

SWEN303

Final Project Part One - Prototypes

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

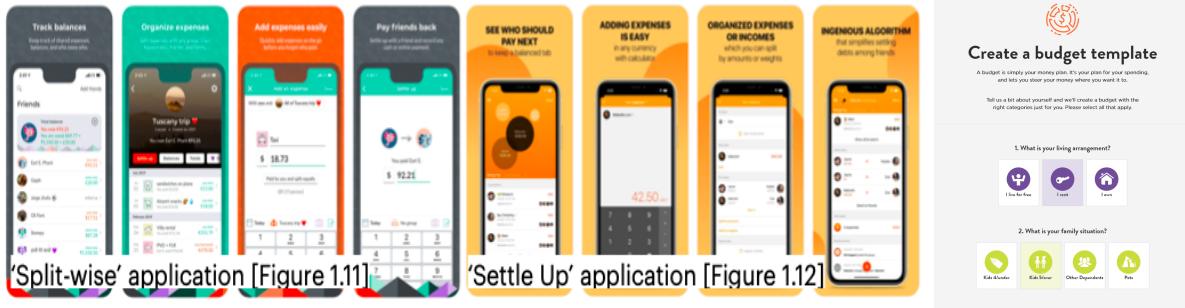
Project Topic:
Managing Finances

[**Figma prototype**](#)
Please use the flow “Login Starting”

BACKGROUND [1.0]

Existing Solutions [1.1]

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’, ‘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]*

Business Objectives [1.2]

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.

Significance to Stakeholders [1.3]

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.

PERSONAS 2.0

Selection method [2.1]

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).

Personas Table of Attributes [2.2]

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	- Set spending limits. - Categorise expenses.	3	Shared Flat	N/A
Takuma Travello	- Set spending limits. - Categorise expenses.	5	Shared Flat	English second language
Oscar Oblivious	- Needs a low amount of user inputs (low financial literacy). - Needs a way to track saving goals.	5	Home	High-functioning Autism

Freya Flata	- Needs a visual way to track saving goals. - Shared and personal expenditure.	5	Shared Flat	N/A
Debbie Detached	- Needs an easy to understand interface (low technological literacy). -Wants convenience when adding expenses.	2	Alone	Low vision
Cameron Careless	-Frequent expenses (needs convenience). -Wants to be able to track monthly spending and saving.	1	Alone	Dyslexic
Samuel Stock	- Easily change saving goals. - Customisable setting presets.	4	Alone	N/A
Paris Pennywise	-Shared/personal expenses. -Wants to closely monitor spending (reports).	1	Home	N/A
Timothy McDermott	- Shared /Personal expenses - Low financial literacy (low user inputs). - Moderate transaction frequency	5	Shared Flat	N/A
Amelia Waldorf	- Shared expenses only - High financial literacy - High transactional frequency (needs easy expense input).	3	Shared Flat	N/A
Sarah Yu	- Medium financial literacy - Personal expenses only -Wants to closely monitor spending (reports).	3	Alone	English second language

Jack Apperly	- Low financial literacy (low user inputs). - Shared expenses only - High transactional frequency (needs easy expense input).	1	Alone	N/A
--------------	---	---	-------	-----

Persona Summary (Organised based on priority) [2.3]

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.

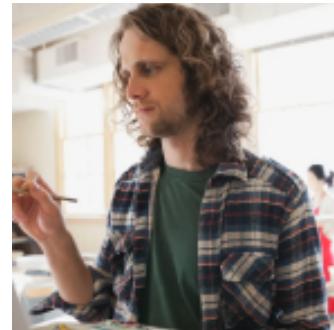
Debbie Détached (Main attribute: Low vision)

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.

Final Personas [2.4]

Name: Dave Dify

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.

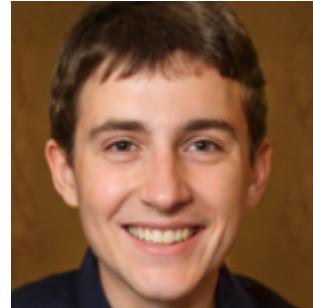
Computer Self-efficacy: Dave is open and able to use new computer systems, so long as they cater to his dyslexia

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

Name: 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.

Name: 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.

Name: 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.

Name: 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.

Name: 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.

SCENARIOS 3.0

Scenario Selection and Merge Process [3.1]

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.

Scenarios Table of Attributes [3.2]

In the first column, we have listed everyone's scenarios from assignment 2. In the second column, we listed scenarios similar to each other. Every member in the group then ranked each scenario from 1 (least important) to 5 (most important) to determine which scenarios we believed were the most important.

		Importance (1 least, 5 highest)							
Scenario	Similar	Jasmine	Todd	Emilia	Joel	Charles	Marina	Avg	
M: Creating an expense	JET: Adding an expense/in	5	5	5	5	5	5	30/6 = 5	

	come							
M: Settle/Manage outstanding balance		1	1	1	1	1	1	6/6 = 1
M: Register/Log in		2	2	1	2	2	2	11/6 = 1.83
M: View spending summary	JET: Expense overview JC: Struggling to pay rent	5	5	5	5	5	5	30/6 = 5
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		4	4	5	5	4	5	27/6 = 4.5

finance								
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

Final Scenarios [3.3]

Adding an expense/income:

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

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.

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.

Enabling/Disabling accessibility options:

Personas involved: Dave Difffy, 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.

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.

Create access-group account:

Personas involved: Dave Diffy, Timothy McDermott, Freya Waldorf, Traveller Takuma, Debbie Detached.

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.

DESIGN IDEATION [4.0]

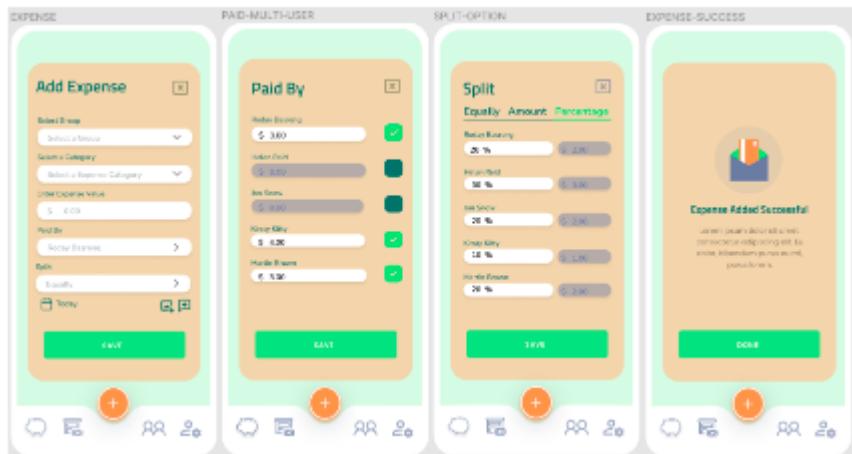
Here is a sample of each team's design of an expense page.

TEAM 1 (Dime) [4.1]

- Kamonchanok Suban Na Ayudtaya

Figma:

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

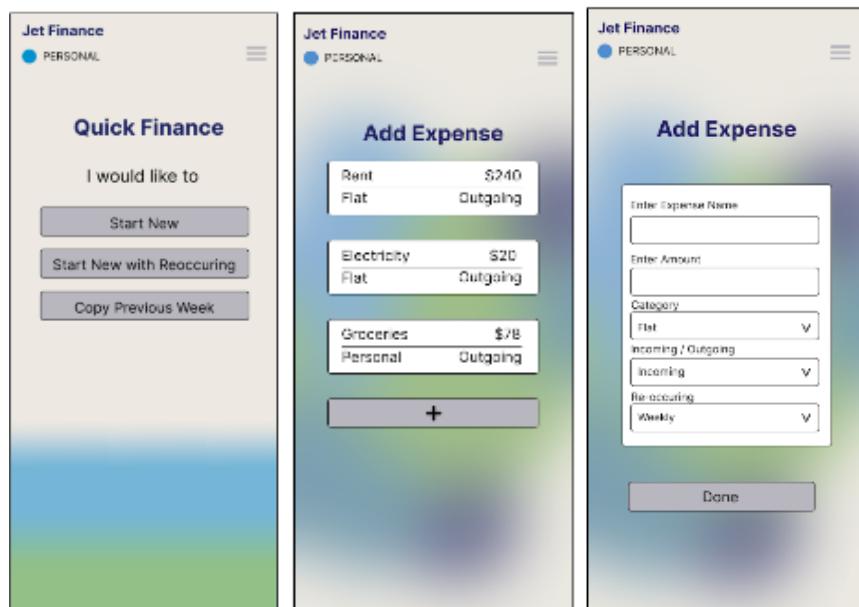


TEAM 2 (Jet finance) [4.2]

- Todd Wellwood
- Jasmine Dong
- Emilia Greve

Figma:

<https://www.figma.com/file/HmAzuXJ6QYgASd8CCFO7TC/SWEN303?node-id=0%3A1>



TEAM 3 (Young & broke) [4.3]

- Joel Crampton
- Charles Hughes

Figma:

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

The image displays six screenshots of a mobile application interface for 'Young & Broke'. The interface is designed for budgeting and consists of the following steps:

- Step 1: Who are you calculating for?** Offers two options: 'Flat' (with a house icon) and 'Personal' (with a smiling face icon). The 'Flat' option is selected.
- Step 2: What is your name?** A text input field labeled 'name' and a 'Continue' button.
- Step 3: What is the time period you're calculating for?** A dropdown menu set to '1 week' with a 'Continue' button.
- Step 4: What is your income?** A text input field labeled '\$ income' and a 'Continue' button.
- Step 5: What are your expenses?** A list of expense categories with input fields: \$ rent, \$ food & living, \$ utilities, \$ transport, \$ leisure, \$ education, and \$ other. A 'Continue' button is at the bottom.
- Step 6: Do you have a long term savings goal?** A text input field labeled '\$ savings' with 'Continue' and 'Skip' buttons.
- Step 7: Here's your results!** Displays a bar chart titled 'Name' with seven bars of varying heights, representing the user's budget components.

DESIGN REVIEW [5.0]

Selection method [5.1]

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 personas 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:

Summary of Group Evaluation [5.2]

TEAM 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. 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. 	<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.
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 information is always numeric, while a selection of contact is done via a dropdown - this helps to prevent errors. Clear DONE button. If you want to save/submit, there is a clear bold 'DONE' button. 	<ul style="list-style-type: none"> Payment cannot be undone. Once payment is processed it is final, there 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. 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 	<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.

	<p>keeps track of any information entered in previous steps. The exception to this is the user's password, which they are expected to remember.</p> <ul style="list-style-type: none"> 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, 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 	<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.

	categories are completely relevant to the app.	
9	<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. 	<ul style="list-style-type: none"> Colour and symbols indicate incorrectness. Currently, a combination of colours and symbols is 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. 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

TEAM 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 the system. Some headers were used to indicate a user's position. 	<ul style="list-style-type: none"> Unclear user position for the entire system. From the depicted system it was difficult to determine a user's position within the system.

	<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>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 clicks on the wrong section, they can easily escape by clicking the hamburger again. 	<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 example 'Start New' instead of 'Copy Previous Week', they can easily go back.
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 	<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

	<p>consistent throughout the design.</p> <ul style="list-style-type: none"> • 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. 	<p>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.</p> <ul style="list-style-type: none"> • 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 button to execute an action is clear to the user. 	<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 have to create a whole other expense. • 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 	<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.

	<p>user does not need to remember anything when inputting data. It is clear on every page what is required.</p>	
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 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. 	<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.
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.
--	--	---

TEAM 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 what they have previously inputted. 	<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 action (essential to reassure users when dealing with money).
7	<ul style="list-style-type: none"> Personalisation through saved calculations. Users can save their calculation to their profile 	<ul style="list-style-type: none"> No editing. Users cannot modify existing overviews, which means users have to repeat the entire input

	<p>to personalise their experience and save time by not having to recalculate for subsequent calculations.</p> <ul style="list-style-type: none"> ● 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. 	<p>process to modify an existing graph.</p> <ul style="list-style-type: none"> ● 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. ● 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.
----	---	---

PROTOTYPE [6.0]

General Design [6.1] (Observed in every domain)

Style/Usability

When generating the style of the prototype, we researched various design theories (e.g. minimalism, white space, contemporary and skeuomorphic). Of these, we found that minimalism, flat design and white space were most applicable to our user's needs as we wanted to emphasise a simple, clean and calming design ('The More of Less').

The application's primary functions should be intuitive enough for people to use without needing an instruction manual or tutorial. From the interviews with multiple students from Victoria University, we found that they expect the solution will assist them in tracking multiple and frequent expenses and that it will be easy to use and accurate.

To ensure consistency, standards, minimalism and aesthetics, we explored the '[Material Design documentation](#)' created by Google. This allowed us to better understand how users interacted with interfaces and the importance of grid-based layouts, padding, and depth effects.

Additionally, we explored the different '[10 Usability Heuristics for User Interface Design](#)' published by Jacob Nielsen. This allowed us to better understand the importance of creating solutions that are 'usable' for our audience and how we could implement that.

Key design ideas implemented in our solution:

Simple colour scheme, Simplified content, Consistent typefaces, Simple functionality and user interaction, Reduced wordiness and Reduced visual distractors.

Usability Heuristic implemented in our solution:

>Match between system and the real world: Simple, concise language allows users to navigate the functions of the solution easily. Icons are representative of real work actions it is associated with.

>Error prevention: limiting user inputs through buttons, drop boxes, and checkboxes where possible.

>*Aesthetic and minimalist design: clean and consistent illustration used to attract our audience (university students), convey complex information and decrease stress associated with finance management without detraction from functionality.*

>*User control and freedom: We give users the option to return from any path where possible and modify any change submitted.*

>*Visibility of system status: headers, nav bar and pop up messages informs the user of their position in the system.*

>*Help users recognise, diagnose, and recover from errors: users are informed when inputs are invalid using symbols, colour and an informative message (delivered in pop up).*

>*Consistency and standards: confirms user has taken the correct action with success screen (affirms positive actions), fields listed in the order in which they should be completed.*

>*Recognition rather than recall: Simplified complex system to depend on icons, simple text and reduced text inputs means that users do not have to remember information from one part of the interface to another, this is handled internally, and users are automatically guided through the process.*

>*Help and documentation: Help button on every screen gives an informative message on the use of the solution.*

>*Flexibility and efficiency of use: shortcuts such as recurring transactions and saved transactions allow users to decrease repetitive inputs. We have also implemented UI customisation to reduce barriers for those with specific visual needs.*

Platform

Based on our user's intended use of the solution, we found that a mobile application was the most effective in distributing our solution. With expenses occurring at various locations and times, access to a website is unlikely to be convenient for our users, which detracts from the core objective of the solution of convenience. For this reason, we decided to implement our solution in a mobile application allowing our users to track expenses on the move quickly.

Icons

Implementing the ideas we had learnt from the 'Material Design documentation' in Figma, we used the 'Iconify' plugin to generate icons and appropriately contrast them using colour to make our design readable, intuitive and easy to pick up. This ensured that users of various technical and language proficiency levels could access our solution by reducing wordiness (removing the language barrier). We compared the icons used with '[Icon Usability](#)' published by Jacob Neilsen to ensure that they were to an appropriate standard.

Fonts

In order to cater for people with learning difficulties and visual deficiency, we did some research into which fonts are better for those with dyslexia and dyspraxia. In the end, the font we used was

FONT
font

FONT
font

FONT
font

'Titillium Web'. This font was selected based on its [style and clarity](#). We believed that this would be sufficient in meeting the agreed style of minimalism while allowing for readability as it is a clear and structured font. Furthermore, in our prototype, [we ensured that fonts were of at least 15pt to ensure that the solution would be accessible for all users with visual deficiencies.](#)

Colour

It is well known and proven that colours affect our mood. In the 21st century, we have a very fixed opinion on what emotions are associated with what colours. Our colour palette consists of five colours: Dark Green, Dark Grey, Light Green, Off White and Orange. The primary colours have the corresponding emotional associations: [Green - Safety/Money, Orange - Energetic.](#)

There are two primary colours in the palette (excluding greyscale). This was because while it has been proven that the use of colours does help people retain information better, [the overuse of colours can also overload people with information.](#) Therefore, limiting ourselves to two primary colours prevents the pages from being cluttered and maintains the solution's primary purpose of easily and conveniently managing finances.

Color Styles

- finance
- finance1
- finance2
- finance3
- finance4

We decided to use 'light green' as the primary colour of the solution due to its [association with money](#). We believe it will allow users to associate our application to finance management. Additionally, it is a [non-aggressive colour](#), and we think it will act to calm users and finance management can be perceived as a stressful task.

The accent colour 'orange' would be used in areas of the website in moderation to direct the user's attention instead of overwhelming the users. This colour was picked as it is a contrasting colour to green, allowing it to be an effective visual aid in directing users to observe important information, e.g. Icon in the navigation tab highlights orange to indicate where a user has navigated to.

We have picked a body colour for text boxes as 'off-white' and the text colour as 'Dark Green' and 'Dark Grey', rather than pure black and white. This colour scheme is easier on the eyes of the user as the contrast between the two colours is reduced.

Navigation

We decided to implement a navigation bar for ease of use. When observing other forms of navigation, such as a 'drop-down' tab, we found that it detracted from our minimalist style, confused users and introduced unnecessary wordiness. Using a navigation bar in combination with icons and colour allowed us to communicate to users more



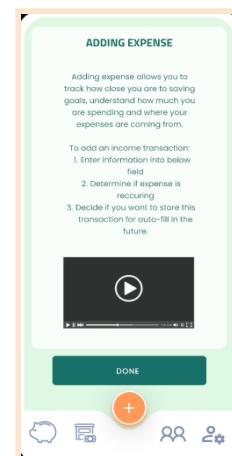
concisely, further driving our purpose of convenience and reducing language and visual impairment barriers.

Logo

While a logo is beneficial for recognising the solution, we decided it should only be displayed in the application image icon. This is because it unnecessarily consumed screen space and detracted from our minimalistic design, reducing functionality and introducing wordiness. Furthermore, observing competitors' solutions' 'Split-wise' and 'Settle-Up', they do not include their logo within the application, so we feel justified in not including a logo within the solution as it negatively affects our primary purpose.

Help and documentation

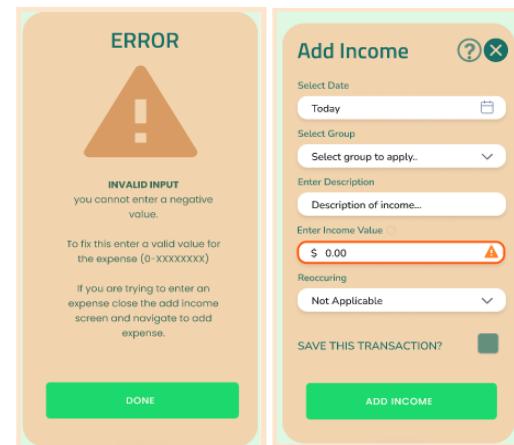
We have decided to add a help button on the top right corner of every screen with user interactions to assist users in navigating the functions of the solution - clicking on it produces a pop-up with a meaningful message. The purpose of this is to reduce learning, risk and knowledge (technological and financial) barriers which may prevent users from accessing our solution.



Error Handling

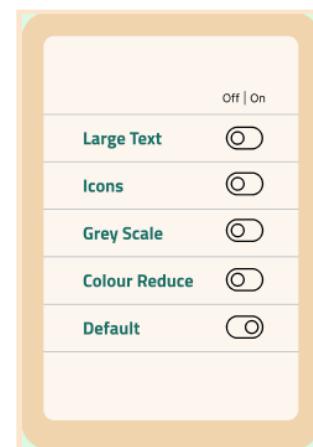
We have decided to use colour, symbols, and a pop-up to indicate an invalid input/error. If an error is found once a user finishes filling in a field and clicks off the input box - the box will highlight orange with an error symbol. Clicking on the error symbol will result in a pop-up with a meaningful message on how to fix the identified error (reduces unnecessary interruptions for experienced users).

This was designed using [Nielsen's design guidelines for errors](#).



Customised Accessibility

We have allowed users to customise the UI to their needs upon opening the app, saving this select customisation on registration or changing this within the account tabs. This is to increase usability for people with visual barriers (low vision, dyslexic), language barriers or require specific needs to be met (autistic).



Information Architecture

An organisational structure is how you define the relationships between pieces of content. Successful structures allow users to predict where they will find information on the site. We have used a hierarchical structure to allow users to determine where to find desired function/information in the solution allowing for intuitive navigation, convenience and ease of use.

Hierarchical Structures use a top-down approach or parent/child relationships between pieces of information. Users start with broader categories (parent) and then drill further down into the structure to find narrower, more detailed information/function (child). This can be observed in the navigation bar - the solution is separated into five parents: Personal finance management, Group finance management, the addition of expense, Interaction management and Account management. From there, users can access the information/functions they require By navigating to the parent categories and navigating from the main pages of these categories to more specific services (e.g. To manage recurring expense-> Go to add expense/income screen -> select saved & recurring -> select recurring tab -> select transaction to modify)

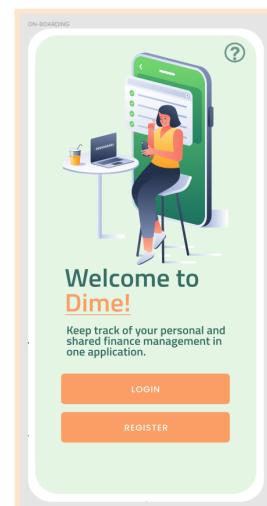
Login & Registration [6.2]

Onboarding:

Our onboarding section needed to have a sleek design that will represent what the experience will be like for users throughout the app. This section is designed to show off the colour options we would be using throughout the app, provide a brief description of our app, and point returning and fresh users in the correct direction.

The point of having this splash screen is to streamline the entry point of the app. Returning users will hit “Login” while new users will hit “Register”. This section was designed based on apps such as Instagram and Facebook, which place the “Login” button first, as it’s more commonly used. This helps with user experience.

Finally, It has a help section in the top right. See “[6.1] Help Sections” for more detail



Registration:

The registration process is something that all **new users** will have to go through; it's designed to be easy to go through and as linear as possible.

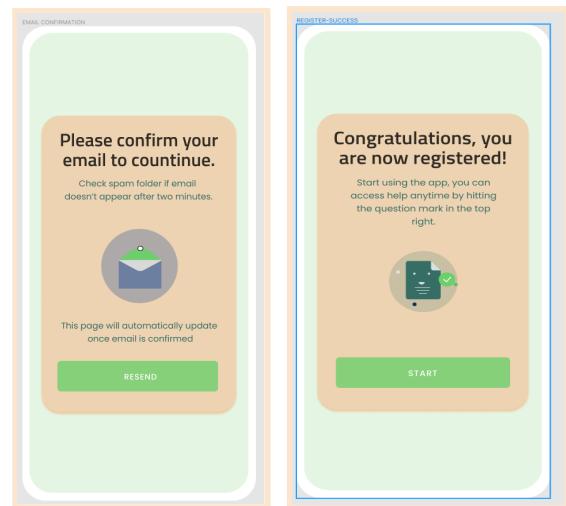
The design used here has field inputs vertically placed. This verticality helps create

Two side-by-side screens. The left screen is a "Registration" form with fields for Name (Roday Basreng), Username (rodbas123), Phone (+64 027465100), Email (Roday43@gmail.com), Password, and Password Confirmation. It has a "GET STARTED" button and a note about accepting terms and conditions. The right screen is a "Terms and conditions" page with dense legal text and a "RETURN" button.

a linear feel and minimises the chance the user skips a step.

If a user does miss a field, the submission button will be greyed out; see “[6.1] Error handling” for more details.

There is also a selector for the area code under mobile; this allows users not to enter an invalid area code as they must select from a dropdown. This leads to better error handling and stops users from forgetting to add it.



We have a big confirm button at the bottom; this button will be greyed out if all the fields aren’t filled in; this can be seen under “[6.1] Error handling” for more details.

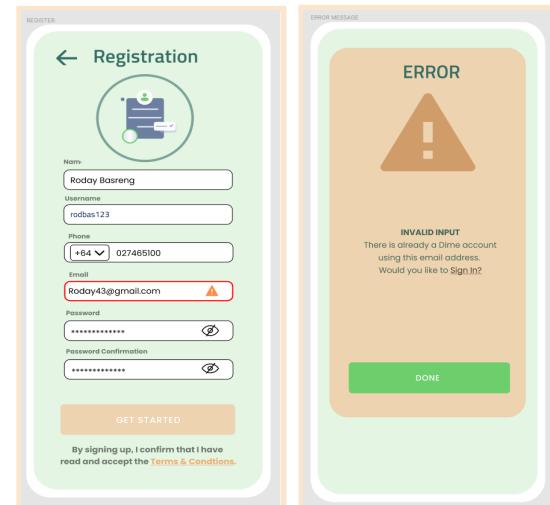
Additionally, there is a terms and conditions page. This page contains placeholder text but is linked to being incorporated into the design. It has been placed below the “Get Started” button as it’s something many users will skip over.

Finally, once the user has submitted, they are required to verify their email. This extra layer of authentication is used to confirm they have access to their account for future logins and politely notify users if they have an account registered with us - see “[6.1] Error handling” for more details.

Error Handling:

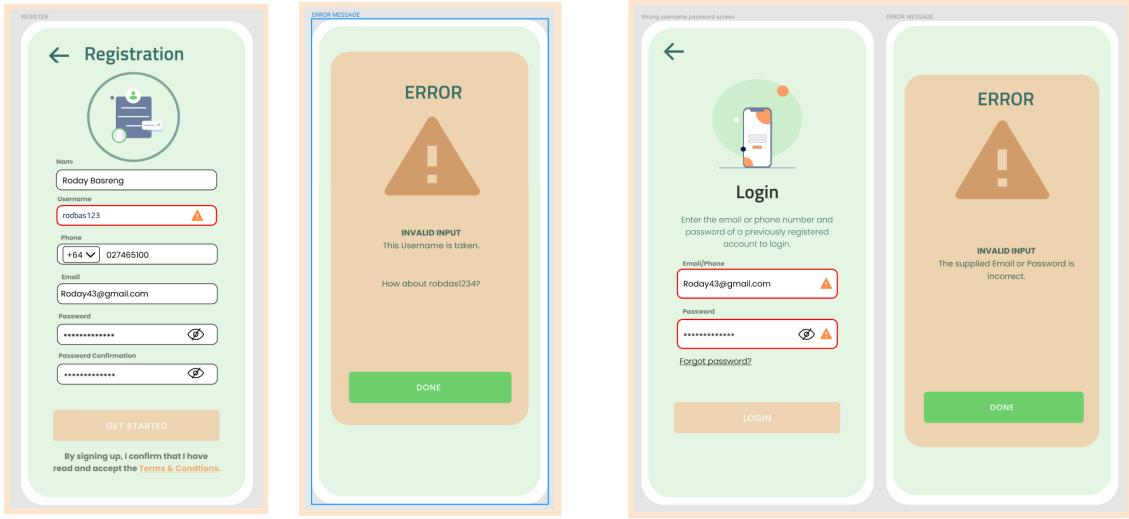
Error handling is a critical part of our system. We have several different types of error handling.

If users enter data that belongs to a currently existing account in the register section, the system will prompt the user to sign in instead. This concept of not only identifying the issue but helping to correct it is good for increasing the usability of our system.



Suppose the user enters an invalid field, such as a username that has already been taken. The system will prompt them with a suggested alternative name. In that case, this helps to lower frustration during the signup process.

Finally, if the user tries to sign in with an invalid username and password combination, the system will not state which specific one is invalid. This was a recommendation by one of the cybersecurity students in our team as an extra security layer.



Accessibility Quiz:

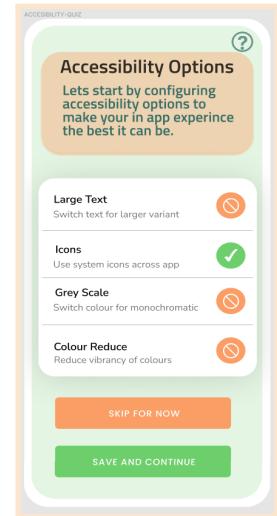
The accessibility quiz is the first page the users are prompted with after having created their account. It enables them to turn on and off accessibility options for new accounts.

In addition, the decision to put this after account creation was to ensure users aren't overwhelmed but also prompt them to use it as soon as possible.

Furthermore, the system has a description to explain what the page is about, as well as clear toggles on the right side from orange to green. It also uses icons to take advantage of users' knowledge of symbol meanings.

Finally, at the bottom, there are two buttons, "Skip for now" and "Save and continue". This allows users who have no interest in it to skip forward. Users can also access the help section in the top-right if needed.

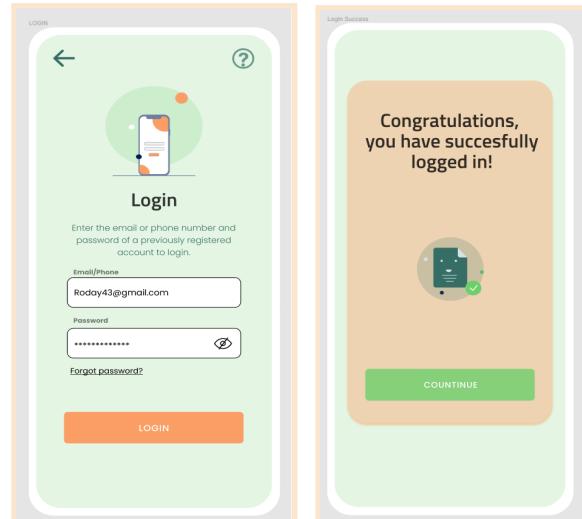
The user can later change these options under the account section.



Login & Password reset:

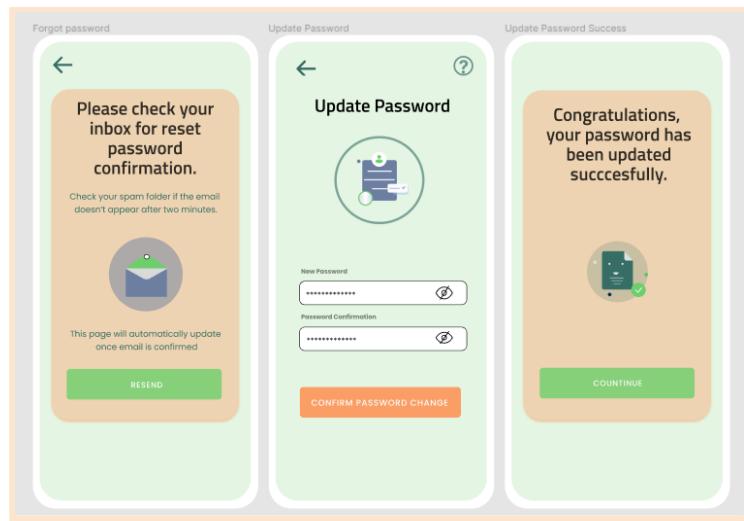
For returning users the login section is where they will start off. We give the option of both phone number and email for login, in combination with a password. We chose not to use a username, as this information is publicly available under the social tab and would be compromising security.

Once logged in, the user is given a confirmation message before being able to continue into the app.



In the event of an invalid username and password combination, see “[6.1] Error handling”.

An additional important feature for users is the ability to reset passwords. We first require a verification of the email and then a two-part confirmation of the new password. This will also have a prompt to confirm that the action has taken place. This will then move them back to the login page.



Help Sections:

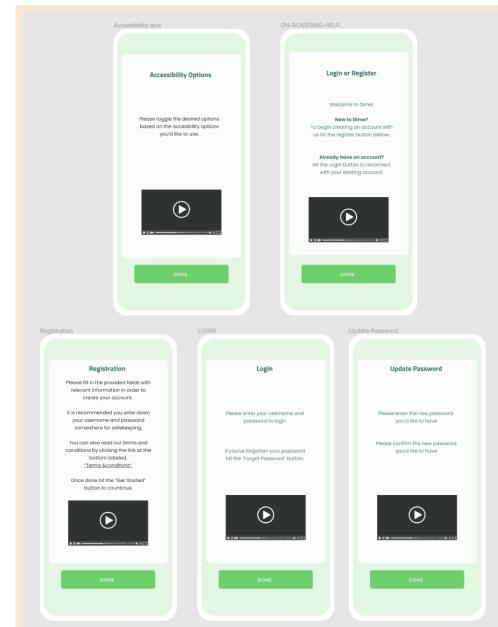
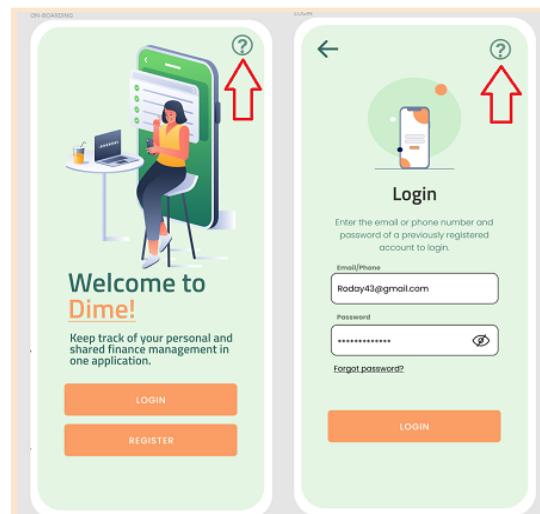
The login domain features help screens on all major screens. The user can access these screens by hitting the top right question mark on all pages that require user input, extra information or explanation.

A title for each screen is shown at the top, followed by a detailed written description.

Finally, we have a video walkthrough for the section.

These multiple options ensure users can get the best possible support for using our app as needed.

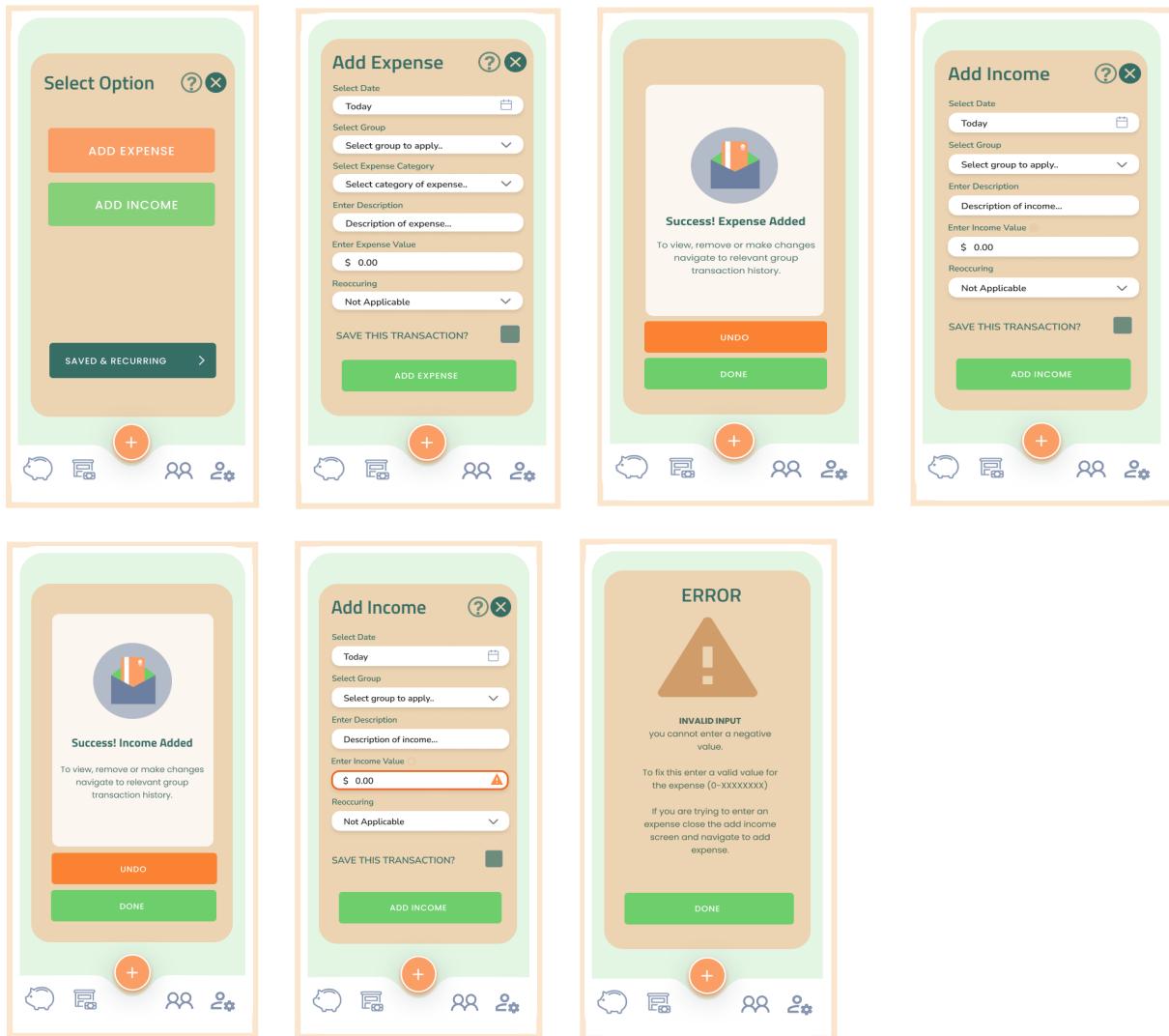
The bottom set of the images shows how to access it from relevant screens.



Manage Expense/Income Domain [6.3]

Adding expense/income

Screen sample:



Adding an expense/income is a crucial function of the solution as our main purpose is to help users manage their finances by tracking expense, income and saving goals.

This domain is used by all personas/users and thus needs to be accessible for users with varying financial, technological and language abilities.

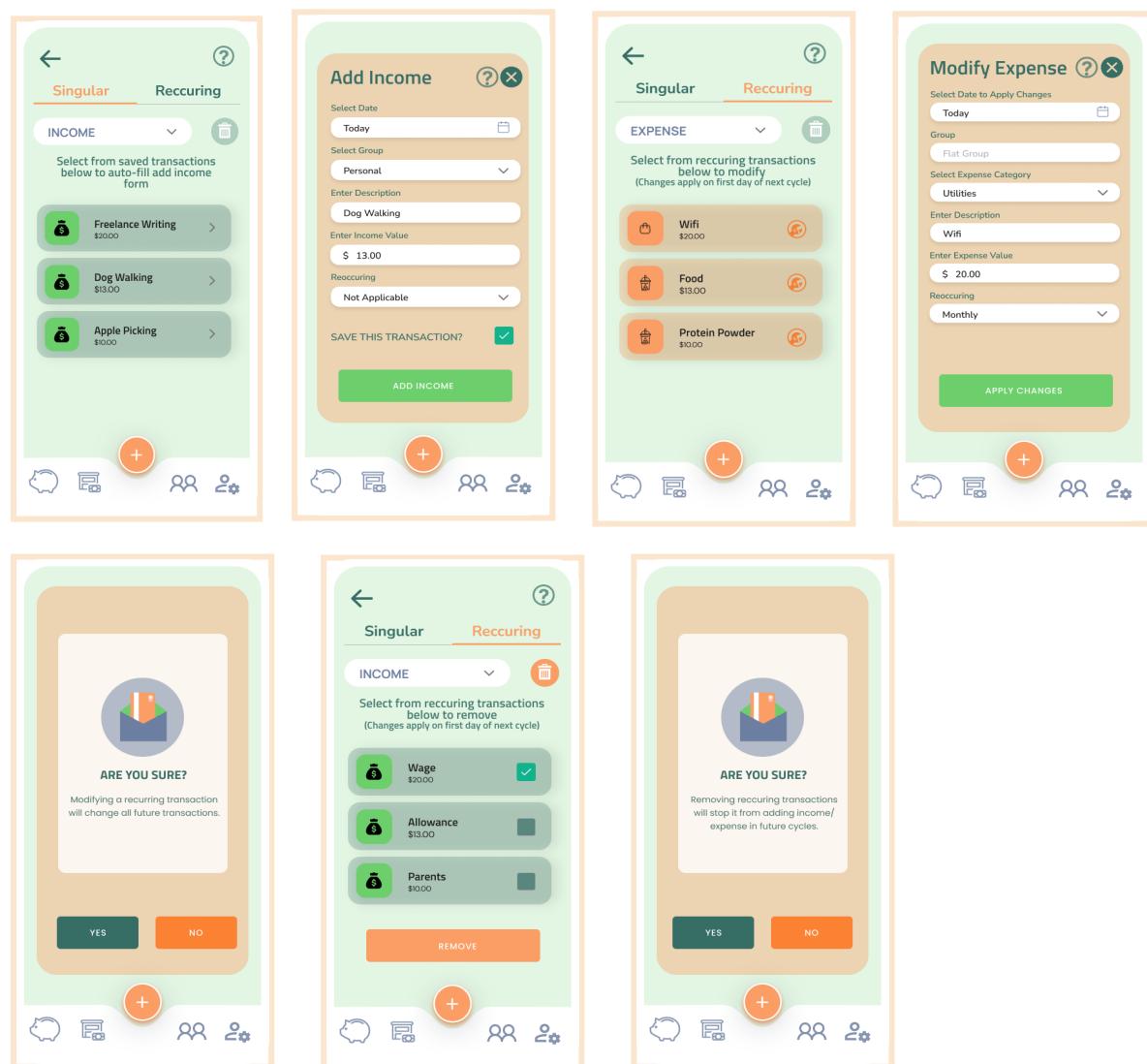
To make this process as convenient as possible, we use clear concise language (Match between system and the real world), we minimise user text inputs by filling as many categories as possible using drop boxes or check boxes (Error prevention), and we allow users to navigate away from the action at any point using the back/exit button at the top of the screen (User control and freedom).

Our layout means that users do not have to remember information from one part of the interface to another, as all inputs are on one screen (Recognition rather than recall).

After a user fills the necessary fields of their selected transaction type, adds the transaction and all fields are found to be valid, it is added to the appropriate account, and the 'Success' screen is displayed (Visibility of system status). Here users are able to proceed or undo the submitted transaction (User control and freedom).

If a field is found to be invalid, colour, symbols, and a pop-up indicate an error. If an error is found once a user finishes filling in a field and clicks off the input box - the box will highlight orange with an error symbol. Clicking on the error symbol will result in a pop-up with a meaningful message on how to fix the identified error (reduces unnecessary interruptions for experienced users) (helps users recognise, diagnose, and recover from errors).

Saved & Recurring Screen Sample:



To increase flexibility and efficiency of use, we have implemented 'Saved & Recurring' when users select saved expenses, a copy of the transaction input is stored here where it can be later selected to auto-fill a transaction or removed when no longer needed (User control and freedom). Additionally, when users

choose a recurring value for a transaction, all recurring transactions are stored here where users are able to modify the details of future recurring transactions or permanently remove the transaction from recurring (User control and freedom). This domain will likely be used by technologically advanced personas and personas seeking convenience.

The functions in this section of the domain uses clear concise language (Match between system and the real world), minimal user inputs (Error prevention), and we allow users to navigate away from the action at any point using the back/exit button at the top of the screen (User control and freedom).

Our layout means that users do not have to remember information from one part of the interface to another, by using filters and tabs to reduce excessive scrolling (Recognition rather than recall).

To auto-fill a transaction users navigate to the singular tab (singular transaction) and select income or expense in the drop down filter. From there all saved transactions under the relevant filters are displayed and able to be selected. After selection users are directed to the same process as adding income and expense with fields auto filled based on selected transaction.

To modify a recurring transaction navigate to the recurring tab (recurring transaction) and select income or expense in the drop down filter. From there all recurring transactions under the relevant filters are displayed and able to be selected for modification. Modification follows the same process as adding income and expense (simply change auto-completed fields). Upon completion users then have to confirm the action in a success screen. Changes are then applied to the next applicable cycle from the date selected.

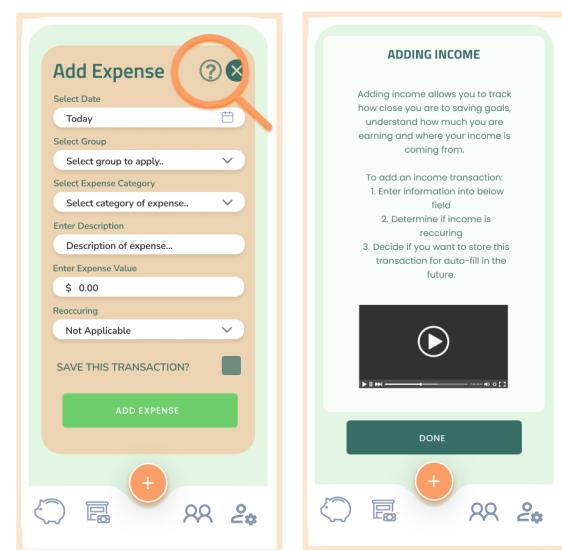
To remove a transaction from singular or recurring, users navigate to the desired tab and filter, select the bin icon at the top of the screen, and select transactions they want to remove. Upon completion users then have to confirm the action in a success screen (Visibility of system status).

Shared Features

Screen Sample:

If a user has any trouble during an interaction with any of these functions they are able to tap the question mark icon to understand the purpose of the function, how to complete the function and watch a video tutorial on its use (Help and documentation).

To ensure consistency and standards we have ensured all screens within this domain share the same icons, placement, and description. In addition we used similar functions where possible (e.g modifying a recurring transaction uses the same format



as adding income/expense) to reduce learning barriers and frustration.

To ensure ease of use we maintained an aesthetic and minimalist design to make navigation and system comprehension effortless for users.

Social / Friends Domain [6.4]

Within our system, we have the option for the user to join groups, whether this be for example a group with the users flat members. This was purposefully added within the system as it allows a user to see the overview of a group setting. Due to this inclusion it was therefore necessary to create a friends domain so that the user can see the friends which they have added and manage this.

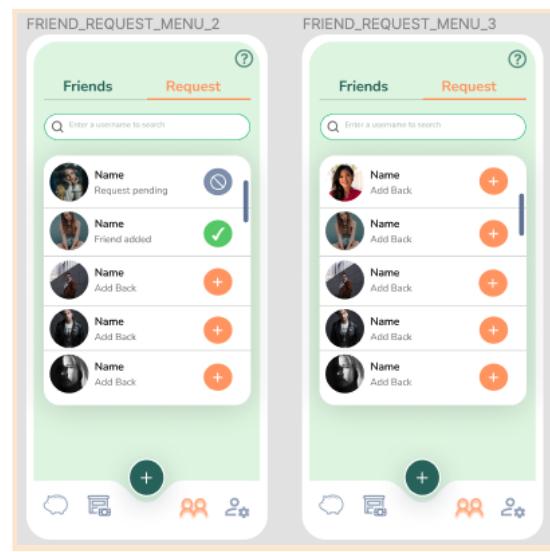
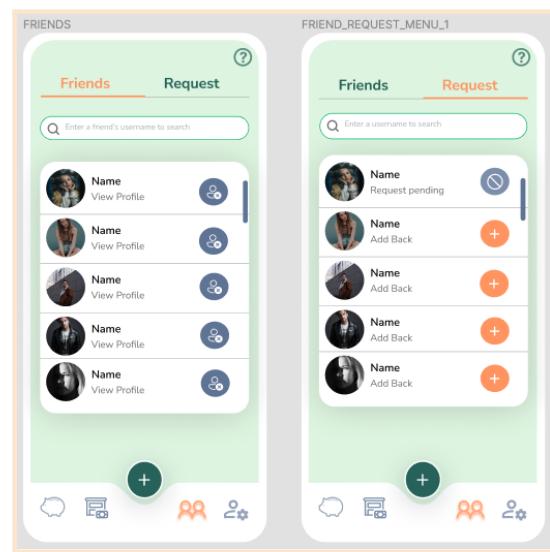
Main home screens:

On the right there are two images and these are the two main screens within the friends domain. The first is the 'Friends' screen and this is the default view when the user enters the domain. This tab allows the user to see all their current friends, view their profiles, and search up their friends.

The second image is the 'Request' screen and this allows a user to accept a request or make one. Like the first screen, the user is also able to search up their requests.

To make these screens as accessible as possible, clear and concise language is used and this corresponds with the heuristic of match between the system and the real world. Additionally, the similar design between 'Friends' and 'Requests' corresponds with the heuristic of recognition rather than recall as it ensures that users do not need to remember the information between each screen. Moreover, user control and freedom has been successfully applied here through the use of the search bar and the ability to scroll on the menu - the user is presented with the option to search for a username, or scroll until they find it. This allows for flexible processes to occur within the system so that the user can choose which one best suits them.

These screens follow a natural hierarchy which allows for easy accessibility for the user to navigate to more options within this domain.



Request Management:

The process of accepting or sending a friend request has been reduced to a very simplified form within this system to ensure convenience for users. The use of icons as well as text incorporates the recognition rather than recall heuristic - many users will be able to identify the purpose of the icon and understand its purpose. The design of this screen also helps the user. By involving green as friend added, orange as friend unaccepted, and grey as pending, as these are colours which the user will typically associate with (green = yes, red / orange = no), this corresponds with the match between system and real world heuristic.

This screen currently requires user input, either to use the search bar or to change the status of a friend. It is very easy for the user to recognise what they have done in relation to the friend status as the user is either prompted with a confirmation (expanded below), or the user can see the state change of the button.

Search Friends and Requests:

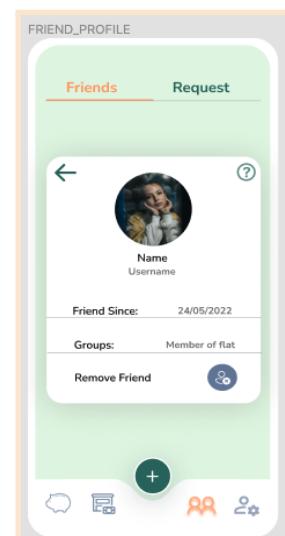
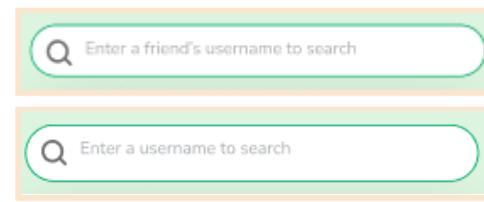
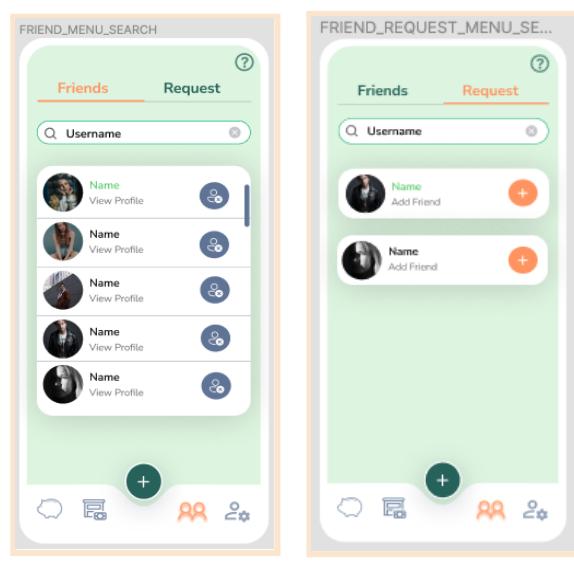
For the sake of convenience and to satisfy user flexibility within the system, a search bar has been included. This search bar appears on both the 'Friends' screen and the 'Request' screen. To help with error prevention, in a light grey text the user is prompted with what to search up. The two images on the right show what happens should a search be successful.

In the 'Friends' section, once a username is inputted the system will display all the friends which meet the inputted username. The most exact match is at the top as this reinforces a good natural hierarchy, and less important results are shown under this. Furthermore, to help convey the search results, the name has been changed to a green colour to demonstrate the success.

The 'Request' section operates very similarly to 'Friends' and this helps with the heuristics of recognition rather than recall, and consistency. Once again, the user is prompted with what to search and should there be a successful result the user is at the top of the list and the name has changed to the bright green colour.

Friend Profile:

This screen is accessible from the main 'Friends' tab. To access it, the user is able to click on 'view profile'. As shown in the image, this screen displays information about a friend. It also allows for the user to delete the friend. Design wise, this screen



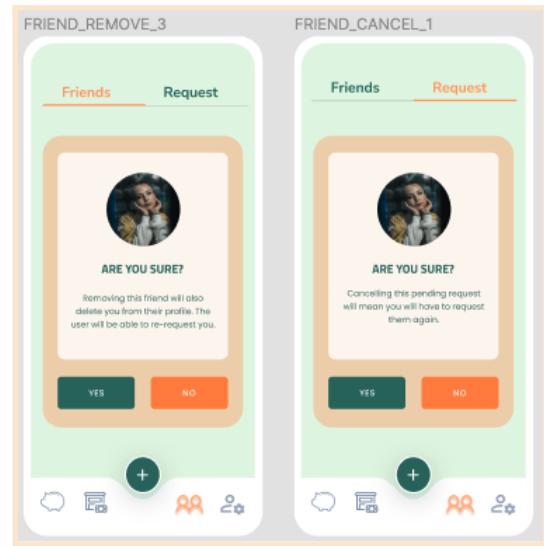
is relatively simple and it only displays the necessary information while still fitting in with the rest of the system's theme. The back arrow in the left corner is a form of user control as it allows for the user to exit the screen and go back in the case this action was performed by a mistake.

Social Domain Confirmation:

These following screens are to ensure error prevention and user control within the social domain. Both screens are relatively similar to ensure consistency, but have small differences as they appear in different parts of the domain.

The first screen is for the 'Friends' tab and it appears when a user wants to remove a friend. The second screen is for the 'requests' and it appears when a user wants to cancel a request which they have sent.

Both screens implement the use of a 'no' button as a form of error reduction, as this allows the user to go back to the previous screen with no changes made in the case that a user has accidentally got to this screen.



Similarities between the two include the use of colour. The green button is for yes, and the orange button is for no. These colours were chosen specifically to ensure the system matches with the real world. Furthermore, both screens have a small description as to what happens should they press the 'yes' button. This more detailed explanation supports error prevention as it guarantees that the user is not confused as to what could occur depending on what button they press. Lastly, in slightly larger text, both screens have the phrase "are you sure?", this helps with the system visibility status as it helps the user to understand what is being communicated to them.

Help Menu:

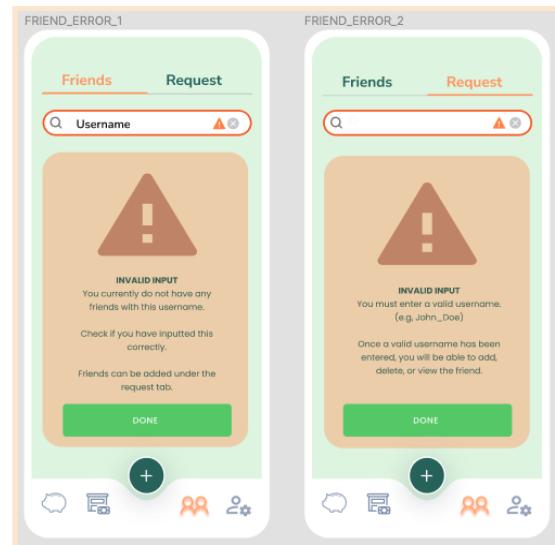
Throughout the social domain the heuristic of help and documentation is met through the use of the following screens. These are accessible by clicking on the question mark which appears on the top right of all screens within this domain (consistency). Recognition rather than recall is also used here as the icon '?' is used to access these screens and the 'x' to close. Concise and simple language is used on these screens to ensure that the user does not get confused and there are clear steps for them to follow next.



Error Handling:

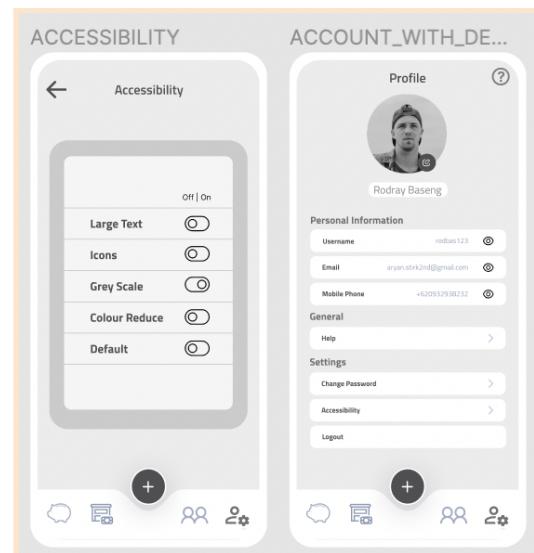
There are two kinds of error messages displayed within the social domain. These both occur when a user attempts to enter a username of a user which they are either not friends with, or the username doesn't exist. Both screens precisely indicate the problem and offer a solution to it.

It is clearly noticeable to the user that something has gone wrong, the use of the colour orange and the error sign which appears twice within each screen.



Accessibility:

Our system has the availability for users to customise their visual experience. This not only allows for use of freedom with the system, but also can be interpreted as a form of error reduction. Users are presented with the choice is set accessibility when they first create their account, but under the profile menu edits can be made. Shown in the images are what the system looks like if greyscale has been toggled on. The accessibility screen was purposefully created to be very simple as it is likely the better who will access this screen with benefit most from its simplicity.



Account [6.5]

Within our system, we have an account page which allows the user to change their name, username, email and mobile phone. This part of the app contains the fundamental features of an account and settings page where you can edit details, change accessibility settings, read help documentation, and log out. This page ultimately influences the way the user logs in as they can change their username and password through this page. We have summarised the help menus in a section as a shortcut for the user if they want to understand how the app works before using it or forget how to do something and want to refer back to it.

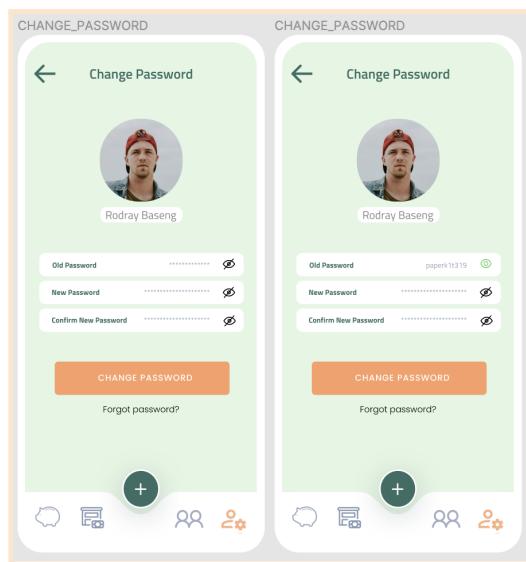


Accessibility

This page allows the user to change the app's visual appearance to satisfy their needs. We added this feature to cater for our personas like Dave, who has dyslexia and focuses on shape, form and colour, as well as Freya who likes to look at a layout that is easy to read and follow, and Debbie who has visual impairment and struggles with small characters and low contrast visuals. Changing the settings of large text, icons, grey scale and colour reduction will hopefully enable these personas to use our app easier. We have kept the screen simple and easy to use by simply clicking the button on and off to switch between the two. The user also has the option to change more than one setting to suit their needs.

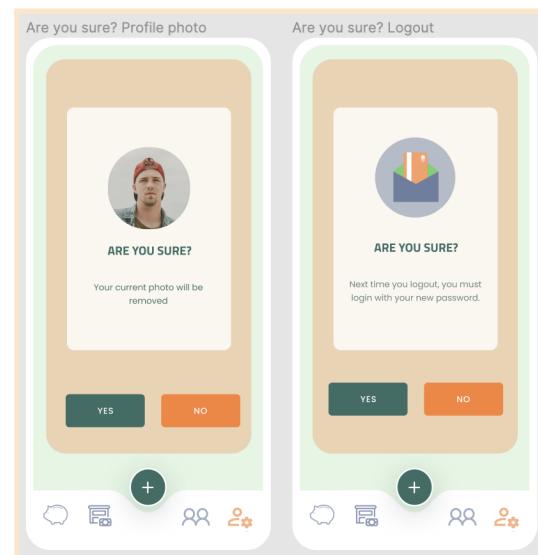
Change password

This section of the app allows the user to change their password. We have kept it simple by having 3 boxes, 1 to input their old password, and the other 2 to input a new password and confirm it. The boxes are white to make it clear to the user what they need to fill out and the big orange button is the focal point which confirms the change of their password. By having the confirm button a different colour, this reminds the user that they must confirm if they want to make changes.



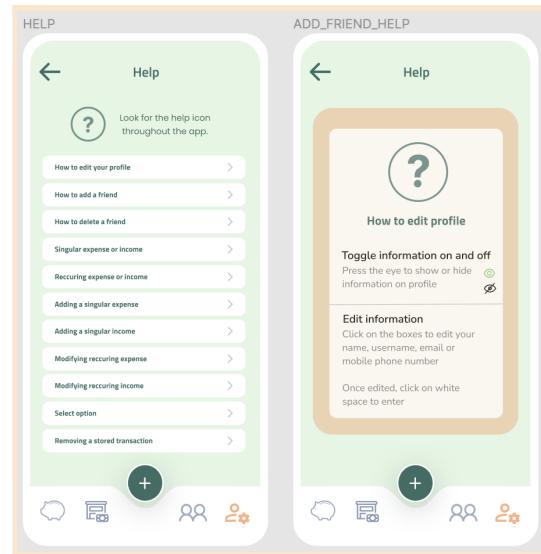
Confirmation (Change profile photo and logout)

These pages are used for error prevention. We have added confirmation pages when the user wants to change their profile photo, password, or logout. In doing so, it prevents the user from making a mistake when they accidentally click on a button that they weren't planning on clicking. It also makes the user confirm their action and ensure that what they are doing is what they want. We have kept the confirmation pages consistent throughout the app by having a big green yes button and a big orange no button. As a user, we would read top to bottom which is why we have placed a relevant image on the top followed by "ARE YOU SURE?" for the user to read first and then answer the question.



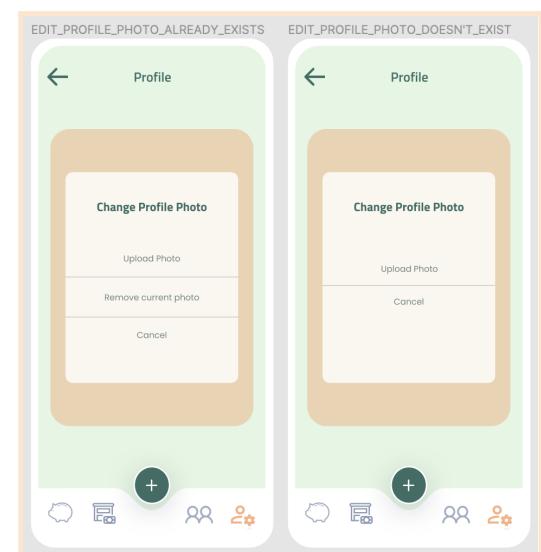
Help Bar

Throughout the app, we have a number of help icons scattered around to help the user navigate their way around the app. Having simple white boxes with an arrow to the right hand side makes it clear to the user that they simply need to click on the box to learn more about a particular section. We have kept all the help pages consistent in this part of the app by having the help icon as the focal point and text underneath explaining what the help paragraph is about. We have kept the design minimalistic, simple and consistent so it is easy for the user to navigate themselves through all the help pages. This section is critical as it provides the necessary documentation for users to perform specific tasks throughout the app. We have kept the help documentation as concise as possible, and we have listed the concrete steps required to successfully perform a task.



Change profile photo

When the user changes their profile photo, there are 2 pages which they will get navigated to depending on if they have an existing profile photo or not. If the user has an existing profile photo, they will have the option to remove it or upload a new one. However, if there is no existing profile photo, they will only have the option to upload one. We have kept the design minimalistic so it is not over complicated for the user. Instead of buttons, we just separated the different elements by a simple line and the user just needed to click the text to pick it as an option. The orange and creme box is to make the page more aesthetically pleasing to look at.



Personal Financial Overview [6.6]

When a user logs in, they are directed to their own personal financial overview section. This is essentially the hub of the entire application. Our personas wanted a few main features in terms of their personal finance management e.g. goal setting, expenditure overview, net income overview etc. These main needs of our personas were what drove the design for this domain of the system.

Personal Management

The personal management page gives an entire overview of the user's personal finances through visualisations and provides some click through widgets for management. This **benefits personas like Samuel Stock who wants a quick overview of his finances** without having to analyse the application too much.

There are three widgets which allow users to manage their personal finances:

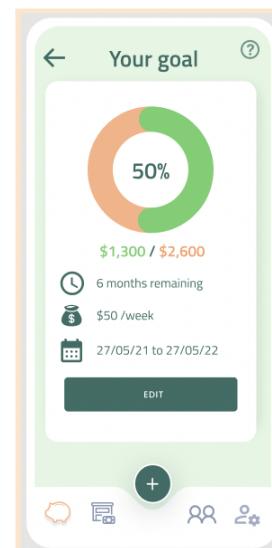
1. Graphical summary of the user's expenses, incomes, and net incomes.
 - o Navigation between these three graphical summaries is easy through the use of the arrow icons below them.
2. Summary of the user's savings goal.
3. Click through to view the user's expenditure.

The navigation bar is also present on every page so that it is always easy to traverse to other sections of the application. The main page is shown here in its three states, showing graphs for; expenses, income, and net income.



Savings Goal

The savings goal page gives an overview of how the user is tracking towards their savings goal. **This was an essential feature of the application for certain personas, such as Takuma Travello** who is saving for a trip home at the end of the year. A feature like this would greatly benefit him. From the home page, the user can see a circular graph of the progress as well as percentage and textual summaries. This means no matter what the accessibility requirements of the user are, they can easily view and use this section. Insights for the time remaining, payment per week and the start/finish dates are also given - along with supporting icons to help those who would benefit from them.

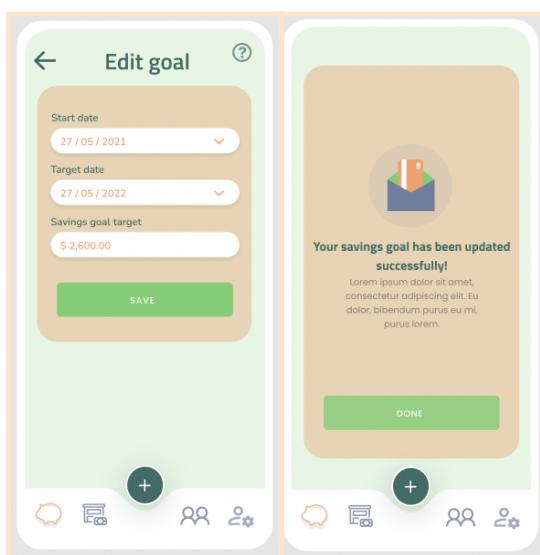


Users can also edit their savings goal from this screen using the “EDIT” button. This takes them to a form where they can configure the details of their goal. The form has error catching built in to let the user know if they enter a wrong savings amount, however the dates use a dropdown input to prevent error in the first place. If the form is submitted with error a page with a message detailing what went wrong is displayed to the user, otherwise if the form is submitted correctly the user receives confirmation.

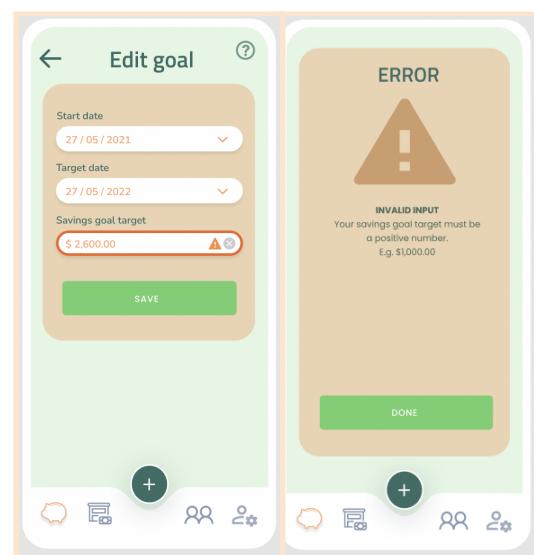
The inclusion of a success message (feedback), error prevention and error handling was inspired by our heuristic evaluation as part of our design review.

These techniques adhere to #1, #5, #6 and #9 of Nielsen's 10 Usability Heuristics. The pages relating to editing the savings goal are shown below.

Success



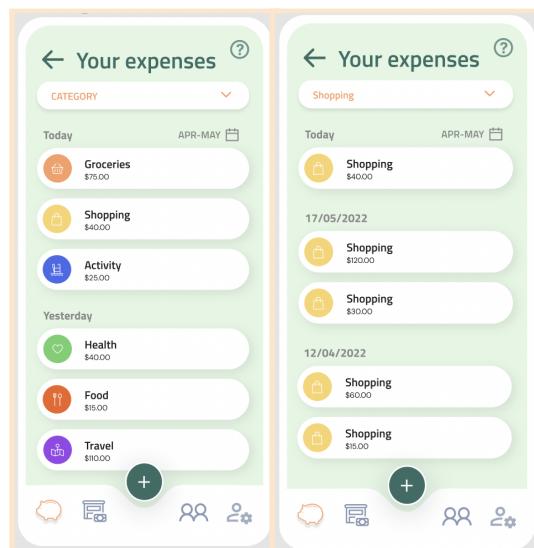
Error



Expenditure

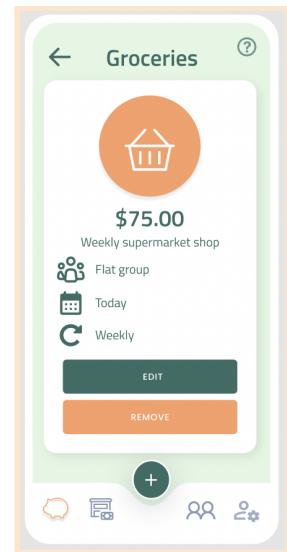
The final main section of the personal finance domain is dedicated to the user’s expenditure. On this page users get a complete timeline of their logged expenses.

On the main page expenses are given as small widgets with a label and corresponding icon for the expense category along with the monetary value of the expense. **The use of supporting icons helps personas with accessibility needs like Dave Difffy who is dyslexic and relies on icons.** Other features include a dropdown menu where users can filter by expense category, to only view expenses under a given category, and the ability to change the date range to only view expenses within a specific timeframe. The expenditure page is shown to the right, with filtering for the “shopping” category applied on the far right.

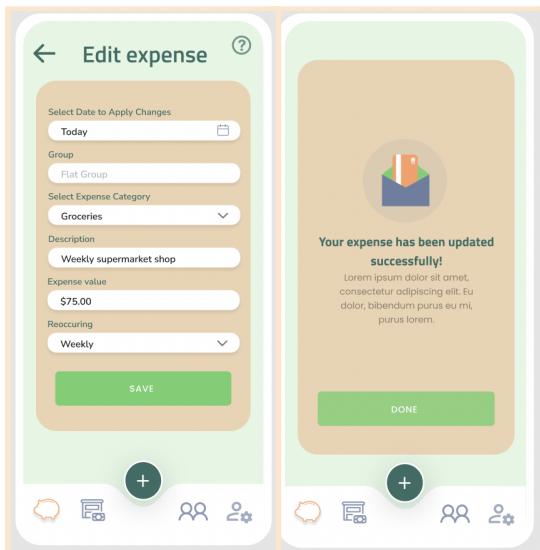


Further to this, users can tap an expense widget to get a complete overview of its individual details. Extra details are given such as; the expense's description, what group (if any) it is part of, the date and whether it is reoccurring or not. Each of these details are supported with an icon to aid quick recognition and help those with accessibility requirements. An example of the individual expense page is shown for a "groceries" expense.

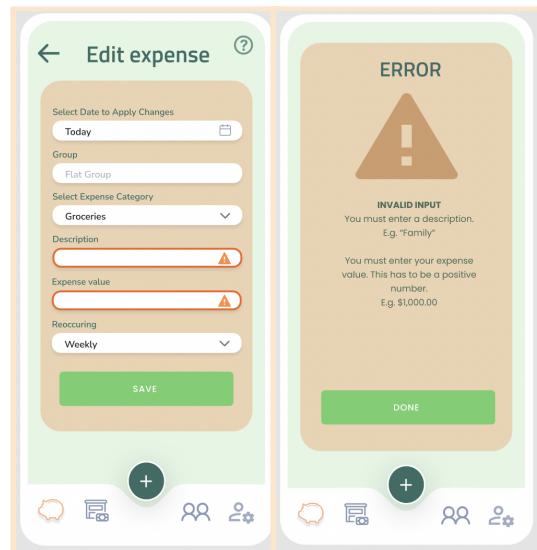
From an expense page users can edit the details of the expense using the "EDIT" button. This takes the user to a page with a form to configure details. The form has error catching for certain inputs, but also makes use of dropdowns to prevent some errors in the first place - **adhering to #5, #6 and #9 of Nielsen's Usability Heuristics**. If the form contains an error when submitted, an error message is displayed to the user, otherwise the user receives success confirmation. Pages relating to editing an expense are shown below.



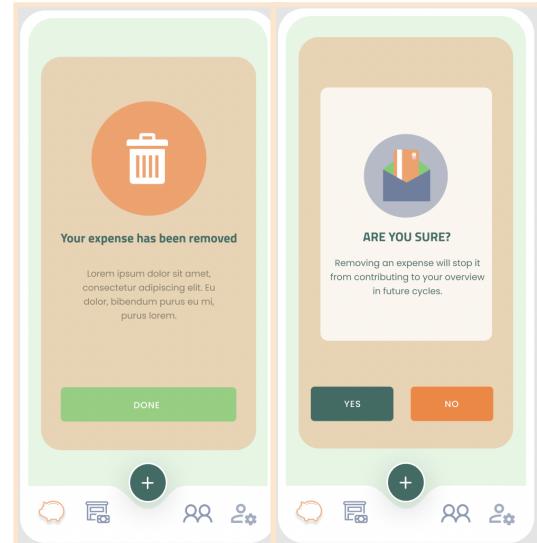
Success



Error



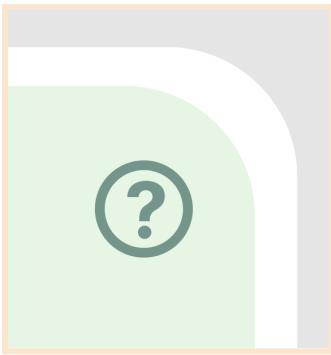
As well as editing an expense, users can also remove an expense using the "REMOVE" button. Users must confirm this action as it is irreversible - which they are reminded of during confirmation. If confirmed, a success page is shown to confirm that the expense has been deleted. The confirmation and success pages for removing an expense are shown to the right.



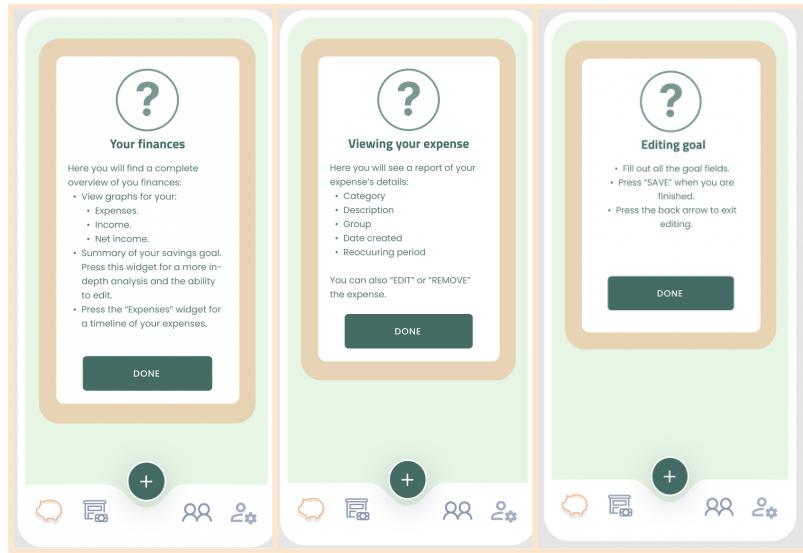
Help Documentation

On most pages of the Personal Financial Overview section there is a help icon in the top right corner. Users can tap this to view help documentation specific to the page they are on. **This satisfies #6 (help in context) and #10 (help documentation) of Nielsen's 10 Usability Heuristics.** For simple pages dealing with confirmation, error messages or success messages, the help icon is not present as they are self-explanatory anyway. Examples of the help icon and accompanying documentation are shown below.

Icon



Documentation



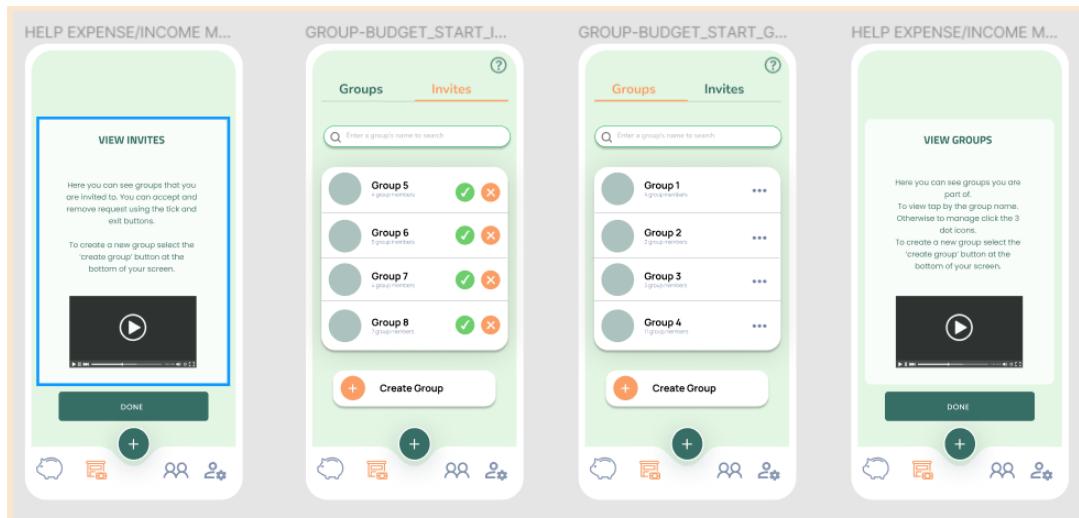
Group Financial Overview [6.7]

In our system we have implemented a way for users to be able to link their financial calculations together into groups. We wanted to include the ability for group financing because our target stakeholders are students therefore a large quantity of them will be in a flatting environment. Having a group section allows for those users to keep on top of all of their joint payments e.g rent, bills etc. We also wanted a group to be able to create a savings goal in order to encourage student saving/ discourage student overspending.

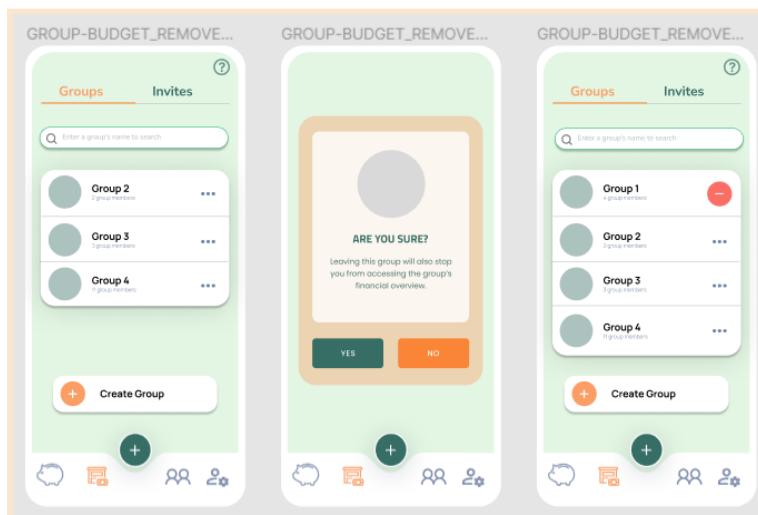
Group Management

The group button (left of the central button) will transport the user to the group section of the system. In this section the user will be able to manage/ create their groups. The start of this section will initially view the user's groups in which they are part of. If the user isn't part of any group they will have the ability to create a new group via the 'Create Group' button positioned at the bottom of the screen. There is a search bar positioned at the top of the screen for users to be able to search up specific groups via their group name. The page can transition to an invites page where the user's invitations to other groups will reside. The user will

be able to choose whether to accept or decline the invite with the green tick accepting and the orange cross as declining. Help buttons are located at the top right of the screens for both pages. They will both have videos explaining how a user may use this section.



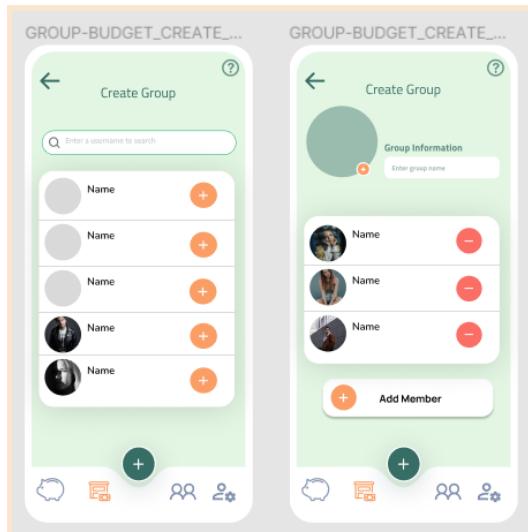
Each existing group can be edited through the meatball icon next to the group name. The dots will allow the user to be able to leave the group. We wanted to hide the leave sign as it would become a big hazard when a user misclicks the leave button. We also placed a warning screen which asks the user if they are sure about leaving the group.



Creating a Group

The create group section is accessed through the 'Create Group' button in the previous section (group management). To keep it simple as well as avoid constant user error we didn't want the user to input too much for a group. However we did want to provide some personalisation so a user is able to choose a name and icon for the group. The initial members of the group can be selected or discarded. A new member can be added via the 'Add Member' button below the member list.

The user will be able to find members through his friends list. Added members will then be sent an invitation in which they will be able to accept or decline. The user can access the create group help page via the '?' icon in the top right corner.



Group Finance Summary

Once a group has been created or selected, the user will then enter the internal group pages. The first page being the summary page. The page will have a basic graph displaying the expenses per each month. At the bottom of the page are icons of expenses that were particularly high at the moment of use. The statement is also available for that month which will include a more in depth/ different visual depiction of the financial overview. The statement will have all the group expenses of that month. The savings target has also been calculated for the user at the bottom of the statement. The group settings section is also accessible from any of the internal pages. To exit back to the main section of the summary there is a green back arrow for a clear user friendly icon.



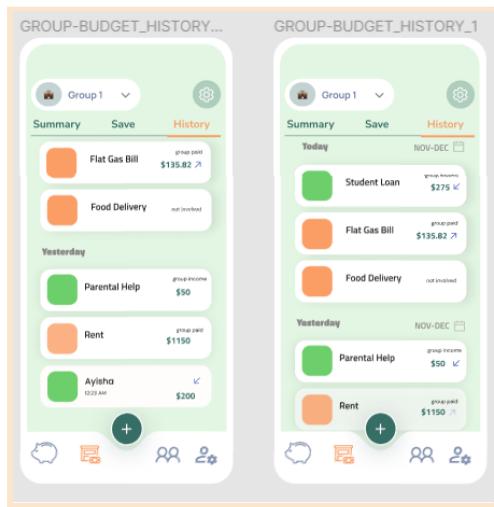
Group Saving

The group savings section is designed for groups to be able to set a financial goal. The user will have the ability to set a new goal or change an already existing one. The start and end date of the goal. The data will show if a goal has been met each day leading up to the final date. The goal will show whether it has been met each day leading to the target date.



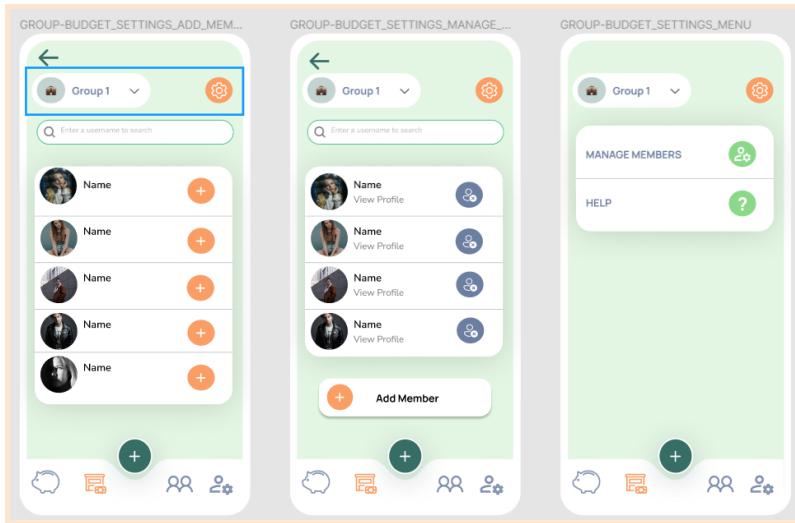
Group History

The group financial history section is designed for groups to see their past transaction history. We can see via the screenshots that it shows incoming and outgoing payments from all the expenses set to this group. We can see the orange square represents payments and green squares represent incomes. Green is a positive colour meaning having it represent income would be an easy recognition for a user. Orange is a similar colour to red therefore it is a great colour from our colour scheme to use as the expense, users would relate losing finances as a negative and gaining as a positive.



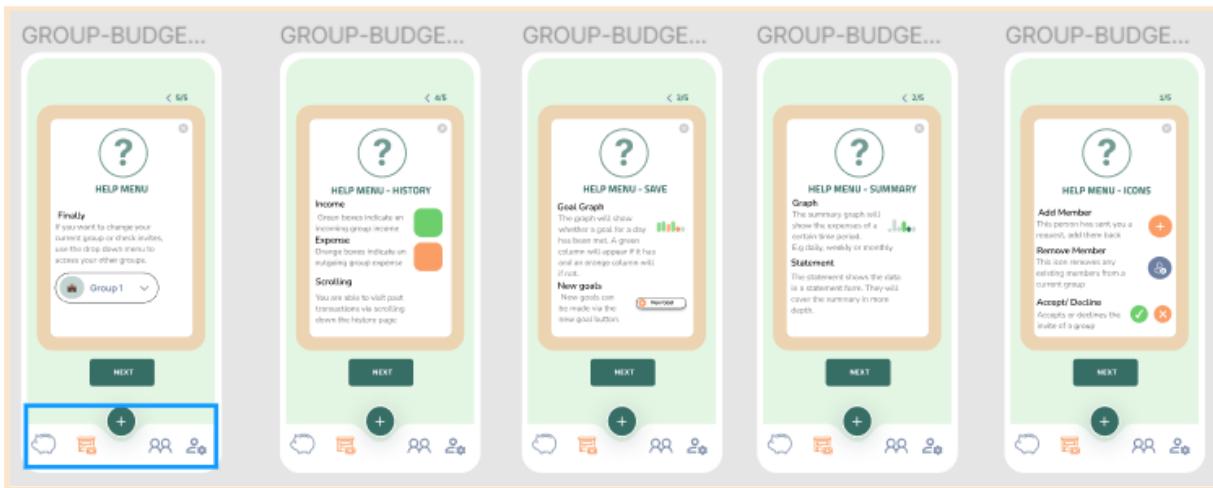
Group Settings

The group settings can be accessed on each page in the internal group sections e.g (summary, save, history). The settings allows the user to edit an existing group or to read a help slideshow. We wanted the group to be adaptable therefore we made sure the group can easily remove and add members.

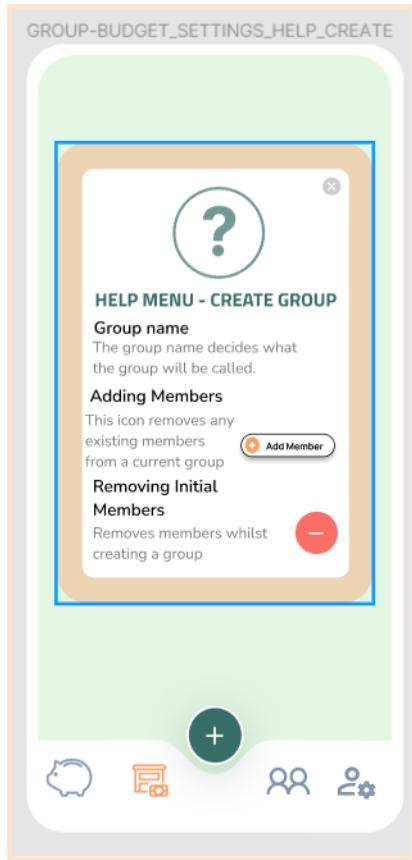


Help Menus

These are the help pages accessed through the group settings. Designed to act as a step by step design, we wanted the user to be able to have a basic introduction to how the group section works. As we can see, each step is a part of the group section. Each help page has been designed to be simple and easy to understand, helping a user to learn each step without an overly long drawn out tutorial.



This help page is accessed through the create group section. The page will aid the user in being able to create a new group. It explains the buttons and inputs the user can interact with whilst being on this page. Keeping the create group simple means the help page wouldn't need as much reading up and a user that won't be overwhelmed.



Error Handling

To try to avoid as much user error as possible, we try to implement most of the user input as drop-down menus. However, we have implemented error screens in some situations where input requirements are more arbitrary. The majority of these errors will be caused by text bars. Included in these errors are two different types of error. One being when a user inputs a name which is not included in the name list or hasn't inputted anything. With both of these errors in different situations, we have implemented an error screen for them.

