Freya Waldorf, Traveller Takuma, Debbie Détached.

Persona's who use the solution to track communal expenses will use this. Users tend to create groups upon account generation. Additional groups may be added when users change flats or become part of new social groups in the solution.

User intention	System
Navigate to the group account page.	
	Display option to manage existing groups.
	Display option to add a new group.
Select add new group.	
	Display required information.

Enter group name and group members.	
	Finish group setup.
	Send an invitation to relevant users.
	Display group creation successful.

4. Design Ideation

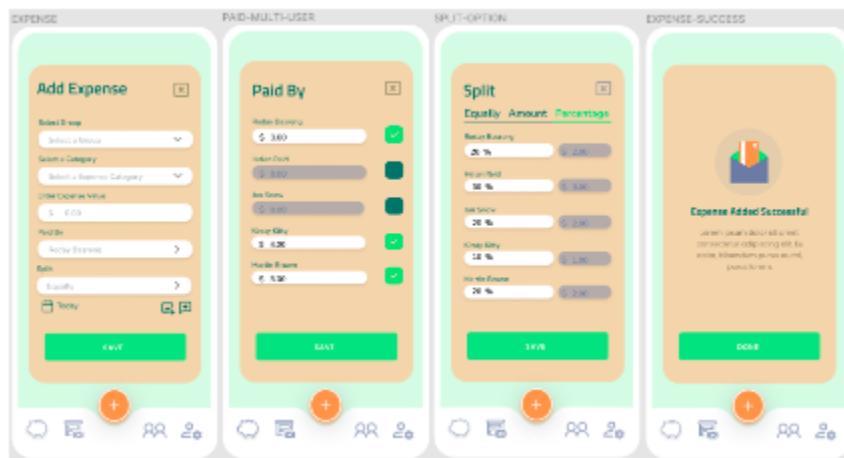
Here is a sample of each team's design of an expense page.

4.1 Dime

- Kamonchanok Suban Na Ayudtaya

Figma:

<https://www.figma.com/file/40UBD7fuEWi98bdIZnULKc/DIME?node-id=0%3A1>

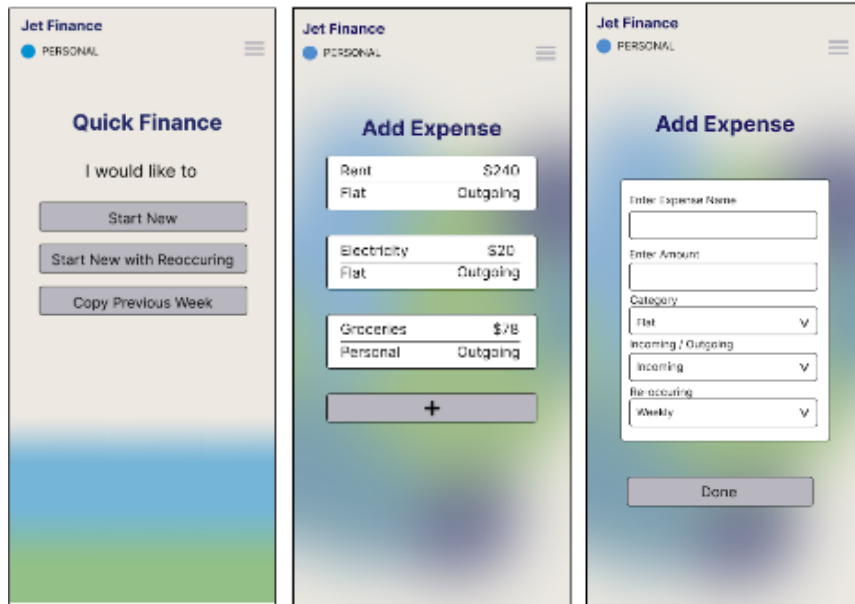


4.2 Jet Finance

- Todd Wellwood
- Jasmine Dong
- Emilia Greve

Figma:

<https://www.figma.com/file/HmAzuXJ6QYqASd8CCFO7TC/SWEN303?node-id=0%3A1>

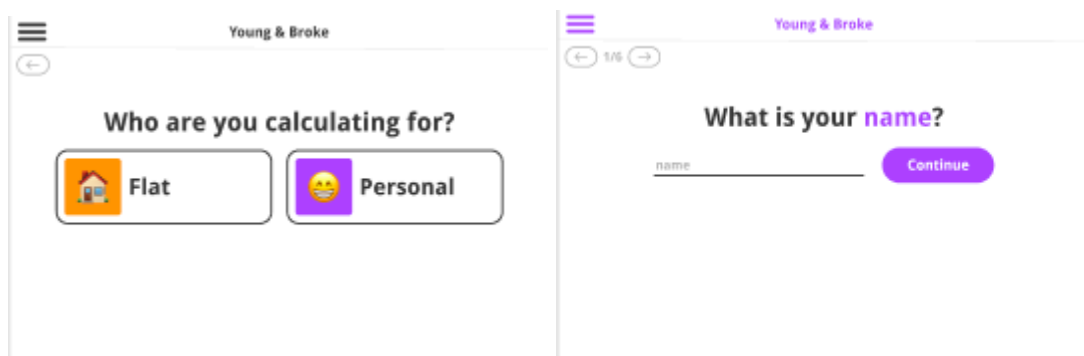


4.3 Young & Broke

- Joel Crampton
- Charles Hughes

Figma:

<https://www.figma.com/file/Cr94gsD2QBdE9OI7BhDps6/design?node-id=0%3A1>



Young & Broke

←

2/6

→

What is the **time period** you're calculating for?

1

week

Continue

Young & Broke

←

3/6

→

What is your **income**?

\$ income

Continue

Young & Broke

←

4/6

→

What are your **expenses**?

\$ rent

\$ food & living

\$ utilities

\$ transport

\$ leisure

\$ education

\$ other

Continue

Young & Broke

←

5/6

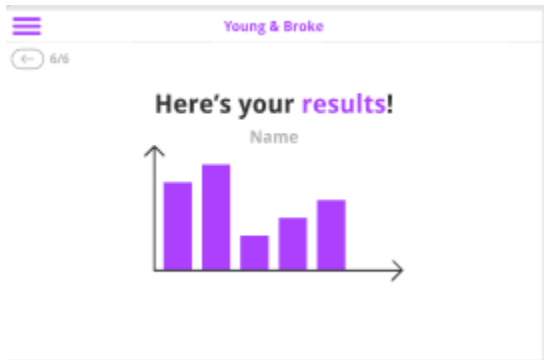
→

Do you have a long term **savings goal**?

\$ savings

Continue

Skip



5. Design Review

5.1 Selection method

Strategy

Before our group began designing our prototype, we performed individual heuristic evaluations against our three existing designs; Dime, Jet Finance and Young & Broke. Nielsen's 10 Usability Heuristics were used as the basis for this evaluation. For each of the ten heuristics, we identified the pros and cons of the designs where possible. Each pro and con was then rated 1 to 5, where 1 is low and 5 is high, for its severity, frequency and impact - a description of these is given below:

- **Severity:** How good/bad is the issue?
- **Frequency:** How much of the system is affected?
- **Impact:** How much of the system's core functionality is affected?

Severity, frequency and impact were also considered in terms of how much they would benefit or worsen our personal experience, as they were who we were designing for.

Individual Evaluation

We then summed these three ratings, averaged them, and ranked all of our ten pros and cons from highest to lowest using the average score. This enabled us to discover some really good features of our designs, whilst also uncovering some features that needed improving, which proved to be very useful for merging designs. Our individual evaluations are linked to (through GitLab) below:

- [Kamonchanok Suban Na Ayudtaya](#)
- [Jasmine Dong](#)
- [Emilia Greve](#)
- [Todd Wellwood](#)
- [Joel Crampton](#)

Group Evaluation

As a group, our individual evaluations were then summarised by merging the pros and cons to give us a more holistic view of how good and bad each design was.

Findings

From this group heuristic evaluation, we found that Dime outperformed both Jet finance and Young & Broke, which contributed to us choosing it as the base for our group

design. During the group design process, the cons we identified for Dime were improved and the unique pros we identified for Jet Finance and Young & Broke were implemented to produce a well-executed group design.

Here are the summarised results of our heuristic evaluation as a group:

5.2 Summary of Group Evaluation

5.2.1 Dime

- Kamonchanok Suban Na Ayudtaya

Heuristic number	Pros	Cons
1	<ul style="list-style-type: none">• Good feedback. After any action; payment, adding expense, etc. there is a confirmation screen that pops up for the user to give visual feedback.• The design indicates present actions. Users know what function they are using through the contrast on an active tab in the navigation bar and know what action they are performing as every screen has a header.	<ul style="list-style-type: none">• No loading indications
2	<ul style="list-style-type: none">• Realistic design. Name section with user profiles is very realistic, and helps to bridge the connection between users and real life.• Easy to interpret. The system terms used to describe possible functions were easy to interpret.	<ul style="list-style-type: none">• Lots of icons. There are a lot of icons which users may get confused about if they don't know what it means.

	<ul style="list-style-type: none"> • Recognisable icons. The icons used throughout the design are internationally recognisable for their action. For example, the "+" icon is used for adding an expense/income. 	
3	<ul style="list-style-type: none"> • Exits used. There are clearly marked emergency exits for user inputs/actions. Simple back arrow for the user to click to go back. A big green button needs to be clicked in order to perform an action. 	<ul style="list-style-type: none"> • Account required. The design cannot be used without creating an account. This limits the user's freedom, and also reduces the potential users. A one-off or occasional user is unlikely to sign up, and would look for alternative designs that do not require an account to use.
4	<ul style="list-style-type: none"> • Consistent layout. The system is designed using Google's material design documentation to ensure the lowest possible learning barrier for users. The colours, icons and layout are consistent throughout the app • Consistent wording. The design makes use of adjectives to label features of the design, but the same adjective is always used for the given feature. This makes it clear and recognisable to the user. 	<ul style="list-style-type: none"> • An unexciting colour theme. • The angle/curve of the boxes should all be the same.
5	<ul style="list-style-type: none"> • Input constraints. Multiple types of data entry where appropriate, financial 	<ul style="list-style-type: none"> • Payment cannot be undone. Once payment is processed it is final, there

	<p>information is always numeric, while a selection of contact is done via a dropdown - this helps to prevent errors.</p> <ul style="list-style-type: none"> ● Clear DONE button. If you want to save/submit, there is a clear bold 'DONE' button. 	<p>is no temporary "undo" button. For something as important as payment, the ability to "undo" or "go back" immediately after the action has taken place could be of great use.</p> <ul style="list-style-type: none"> ● There is no pop-up such as "Are you sure you want to do this?", so once the user clicks the button, the action will be processed.
6	<ul style="list-style-type: none"> ● All related inputs are on the same screen. This means users do not have to recall previous inputs to confidently submit an action (essential to reassure users when dealing with money). ● No information to remember. Whether the user is adding an expense, or settling a payment, there is no information that needs to be remembered between steps. The application keeps track of any information entered in previous steps. The exception to this is the user's password, which they are expected to remember. ● The app is well-titled and allows the user to easily follow through with what is needed to perform an action such as adding an expense. On top of every box which requires input, 	<ul style="list-style-type: none"> ● No help in context. There is no help in context throughout the application. The only help that exists is in the profile section, which would be a full tutorial for the entire application. It would be beneficial to have smaller help documentation in context - when and where the user needs it.

	there is a title of what is needed.	
7	<ul style="list-style-type: none"> • Efficient navigation. Having the bottom taskbar allows users to quickly navigate between pages, and access anything from anywhere. • Good customisation. Users can customise the application by adding incomes and expenses, adding friends, updating their profile picture, etc. • Search bar. Users may take advantage of this and use it as a shortcut to quickly find needed information 	<ul style="list-style-type: none"> • Lack of input efficiency and customisation. There are currently no methods to make inputs more efficient or customise the content to users. • No accessibility features. The user doesn't have the ability to personalise the application with accessibility features. This limits the kind of people who can use the design.
8	<ul style="list-style-type: none"> • Good categorisation. Each tab separates the functions by its associated category. There is sufficient space for each function on the page to reduce wordiness and barriers to use (e.g. language, dyslexia). • Minimal design. In terms of each page, anything on the page is of use and laid out in a simple manner. • All the information is all relevant. All the different categories are completely relevant to the app. 	<ul style="list-style-type: none"> • Unnecessary feature. Includes payment system which is outside of the necessary scope for the design. This feature is a main part of the design, so is a constant issue. • The app is a bit cluttered. A solution would be to have more white space and consider scrolling down the page rather than having all the context at first sight.
9	<ul style="list-style-type: none"> • Colour and symbols indicate incorrectness. Currently, a combination of colours and symbols is 	<ul style="list-style-type: none"> • Colour and symbols indicate incorrectness. Currently, a combination of colours and symbols is

	used to indicate incorrect fields. This increases usability by ensuring all users' inputs are valid.	used to indicate incorrect fields. Users are unable to progress through the input process until this is amended. Due to barriers such as insufficient technological knowledge, colour blindness and low vision, it may be better to indicate errors in a pop-up box with a more descriptive message. <ul style="list-style-type: none"> ● No error messages. If a user gives an invalid input there is no error message to tell them what is wrong.
10	<ul style="list-style-type: none"> ● Help within profile. There is help documentation in the account tab (help screen not depicted). ● Drop down menus. This allows viewers to see what is meant to be inputted as the choices are already there for them. This helps the user understand what is needed. 	<ul style="list-style-type: none"> ● Insufficient help and documentation. There is one place help is available in the profile section, but help is not available elsewhere. Help in context for each screen could improve usability for users with low technological skills

5.2.2 Jet finance

- Todd Wellwood
- Jasmine Dong
- Emilia Greve

Heuristic number	Pros	Cons
1	<ul style="list-style-type: none"> ● Some indication of the user's position in parts of 	<ul style="list-style-type: none"> ● Unclear user position for the entire system. From

	<p>the system. Some headers were used to indicate a user's position.</p> <ul style="list-style-type: none"> • Minimal feedback. There are only a few sections where you receive feedback from the system. For example, how far away from the savings goal you are. 	<p>the depicted system it was difficult to determine a user's position within the system. This is a result of some headers being missing for some functions, No visible navbar (the team used a hamburger navbar) to visually display a user's position, and no confirmation is given after submitting an input.</p> <ul style="list-style-type: none"> • No confirmation. There are no explicit confirmation messages or popups after an action has taken place.
2	<ul style="list-style-type: none"> • Realistic wording. Most of the terms used are related to real-world actions. This positively affects the usability of functions by reducing learning and risk barriers. • Simple wording. The choice of wording is simple and understandable to the user. 	<ul style="list-style-type: none"> • Some unfamiliar terms. There are system terms used to describe possible functions, such as 'Pen and Paper' and 'Quick Finance' were hard to interpret. • Lack of personal wording. The choice of wording is very generic and doesn't feel personal when reading. • Hamburger usage. If users aren't aware of the purpose of the hamburger, they may be confused when they first look at the app.
3	<ul style="list-style-type: none"> • Ability to reset savings goals. The user can reset their savings goal, but only once the calculation has been done. • Easy navigation to hamburger. If the user 	<ul style="list-style-type: none"> • Lack of reversibility. Actions don't have explicit "back" or "undo" buttons. • No option to go back. The app needs to have back buttons so if the user clicks on the wrong button for

	<p>clicks on the wrong section, they can easily escape by clicking the hamburger again.</p>	<p>example 'Start New' instead of 'Copy Previous Week', they can easily go back.</p>
4	<ul style="list-style-type: none"> • Consistent formatting. The solution is relatively consistent in terms of formatting. • Consistent colour theme. The colours blue, green and white are consistent throughout the design. • App is consistent. The app is consistent and the navigation of the hamburger does not change. If the user switches account, the only aspect that changes in the title at the top of the app. The app doesn't use many icons for the user to get confused about. 	<ul style="list-style-type: none"> • Hamburger navbar is not recommended by some. UX professionals (https://uxplanet.org/the-ultimate-guide-to-the-hamburger-menu-and-its-alternatives-e2da8dc7f1db) do not recommend implementing a hamburger navbar for mobile solutions. This is because hamburger navbars do not showcase features well (often used for unimportant functions), are hard to reach at the top of the screen (not often explored), and are incompatible for ios as it clashes with ios navigation. • Hamburger icon positioned top-right. Typically the hamburger icon to access navigation is positioned top-left. The opposing position of the top-right might confuse the user and make an usually automatic action be an action where they have to think about where to press.
5	<ul style="list-style-type: none"> • Dropdown inputs were used. The solution reduces user inputs by implementing dropdown input boxes. • Clear execution. The 	<ul style="list-style-type: none"> • Unable to edit (undo) an expense. Once an expense is created, there is no option to undo or edit any part of it. You would

	<p>button to execute an action is clear to the user.</p>	<p>have to create a whole other expense.</p> <ul style="list-style-type: none"> • No error message. The app does not have any error messages if the user inputs the wrong thing. There is no pop-up such as "Are you sure you want to do this?", so once the user clicks the button, the action will be processed.
6	<ul style="list-style-type: none"> • No information to remember. All user inputs are displayed on the same screen. This means they do not have to recall previous inputs to confidently submit an action (essential to reassure users when dealing with money). • Simple to follow. The app is simple and easy to understand so that the user does not need to remember anything when inputting data. It is clear on every page what is required. 	<ul style="list-style-type: none"> • No help in context. There is no help in context, or anywhere at all in the design. • Lack of explanation. More text could be added in quick finance as the app currently has headers but it might not be that clear to the user what each of them means.
7	<ul style="list-style-type: none"> • Save expenses. Save previous expense submissions to reduce user inputs. • Accessibility. Users can configure accessibility options to make using the app easier to use - especially for those with disabilities. • Tailor frequent actions. The app has a feature 	<ul style="list-style-type: none"> • Cannot add income. While the system allows users to add expenses, they cannot add income. Therefore, calculations for net income cannot be done - which would be of use to users.

	where you can copy the previous weeks. Therefore, the app allows users to tailor frequent actions and does not have to repeat adding an expense every week.	
8	<ul style="list-style-type: none"> • Simple colour scheme. The simplicity of the green, blue and white colour scheme doesn't overwhelm the user. • Contains relevant information. The app only contains necessary information and the design is minimalistic and aesthetically pleasing to look at. 	<ul style="list-style-type: none"> • Overuse of grayscale. The application's dependency on grayscale tones (grey, black, and white are the main colours observed) may not suit our younger audience and may act as a deterrent for a task already perceived as "boring". • Text overload. Some of the pages tend to be overloaded with text. Can be simplified.
9	<ul style="list-style-type: none"> • Hard to make an error. It is hard for the user to make an error that the app will pick up on. For example, if the wrong number is recorded, that is on the user rather than the system itself. 	<ul style="list-style-type: none"> • No error messages. If a user gives an invalid input there is no error message to tell them what is wrong.
10		<ul style="list-style-type: none"> • No help and documentation. There is nowhere to receive help if the user needs it.

5.2.3 Young & broke

- Joel Crampton
- Charles Hughes

Heuristic Number	Pros	Cons
1	<ul style="list-style-type: none"> • Steps remaining shown. The design shows what step of a calculation the user is on, and allows them to step back/forward. This form of feedback lets the user know their "position" in the calculation. • Clear navigation to the next step. The app has clear bold buttons saying "Continue" if the user wants to continue with the next step. 	<ul style="list-style-type: none"> • Unclear user position for the entire system. It was difficult to determine a user's position from the depicted system. This results from no headers indicating where a user was within the system (e.g. creating a personal/shared finance overview) and no visible navbar (the system used a hamburger navbar) to visually display a user's position.
2	<ul style="list-style-type: none"> • System speaks the user's language. Casual and personal wording is used, for example "Does anyone have a long term savings goal" rather than something impersonal like "Savings goal: ". 	<ul style="list-style-type: none"> • Lack of visuals. The app could have more visuals/images to explain certain steps as some users may be visual learners.
3	<ul style="list-style-type: none"> • Back/forward buttons. Allow for easy traversal back to a previous step or forward to the "next" one. Form of "undo"/"redo" buttons essentially. 	<ul style="list-style-type: none"> • No exits. There are no clearly marked emergency exits for user inputs. • Can only go back/forward by one step. The user cannot traverse multiple steps forward or back. For example, if the user wanted to go from step 4 to step 1, they would have to first go through step 3 and 2.

4	<ul style="list-style-type: none"> • Consistent design. The solution is relatively consistent and meets the standards of a website. • Consistent colour theme. Depending on the calculation type; flat or personal. The theme is set to orange or purple. This is consistent throughout the calculation and reminds the user of the calculation type they are performing. 	<ul style="list-style-type: none"> • Home page is inconsistent with the theme. The home page does not fit into the overall theme of the site. It uses different colours not seen anywhere else, and is quite cluttered compared to other pages.
5	<ul style="list-style-type: none"> • Dropdown inputs used. For some text-based inputs, dropdowns are used so users cannot enter invalid values. • Clear execution. The button to execute an action is clear to the user. 	<ul style="list-style-type: none"> • Text inputs have no error prevention. The solution requires an excessive amount of inputs. While it has implemented drop-down boxes in some inputs, there are thirteen different pages to enter a group budget overview and six different pages to enter a personal budget overview. This increases the probability of errors
6	<ul style="list-style-type: none"> • No information to remember. Between steps there is no information to remember. The application keeps track of any information entered in previous steps. The exception to this is the user's password, which they are expected to remember. • Back arrow as an option. The back arrow allows the user to go back and see 	<ul style="list-style-type: none"> • No help in context. There is no help in context, or anywhere at all in the design. • Lots of screens. All user inputs are displayed on separate screens (thirteen different pages to enter a group budget overview and six different pages to enter a personal budget overview). This means they must recall previous inputs to confidently submit an

	<p>what they have previously inputted.</p>	<p>action (essential to reassure users when dealing with money).</p>
7	<ul style="list-style-type: none"> • Personalisation through saved calculations. Users can save their calculation to their profile to personalise their experience and save time by not having to recalculate for subsequent calculations. • Save calculations. Save previous shared/personal overview submissions, which reduces user inputs. • Clear process to complete tasks. The app is good for inexperienced users as each step is clearly titled as to what they need to input. 	<ul style="list-style-type: none"> • No editing. Users cannot modify existing overviews, which means users have to repeat the entire input process to modify an existing graph. • No shortcuts. There are no shortcuts that have been enabled for regular users to speed up their experience. • Repeating process. The app doesn't allow experienced users to receive what they have entered in the previous weeks. In order to input another expense, they have to go through the step by step process again.
8	<ul style="list-style-type: none"> • Minimal design. Anything that is unnecessary is not included. Easy and simple for the user to use. • Contains only necessary information. The app is very minimalistic and only shows what is necessary. For some users, this will make it way easier to use and they won't be distracted by unnecessary features. 	<ul style="list-style-type: none"> • Lack of colour. The colours used in this application are mainly grayscale tones (grey, black, and white are the main colours observed), may not be suitable to our younger audience and may act as a deterrent for a task already perceived as 'boring'. Additionally, the use of 'emoji' images does not convey an appropriate level of professionalism, especially in dealing with finances and personal information.

		<ul style="list-style-type: none"> ● Lack of visuals. There is a lot of white space which they could utilise more. It currently only has a title and input boxes so they could add more visuals so the user is more drawn to the app.
9	<ul style="list-style-type: none"> ● Hard to make an error. Like the other designs, it is hard for the user to make an error which the app will pick up on. For example, if the wrong number is recorded, that is on the user rather than the system itself. 	<ul style="list-style-type: none"> ● No error messages. If a user gives an invalid input there is no error message to tell them what is wrong.
10	<ul style="list-style-type: none"> ● Highlighted keywords. Helps users to easily know what is required for the given section. Reduces the reading required. 	<ul style="list-style-type: none"> ● No help and documentation. Does not contain a help screen nor documentation to assist users in navigating the functions of the solution.

6. Usability Test plan

6.1 Methodology

Given the designated group of five users, we intend to test the usability and functionality of our prototype system. We hope to identify areas of improvement and have outlined a testing plan that involves assessing the time it takes for users to navigate through a task, feedback from the system interactions generated from completing the task and an interview discussing the overall system.

Users

- Ella Spies
- Antoinette Trott
- William Kilty
- Boyng Wang
- Anish Sodhi

6.2 The Script

As a group, we agreed that rather than having a word-for-word script for the testing process, we would instead have a list of bullet points we needed to bring up. Our justification is that it would allow a more natural and better flow. Furthermore, as we had one more member in our team we had to have two people test one user, and the lack of a strict script allowed for more flexibility. Our bullet points of information we needed to say were as follows:

- Briefly introduce our financial system
- Let the user know we would be timing them, but reinforce not to rush the testing
- Introduce each issue and explain to the user what they were required to complete
- Ask the questions listed in Transcript and Interviews (7.2)
- Thank the user for being a part of the testing process

Our script, although simplistic, allowed us to cover everything we needed. A benefit of having an uncomplicated script is that we were able to communicate with our tester more efficiently. Using technical language may have resulted in miscommunication and thus confusion, where the tester may have misinterpreted what they were required to do. Furthermore, by taking a more colloquial approach, we were able to ask each tester

to expand and clarify their remarks. This was particularly useful in 7.2, Transcript and Interviews, as we were able to receive more specific feedback to apply.

It should be mentioned that when each user began testing our issues, we remained completely silent so that we would not interfere with the process in any way and not affect the results of the testing. This ensured the integrity of the testing process.

6.3 Roles

After discussing possible opportunities for testing with our users, we identified that due to our varying schedules it was best to test the prototype one-on-one.

We had outlined the task, generated a format for recording the results and had a structured script in which all team members were involved in developing - as a result all members were equipped to collect test results individually.

Each team - member was then responsible for summarising their findings and populating the 'Issues' table (Issues [7.3]).

User and team member pairs

	Team member(s)	User
1	Marina Suban	Antoinette Trott
2	Todd Wellwood	Ella Spies
3	Joel Crampton & Emilia Greve	Anish Sodhi
4	Jasmine Dong	William Kilty
5	Charles Hughes	Boyng Wang

6.4 Tasks

What are we testing?

We want to test the full functionality of our app. We plan to use our scenarios as a guideline and break them up into tasks for the user to complete. E.g, "The user should create and register a new account". We also have added some areas to test in order to make sure we are covering each domain of our application. We believe having the most data possible, will allow us to maximise coverage of our apps, and identify as many issues/areas of improvement to be covered. We plan to only intervene if the user asks

for help, this will allow us to maximise the data we receive from the users and identify the most issues.

Finally, we plan to ask the users some generic questions, including if they have any additional feedback for us. Combining this with our specific question will allow us to get as much information out of the testers as possible, as our question scenarios will make them think of specifics, and us asking for any generic feedback will give them the opportunity to say things they otherwise may have reserved, not felt comfortable mentioning or only thought of after the user testing.

6.5 Metrics

How are we measuring? (e.g. SUS, NASA TLX)

Reference:

<https://xd.adobe.com/ideas/process/user-testing/sus-system-usability-scale-ux/>

We used **S**ystem **U**sability **S**cale (**SUS**) and gave our users a 10 question SUS questionnaire to fill out after looking through our app. SUS was beneficial for us as it enabled us to get reliable and repeatable feedback and gave us an indication of how usable our app is. SUS is a cheap and quick test as it requires little to no resources and the quiz can be completed within a couple of minutes. We followed the rating system metric which ranged from strongly disagree to strongly agree. We also measured the SUS score by the following method:

- Add up the total score for all odd-numbered questions, then subtract 5 from the total to get (X).
- Add up the total score for all even-numbered questions, then subtract that total from 25 to get (Y).
- Add up the total score of the new values (X+Y) and multiply by 2.5.

Rating system

1. Strongly disagree
2. Disagree
3. Neutral
4. Agree
5. Strongly agree

SUS questions		Users					Average	Total
		Ella Spies	Antoin ette Trott	William Kilty	Boyng Wang	Anish Sodhi		
1	I think that I would like to use this app frequently.	4	5	5	4	5	4.6	23
2	I found the app unnecessarily complex.	2	2	1	1	2	1.6	8
3	I thought the app was easy to use.	5	4	3	4	5	4.2	21
4	I think that I would need the support of a technical person to be able to use this app..	1	2	1	1	1	1.2	6
5	I found the various functions in this app were well integrated.	5	4	4	4	4	4.2	21

6	I thought there was too much inconsistency in this app.	2	3	2	2	2	2.2	11
7	I imagine that most people would learn to use this app very quickly.	4	5	5	5	5	4.8	24
8	I found the app very cumbersome to use.	1	2	1	1	2	1.4	7
9	I felt very confident using the app.	4	4	4	4	4	4	20
10	I needed to learn a lot of things before I could get going with this app.	2	1	2	2	3	2	10

SUS Scores

SUS scores		New values
1	$23 - 5 = 18$	$(18 + 17 \times 2.5) = 87.50$
2	$25 - 8 = 17$	
3	$21 - 5 = 16$	$(16 + 19) \times 2.5 = 87.50$
4	$25 - 6 = 19$	

5	$21 - 5 = 16$	$(16 + 14) \times 2.5 = 75$
6	$25 - 11 = 14$	
7	$24 - 5 = 19$	$(19 + 18) \times 2.5 = 92.50$
8	$25 - 7 = 18$	
9	$20 - 5 = 15$	$(15 + 15) \times 2.5 = 75$
10	$25 - 10 = 15$	

Having an overall SUS score between 75 - 92.50 illustrates that the app is good in regards to usability and they would suggest it to their friends. A score of 75 indicates that work could be improved. However, by discovering the issues that the user has told us about through user testing, once these issues are implemented, hopefully, our score will increase.

6.6 Usability Goals

What is the level we are happy with on the above metrics?

The usability goals stated above are in line with the System Usability Scale (SUS). We made sure to ask a variety of questions in our feedback analysis, corresponding with the list of objectives to be performed by the testers. allow us to gain a significant understanding of their overall experience. To calculate the scores obtained in our analysis, we will perform a simple mathematical function on each score to obtain a percentage of satisfaction. Each answer to the question will be given a numerical value as follows, and this would depend on the nature of the question.

Example: Question 1	
I think that I would like to use this system frequently	<ol style="list-style-type: none"> 1. Strongly Disagree 2. Disagree 3. Neutral 4. Agree 5. Strongly Agree

The sum of the values of all ten questions will then be multiplied by 2.5, which will convert the answers from a range of 0-40 to 0-100. A SUS score above 68 is average and below this is below average. Considering the group who will be testing our

prototype will be small. It is certain that our results will not be able to meet an accurate representation of what a larger population would. Therefore, it is important to expect the efficacy of our design in terms of the user's experience. We want to have the best possible experience for our prototype therefore we will aim for a SUS score around 70-75 to be the desired result. We believe that setting the bar high will show that although not at all perfect, the experience was at an initially good standard. Therefore, allowing our group to strive further to make our application as good as it can be.

6.7 Problem Severity

How bad are the problems?

We determined how bad our prototype's problems were in a similar way to how we identified what level we were happy with for our SUS metrics in the above section (6.6). As previously stated in that section ***“we will aim for a SUS score around 70-75 to be the desired result”***. Therefore, any SUS score below 70 should be considered undesirable, or bad. If we take a look at the SUS scores from the “How are we measuring?” section (6.5), we can see that the SUS scores achieved were: **87.50, 87.50, 75, 92.50 and 75**. Against our strict metrics, none of the problems that our testers identified were bad, which is encouraging and shows us we successfully developed an initial prototype for users to manage their finances at a desirable level.

While none of our problems are considered bad, we didn't achieve a SUS score of 100 in any areas. Therefore, we cannot say our prototype is perfect in any way, and it still needs work. This is an important way to think, it would be a poor attitude for designers to build a product which only satisfied some users (SUS score of 70-75). We instead want to satisfy everyone who uses our product (SUS score of 100), and therefore even problems that aren't considered bad should be improved.

7. Usability Test Results

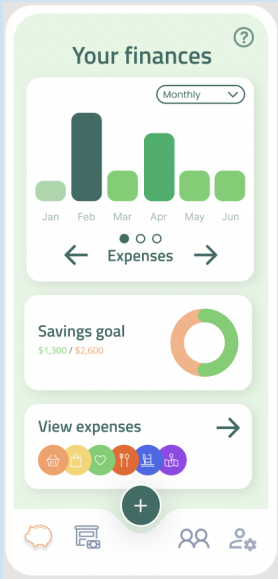
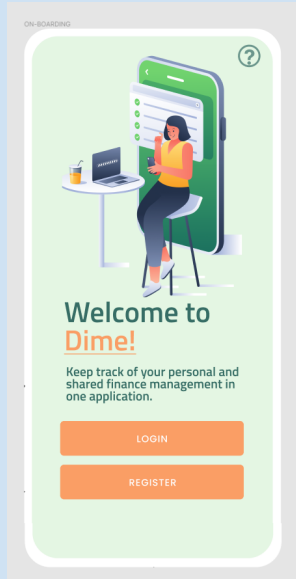
For our testing, we decided to break the process down into during and after sections - with certain steps in each.

During

1. Define actions for a main task.
2. Record what subactions were done during the scenario.
3. Record time for actions
4. Record anything said
5. Ask questions

After

1. Summarise into issues
2. Assign severity value to each issue
3. Create solutions for each issue.

Starting points	
<u>For all (except create an account)</u>	<u>To create an account</u>
	

7.1 Task duration

Main task	Actions	Times (seconds)				
		Ella Spies	Antoinette Trott	William Kilty	Boyng Wang	Anish Sodhi
Create an account (Login Domain)	Press “register” button	0:03	0:02	0:02	0:04	0:01
	Complete register form	0:22	0:15	0:03	0:05	0:20
	Email confirmation (hidden button)	0:33	0:10	0:05	0:08	0:03
	Press “start” on the registration confirmation screen	0:33	0:05	0:05	0:06	0:15
	Complete accessibility quiz	1:00	0:30	0:17	0:24	0:10
	Total	2:31	1:02	0:32	0:47	0:49
	Notes	Accessibil ity should work	Liked how the accessibili ty button was available in the registratio n. Could be improved by adding an accessibili ty button	Pretty comprehe nsive	Good colour use. Overall design is very clean.	Was confused when he encounter ed error messages through the hidden button (unintend ed)

			on every screen in the domain.			
Adding an expense	Press the “+” button	0:17	0:04	0:20	0:11	0:01
	Press “add expense”	0:22	0:02	0:08	0:09	0:01
	Complete expense form	0:31	0:14	0:10	0:11	0:06
	Press “done” on the expense confirmation screen	0:35	0:02	0:03	0:04	0:03
	Total	1:45	0:22	0:41	0:35	0:11
	Notes	<p>Couldn't find it at first, felt the save transaction button to be confusing</p> <p>Centre the buttons on the add expense screen</p>	<p>Thought + icon was intuitive personally but felt that some may find it hard if they have low technological knowledge. Adding options for button labels in accessibility may be helpful.</p>	<p>Make it more clear what the add button does and that it is even a button</p>	<p>Some buttons are vague. Can be seen as too much information on one page. Add a back arrow to go back to the main expense page.</p>	<p>Confused on how to return back to the personal finance screen after the task (piggy bank icon).</p> <p>Potentially add labels to help users identify navigation. One-off</p>

			Overall she thought the function labels were clear and the process easy to follow.			(tutorial at start)
Update personal savings goal	Press the "savings goal" widget	0:02	0:08	0:02	0:06	0:01
	Press the "edit" button	0:03	0:02	0:04	0:05	0:02
	Complete savings goal form	0:05	0:05	0:03	0:06	0:04
	Press "done" on the savings goal confirmation screen	0:22	0:03	0:15	0:11	0:03
	Total	0:32	0:18	0:24	0:28	0:10
	Notes	Error correction had no way out of it. Can't go back to piggyback without first clicking	Icon might need to be changed to reflect the functions in the tab better. Missing function to set	Perfectly clear - clicking on error button means that you can't save anymore but that is a figma issue	Really well done. Error button glitches. Text to tell the user what each colour means. Keep the group and personal	Found nothing wrong - occasionally got mixed on on error messages but this is simply due to our prototype

		off something	contribution rate for the saving goal. There is no function which allows users to input if they met a calculated contribution rate for a time period.		sections similar in style. Both have completely different layouts.	
Enable grayscale in the accessibility menu	Go to the settings menu	0:02	0:03	0:04	0:06	0:02
	Press the "accessibility" button	0:06	0:03	0:03	0:05	0:02
	Switch on grey scale	0:11	0:03	0:04	0:08	0:03
	Total	0:20	0:09	0:11	0:19	0:07
	Notes	Expect the whole button to be a button, instead it's only the tiny arrow.	Where it's expected to be (Easy to access) - found labels helpful. Dark mode is an	Easy to access - likes the option for accessibility when you log in	Pretty good design. Clear, easy to get to.	Said it was easy to navigate to, pretty straightforward

			increasing ly popular feature this could be added to the 'customis e accessibili ty' function.			
Accesses saved recurring expenses	Press the "+" button	0:39	0:26	0:21	0:30	0:45
	Press the "saved & recurring button"	0:26	0:04	0:26	0:02	0:01
	Press the "recurring tab button"	0:04	0:07	0:20	0:10	0:05
	Switch from "income" to "expenses"	0:04	0:03	0:15	0:05	0:01
	Total	1:05	0:40	1:22	0:47	0:52
	Notes	Expected it to be under the "view expenses" reachable by the piggy bank.	Didn't expect it to be under add expense. Felt that recurring expenses should be in the group/per sonal account that it's	Abit obscure to find it. Have another link to it rather than in the '+' button	Drop down menu in singular is pointless for only two options. Cross doesn't work .	Got confused between 'view expenses' on the home screen and thought it would be here, rather under the

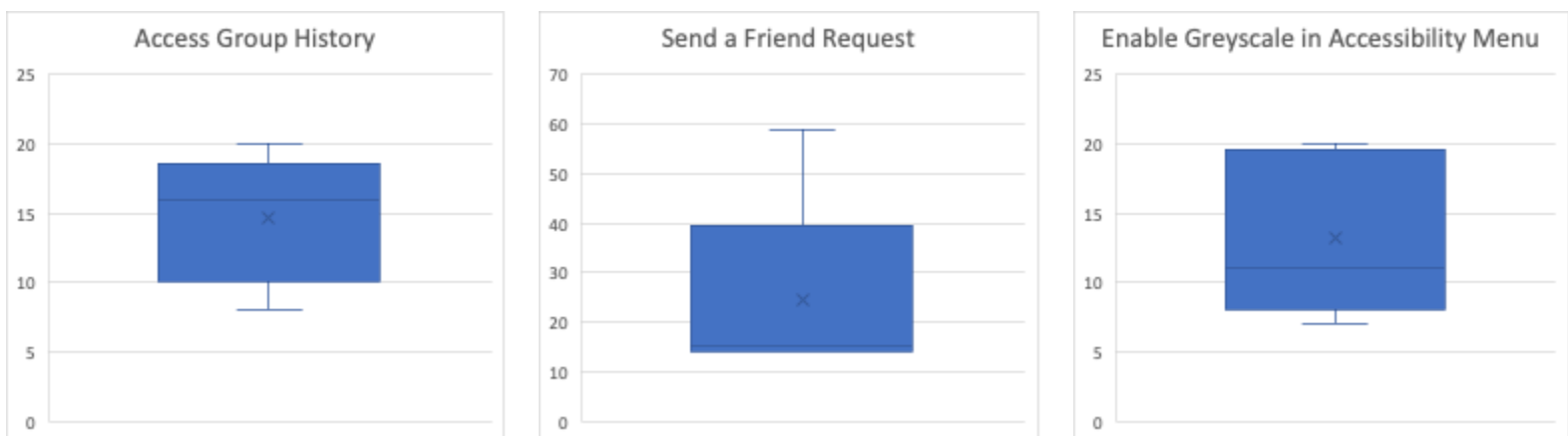
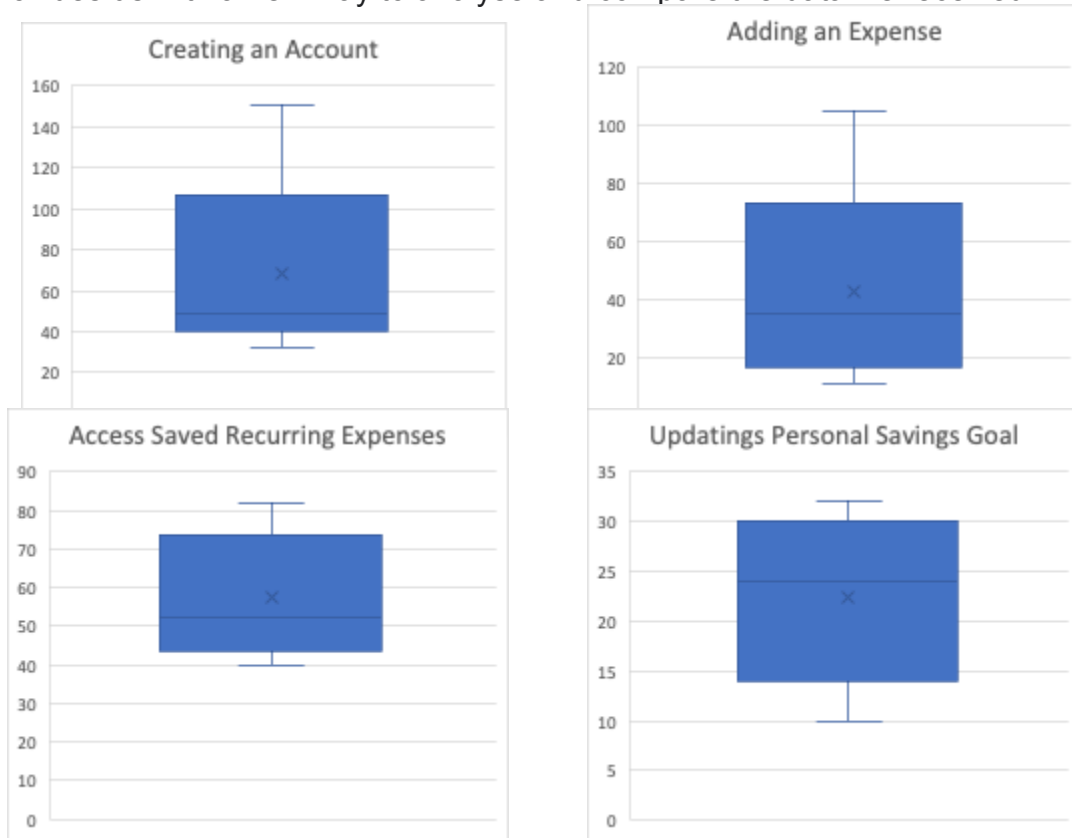
			relating to.			'+' icon. Once here was able to navigate well.
Access group history	Press group button on navigation bar	0:04	0:07	0:07	0:05	0:06
	Press a group widget from the menu	0:06	0:05	0:04	0:06	0:04
	Press the "history" tab from the header bar	0:08	0:08	0:05	0:06	0:02
	Total	0:08	0:20	0:16	0:17	0:12
	Notes	Needs a better button for the group, the two people associated more with groups. What is the button meant to be? Three people for groups?	Self explanatory and easy to navigate. However, the user felt That the function should be represented with a different icon as it could be mistaken for the friends tab.	Fairly straight forward	Larger text on the income labels. Icons in the squares. Make the colours a little darker. Keep the buttons similar.	Got confused between friends and group icons. Went to friends first. Once in groups, was able to navigate well and "well laid out".

			<p>No function to filter expenses by category. This is a functionality agreed upon by the group.</p> <p>Irrelevant transactions in history tab ('not involved' transaction is irrelevant) - unnecessary.</p>			
Issue a friend request	Press on the friends button on the navigation bar	0:01	0:02	0:03	0:05	0:02
	Switch to the requests tab	0:02	0:04	0:04	0:05	0:02
	Search for a username in the search bar	0:26	0:04	0:03	0:04	0:05
	Press the + button to	0:30	0:04	0:04	0:06	0:06

	create the request					
	Total	0:59	0:14	0:14	0:20	0:15
	Notes	Clicking request, while having searched for a user, should take back to currently pending friend requests	Intuitive. However, the user felt That the function should be represented with a different icon as it could be mistaken for the friends tab.	Perfect - exactly what you would expect	Really well done clean design.	Originally tried to add a friend request in the friends table, however quickly realised the mistake and changed to request.

Graphical Representation

We have collated our timed results for each issue into a box and whisker graphs, where the y axis is measured in seconds. Turning out results into a graphical representation of data provides us with a new way to analyse and compare the data we received.



7.2 Transcript and Interviews

Questions	Answers				
	Ella Spies	Antoinette Trott	William	Boyng Wang	Anish Sodhi
General feedback during testing	<p>Buttons on taskbar should be a tad higher</p> <p>Home piggy bank should be the centre if that's our home main home.</p> <p>Excellent colour scheme,</p> <p>Likes the dimple.</p>	<p>Clearer icons - maybe more similar to existing apps (friends/group).</p> <p>Likes add expense being in the middle because can be accessed on every page.</p> <p>Maybe move icons around to distinguish between group/personal tab.</p> <p>Likes help button - personalised to every page.</p> <p>App is extremely clear.</p>	<p>Make navigation bar all line up - consistency</p> <p>Clear up plus button to make it more obvious what it does</p>	<p>Navigation is pretty good. Has good. Can access any page from the taskbar.</p> <p>Colour scheme is the best part of the prototype.</p>	<p>A 'pretty good prototype, not a lot of flaws'. A bit of confusion with the menu icons.</p>
Do you have any questions	N/A	N/A	N/A	N/A	How professional are you

about the prototype?					trying to make this?
Tell us your thoughts whilst you were testing the prototype?	<p>Error handling in a weird spot <- figma being weird.</p> <p>Noticed buttons jumping around</p>	Good experience - found the application easy to use.	Have to think about what the buttons mean. Once navigating through it more, it became more clear what each icon meant.	Really easy to use. Not loads to improve upon.	<p>Found no irregularities.</p> <p>Pretty consistent layout. Was really impressed with the prototype.</p> <p>Execution wise every has been implemented .</p>
Rate your experience testing the prototype from 1 to 5, where 1 is very negative, 3 is neutral, and 5 is very positive.	4 but I am mainly blaming figma here.	4-5 i don't see any major issues - just minor changes.	4 for reasons I have discussed prior	5, Really intuitive to use.	4, based on those mentioned earlier, would have given a five but marked down due to navigation confusions.
What are some parts of the prototype that you thought were difficult to use or were not clear?	<p>View recurring expenses should be moved.</p> <p>Why is their x on the select option screen</p>	<p>View recurring expenses should be moved.</p> <p>Group and friends icon unclear.</p> <p>Missing function to set contribution</p>	<p>View recurring expenses should be moved.</p> <p>Another minor - make it more clear that it is groups (change icon)</p>	Not really, most of the sections were easy to understand.	<p>Some of the labelling was misleading -naming of it</p> <p>Icons were misleading in the sense of groups and friends</p> <p>Menu bar changes in size.</p>

		<p>rate for the saving goal.</p> <p>Missing function to see income summary or saving summary in personal and group tab.</p> <p>No function to filter expenses by category in the group tab.</p> <p>Personal domain missing income statement function (observed in 'Group Finance Overview').</p>			<p>Piggy bank isn't clear what it should leads to.</p> <p>Likes the idea of labels for the icons.</p>
What are some parts of the prototype that you thought were easy or clear?	Adding expenses and income plus is very central.	Clear and big font, illustration and icons.	Piggybank is clear. Very intuitive use of arrows and drop down buttons.	It was hard to find sections that were not easy or clear.	<p>Adding new expenses / incomes was pretty clear.</p> <p>Viewing expenses and savings goals was very clear, directly there. Home Screen shows all</p>

					<p>correct information.</p> <p>Liked graph on main home screen.</p> <p>Like setting sbe easily accessible.</p>
<p>If the prototype was fully functional, would you use this system to manage your finances? Why or why not?</p>	<p>I think I would probably give it a try for sure, but most likely wouldn't because I don't like to use apps.</p>	<p>Currently I don't use a solution but I'm quite happy to use this as its easy to use and aesthetically pleasing. I would recommend it especially due to the accessibility features.</p>	<p>Personally, I would not as I don't really use an app to manage my finances.</p> <p>If I did, I would use this app.</p>	<p>Yes I would use it. Seems like a very simple and useful app.</p>	<p>Wouldn't use it, but only because he doesn't manage his finances.</p> <p>For the sake of the project if he needed a financial plan he definitely would.</p>
<p>Anything else you'd like to add?</p>	N/A	N/A	N/A	N/A	<p>Loved the help screens!</p>

7.3 Issues

U = how badly does it affect usability

F = how badly does it affect the core function of the solution

Issues	Team members						Average
	Marina	Todd	Jasmine	Joel	Charles	Emilia	

View recurring expenses should be moved (Hard to find).	U: 3/5	U:5/5	U: 4/5	U: 5/5	U: 3/5	U: 4/5	U: 4
	F: 1/5	F: 1/5	F: 1/5	F: 1/5	F: 1/5	F: 1/5	F: 1
	If users can't find the function this demonstrates a strong usability issue but the functions are still fit for purpose. This function is only used by experience persons	My user felt frustrated, and didn't know the help section existed.	My user struggled to get to this section and had to look through most tabs. Doesn't relate to adding an expense or income.	Anish did struggle to find "recurring expenses" at first. He thought it was under the "expenses" section for personal finance. Distinction between these two should be given.	My user had minor difficulty finding the recurring expenses. But although not a detrimental problem it is still an issue which needs to be sorted to get the desired product.	My user firstly struggled to access the main page, but once he was able to navigate to the recurring expenses page he was able to navigate better. Some ways to better display this information to the user is needed.	
Group and friends icon unclear. Missing function	U: 5/5	U: 5/5	U: 5/5	U: 5/5	U: 4/5	U: 5/5	U: 4.83
	F: 1/5	F: 1/5	F: 1/5	F: 1/5	F: 2/5	F: 1/5	F: 1.3

to set contribution rate for the saving goal.	If users can't find the function this demonstrates a strong usability issue but the functions are still fit for purpose. This function is used by all personas	Functionality is there just hard to find, currently the button doesn't look like a button	Could change the icon to make it more clear that it is for groups. My user knew it was a button but was unsure of the purpose till he clicked it.	Anish also got confused between these two icons. He suggested alternative icons of an icon with three people for group, and an icon with one person and a plus (+) sign for friends.	The function is reasonably hard to find. The user found some of the buttons misleading and could be improved upon. This is a vital part of the system.	My user found the icons within the menu hard to differentiate, as in that some icons could correspond to other domains. This was especially an issue between groups and social.	
Missing function to see income summary or saving summary in personal and group tab.	U: 5/5	U: 2/5	U: 4/5	U: 3/5	U: 2/5	U: 3/5	U: 3.17
	F: 5/5	F: 5/5	F: 5/5	F: 5/5	F: 4/5	F: 5/5	F: 4.83
	The objective of the solution is to summarise financial information without it	We should definitely have this functionality, while it didn't show up under my testing it	My user did not identify this issue when navigating through the app. However	This issue was not commented on by Anish, however it is something that	This issue wasn't a big problem with my user. He didn't have any major problem	This was not identified as an issue with my user, however, this should be a	

	users may not use the app as it does not meet their expectations.	feels very inconsistent with expenses.	, I believe we could implement this function.	should be addressed.	s with it. Although it is still an issue that we can't look to be able to meet the user expectations	necessary feature to implement.	
No function to filter expenses by category in the group tab.	U: 2/5	U: 3/5	U: 2/5	U: 2/5	U: 1/5	U: 3/5	U: 2.17
	F: 2/5	F: 2/5	F: 2/5	F: 2/5	F: 2/5	F: 2/5	F: 2
	This is inconvenient and inconsistent with the personal tab.	Inconsistent with rest of the app	Inconsistent and I like the way the categories are listed on the app.	This was not commented on by Anish, and also wouldn't benefit usability and core functionality much.	This problem wasn't addressed by the user. However would be a nice touch to add in order to help sort out the section.	This is inconsistent within our system.	
Personal domain missing income statement function (observe	U: 5/5	U: 5/5	U: 5/5	U: 5/5	U: 5/5	U: 4/5	U: 4.83
	F: 5/5	F: 5/5	F: 5/5	F: 5/5	F: 4/5	F: 5/5	F: 4.83

d in 'Group Finance Overview').	The objective of the solution is to act as an income statement generator without it users may not use the app as it does not meet their expectations.	Can't generate income statements is pretty bad, feels weird when we make an expense one.	My user did not pick up on this. However, it is important that the user can generate an income statement to see an overview of how much they are earning.	This wasn't noticed by Anish, but needs to be implemented.	My user found the difference confusing between the group having a statement but the personal not. Is an essential aspect of viewing finance.	This was not noticed by my tester, however to ensure the system is cohesive this needs to be implemented. It is an important function as it corresponds to the ability to overview the budget.	
Help section isn't introduced to the user	U: 2/5	U: 3/5	U: 2/5	U: 1/5	U: 1/5	U: 2/5	U: 1.83
	F: 1/5	F: 1/5	F: 1/5	F: 1/5	F: 2/5	F: 2/5	F: 1.3
	Users should assume from the help icon however this may be better highlight	Currently the system doesn't introduce the user to this menu at	The help icon is clear and as a user, I would assume that I need to	My user didn't point this out as an issue, and I don't believe it needs to	My user did recognise this issue however he mentioned it	My user did not immediately notice the help screens, but this was	

	ed in a different colour or with a pop up upon registration. This may mean users are not able to find the functions they need.	all, this means when they're frustrated they have no idea where to go for help in the app.	click it if I need help. However, we could introduce it when they make an account.	be addressed either.	being a very minor aspect. In most applications the system assumes the user already knows how to view a help section.	primarily because he did not need any help. However, once it was pointed out to him he really enjoyed this feature. To help solve the issue of the help icon not being clear a brighter colour for example could be used.	
Friends search isn't properly implemented	U: 5/5	U: 5/5	U: 5/5	U: 4/5	U: 3/5	U: 5/5	U: 4.5
	F: 3/5	F: 3/5	F: 3/5	F: 3/5	F: 4/5	F: 3/5	F: 3.17
	Reduces functionality of the friends tab and dead ends will confuse users.	Users should be able to add friends easily, this	The user had no issue searching and adding a friend. I believe it just	The user had no issue with this, but if this is an issue it should be	If the friend section isn't functional then there would be a	My user did not experience any issues with the friend search, however,	

		section currently has inconsistent back button behaviour	wasn't finished and figma had some issues as we wanted to emphasise the error message for the prototype	addressed.	problem with the overall system's efficiency. Would confuse many users as to why this section doesn't work.	this could be because we mentioned that a hidden button was implemented and to avoid it as this was not relevant to the testing process.	
--	--	--	---	------------	---	--	--

8. Solutions to Issues

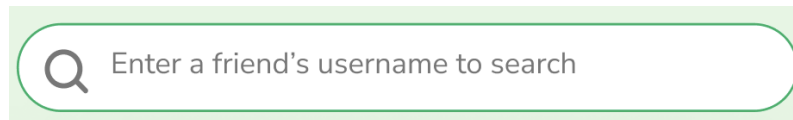
8.1 Friends search isn't properly implemented

Emilia Greve

The rationale for this update and how it benefits the project

The first issue which I was required to update was to fix the subtext within the search bars. Previously, the text was in a small, light grey font, with a brief description of what the user should input as a way of error prevention. However, after reviewing the feedback from the testing process it was found that this needed to be portrayed more clearly. An additional issue that was found within this search bar was how error screens were implemented. It occurred a few times within the process where when the users tried to search for a username they were instead shown an error screen.

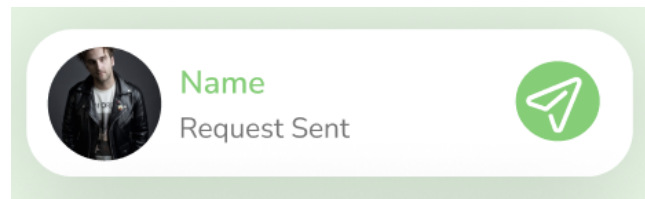
To fix this, I first changed the colour of the text in the search bar. What was once a light grey, I changed to the matching colour of a darker grey within the system. This helped with resolving the error but I was also able to ensure that the cohesiveness remained. The fix is shown below.



In regards to the second issue within the search bar, there is no fix required. The particular problem that users encountered in testing was the hidden button which I had previously implemented for project 1. The purpose of this button was to show how errors could be handled within the system. If a user were to use this system they would not be taken to the error screen simply by clicking the search bar, but only if an actual invalid username was inputted. Thus, this issue originated from a lack of communication with the testers, and I believe is not a system issue to be resolved.

Another issue which was identified during user testing was the icons and subtext within the request section. It was identified that the subtext did not accurately reflect the action which was to be completed. This was an important issue to fix as it currently created confusion with the users. Furthermore, the need to change the icons to reflect the action was required. Here, I have used an aeroplane icon which is commonly associated with things being sent as a way to identify that the request has been successful.

The last issue to which I was assigned was fixing the screen changes when a user clicks elsewhere. To expand on this, we discovered that should a user search for a friend for example, and then click on the requests tab, should they go back to the friends tab instead of being taken to my main screen, it should take them back to the friend search. I found this a really interesting issue which was identified as it was not something which I had previously recognised. However, I understand how it will help prevent unnecessary repeated actions for the user, and also ensure that the states within this domain remain consistent.



8.2 View recurring expenses should be moved

Todd Wellwood

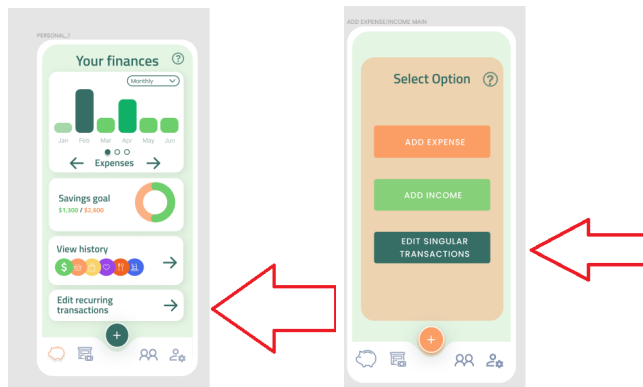
The rationale for this update and how it benefits the project

The current problem our users faced during testing is not being able to locate the recurring expenses and singular transactions. Before any changes, they were both located under the “+” tab, which our users didn’t find very intuitive. This lack of intuitiveness is likely caused by our users seeing that as an add action, that shouldn’t be gone to for viewing already processed information.

To remedy this, I moved out these menus for editing expenses to the personal, and group screens. The reason for selecting these screens is they already act as our “viewing” screens for information. This change makes it simple for users to find and use, as well as using the design pattern of “Input and output” for clear segmenting between “features” and “viewing” in our app.

After updating the Figma and user testing this again some other users found it was better to keep the singular transactions under the addition menu and recurring under a separate menu, as the singular transactions could already be edited in the view history section with the current implementation.

After changes



8.3 No function to filter expenses by category in the group and personal tab

Charles Hughes

The rationale for this update and how it benefits the project

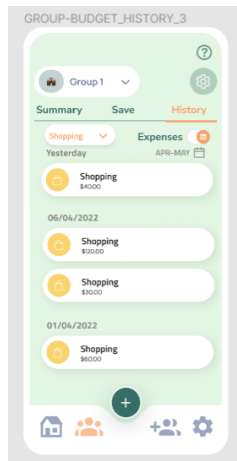
History filter expenses feature

The issue in which we found users experiencing the group section of the prototype would be the ability to select specific categories. They found that the history would become very unorganised and difficult to visualise once a lot of expenses and income have been added into the account. Therefore as a group, we decided to solve this by:

- Sorting by Expense/Income

As seen in the new history screen we have added a button to switch from expenses to incomes.

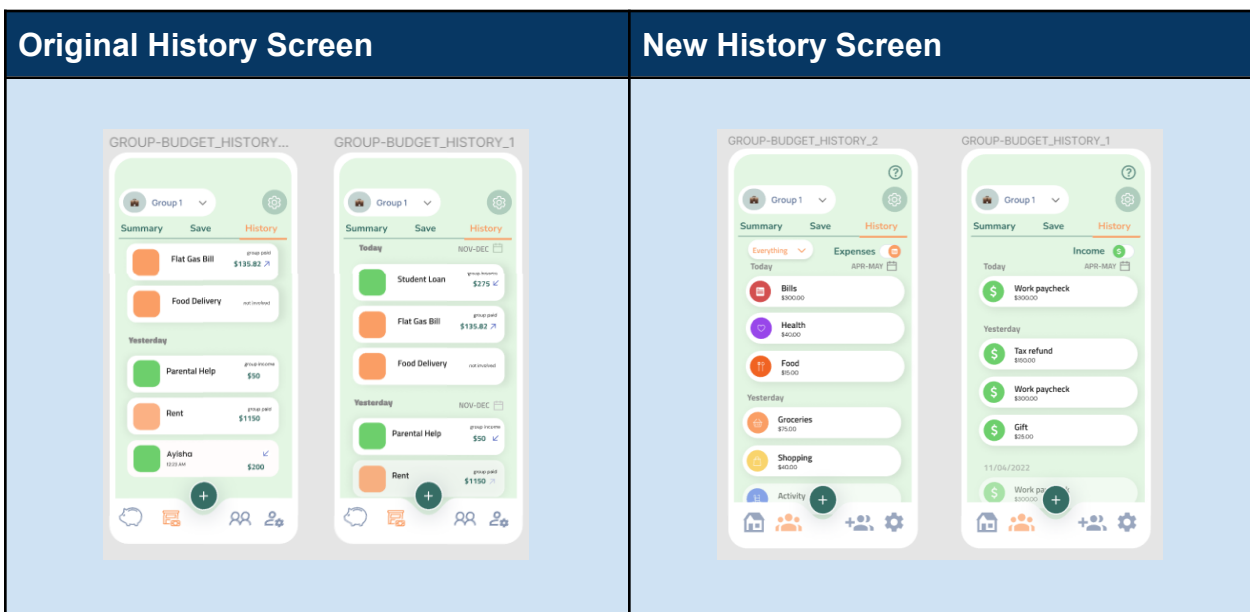
- Sorting by Categories:



Going even further there can also be a filter that allows the user to see the history with a selected expense. This adds to the modularity of the section giving the user more functions to help them view information more efficiently.

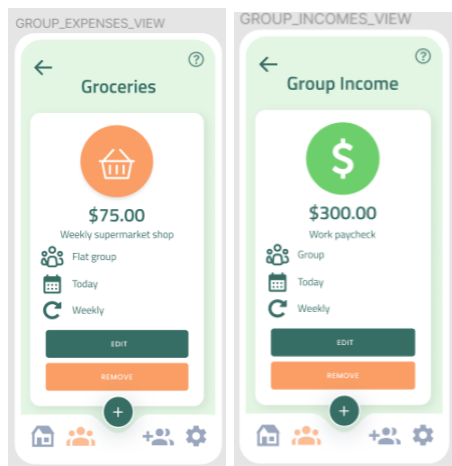
- Style of the payment widgets.

Keeping it similar to personal I changed the squared widgets of the income and expenses to a more rounded design. We decided that the new design fit more with our prototype's new design as well as keeping it similar to the personal history section. Icons have also been added to the widgets to further decrease user confusion about what to do.



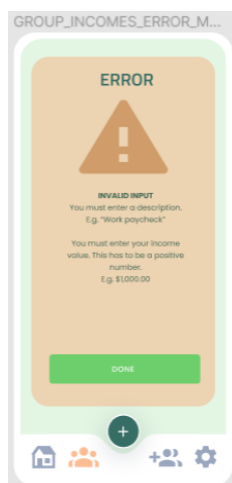
Selecting and editing group expenses / incomes

One of the big aspects of the feedback from the testing is improving the modularity of the prototype. We improved this by increasing the accessibility of areas the testers highlighted. For the group/ personal section, it was being able to access and edit the expenses/incomes. For this, we made the widgets in the history section transition to an expense in which the user can have the option to remove or edit. I made sure that only group expenses are accessible in the group section and that personal expenses are accessible in the personal section. This will help the user to identify which expense is for which type of finance.



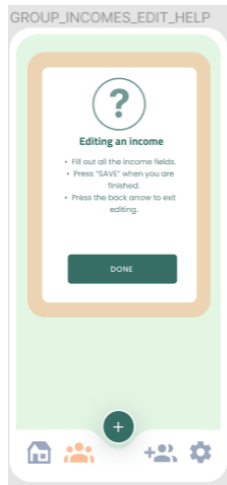
Additional user error prevention for these edits

New error screens have been included in the system to avoid user errors. All the new user inputs which can have user errors have screens to stop this. These error screens are similar to our old error system which had good feedback from the testing.



Additional support/ help screens

New help screens have been added throughout the group section to help the users with each page they are on. The new modularity of editing incomes from the group and personal means new help pages for these sections.



8.4 Personal domain missing income statement function (observed in 'Group Finance Overview')

Marina Suban

The rationale for this update and how it benefits the project

When asked to explore the system users identified that the income statement generated was not aligned with the functionalities of our solution, inaccessible in the personal tab nor was it formatted to reflect an industry-recognised 'Income Statement'. The purpose of our application is to make finance management convenient and accessible. Because our 'Income Statement' is not meeting its purpose it negatively impacts the functionalities of our system.

To resolve this issue I analysed our old 'Income Statement' screen

Firstly, I identified that the original screen was missing a header. This creates a barrier of use for users who are risk-averse or have low technological or financial knowledge as users may not understand the purpose of the screen which is

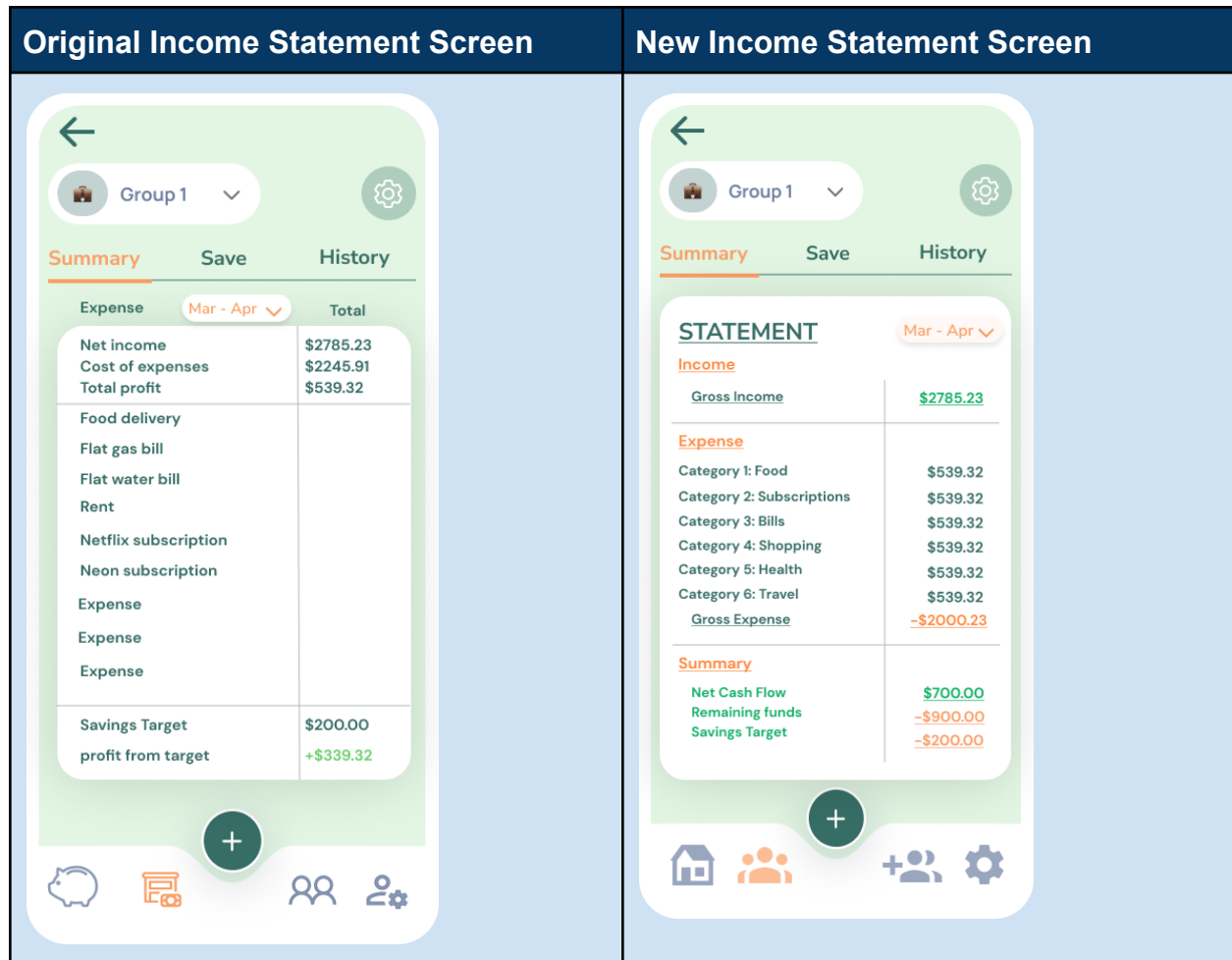
a core function of our system. To fix this I added a Header at the top of the screen reading 'Statement' to reflect its purpose. This increases the usability of the screen as users have a greater concept of its purpose.

Secondly, I found that the information within the 'income statement' did not reflect the functions of our application nor did the fields correctly depict the values they represented. By not aligning our income statement with the system's function we make it more difficult for users to manage their finances and make functions within our system redundant. To resolve this issue I combined expenses into the categories offered to users when adding an expense e.g 'Netflix subscription' to 'Category 2: Subscriptions'. This makes information regarding expenses easier to consume as it's summed into categories instead of individual transactions. Additionally, I changed the name of the fields to reflect the value it represents e.g 'Profit from target' to 'Remaining funds' which depicts how much funds users have remaining for a period after expenses and saving goals have been deducted from their income.

This makes the statement easier for users with low financial knowledge to understand increasing the usability of the system.

Thirdly, I found that the income statement did not meet industry criteria. This may make our system appear 'unprofessional' to users with finance knowledge or decrease trust in our system. To remedy this I researched formats of income statements and modified our fields to reflect these industry criteria. By ordering the statement by 'Income, Expense, Summary' instead of 'Income, Expense, Summary, Expense, Summary' the statement is much easier to follow (reduces barriers for those with low finance knowledge) and appears much more professional.

Lastly, I ensured that both the 'Personal' and 'Group' tabs had buttons to access this screen. Previously this screen was inaccessible by the 'Personal' page resulting in reduced functionality of the overall system.



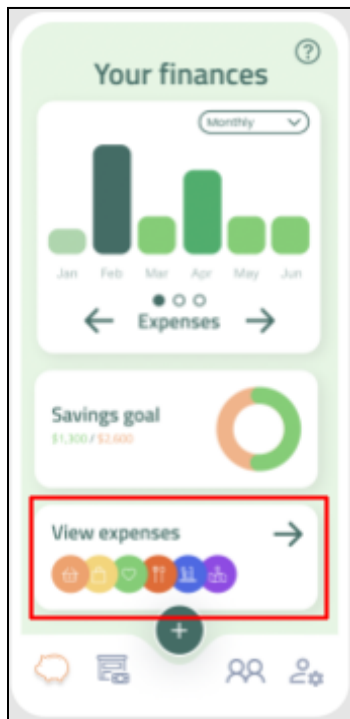
8.5 Missing function to see income summary or saving summary in personal and group

Joel Crampton

The rationale for this update and how it benefits the project

From our testing (7.1) we developed a transcript (7.2) which addressed issues users faced when testing our prototype which we used to summarise the main issues of our prototype (7.3). One of these main issues was that the personal finance page did not include an income summary. This was discussed in the transcript (7.2) by one of the testers, **Antoinette Trott**, who spoke of the prototype saying it was **“missing function to see income summary ... in personal [page]”**. The personal finance page did however have an expense summary, and this was commented on as being a good feature by those testing the prototype - so it was chosen to extend this expense

summary to cater for incomes too. The current expense summary can be accessed by a widget on the main page - outlined below in red.



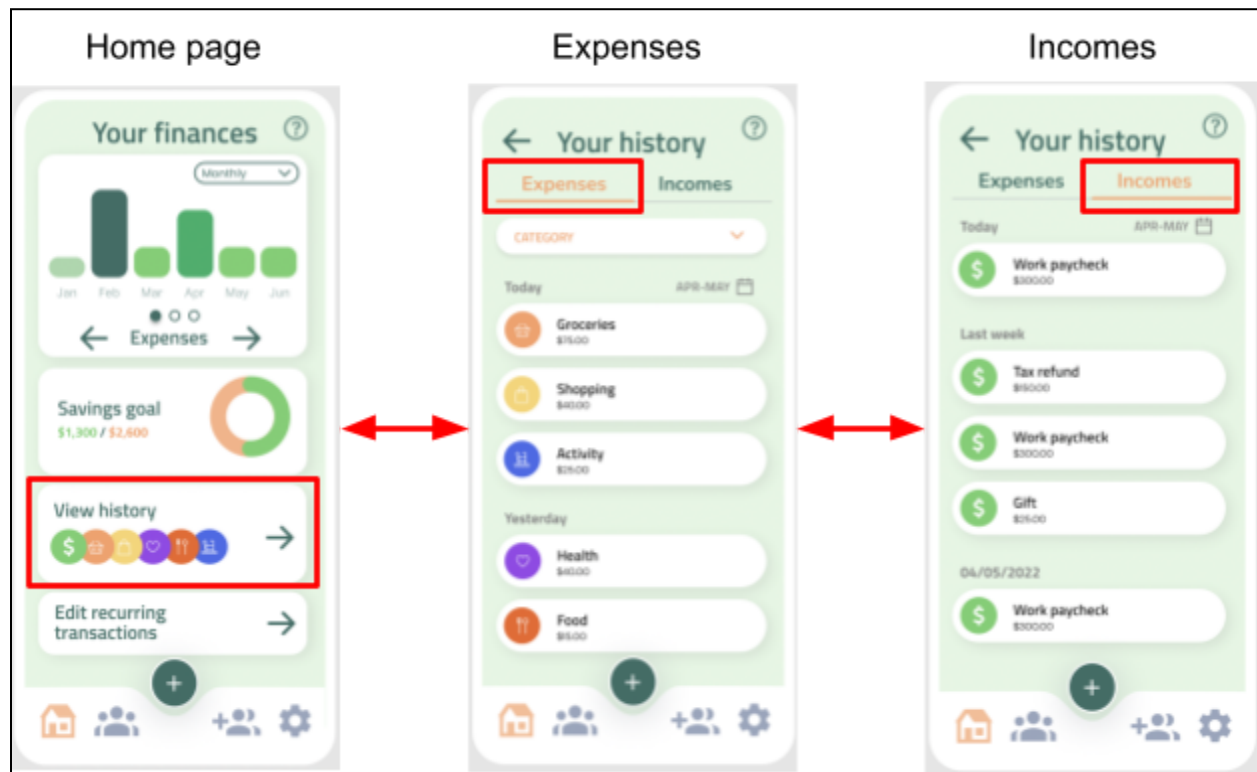
We thought about duplicating this widget for incomes, however, this would make the home page even busier, and we wanted to keep the simple layout that was currently present. So instead of the existing “View expenses” widget, we used a “**View history**” widget. The history widget **encompasses both expenses and incomes** in one section. When users click this widget they can toggle between expenses and incomes. The original expense summary features remain the same such as:

- Filtering by expense category
- Viewing an individual expense
- Editing an expense
- Etc.

But those features have now also been extended to incomes (where applicable) such as:

- Viewing an individual income
- Editing an income
- Deleting an income
- Etc.

Furthermore, the original layout of the expenses summary was used for the incomes section to make the new addition fit right into the current system. Features such as rounded corners, circular icons, identical font styles, etc. were used to achieve this. Through these new history features, the personal finance page now provides a complete and manageable overview of the user's finances - expenses and incomes. A brief overview of the functionality of the new history widget is shown below.



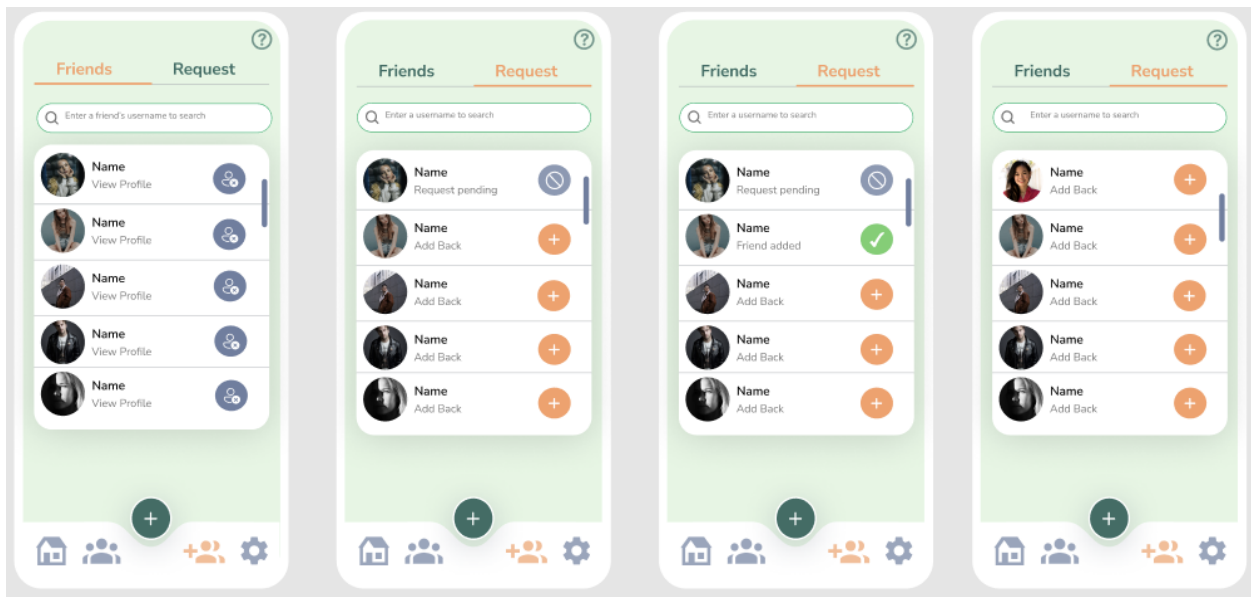
8.6 Change the personal, group and friends icons as unclear + fix all nav bar inconsistency

Jasmine Dong

The rationale for this update and how it benefits the project

As the users navigated through our app, they noticed that our navigation bar was inconsistent in regards to the axis that they were all sitting at. Some navigation bars were higher than others as they navigated through the different pages. This made it very distracting for the user and made the design of our app inconsistent. To fix this, I looked at one of the screens and moved the navigation bar so it was perfect and aligned exactly where I wanted it to go. Once that was done, I took note of the measurements which were followed by $X = 12$, $Y = 707$, $W = 375$ and $H = 118$. By having one screen

perfect, I was able to go through all of the other screens throughout the app which had a navigation bar and ensured that all the navigation bars followed the exact same measurement. In doing so, this made sure that it was the same on every screen throughout the app, and when the user moved from screen to screen, the navigation bar remained in the same place. The image below illustrates the updated navigation bar on 4 different screens.

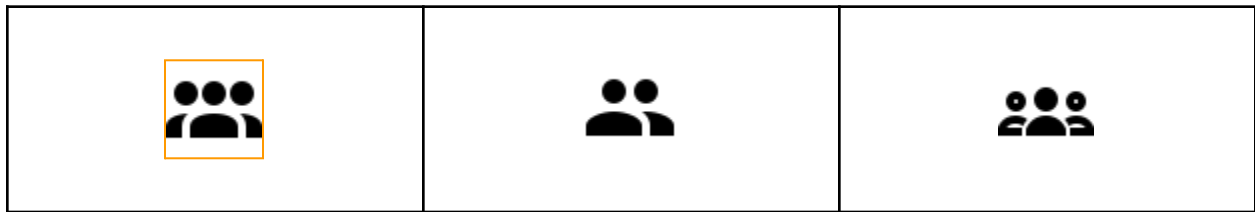


Another issue that the users found difficult when they tested our website was accessing group history and issuing a friend request. The task itself was not an issue. However, navigating the page to complete the task was a challenge. This is because our group and friend request icons were not clear to the user, and they had to navigate through the app a few times to get to where they wanted to be. A solution to this was to change the icons on the navigation bar to make it more obvious to the user what the purpose of the button was. In order to find appropriate icons which suited the purpose, I installed a few different plugins on Figma that I could look through and pick from. There was a wide selection of icons and the images below are the 3 icons for each button that I narrowed it down to.

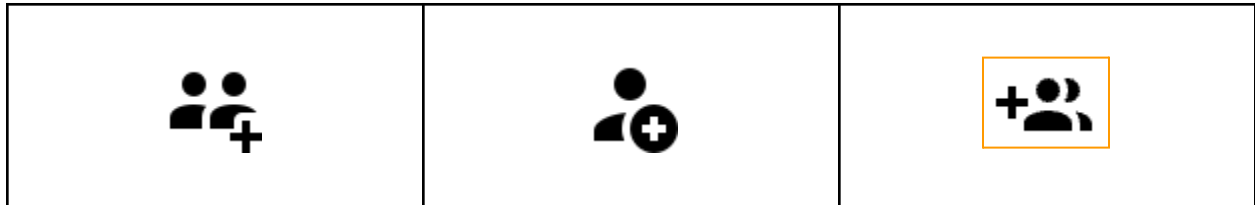
Personal



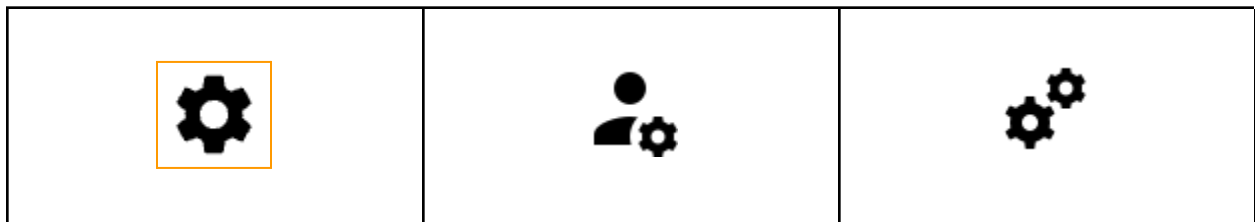
Group



Friends

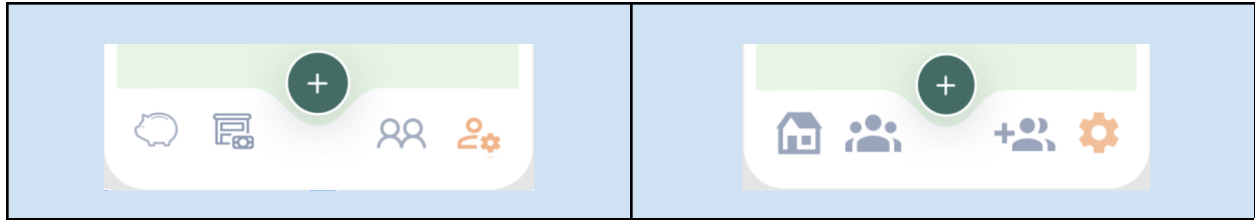


Settings



All options were perfectly suitable and conveyed the right purpose for the button. However, I picked the icons based on what looked best on the app and looked aesthetically pleasing next to each other. My chosen icons are the ones highlighted by an orange box. Once I picked the icons, I also had to evenly place them on the navigation bar, and then ensure that they were all in the same place throughout every screen so the inconsistent navigation bar issue did not occur again. Just like the navigation bar, I lined up the icons perfectly on one screen and then noted the measurements down to change all the other screens. The measurements were:

House	Group	Friends	Settings
X = 27, Y = 756, W = 55, H = 55	X = 100, Y = 756, W = 55, H = 55	X = 247, Y = 756, W = 60, H = 60	X = 325, Y = 758, W = 50, H = 50
Part One		Part Two	



9. Usability Test Discussion

9.1 What were the most interesting things you found from user testing?

User testing revealed some interesting aspects to our group members that we had not previously thought about, the most noticeable of these was how some users took a substantial amount longer to complete a particular task, and also how some users interpreted a particular task in a different way to other users despite having the same instructions. To expand, the user Ella Spies often took a lot longer to complete a task in some issues taking 1:29 longer than the next user. This was really interesting to us as a group as it demonstrated how some users may struggle to complete a task with our system.

The next interesting aspect that we gathered from user testing was how different users interpreted how to complete a task despite being given the same set of instructions. This was evident in our issue where we got the users to send a friend request. While most users followed the steps exactly how we expected them to, by accessing the request tab in the friends domain one user, Anish, originally tried to send this request under friends.

While some issues that users discovered while testing were unique and ones that we hadn't thought of, the majority of issues discussed by users were ones we too had considered. Our team has become very familiar with the system after spending time developing it, so it was interesting that people brand new to the system pointed out similar issues. This showed us that most of the time users have similar intuition as to how a system should function, and in this case testing showed us that the collective intuition of users was not met all the time. Although users typically have a similar understanding to one another of what a system should do, our testing also showed there are outlying features that certain types of users had a desire for - and these shouldn't be ignored. No matter how insignificant a feature is, as developers, we should strive to design a system that caters for everyone. We have made good strides in our prototype to align with this view through features such as the accessibility section,

however, testing showed us there is still room for improvement, in order to create a system everyone is completely satisfied with.

9.2 What did you learn from conducting the user testing?

We gained knowledge on the importance of user feedback/communication. Our initial prototype made assumptions that:

- Users would be able to interpret the difference between the 'Group' and 'Friends' tab.
- Users would assume recurring expenses would be found in the 'Add transaction tab'.
- The layout of the 'Group' and 'Personal' tabs contained all applicable functions.
- Our financial tools met industry standards.
- All functionalities were operating.

We as a group found that the group we collaborated with for testing struggled with tasks we thought met the system objectives and usability heuristic. Having a thorough look into our system from an outside perspective highlighted many areas that we didn't realise needed improving. An instance of this would be the position of the viewing 'Recurring incomes' widget initially in the expenses section. Many users highlighted the fact that it was awkward to navigate and that they would've liked to have seen it in other areas of the system. This small improvement allowed us to improve the overall accessibility of the functions and further increase the modularity of our prototype.

If we were to repeat this process, we would allow more time for a greater number of users to test our prototype, this would mean that we are more likely to identify a broader range of opportunities for improvements. Improving areas weakened by personal bias enabled us to develop a stronger product that met both the product's objective and usability expectations. From user testing, we learnt that our assumptions did not always align with users and that it was important to conduct testing to mitigate bias generated from personally developing the system.

9.3 What were the most significant results you found from user testing?

When testing we found various issues that we had originally missed in our own testing. In particular, we identified inconsistencies in the flow of our application. As we developed the solution, we all had a thorough understanding of our app and how it functioned so when users were testing our solution, they were able to identify small

features that needed to be changed. This is because we knew how our system worked and took for granted where to go for each feature. With user testing, our testers didn't have this knowledge and it became abundantly clear that features weren't in the optimal locations when users were unable to find them. For example, William struggled to find where to add an expense, because although we had a "+" button in the middle of the screen, he navigated through the icons on the navigation bar before realising that the "+" button was the button that needed to be pressed to complete the task.

Our users identified several small inconsistencies with titles and names for our app sections. These inconsistencies were something that had slipped our minds due to how small they were. However, they were quite obvious to the users as they were reading each thing on every new page as a first-time user.

Additionally, the users who had tested our app identified that our navigation bar was inconsistent on each screen, creating a disorienting feel when used in quick succession. This was a critical issue that was missed and only found through user testing. This really helped to cement its value for us, and highlighted the importance of having a clear intuitive app design.