COSC368 Assignment
Flatting App Design Concept

Authors:

Kaishun Yang (kya31)
Tawatchai Holmes (tho78)
Yoseph Bogale (ymb12)
Chathuranga Alwis (vca35)
Ziling Huang(zhu51)
Yiyang Yu (yyu69)

Abstract

Our design team will present the most effective flatting app concept to satisfy the tasks users' needs. In terms of app design, we will first introduce our representative user groups and each specified task. We will then create corresponding sketches, from which one will be carefully selected to be part of the final design. To improve the app quality and efficiency, We will describe and refine our early sketches and then extract the most predominant features from alternatives and incorporate them into the selected design. In the final section, we will show our final design with its explanations and conclude the entire project.

Introduction

At the request of FlatMate Ltd, our team will design a mobile app to help flatmates effectively manage their flat. The assignment will identify the app's representative users and discuss their main featured tasks and associated priorities. This knowledge was at the front of our minds when exploring a wide range of design sketches that can support the users in accomplishing those tasks. We will introduce the design rationale that was used to produce the final design. To build this app to its full potential, A task-centred system design process was used which helped our team identify users' goals and evaluate design decisions regarding any system improvements.

User Identification

In the design brief, the CEO from FlatMate Ltd expressed the goal for the final application to "become indispensable for managing and coordinating flat activities". These activities are highly likely to be needed by students and/or young professionals living in flatting situations. Hence, the software will be designed for these sets of users in mind. Tenants and Managing Tenants are the representative users/stakeholders identified, and the importance of supporting each user group is discussed below.

Tenants:

A tenant is a user who rents and lives in a flat with other tenants. Being a tenant, what matters the most is managing their task and paying their rents on time in a flat. These interface functions need to be designed in a way that helps the tenants learn how to use the app effectively and comfortably. There must be one tenant authorized as the Managing Tenant, as detailed below.

Managing Tenant:

The Managing Tenant is the user who is the head of the flatting group, with more responsibilities managing the flat. As the person responsible for the flat, they have full power to set and manage payments and tasks required to complete in their flat. A Managing Tenant is also a tenant of that flat; therefore, the interface must communicate distinctions between these two modes and which mode a user is currently on.

Task Identification

As this app's main goal and purpose is to improve the flatting experience of tenants and fulfil their requirements, we defined multiple tasks that support this. These tasks can be categorized into many subtasks as defined below.

Please note that to exemplify the tasks specified below, tenants Alice, Bob, Eve, Ben (Managing Tenant) who reside in the same flat will be used.

1. **Create and Manage Flat Tasks**

- **Create Task**
    - Creating a task that needs to be completed by a flatmate or a group of flatmates. Each task will contain a description, timestamp and the name of the user who created it.
    - Provides functionality to tag other flatmates to the task such that it notifies those tagged flatmates once the task has been created and assigned.
    - Provides functionality to set a deadline for each task, but it is optional.
    - ***Scenario:*** *Alice creates a task '@Bob please vacuum the living area.' and assigns it to Bob. All flatmates will get a notification and Bob will be informed that he has been tagged on a task.*
    - ***Scenario:*** *Alice needs everyone to clean their rooms, so she creates a task '@Everyone, please clean your rooms' and assigns it to the whole flat. All flatmates will get a notification and be informed that they have been tagged on a task.*
- **Edit / Delete Task**
    - Provides functionality to modify/edit a task created by the user.
    - ***Scenario:*** *Alice decides to edit the description of the task created to "@Bob please vacuum the living area asap." and sets a deadline to '27 October 2021'.*
- **View Task**
    - Provides functionality such that flatmates can view tasks created by another flatmate.
    - ***Scenario:*** *Eve sees the notification indicating that Alice has created a task and decides to view it.*
- **Mark as Committed**
    - Provides functionality to the user to mark a task as committed. However, if the task is tagged and assigned to one or more flatmates, other flatmates who are not tagged cannot mark the task as committed.
    - ***Scenario:*** *Bob sees the notification indicating that Alice has tagged him on a task and marks it as 'Committed'.*
- **Mark as Completed**
    - Provides functionality to the user to mark a task as completed. However, this feature is only available to flatmate(s) who have committed to the task.
    - ***Scenario:*** *After vacuuming the living area, Bob views the task and marks it as 'Completed'.*

2. **Manage user account**

- **Sign out**
    - Provides functionality to sign out of the account.
    - ***Scenario:*** *Alice wants to sign out of her account.*
- **Set up/modify/view account info**
    - Provides functionality to set up, change and view account info by users.
    - ***Scenario:*** *Bob wants to add his bank account number and change his profile picture in his account.*
    - ***Scenario:*** *Alice wants to check her account number.*
- **Delete user account**
    - Provides functionality to delete accounts by users if the account doesn't have flats that have tenants living inside.
    - ***Scenario:*** *Eve wants to delete her account.*
- **Save Bank Details**
    - Provides functionality to add and save your bank details to be remembered for future payments.
    - ***Scenario:*** *Bob uses the same bank account to make payments and is tired of entering her details again and again. She wants to save time in future payments by having her bank details saved for quicker selection.*
    - ***Scenario:*** *Ben prefers to use one bank account for her weekly rent and another for one-off payments. Kat can save both bank account details from choosing between for quicker selection in future payments.*

3. **Managing Payment**

- **View bill**
    - Provide functionality for Managing tenant/tenants to view the payment bill.
    - ***Scenario:*** *Alice wants to view the bill, so she knows what she's paying.*
- **Pick payment method**
    - Provide functionality for Managing tenant/tenants to pick different payment methods (e.g. Credit card, Debit card, etc.).
    - ***Scenario:*** *Ben wants to pay rent using Internet Banking instead of his Debit card.*
    - ***Scenario:*** *Bob wants to pay rent using his Debit card.*
- **Payment history**
    - Provide functionality for Managing tenant/tenant to view their history payment.
    - ***Scenario:*** *Eve wants to view past payments.*

4. **Manage Flat group**

- **Create Flat group**
    - Functionality the Managing tenant to create a flat group.
    - ***Scenario:*** *Ben (Managing tenant) wants to create a flat group to manage the flat.*
- **Add tenant to flat group (Flat ID)**

    - Generate a random unique Flat ID, once the Managing tenant creates a flat group for the tenants to use to be able to join the flat group
    - ***Scenario:*** *Alice wants to join the flat group, so she asked Ben for the Flat ID*
    - ***Scenario:*** *Ben send the Flat ID to Alice, Eve and Bob, so they can join the flat group*
- **Delete Flat group**
    - Functionality to delete the flat group (Only for Managing tenants).
    - ***Scenario:*** *Ben (Managing tenant) wants to delete the flat group because he moves to Auckland and buys a house.*
    - ***Scenario:*** *Ben (Managing tenant) wants to delete the flat group because he won't be flatting anymore.*
- **Set up/modify/view flat details**
    - Functionality to set up/modify/view flat details for all tenants, where the tenant can only view the flat details.
    - ***Scenario:*** *Ben (Managing tenant) wants to change/modify some details about the flat.*
    - ***Scenario:*** *Alice wants to view the details of the flat.*
- **Set up/modify renting fee**
    - Functionality for the Managing tenant to set up rent fees for themselves and the tenants.
    - ***Scenario:*** *Ben (Managing tenant) wants to set up rent fees for the entire flat*
    - ***Scenario:*** *Ben (Managing tenant) wants to modify the rent fees, as Ben manages to get the rent cheaper.*
- **Remove tenant flat group**
    - Functionality for the Managing tenant to remove tenants from the flat group
    - ***Scenario:*** *Ben (Managing tenant) wants to remove Bob from the flat group, because he found a new flat to rent.*

5. Communication between Tenants

- **Group messages**
    - Communicate with the entire flatting group.
    - ***Scenario:*** *Ben wants to message Alice and Bob in the group chat about the study session "@Bob @Alice are we still doing the study session today at the library?" in the group chat.*
    - ***Scenario:*** *Ben wants to message Alice, Bob and Eve about the flat task he set for them "@Ben @Bob @Eve I have set up some task for you guys to do around the flat" in the group chat.*
- **Directly messages**
    - Communicate directly to another user
    - ***Scenario:*** *Bob wants to send a direct message to Ben (Managing tenants) about his payment issues.*

    - ***Scenario:*** *Ben (Managing tenants) wants to send a direct message to Alice about late payments.*
- **Remove/Add tenant group chat**
    - Add functionally for Managing tenant or Admin of group chat to add/delete tenants from group chat
    - ***Scenario:*** *Ben (Managing tenant) wants to add Alice to the group chat*
    - ***Scenario:*** *Ben (Admin) wants to delete Bob from the group chat*

6. **Security**

- **Sign up**
    - Signing up by email address for the app.
    - Can sign up with either Facebook, Google or Apple accounts.
    - After signing up, the user will be asked to verify themselves by uploading their IDs(either passport or driving licence).
    - ***Scenario:*** *Alice wants to create a new account, 'alice123', by email address:alice@gmail.com, password:'alicepassword'*
- **Login**
    - Logging in directly to the app
    - ***Scenario:*** *Alice wants to log into the app from her new phone.*
    - *Receives a text code for authentication when logging in.*
    - *The **password** form is only revealed when you enter the correct email address.*
- **Password resetting**
    - Resetting password for an account.
    - ***Scenario:*** *Alice wants to reset her password.*

Priority Analysis of Tasks

We performed the priority analysis by considering the frequency of use and importance of each task and its subtasks. In carrying out this analysis, we intend to assign priority scores for all tasks defined above to enable us to make design decisions. The priority scores for each of the main tasks are assigned by taking the average score of its subtasks.

The following table is used as a reference when assigning the priority scores for tasks.

		Frequency of use		
		Low	Medium	High
Importance	Low	C	C	B
	Medium	B	B	A
	High	B	B	A

Explanation of Priority Scores	
A	High Priority: Indicates a crucial part of the app which needs to be designed well to ease the functionality and needs to be implemented.
B	Medium Priority: Indicates an essential part of the app which needs to be designed well. Recommended to be implemented.
C	Low Priority: Indicates a part of the app which is not vital. Optional.

1. Create and Manage Flat Tasks		**Priority Score: A**
Create Task	Priority Score	A
	Frequency	This task will be performed often by flatmates whenever something needs to be done. Thus, the frequency of use is **high**.
	Importance	Crucial selling point of the app. Users would need to use a different app or a manual method to assign tasks to flatmates otherwise. Thus, importance is **high**.
Edit / Delete	Priority Score	B
	Frequency	This task will be only used if the user wishes to edit/delete a task. Therefore, frequency is **low**.
	Importance	Importance is **medium** as it is vital to provide a backdoor to undo an action such as creating a task by delete and editing options.
View Task	Priority Score	A
	Frequency	This task will be used by flatmates each time someone in the flat creates a task. Therefore, the frequency is **high**.
	Importance	Importance is **high**, as without being able to view a task, the user can't get more information about the task to make further decisions.
Mark as Committed	Priority Score	A
	Frequency	Frequency of use for this task would be **high** because most of the tasks would require a flatmate to do some task in the flat as 'washing dishes'.
	Importance	Importance is **high** because without it the app cannot provide feedback on the status of a task.
Mark as Completed	Priority Score	A
	Frequency	Frequency of use for this task would be **high** because most of the tasks would require some task to be done in the flat.
	Importance	Importance is **high** because without it, the app cannot provide feedback on the status of a task.

2. Manage user account		Priority Score: B
Sign out	Priority Score	B
	Frequency	The task will not be frequently used, so it will be **low**, as users won't be signing out as often.
	Importance	The importance of this feature is **high** as it's a feature to offer users to be able to sign out of their account.
Set up/ modify/ view account info	Priority Score	B
	Frequency	This task will be used often(**medium**) when the user wants to view/modify their account or set up their account when they first sign up/login
	Importance	**Medium** importance, as the user won't be viewing their account info or modifying their account as often. However, it's still an important feature to have for customizing their account.
Delete user account	Priority Score	B
	Frequency	The frequency of this task will be **low**, as there's low likelihood for users to delete their account, and if so, they can only do so once.
	Importance	Importance is **high**, as we must provide this exit for users to remove their accounts.
Save and modify bank details	Priority Score	B
	Frequency	The frequency of this task is **low**, as the user only needs to complete this task once to set up bank details for payment, and it's not often for users to change bank details.
	Importance	Importance is **high**, as it is a payment process required for users to fill up their bank details. Users should have freedom to change their bank details however they want.

3. Managing payment		Priority Score: A
View bill	Priority Score	A
	Frequency	It is arguably the most frequently used task in the app. In addition to occasional checks, each time before and after making a payment, users will very likely use this to check on their bills. The frequency is exceptionally **high**.
	Importance	Along with paying bills, this is the most powerful and important (**high**) feature app will provide to users. Users need to be able to view how much they are going to pay.
Pick payment method	Priority Score	B
	Frequency	The frequency is **medium** for this task as it depends on how many payment methods and how often the user would switch them.

	Importance	The importance is **medium**. It's a crucial task that allows users to change their payment method if they have more than two.
View payment history	Priority Score	A
	Frequency	The frequency is **high**. Similar to viewing bills, this task would be used quite frequently as users need to check if payment has been changed much.
	Importance	The importance is **high**. Users should have the power to check their payment history whenever they want.

4. Manage Flat group		**Priority Score: B**
Create Flat group	Priority Score	B
	Frequency	This task is not used that often (**medium**). It's only when the managing tenant wants to create a flat group.
	Importance	The functionality of this task is really important (**high**) because if a tenant wants to join a flat then there needs to be an existing flat group.
Add tenant to flat group (Flat ID)	Priority Score	B
	Frequency	The Frequency of this function will be very **low**, because the managing tenants add tenants to a flat group a few times a year. Once a tenant is assigned to their flat, then there is no need to join another flat group.
	Importance	Importance is **high**, because if tenants are not added to a flat group then they are unable to view or complete their required task.
Delete Flat Group	Priority Score	B
	Frequency	This task won't be used as frequently(**low frequency**). This task can only be performed by the managing tenant and it's unlikely that the managing tenant will delete the flat group, at least not in a few years.
	Importance	This functionality is very important because if the managing tenant wants to rent another flat or buy their own house, they will have to leave their current flat group first.
Set up/ modify/renting fee	Priority Score	B
	Frequency	The frequency of using this task is medium. The managing tenant changes the flat detail not that often.
	Importance	This task is critical (**high**). For example, the flat rent is $230, and the landlord made a discount and decreased the price to $200, but the managing tenant did not update the rent to $200. The tenant will be paying the original amount and did not benefit from the discount.

Remove tenant from flat group	Priority Score	B
	Frequency	This task won't be used frequently (**low frequency**) because a tenant will not leave their flat often(they will stick around for a few years at least).
	Importance	This functionality is important (**high**) because in order to join another flat, you will have to request to the managing tenant to be removed so you can go and join another flat. If you are unable to leave your current flat, you will be stuck in that flat and it will be hard for you to join another flat.
Set up/modify/flat details	Priority Score	B
	Frequency	The frequency of using this task is **medium** because the flat detail is not changed often.
	Importance	The functionality of this task is really important (**high**), because the tenants rely on it to complete their task and view their flat details.

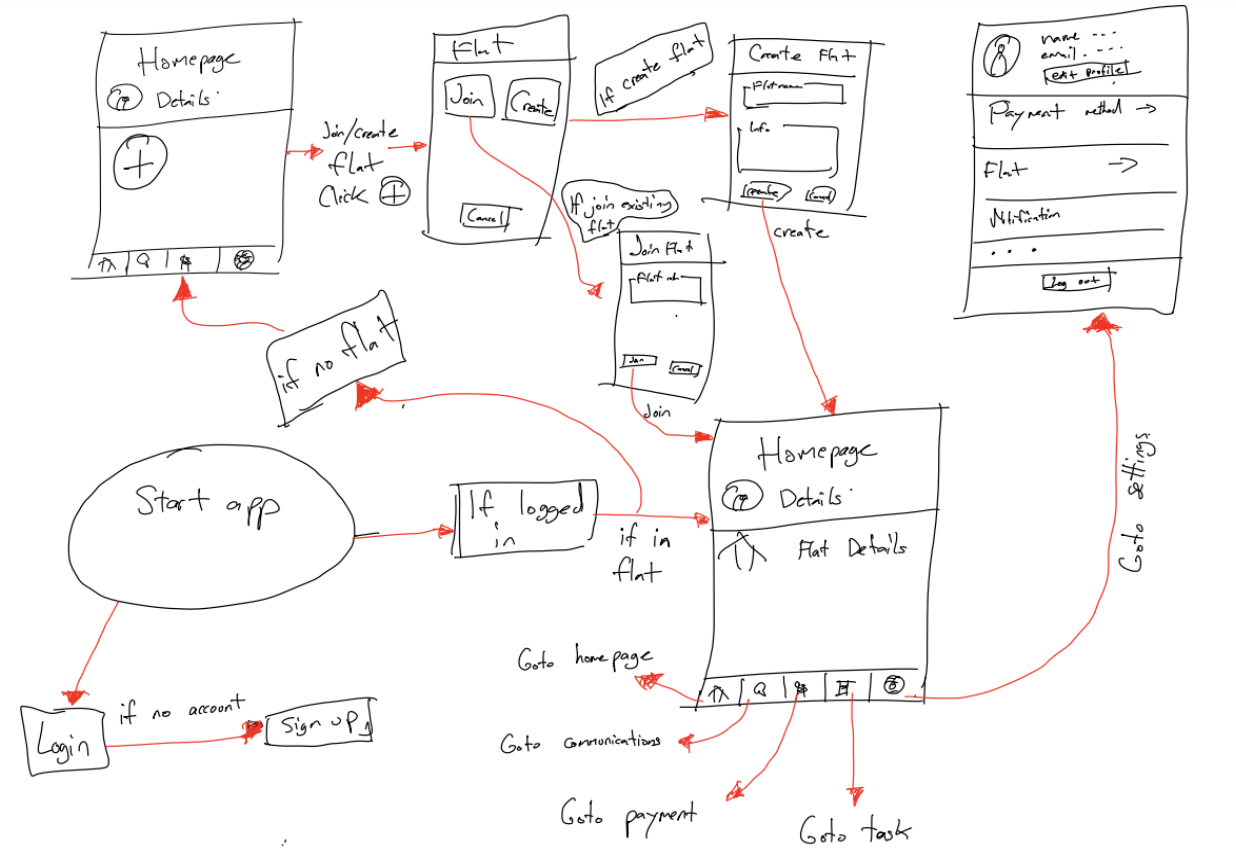
5. Communication between tenants		**Priority Score: B**
Group messages	Priority Score	C
	Frequency	**Medium** Frequency, as this will only be used if all the tenants have a problem within the flat or when one tenant wants to send an important message to other tenants.
	Importance	**Low** Importance, because most tenants would use direct message to communicate with one another and this will only be used if the Managing tenants want to announce to everyone about payments or tasks within the flat
Directly messages	Priority Score	A
	Frequency	The Frequency of this function will be very **high**, because this will be the number source of communication between the tenants or amongst the tenants
	Importance	Importance is **high**, because this will be the main source of communication directly between tenants
Remove/Add tenant group chat	Priority Score	B
	Frequency	This task won't be used as frequent(**low**), this task will only be used when the admin/managing tenant in the group wants to add or delete a tenant
	Importance	**Medium** importance, as the admin/managing tenant would like to remove/add tenant to the group chat depending on the situation

6. Security		Priority Score: A
Sign up	Priority Score	A
	Frequency	The task will not be frequently(**low**) used since each tenant usually only needs one account
	Importance	The importance of this feature is incredibly **high** as it's the feature to offer users safe and legitimate accounts for approaching all other features. Without this feature at first, users literally don't have accessibility to the app.
Login	Priority Score	A
	Frequency	The task will be often(**high**) used. Each time the tenants want to check their profile, they have to log in first
	Importance	Really important(**high**) since the tenant will not be able to check our app without their account info
Resetting password	Priority Score	B
	Frequency	This task will only be used when the user cannot remember their password(**low**). Some of the users may use it once or twice, and some of the users may never use this function. Besides, nowadays the browser can remember passwords automatically.
	Importance	Importance is **high**. When a user needs to reset their password, it is really important to let the user have control as soon as possible.

Preliminary Sketches and Alternatives

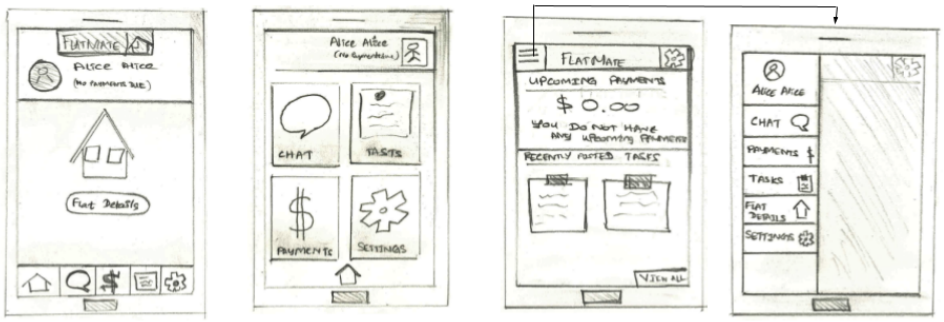
In this section, we will introduce every early sketch and its rationales and add each sketch's pros and cons. Among all these designs, we will have a principal design elaborated in the final design. The rest of the designs will be its alternatives used to accentuate its predominant features by making comparisons. This section is to help FlatMate to be able to recognise what traits a good design has and what these design rationales help with app creation.

Main Menu Design



- When the Tenant opens the app, they will be sent to the login page if they aren't logged in or be sent to the Home page if they are logged in.
- The tenant can navigate through different pages using buttons on the home page (e.g. communication, payment, tasks and settings).
- If the tenant is not part of a flat, they can join or create a flat using the "Add Button" (Tenant can only be a part of one flat at a time).
- When Joining a flat the tenant will need to input a flat code. This will allow them to join a flat.
- When the setting button is clicked, the tenant can edit/modify their account, payments, flat, etc.

Homepage Designs



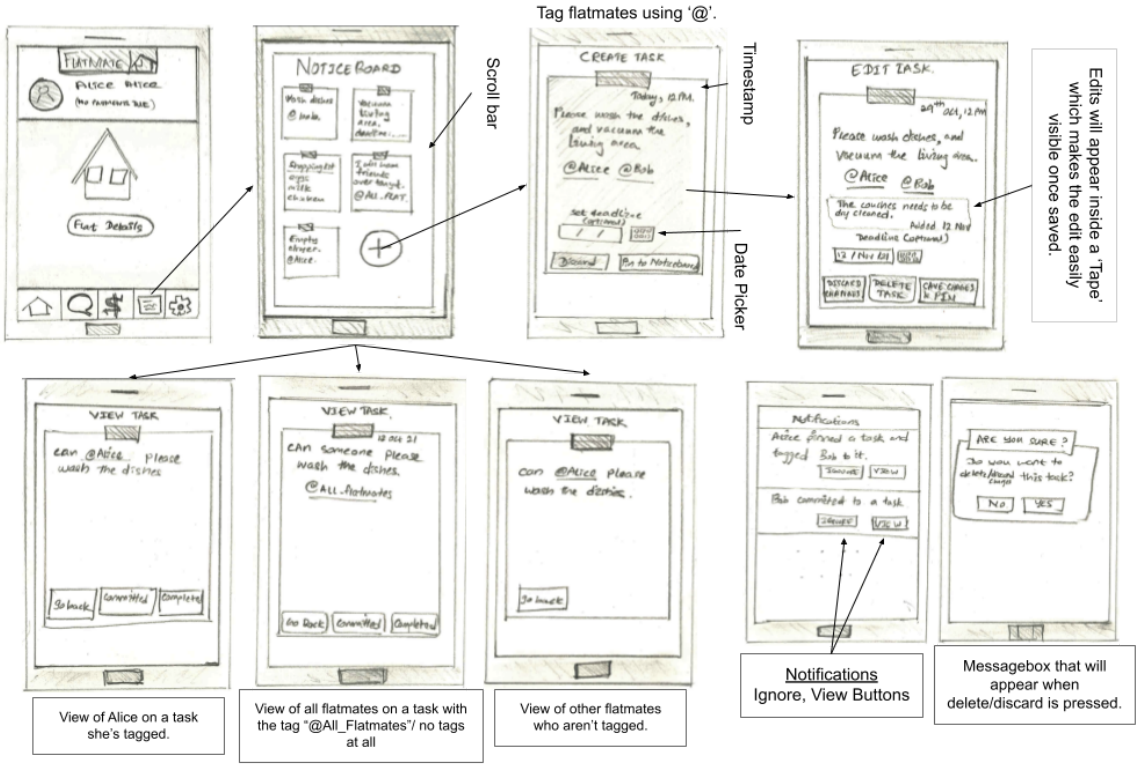
Simple and Generic | Easy to use, Tile Based | Modern, Informative

The homepage is used each time a user opens the app, and so it's design needs to be as simple and user friendly as possible. Thus, we sketched multiple different designs, and the above are the best three designs we selected.

1. Create and Manage Flat Tasks

1.1. Noticeboard Flat Tasks Design

This design is simple and easy to use. The users can just type the message/task on the note together with '@' tags in between tagging flatmates to those tasks.



Pros:

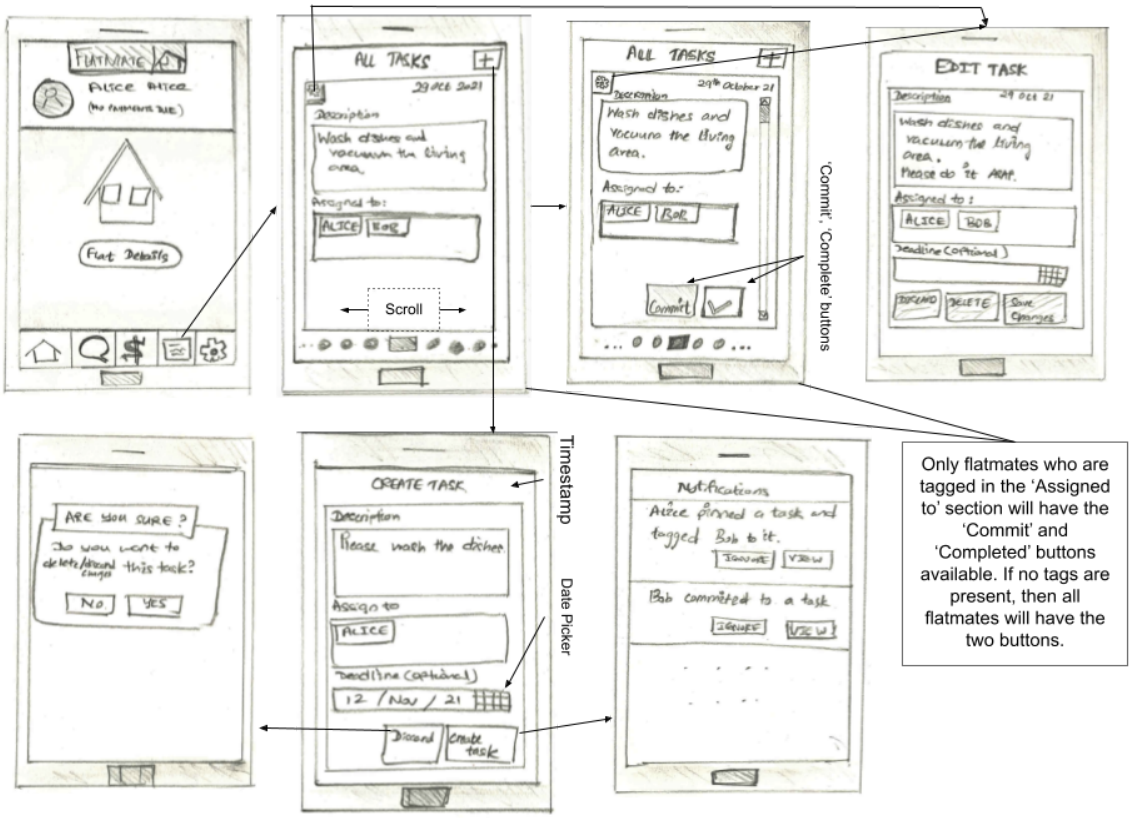
- Simple and natural dialogue and easy to use as it is unstructured.
- The noticeboard design, which speaks the user's language, makes it convenient and natural.
- Viewing all the tasks is easy and fast through the vertical scroll feature; thus, it increases app efficiency and flexibility.

Cons:

- There is no edit button. You can just type on the note to edit, and so it may cause confusion.
- The 'Create Task' button needs to be shifted to the top. Otherwise, the user has to scroll all the way down to create a task.
- Message Box buttons need to be named according to the task to avoid misclicks and errors.

1.2. Horizontally Scrollable Card of Flat Tasks Design

This design is structured and organized such that creating, editing tasks will have their own specific text fields to add a description, tag flatmates, set deadlines etc.



Pros:

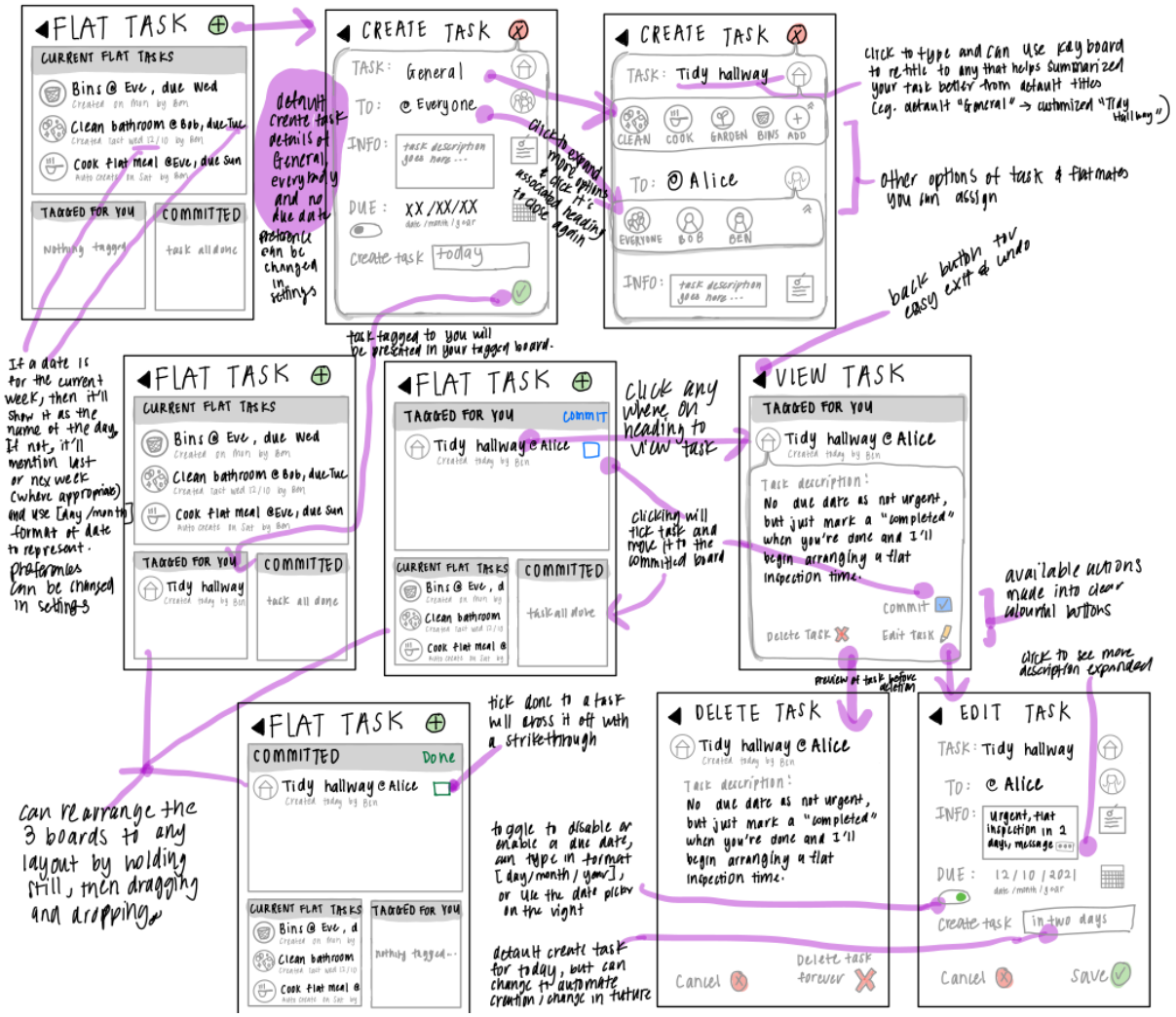
- Structured, and organized as it is divided into different subsections with text boxes. Hence, it provides better visibility for users to operate.
- Easy to view a task without having to open it. So users can focus on more essential features without getting interrupted or distracted.

Cons:

- Quite slow to search for a task you want to view, as you need to scroll through each one. (Not efficient)
- Since it is structured, creating and editing would take more time compared to the previous design.
- Message Box buttons need to be named according to the task to avoid misclicks (Current design is not error preventive.)
- The description text field is too small and will need its scroll bar, thus making it more complicated and packed.

1.3. Progress Board Flat Task Page Design

Inspired by scrum boards, this design uses three different boards to track the progress of their flat task. The boards can be customized to any size and position on the screen.



Pros:

- Creative new task management system, with separation of user's task.

Cons:

- May have trouble speaking the user's natural language as scrum boards are mostly only familiar to a small group of users.
- The presentation of the interface is not the most natural and straightforward, with lots of boxes and windows that will overwhelm the user's visual input.

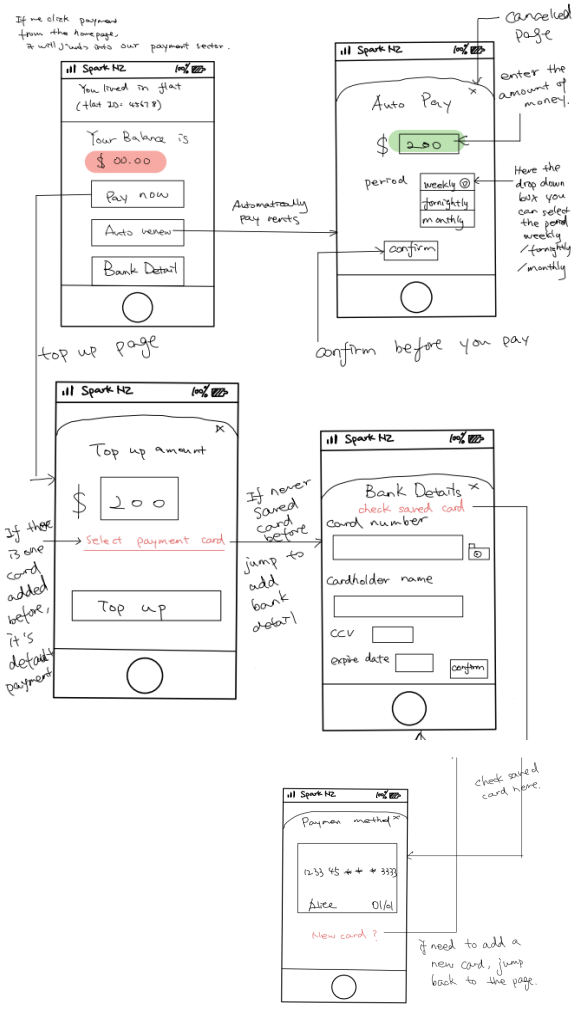
Conclusion for Flat Tasks Page Design

After going through all the sketched designs, we decided to go with the noticeboard design. It is much more understandable and natural to users. Nevertheless, we will include the good features from the other two designs in our final UI design.

2. Payment Page Sketches (Manage payment)

2.1. Minimalist Payment Design

On the page of the payment sector, on the top, it shows the flat you lived in and the flat ID as a reference. Below that, you can see the balance of your account is in the middle. Then there are three buttons, "pay now", "auto-renew", "bank details" which allows you to pay the bill, set an automatic top-up or check the card's details, respectively.



Pros:

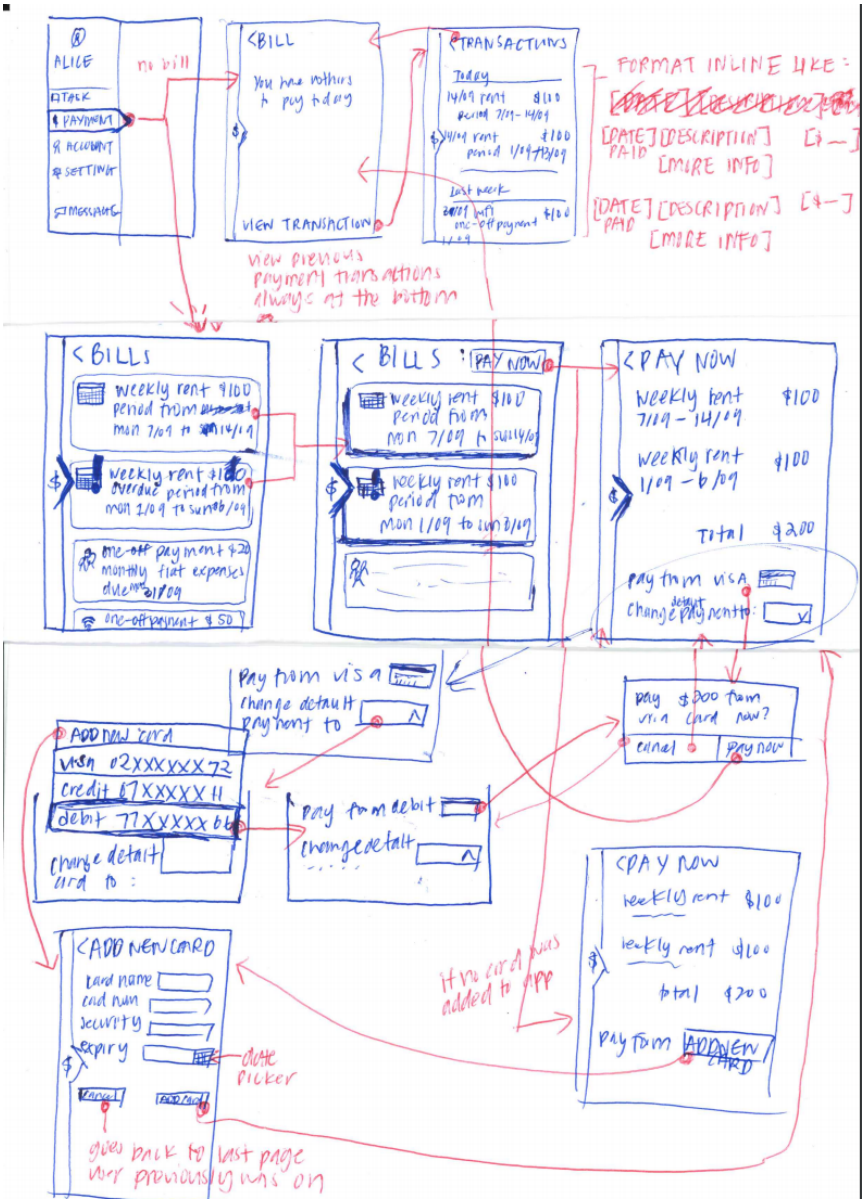
- Simple and nice UI design with three clear payment decisions to make all collected in the same page.
- Intuitive colours to communicate a low balance in red and a valid balance in green.

Cons:

- There is no indication of where you are at in the payment process or how close you are until completing. If users want to jump from one page to another, they have to cancel the whole payment page and start from the beginning of the payment sector.
- There is no clearly marked exit. If a user misses clicking one button, it is hard to go back to the previous page.
- The user's task is to "make a payment". Therefore they are unlikely to be interested in the flat information (such as flat ID), which negatively eats up precious screen space for them to complete the task at the front of their mind - making a payment.

2.2. List Structure Payment Design

Users can skim down a page to find the information they need quickly. Viewing current and previous transactions, your bills, your cards will have their details organized in a structure where related information can be found by scanning down a page.



Pros:

- Organized structure helps user's visual searching and processing of information into manageable chunks.
- A page with less clutter when there is no bill to pay, which is tidier and less noisy, and may encourage users to achieve this cleaner state by paying their bills asap.
- Easy to recognize icons such as a calendar, exclamation mark, group, wifi, to quickly identify what type of bill they are receiving
- Clear indication of what mode/page you are on with a side panel for navigation that has it's tab out with a payment "$" icon.
- Each page has consistent titles, page layout and back "<" button to provide clearly marked undo and exit of the current page you are in. Easy to find undo and exit is

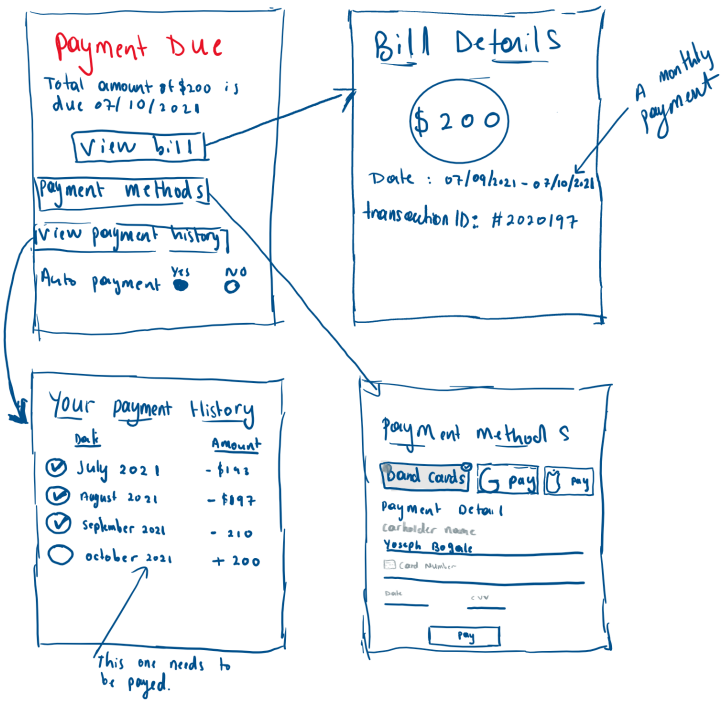
important for users to know they can make actions, explore the environment, learn how to use new tools, without consequences (especially important ones like making payments).

- Doesn't trap users, forcing them through the designer's task. Provides flexibility to choose what bill to pay and from what bank account card based on their unique user task.

Cons:

- Too much visual information is presented at once on the billing page that will be overwhelming for the human input. Hick/Hymans Law of Decision Time is evidence that a higher information entropy (H) leads to a longer choice reaction time (T) as T = a + b * H.
- Side panel eat's up some screen space.
- Navigating the interface to pay a bill is not immediately intuitive (You need to select a bill first, and then the pay now button will be activated to do that action).
- It may not be intuitive to learn shortcuts to make multiple payments at once (You need to select multiple bills, and then the pay now button will be activated).

2.3. Quick View Payment Design



Pros:

- Simple and natural design. Such a minimalistic design can increase the efficiency as design diminishes trival page transition.
- Speaks users' language.

Cons:

- Doesn't minimize the users' memory load (no input format shown in the date section.
- Trapping the users in the payment design (only one button, which is pay).

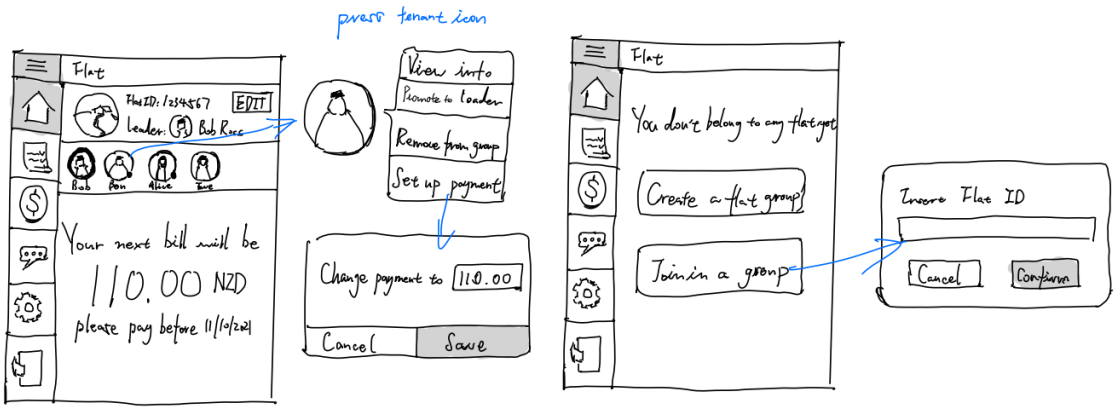
Conclusion for Payment Page Design

The final payment design would be a combination of the *design 2.1* and *design 2.2*. *Design 2.1* presents an accurate overview of a user's task categories when they arrive at the payment homepage. While the *design 2.2* efficiently uses a tidy list structure to present more information when users choose to (e.g. payment history, viewing all bills etc.). A combination of *designs 2.1 and 2.2* would take advantage of their presentation of visual information to show only the necessary information and choices needed/wanted through each step of their payment process.

3. Flat Management Page Sketches (Manage Flat group)

3.1. Dropdown List Flat Management Design

Simple and effective dropdown menu allows the group leader to manage tenants quickly. It's easy to have only two buttons for tenants to join or create a flat if they don't have one.



Pros:

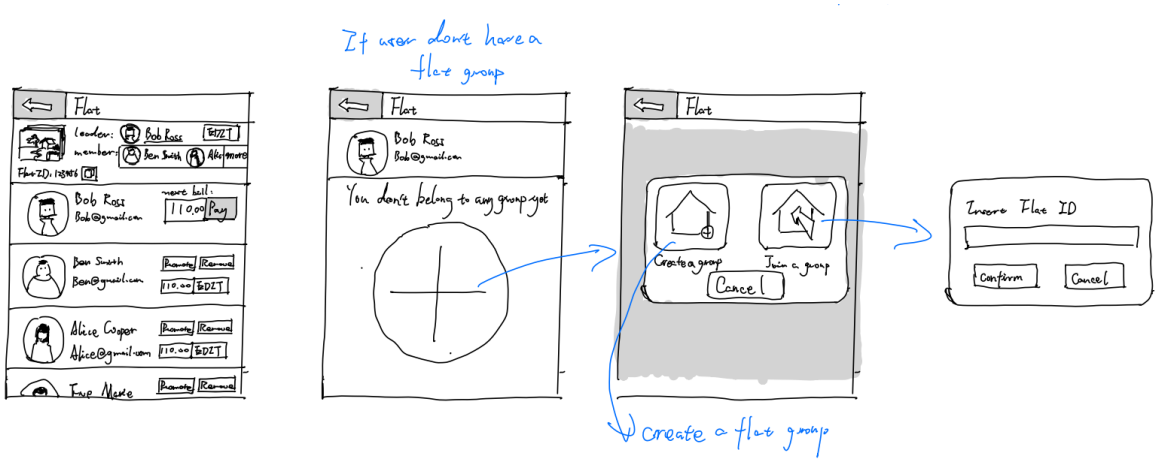
- Dropdown menu gives group leaders a shortcut to quickly manage tenants, which increases the efficiency of the app.
- Speaks users' language, which reduces the learning time for users as it fits the actual world image of users.

Cons:

- There are not many instructions or visual cues for helping users to recognize most of the features. Thus, a lack of such vital information might lead to misclicks.
- Logout button as an inessential feature should not be included in the main taskbar.

3.2. Display All Flat Management Design

Every management option is shown in the tenant's bar; it's crystal-clear for the group leader to manage tenants. Similar to *design 3.1*, users will easily create or join a group, but what makes this even better is the button has added illustrations.



Pros:

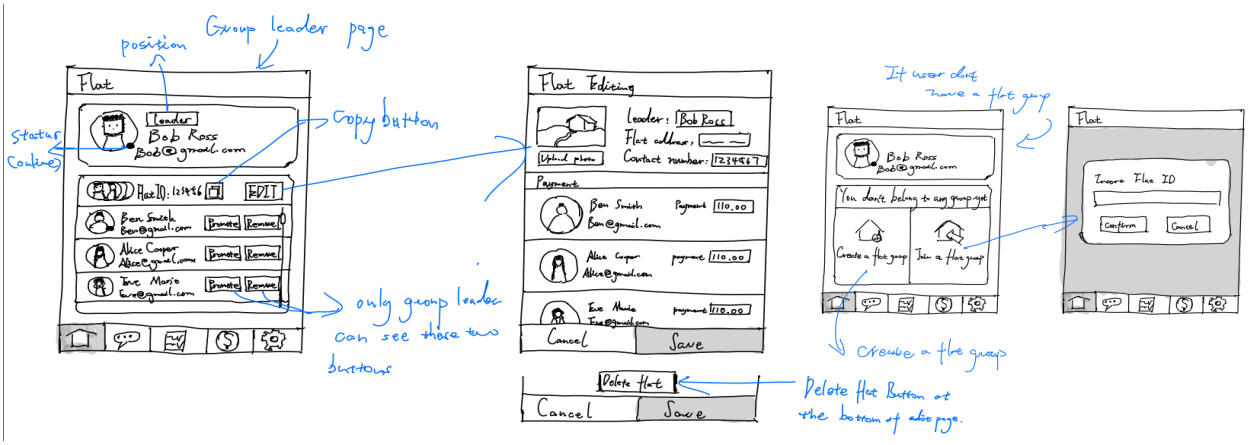
- It's more efficient if there are not too many tenants in the flat.
- All the essential information of tenants are shown in one block; users do not need to recall all processes of tasks as it is being displayed there.

Cons:

- So much information on one screen causes the rise of memory load.
- There are no good error prevention procedures for misclicks.
- There is no scrollbar shown on the side of the list. This may cause confusion as it is not consistent with other pages of the system.

3.3. Categorized Task Flat Management Design

This design is more organized and detailed as each feature is placed depending on their frequency. The large proportion of creating and joining a group buttons are more error-protective.



Pros:

- It contains many useful side features that can increase efficiency, for example, the copy button of flat ID and member status(group leader).

Cons:

- Payment management belongs to the flat editing page, which is contradictory to users' real world image. Thus, it does not speak the user's language.

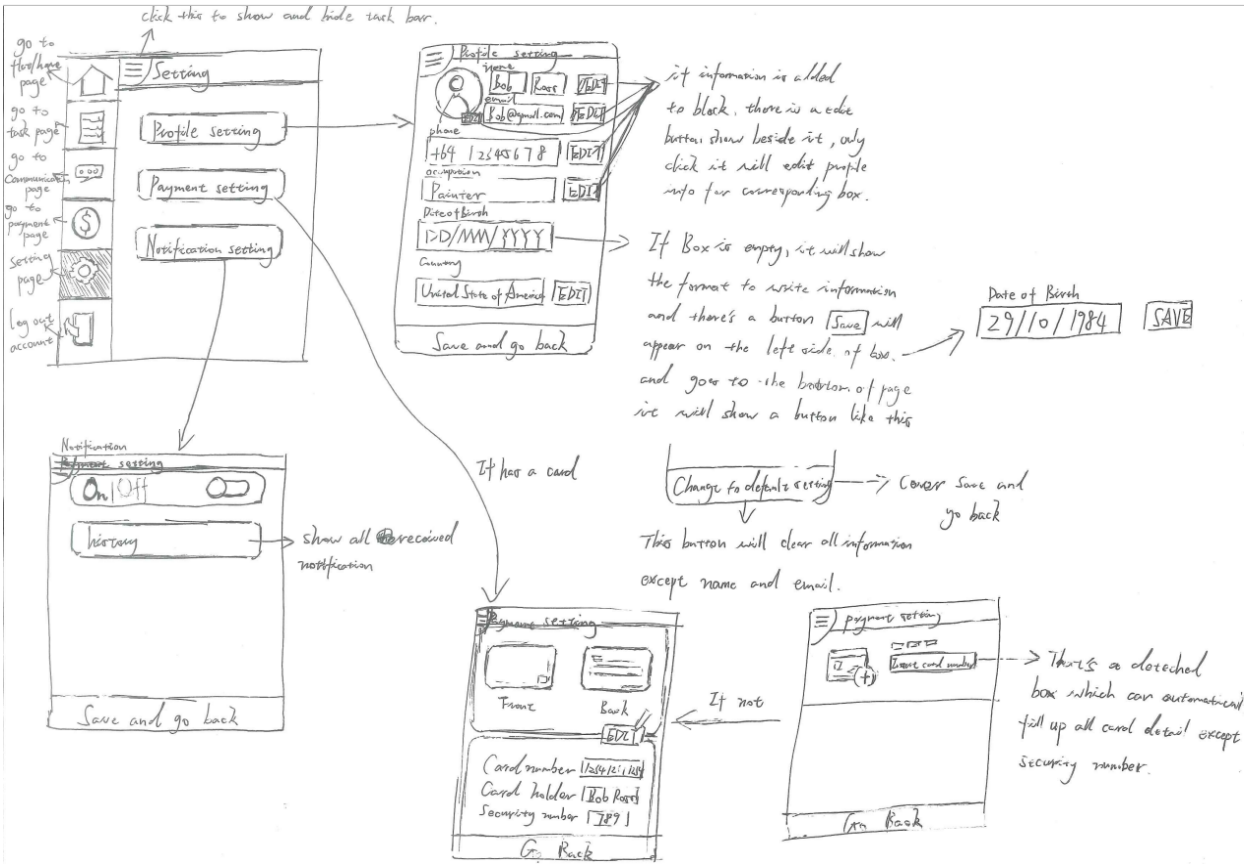
Conclusion for Flat Group Management Page Design

After thoroughly analysing through three designs, we decided to proceed with the second design(*design 3.2*) for the final UI because it covers all features the group leader needs in one page and it has a better annotation than *design 3.1* to address features, and a more brief and convenient design compared to *design 3.3*.

4. Setting Page Sketches (Manage account)

4.1. Fast and Straightforward Setting Design

This design is mainly created for users to navigate directly through multiple pages and modify and edit individually. An expandable taskbar consistently on the top left corner allows users to fast travel from any page back to the main task page, where signout as a button in the taskbar can be selected directly.



Pros:

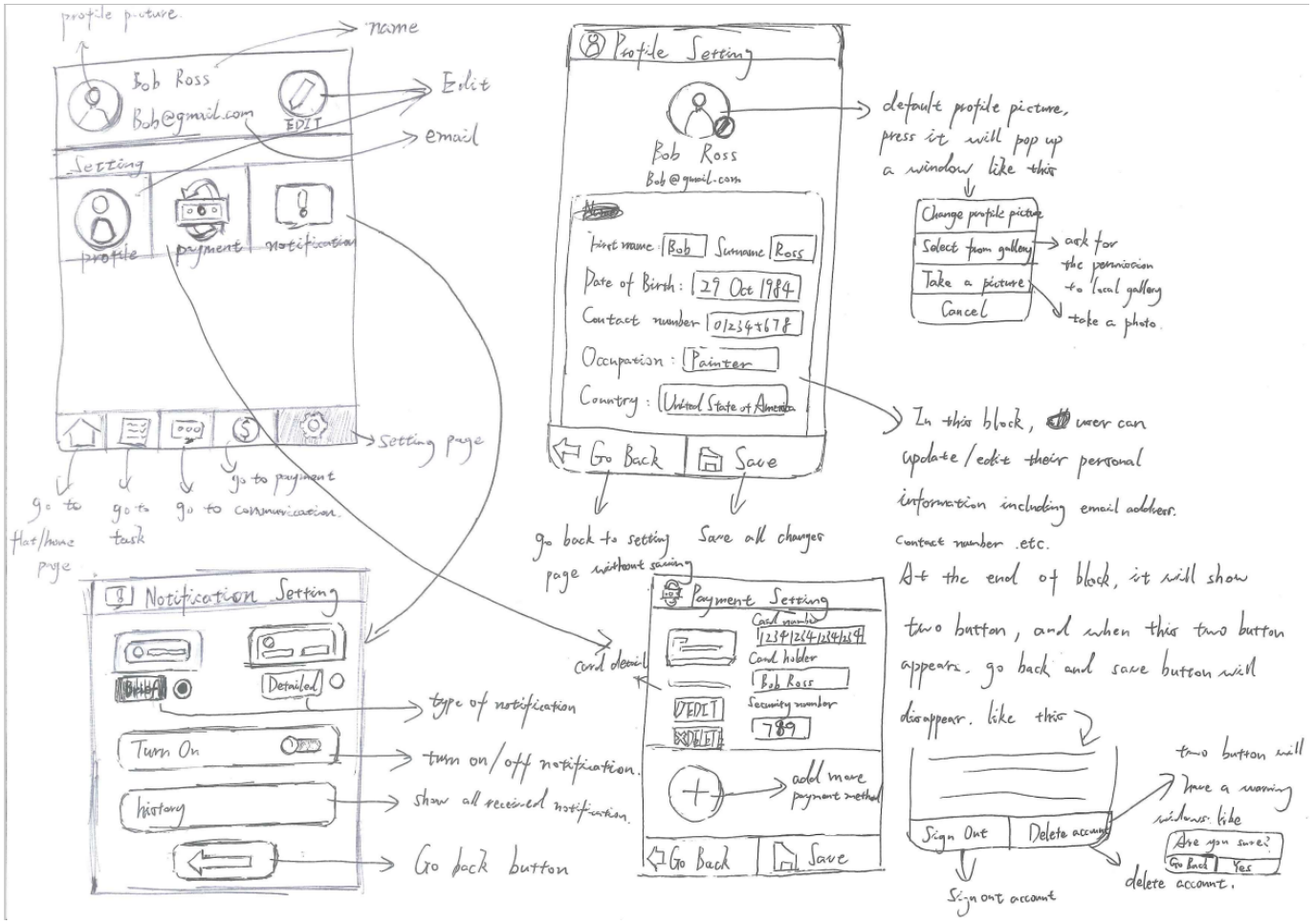
- Each page has a clear name bar on the top that speaks the natural user's language, and that's also good feedback to remind users which page they are currently in.
- Each setting page has a go back button that allows users to return to the previous page quickly. (There is a clearly marked exit)
- Expandable taskbar offers a shortcut for users to quickly enter the main task page, which increases the efficiency of the app.

Cons:

- Log out as a rarely used function is shown in the main task bar, an improvement can be replacing log out with a brief user account info box.
- Due to using only one button to save and go back, it may cause confusion among users between the edit button and save and go back button. Besides, there's no good error prevention in this case.

4.2. Adaptable and Organized Setting Design

This design mainly focuses on providing the most concerned information adequately in each page and supporting users to use it on different platforms.



Pros:

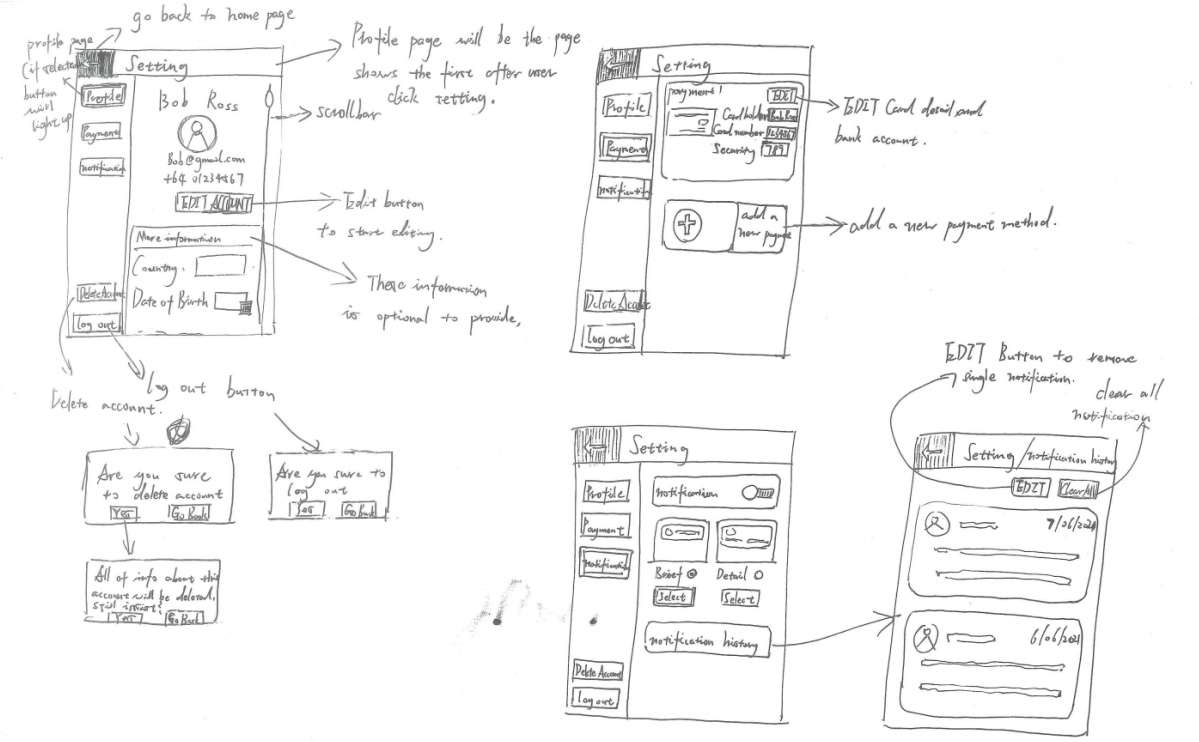
- The UIs speak easy and simple user's dialogue and have clearly named features minimising user's memory load.
- This structured design bestows clear visibility for users to perceive and learn.

Cons:

- It is a slightly more clustered design, which increase the memory load of users
- Each page contains a lot of information. Thus, it is not a minimalist design.

4.3. Compact and Multipurpose Setting Design

This design will have a sidebar constantly shown on the left side of the setting page. The purpose is to simplify the task procedure and allow users to do multiple tasks simultaneously. Each setting page can automatically save all modifications that allow the user to navigate to another page without affecting made changes. There are some parts in profile and payment settings requiring an edit button to make changes, else the go back button will save all.



Pro:

- All remove/ delete buttons have warning windows to reconfirm users' decisions.
- Users could use the main taskbar to easily navigate through different settings pages. It increases the efficiency of the app and lowers the memory load for users..

Con:

- The main task bar occupies too much space in a page and so it is not consistent with other pages.

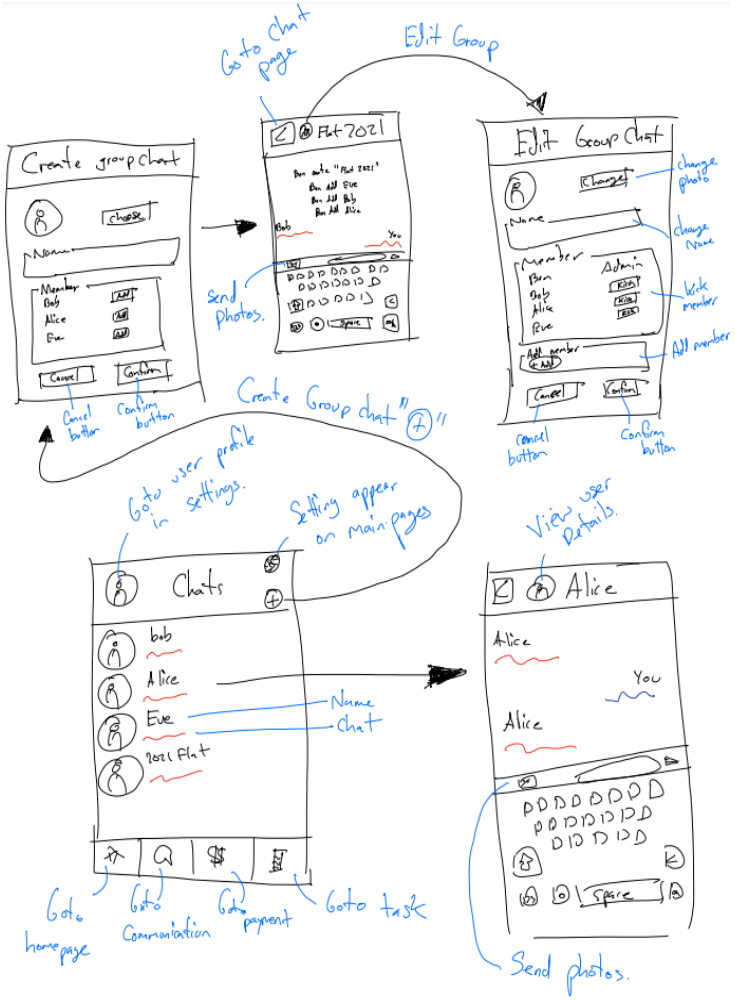
Conclusion for Setting Page Design

After reviewing all sketches, we decide to pick out each design's advantages to make a final setting page. In the final version, we decided to combine the expandable bar from design 4.1 and 4.3 to make it into the landing page, which also includes the main taskbar at the bottom from design 4.1. Then we will design specified task pages based on the format of design 4.2 and logics of 4.1.

5. Communication Page Sketch

5.1. Minimal/Easy to use Communication Design

Easy/simple to use chat design. Easy to navigate through the page to find what the user wants. The main chat page displays other users profile pictures, names and last message they sent/you sent.



Pros:

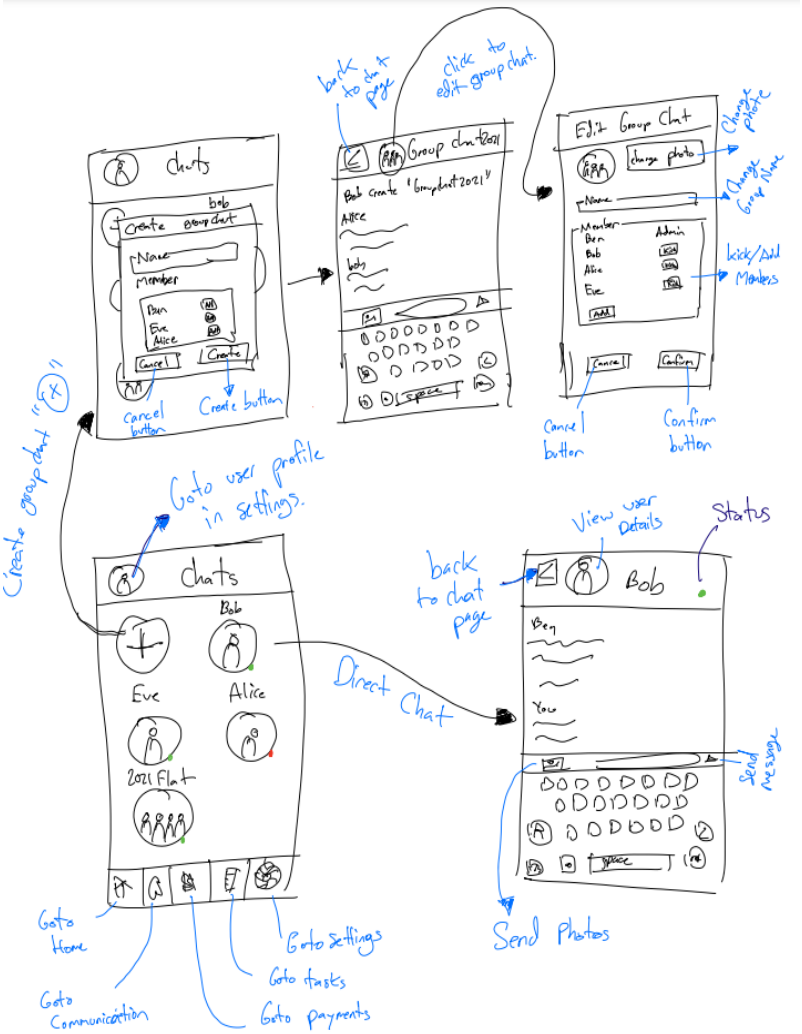
- Can see a lot of information at once on the main page.
- Let the user see the last message that was sent in the chat rather than opening the chat to see what was the message.

Cons:

- Hard for the user to see the “Make a group chat button” on the page. The shortcuts are not enough. It may cause inconvenience.
- The user can accidentally click on settings when wanting to create a group chat, because of the button placement, which causes an error.

5.2. All in One Modern Communication Design

Modern pop up design / Clear buttons to do tasks. Design where when a user clicks on Create group chat on the chat page, pop up create group window it shows instead of going to a new page. Easy to use Group chat setting to Add/Delete users, rename group name and change group photo.



Pros:

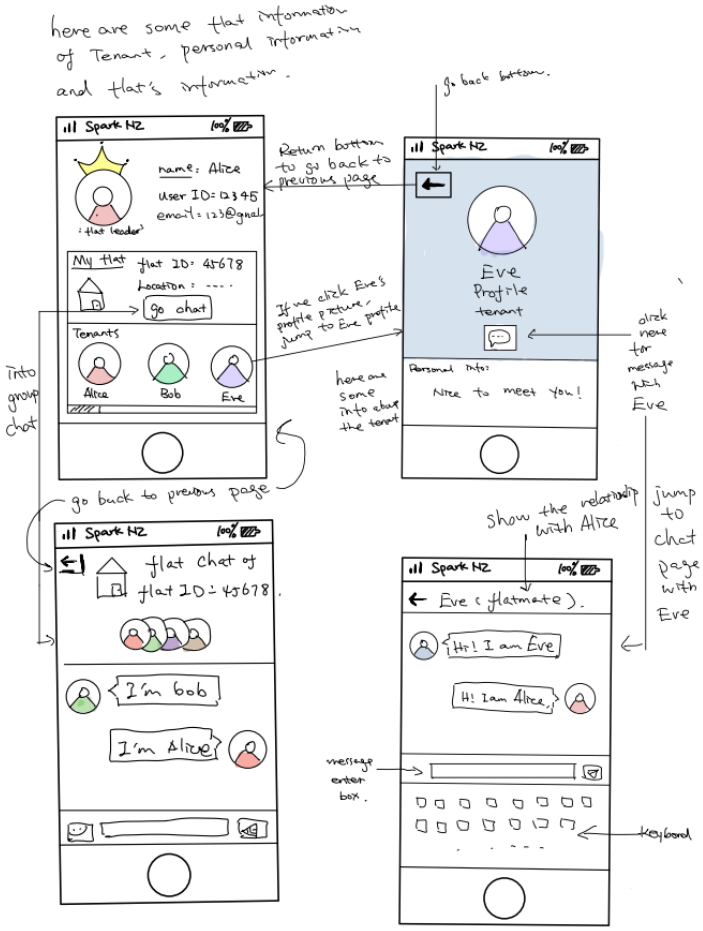
- Pop up create a group chat instead of opening a new page (less memory to generate)
- Clean/simple look on the main chat page, simple and easy dialogue for users, there is not a lot of information to look at.
- Good visibility for buttons on the page

Cons:

- Can't see previous messages from other users, like in design 1. There is no clear way to exit.
- Text in the chat/group chat window on one side of the page, can be confusing for users/not user friendly.

5.3. Chat Built into the Flat Information Communication Design

From the profile page, we can see a section below showing all the information about the flat you lived in and the roommates in it. There is a group chat button that you could click and go straight to the flatmates group chat. On the top section of group chat, you can see the flat ID and all the flatmates' profile photos there. Or you could click into one of your tenants profile pages, and in the middle, there is a chat button that you could go straight into the message sector.



Pros:

- Well-organized profile page with all the information about your flat and roommates in the profile page.
- Simply a page about communication with roommates or a group. There is no need for another action to create a group chat, the tenants in the same group chat will be added into the same group chat automatically. (the group chat of one flat is the default)

Cons:

- Too much information about the flat rather than personal information in the profile page. All of this unnecessary information may increase the user's memory load.
- The layout of the personal page is not consistent with the profile page of other people.

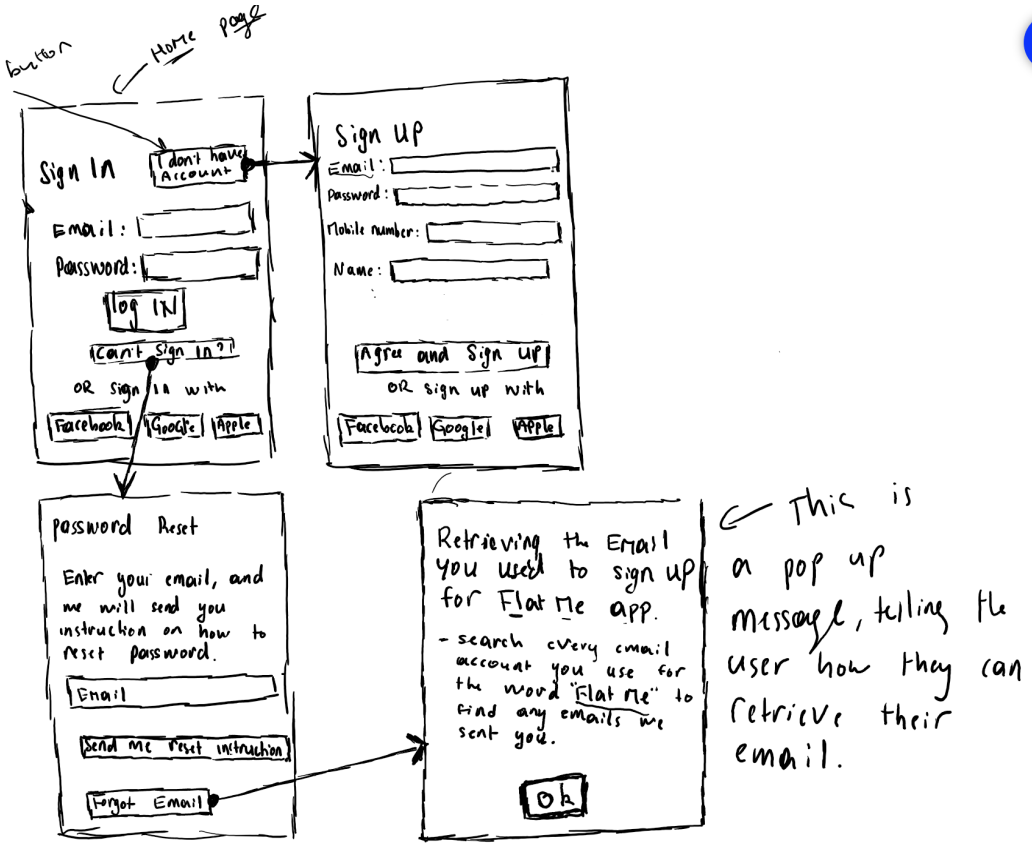
Conclusion for Communication/Chat Page Design

After going through the pros and cons of the three communication designs, we have decided to combine *design 5.1* and *design 5.3*. *Design 5.3* has some good features, the group chat and direct message page, this looks to be visually clean and nice compared to *design 5.1* and as for *design 5.1* the main chat page, create group page and edit group are much better than *design 5.3*, so combining these two designs will make our final design better.

6. Login Page Sketch (Security)

6.1. Simple Login Design

When tenants/users open up the Flat Mate app, the login/home page will appear and a tenant can go ahead and login with their email and password.
I have taken into consideration that some people might want login as fast as possible. Therefore I have given them an option of logging with their external account such as Facebook, Google or Apple.
When signing up, it is necessary to include all the information in the sign up page such as email, password, mobile number and name. Again, there will be an option if a tenant wants to sign up using their Facebook, Google or Apple account.



Pros:

- Simple design which speaks the user's language.
- Efficient as login, register and reset buttons are in one page, less memory load.

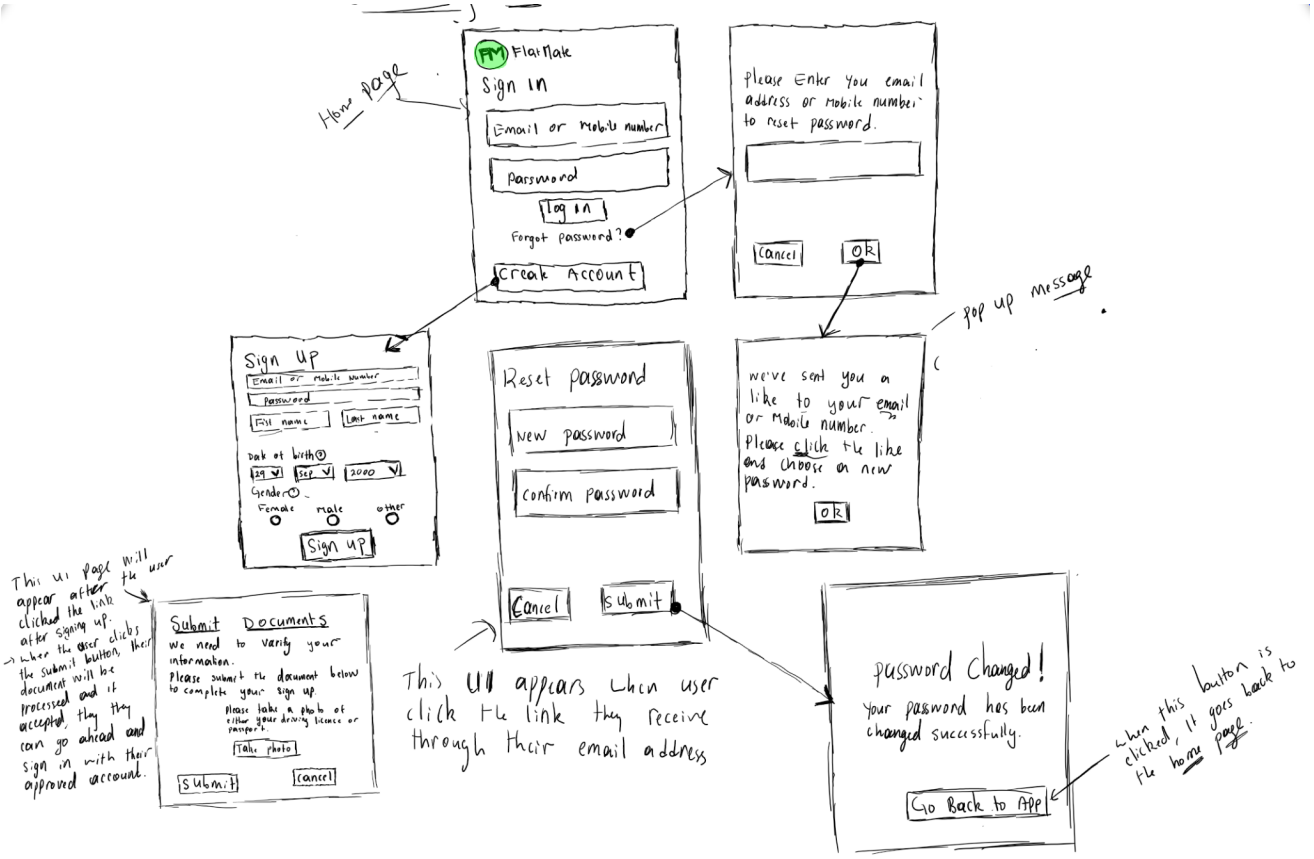
Cons:

- The signup page doesn't contain more information needed by the landlord, eg. the landlord would want to know the date of birth or gender of the user.
- The name input is unclear. It doesn't specify whether or not a user needs to enter their full name(first name and last name) or just the first name.

6.2. Well Structured Login Design

This design is almost similar to the first design of the login page except that it has security and the signup and reset password button is slightly changed to make it clearer and to get more information from the user.

Before a user/tenant can go ahead and sign in with their new account, they will have to verify their identity by uploading a photo of either their driving licence or passport.



Pros:

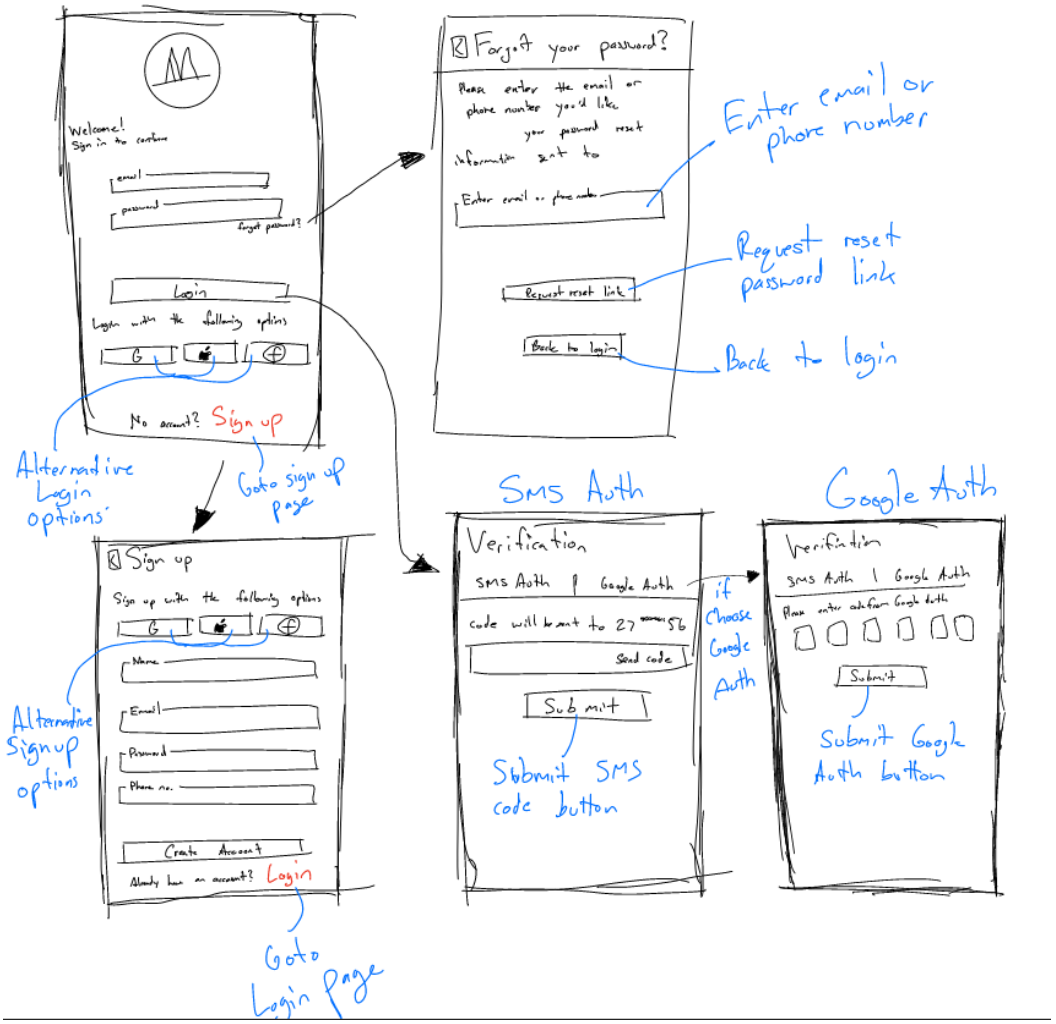
- Much clearer compared with *design 6.1*.
- Provides feedback to users before they process the next step.

Cons:

- More clustered compared to *design 6.1*.

6.3. Modern/Minimalistic Login Design

A modern/minimalistic login page, appealing to the user's eyes. Straight forward tasks, showing user where to go if they need to do certain task (eg. Sign up, Sign in with other options, forgot password, etc.). Design has a Verification, using SMS/Google Authenticator, every time the user login (For safety).



Pros:

- User interface looks nice and Modern user interface.
- Let users Signup/Login with other options, giving users more flexibility.
- Pleasing to the user's eye, allows the user to maneuver around the page easily .

Cons:

- Every time the user logs in. They will have to go to the verification page, which can be annoying and can be an unnecessary step which may trap the user.

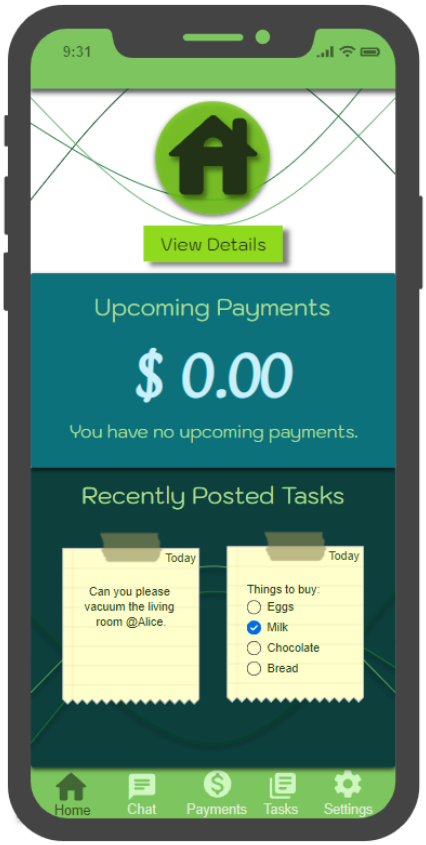
Conclusion for Login Page Design

After going through the pros and cons for the 3 designs, we have decided to merge *design 6.2* and *design 6.3* together. Both designs have some really good features that the other design doesn't have, so combining these two designs would be the best option.

Final User Interface Specification:

In this section, we will present the final design built from selected designs of each task section. We will describe how each UI design functions and make a conclusion in the end to summarize the final design.

Homepage UI Design



This is the homepage design of the application. We arrived at this design by merging two sketched designs. As the homepage is used each time a user opens the application, we decided to include the most relevant and important information by considering the priority scores for each task.

The user can view flat details by clicking on the **View Details** button, which would take the user to the relevant flat details page.

The middle section of the homepage will display any **upcoming payments** the user has to make. The user can simply click on this section or click on **Payments** on the taskbar at the bottom to open the payments page.

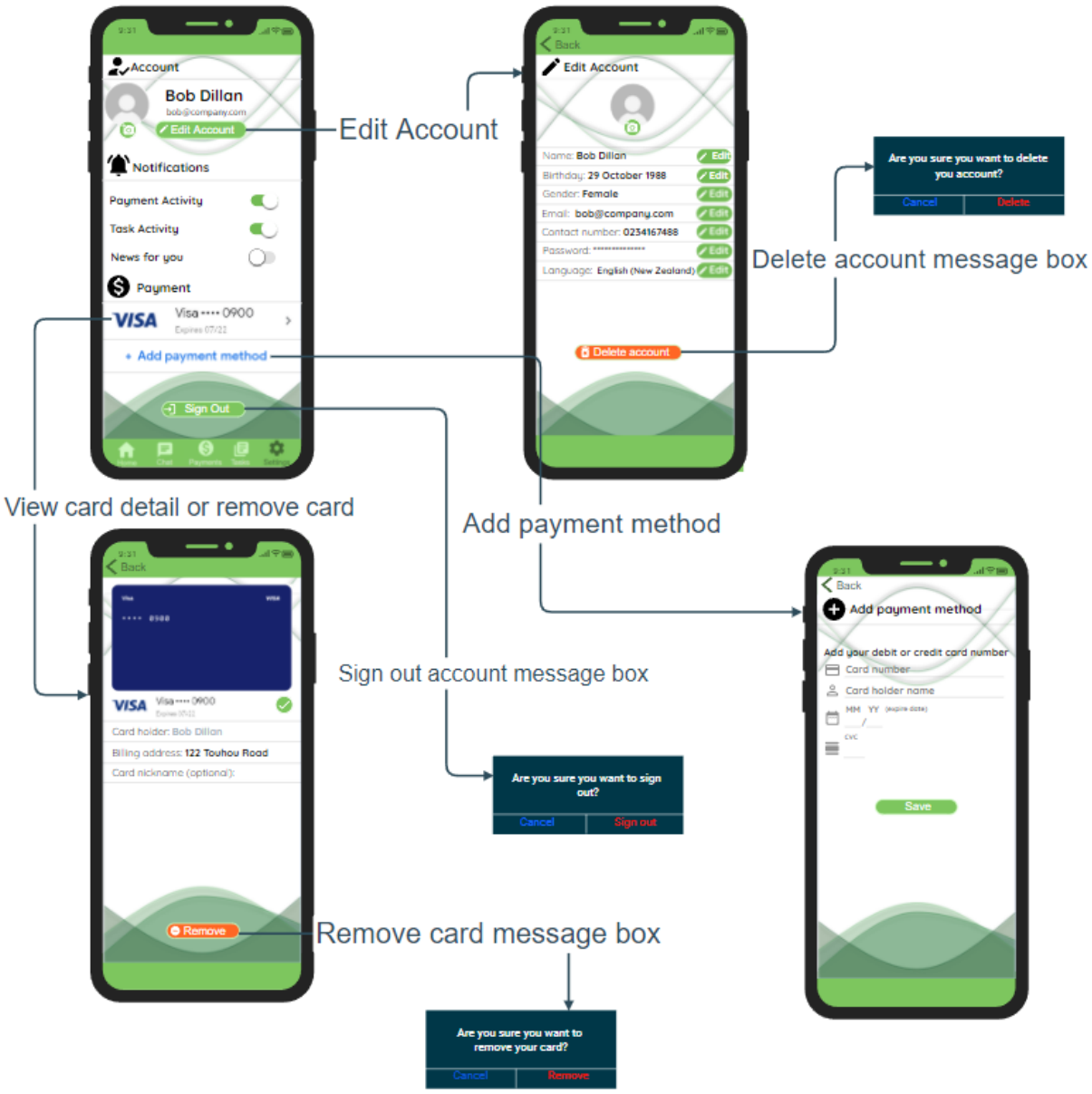
The bottom section of the homepage contains two of the most recently posted tasks. To expand/view the task, the user can simply click on them.

1. Create and Manage Flat Tasks UI Design

When the user clicks on the **Tasks** section from the taskbar at the bottom of the homepage, the **Task Noticeboard** page will appear. This page includes all the tasks that are currently available and has a **scroll bar** to scan through all tasks. A user can simply click on a task to view it, and the **Add** button on the bottom right can be used to create a task. As shown in the above diagram, the functionality of the design is straightforward as it uses commonly used terms and icons.

As mentioned in the conclusion for section 1 in the preliminary sketches and alternatives, we decided to go ahead with the noticeboard design together with a few other good features from *design 1.2* and *design 1.3*. Furthermore, we changed the **Add** button and it's location in order to make our system consistent with other real world UI designs.

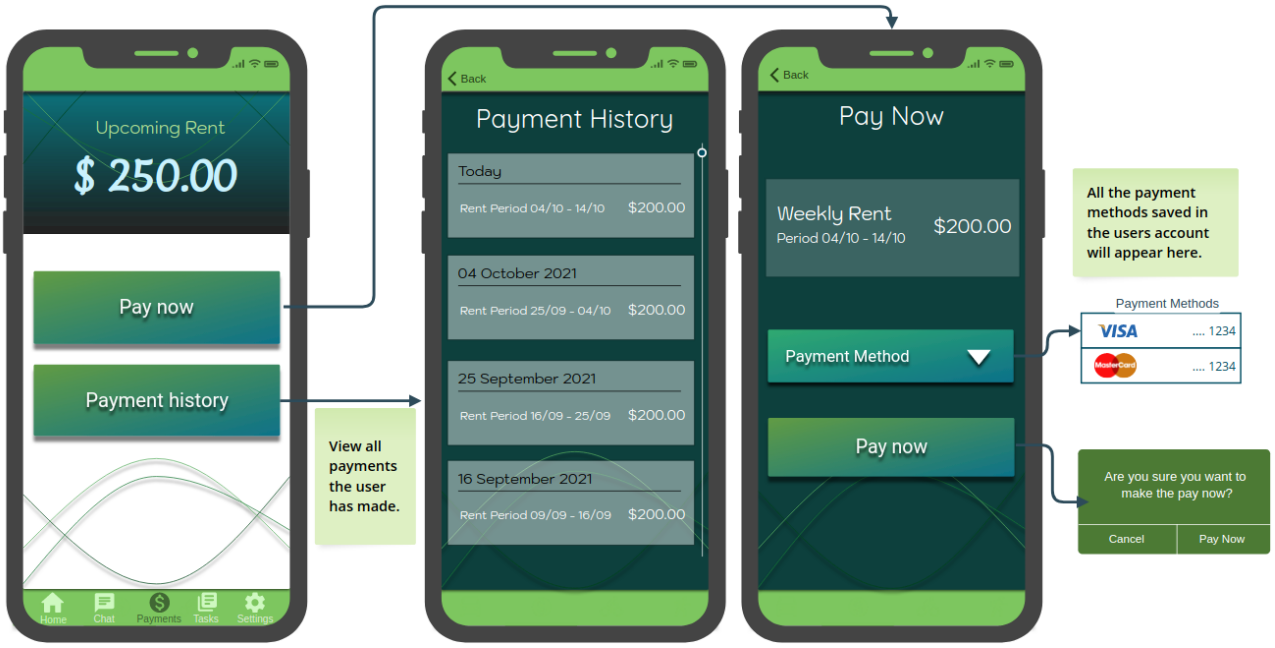
2. Manage User Account UI Design



hAs what we have promised, the main page inherits the feature of both expandable bars from *design 4.1* and *4.3*, and each task page is based on *4.1* with format of *4.2.*

This page is presented after the users click the setting icon from the main taskbar at the bottom. Among all these tasks, **notification** settings can be directly modified on the setting page. Users **can edit their profile and account information** independently on the Editing account page because of each edit button. **Viewing or Removing payment methods** can be done quickly by clicking the card, and users only need to fill up three required information to **add a payment method**. **Sign out** button is directly shown at the bottom of the setting page, whereas the **delete account** button is located on the editing account page due to the low frequency. All task pages have a **back** button at the top left corner, used to return to the setting page. **Delete**, **remove** and **sign out** buttons have a message box to prevent users from mistakes.

3. Payments Page UI Design

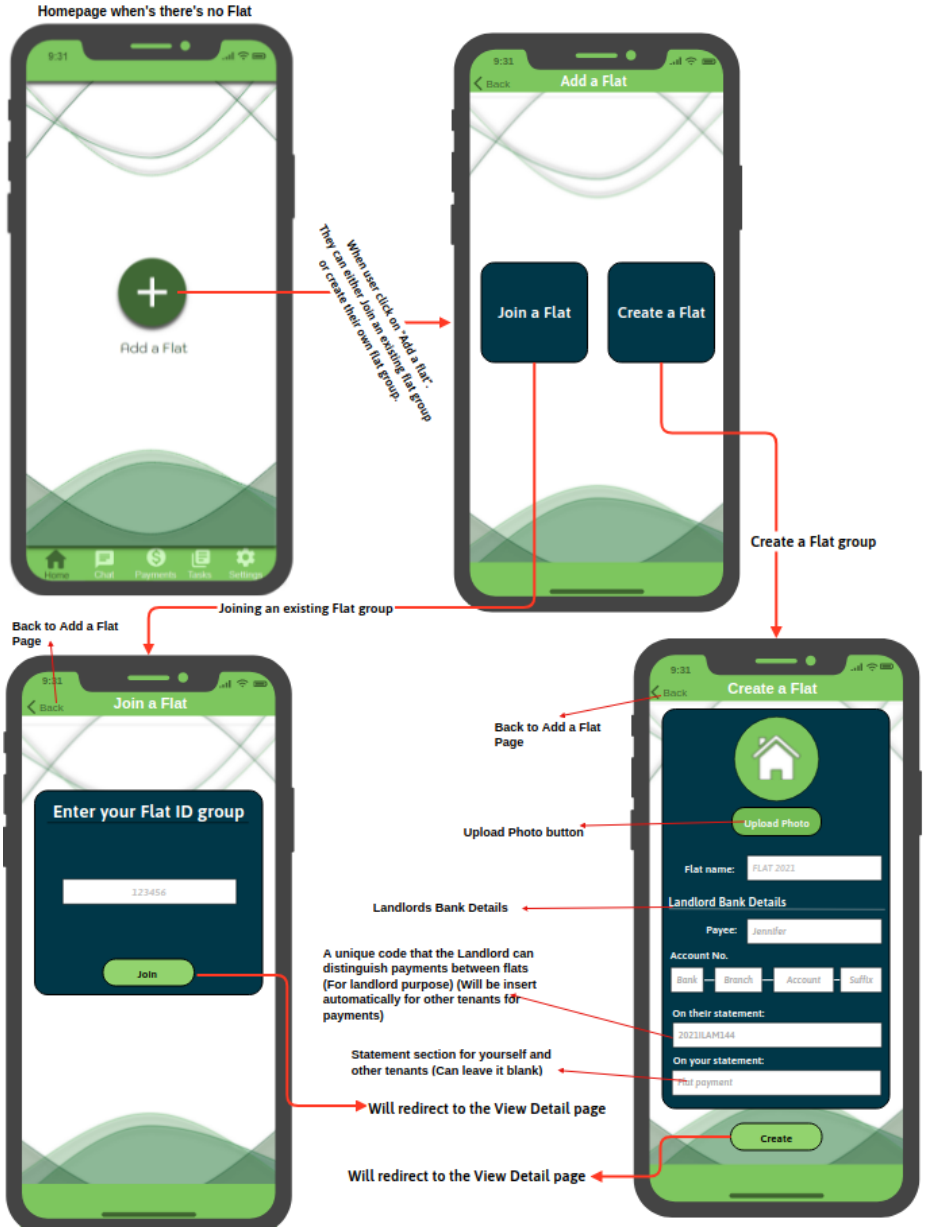


In the top section of the main **payments** page; the user can clearly see the **upcoming rent**. Below that part, two buttons allow users to **pay** the rent and view the **payment history**. The user can **view all previous payments** made and the details on the payment history page, including pay date, renting fee information, and payment number. It also has a scroll bar allowing the user to scan all the previous payments efficiently.
On the Pay Now page, users can **choose saved payment methods** to **pay the bill.** There is a clearly marked exit on each page. After the user presses the pay now button, it will show a message box to confirm the user's payment decision in case the user misclicked the button.

As mentioned in the conclusion of section 2 in the preliminary sketches and alternatives, the final design is a combination of *designs 2.1* and *2.2*. However, in order to maintain consistency throughout our system we decided to remove unnecessary features such as being able to add a new card from the Pay Now page as that functionality is already available in the Manage user account page. By doing so, we were able to design an aesthetic and minimalist UI for the Payments page.

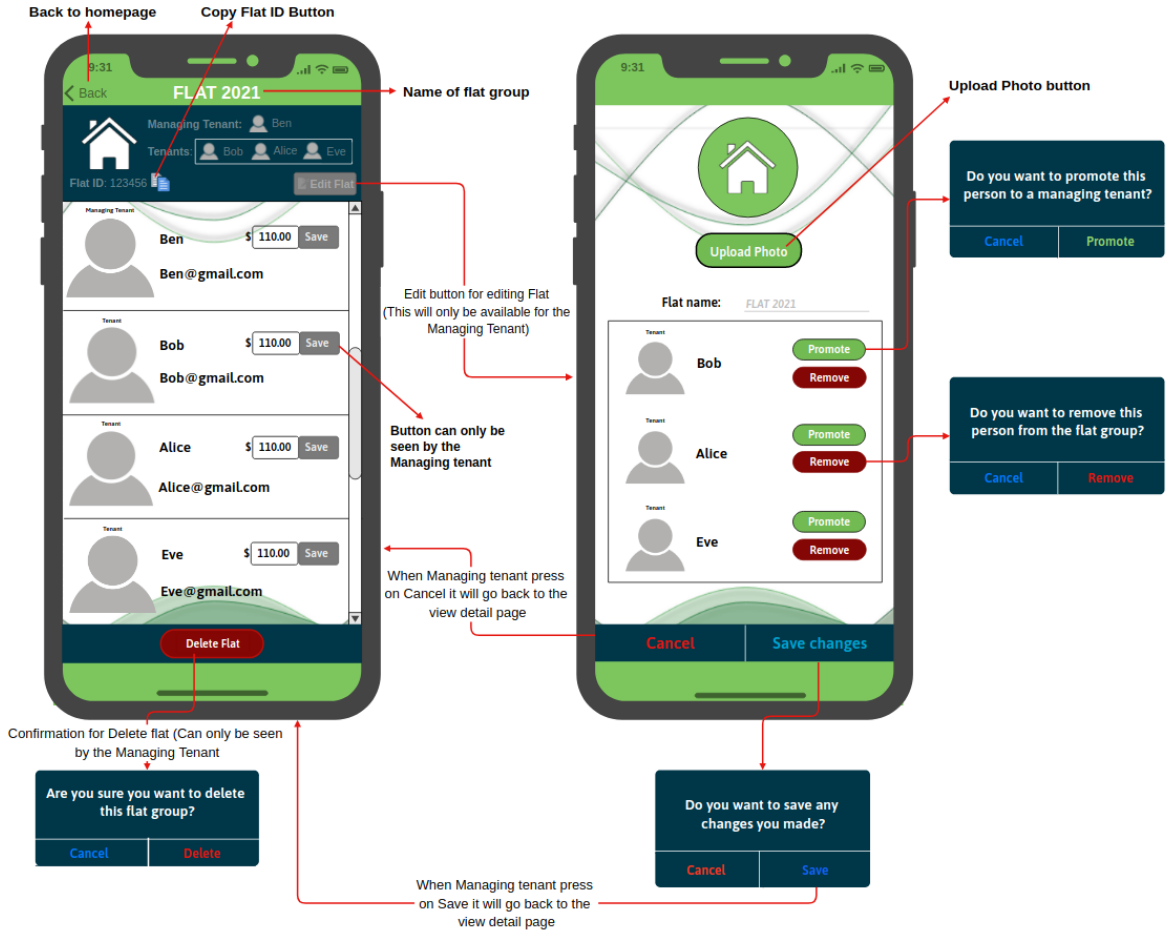
4. Manage Flat Group UI Design

4.1. Create/Join a Flat group UI Design



When the user login or Signup, they will be redirected to an empty homepage where an Add a Flat Button will be shown. The user will then have to click on the button and can decide whether to join or create a flat. If the user clicks on **Join a Flat** they will be redirected to a Join a flat page where they will have to enter their Flat ID from their group. If the user clicks on **Create a Flat** they can create their own Flat Group, where they can **upload a photo**, **insert a flat name,** and they will have to enter the **Landlord Bank details for payment,** seen in the design. After they Create or Join a Flat they will be redirected to the Flat view details page, seen on the **Flat view details UI design.**

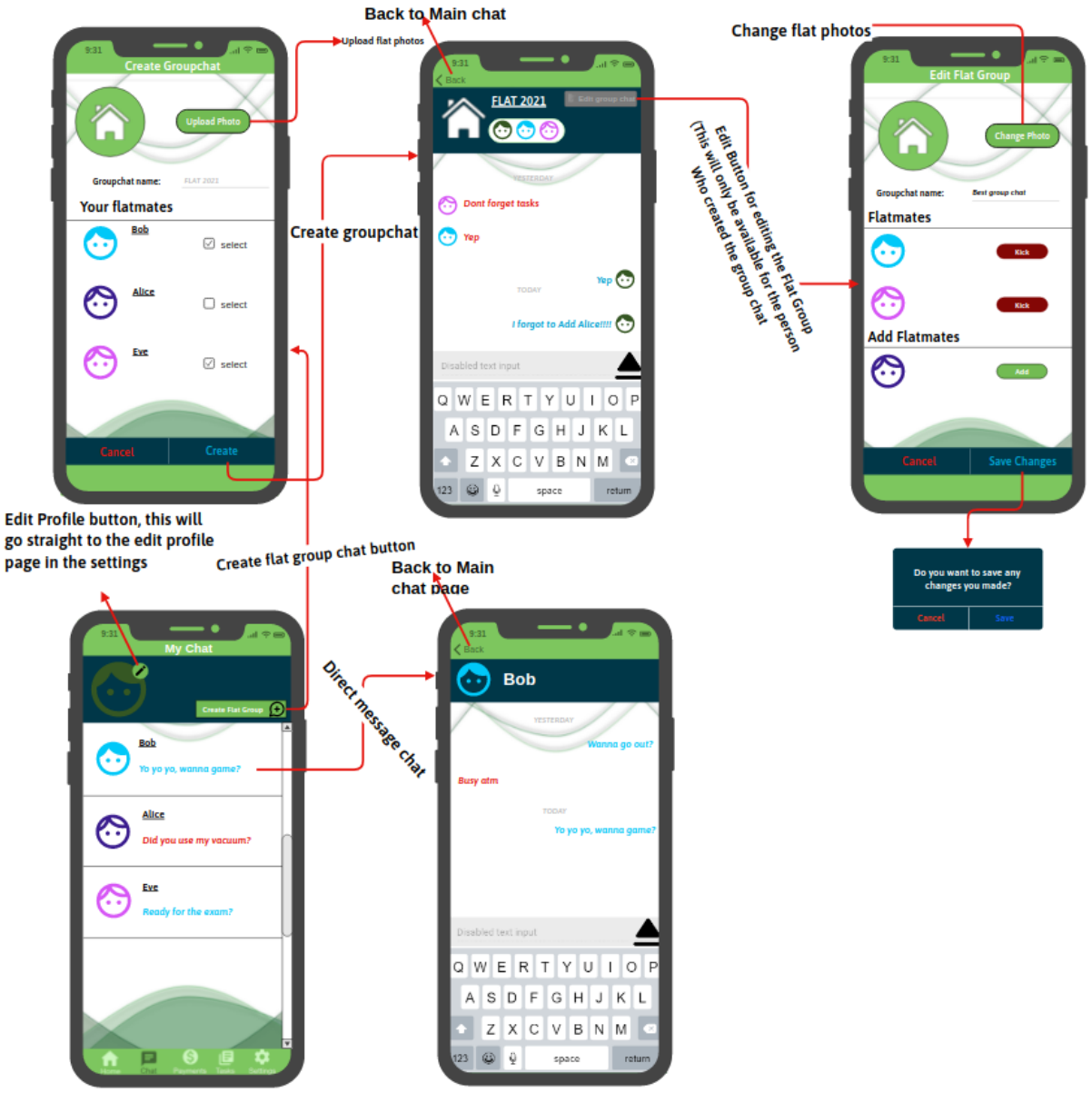
4.2. Flat view details UI Design



When the user clicks on the **View Details** button on the Homepage, seen in the Home page final design, the user will be redirected to the design on the left where it will display the **Managing Tenant**, **Tenants**, **Flat ID** and the **payment** for each user. Depending on the user, if the user is a Managing Tenant, then he/she can Edit the Flat (Promote/Remove/Change flat photo) using the **Edit Flat** button on the page, **Change payment amount** and **Delete** the Flat group, If the user is a Tenant then the only thing they can see is the Managing Tenant, Tenants, Flat ID and the payment for each user, they won't be able to change the payment or edit the Flat. A Copy Flat ID Button next to the Flat ID for any user to send the Flat ID to their Flatmates to join the Flat Group.

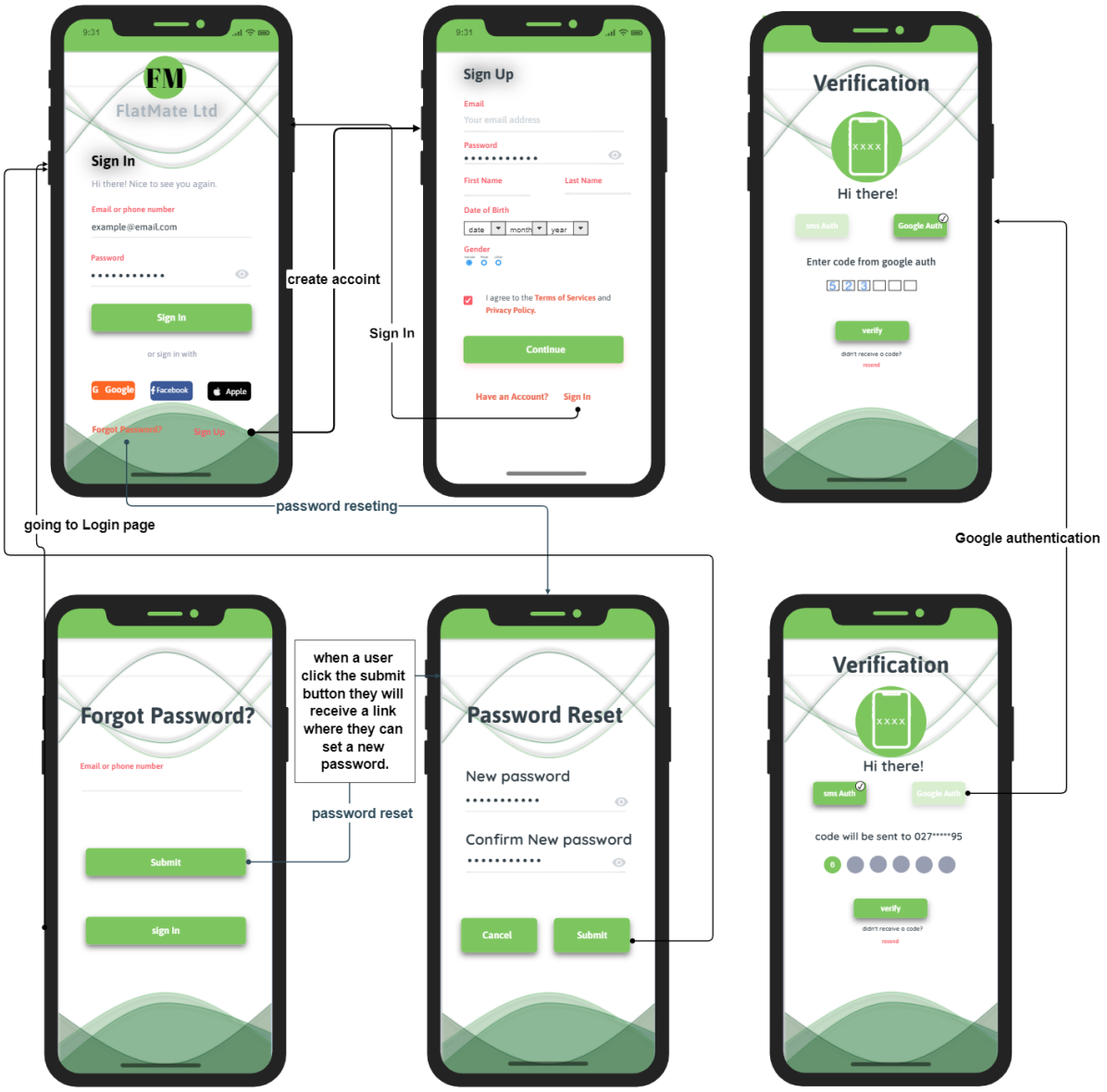
As concluded in the Conclusion for section 3 in the preliminary sketches and alternatives, we went with *design 3.2* with some slight modifications. As seen in the final design for the Create/Join Flat group and Flat view details we modify our final to make it more understanding and easier for the user to use. For the Create/Join Flat group, we decided to add a section in the Create flat group to add the landlords bank details and for Flat view details we decided to create a new page for the **Edit Flat** for the Manage tenant to make it easier to Promote/Remove tenants, rather than having everything in one page like in the sketches.

5. Communication UI Design



From the bottom bar, we can **select** the message icon and go to the main page of **My Chat**. On that page allows tenants to **create** a group chat or **start** a chat with any flatmates according to our preliminary *design 5.1*. There is a **scrollbar** to let users scroll up and down for more chats. Also, on the **group chat page**, tenants could **edit** the name, profile photo, and group chat members. All the past conversations will be shown on the **My Chat** main page regarding preliminary *design 5.3*.

6. Security UI Design



When a user opens the app for the first time or logged off, the **sign in** page will appear. Once the user enters their email/mobile number and password correctly, they will be directed to the **home page** to view their flat details, upcoming payment, tasks, etc.
If a user /tenant has no account, they can create one using the **sign up page**. They will also get to reset their password if they forget it somehow.
When a user/tenant completes signing in, they will be asked to authenticate their account using either **Google auth** or **SMS auth**.
As stated in the conclusion of **security page sketches**, I derived the security UI design by combining *design 6.2* and *6.3*.

Conclusion

We have designed an app for FlatMate Ltd, which includes features for managing flat tasks, payments, tenants, personal account details, and communication with other flat members.

The user will be initially greeted by the homepage, which provides an overview of the information easily identified to their current task in mind. These are the most relevant and important information based on our priority scores analysis of tasks and will provide a useful starting point in our system.

Overviews can be found throughout other areas of our system, such as flat task, flat group management and payment page. This provides an obvious initial translation of the user's task language observed in the system image. Details can be expanded on command by a click, speaking the users' language when interested in a section.

The functionality of the design is simple and natural to the users' model in mind as it includes commonly referred to graphics to represent different sections of the app in the taskbar, and actions for adding, editing, paying etc. This is also consistent with other successful real-world applications (such as visa card icons, sticky notes, and notification bells) from which the user has built their model. Our system image and its affordance appear close to the potential user models students/young professionals will use to look for what they need in our system application.

During its production, we restored early sketches and design rationales for the sake of FlatMate Ltd, hoping further to increase the app's efficiency and performance; emphasize its crucial factors; and identify potential loopholes for later enhancement.

Contributions

Kaishun Yang (kya31)

- Designed the **Manage account**, including the final UI design and sketches/alternative sketches.
- Designed sketches/alternative sketches for **Manage Flat group**
- Did the task identification of **Manage user account**.
- Helped review pros and cons for each sketches/alternative sketches
- Worked on **Conclusion** and **Introduction**

Tawatchai Holmes (tho78)

- Worked on **Communication between Tenants** of task identification and its Priority Analysis.
- Designed sketches/alternative sketches for **Communication between Tenants**
- Design alternative sketch for **Security** (Login/signup)
- Helped Design Final design UI for **Communication between Tenants** and Designed Final design UI for **Manage Flat group**
- Helped review pros and cons for each sketches/alternative sketches

Yoseph Bogale (ymb12)

- Worked on the final design of the security (login/signup) as well as the sketches.
- Helped review pros and cons for each sketches/alternative sketches
- Designed an alternative sketch for **payment.**
- Worked on the security of priority analysis of task section.
- Helped design the **Manage Account** task.

Chathuranga Alwis (vca35)

- Worked on **Create and Manage Flat Tasks** of task identification and it's priority analysis.
- Designed sketches/alternative sketches for **Create and Manage Flat Tasks** and **Homepage**.
- Designed the final design UIs for **Homepage**, **Create and Manage Flat Tasks**, **Payments page**.
- Helped review pros and cons for each sketches/alternative sketches

Ziling Huang (zhu51)

- Design the **Security,** including the priority analysis and task identification.
- Designed sketches for **communication between Tenants.**
- Designed sketches for **Payment.**
- Designed Final design UI for **Communication between Tenants**.
- Concluded for the final design of **Payment**.
- Helped review pros and cons for each sketches/alternative sketches

Yiyang Yu (yyu69)

- Worked on **User Identification** for **Tenants** and **Managing Tenants**
- Designed sketches for **Payment.**
- Designed sketches for **Create and Manage Flat Task.**
- Helped review pros and cons for each sketches/alternative sketches
- Worked on **Conclusion** and **Introduction**