How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

Description
Intended User
Features
User Interface Mocks
Screen 1
Screen 2
Key Considerations
How will your app handle data persistence?
Describe any corner cases in the UX.
Describe any libraries you'll be using and share your reasoning for including them.
Describe how you will implement Google Play Services.
Next Steps: Required Tasks
Task 1: Project Setup
Task 2: Implement UI for Each Activity and Fragment
Task 3: Your Next Task
Task 4: Your Next Task
Task 5: Your Next Task

GitHub Username: CaudellDev

Household

Description

Write a brief summary of what your app does. What problem does your app solve?

Organize and streamline your home life by collaborating tasks with your family members or roommates!

Need something to help organize your home life? Already do, but is it hard to relay that to your family or roommates? This app can help you organize your personal life, but also connects with the people you live with. By having a synced list of tasks, you and your comrades can keep your home going smoother than ever.

Add a task when you run out of milk. Add a task to mow the lawn. Complete the task to change the cat litter (and preferably the chore, too). In addition to basic tasks, it can help organize events like a sleepover or a Friends-giving.

Intended User

Who is your intended user? (For example, is this an app for dog owners? Families? Students? Travelers?)

Families and roommates for the most part, but it can also benefit people living by themselves.

Features

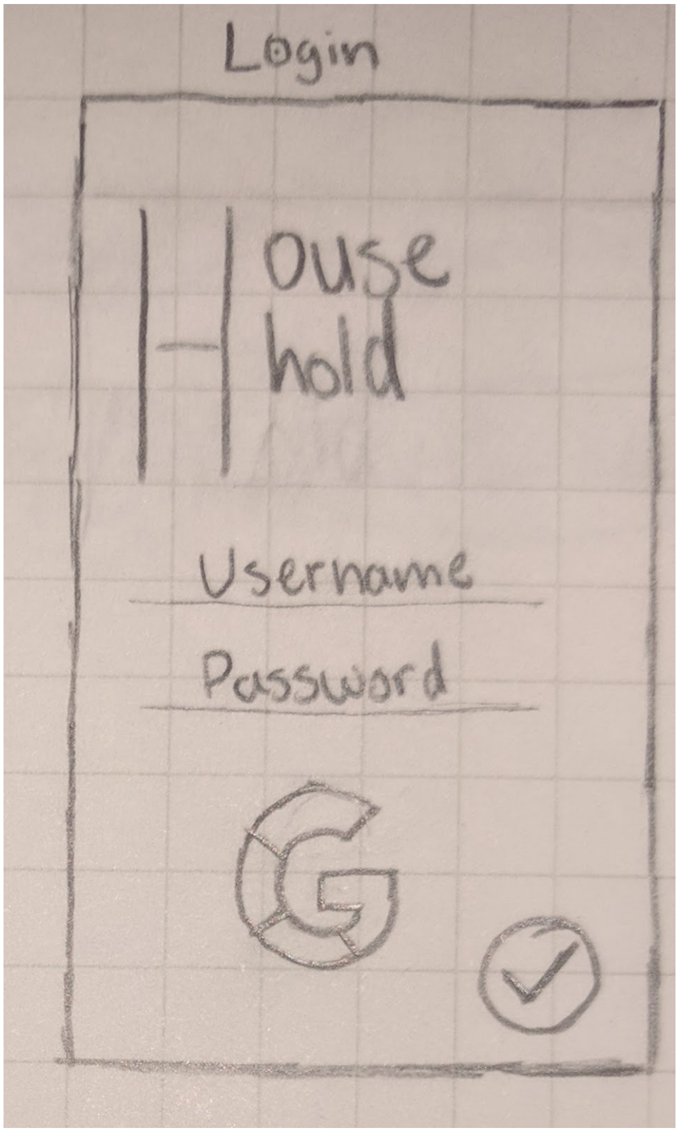
List the main features of my app:

- Create a task for house chores, items, or other things
- Have a synced list of tasks your family members can see and complete
- Organize and plan with roommates

User Interface Mocks

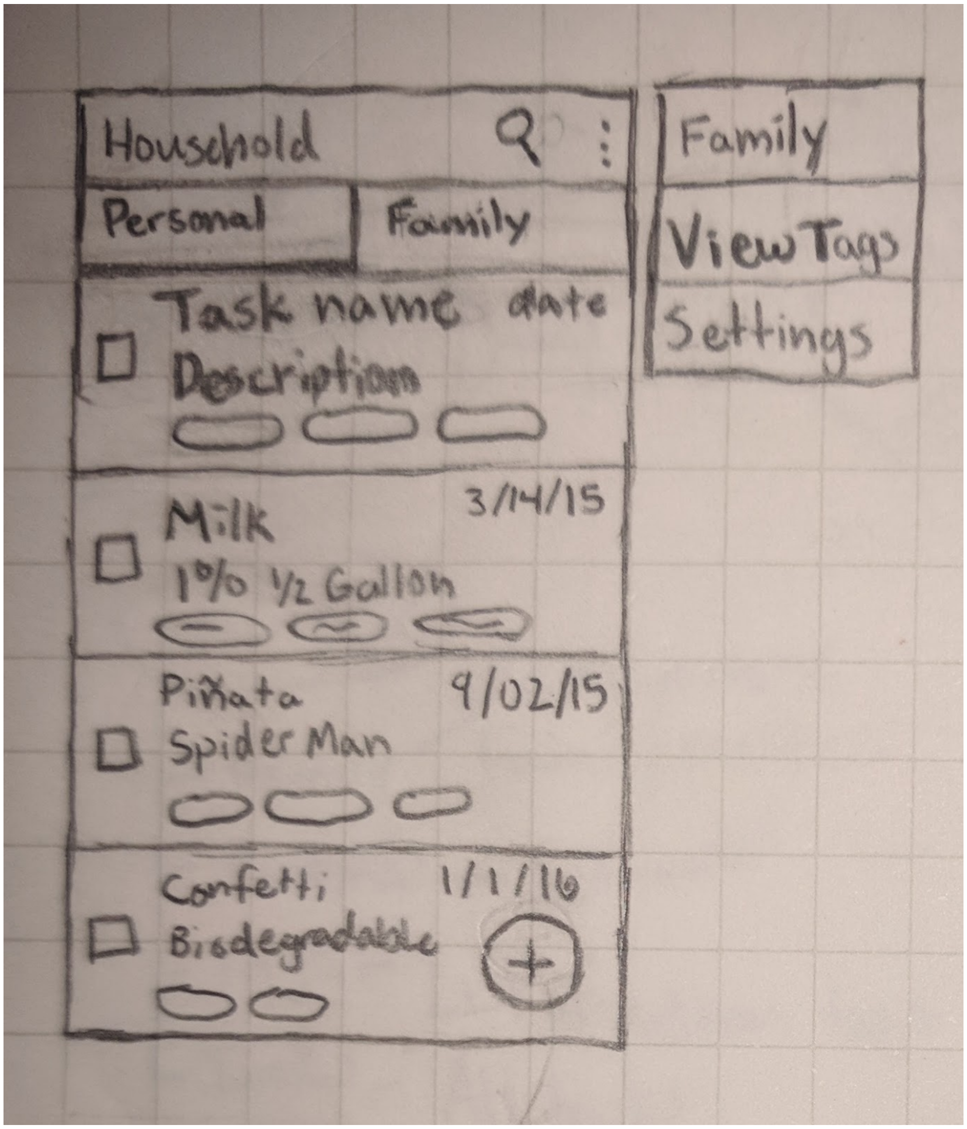
These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

Screen 1 - Mock Login Screen



Simple login screen that used Firebase Authentication. Using email and FAB to submit.

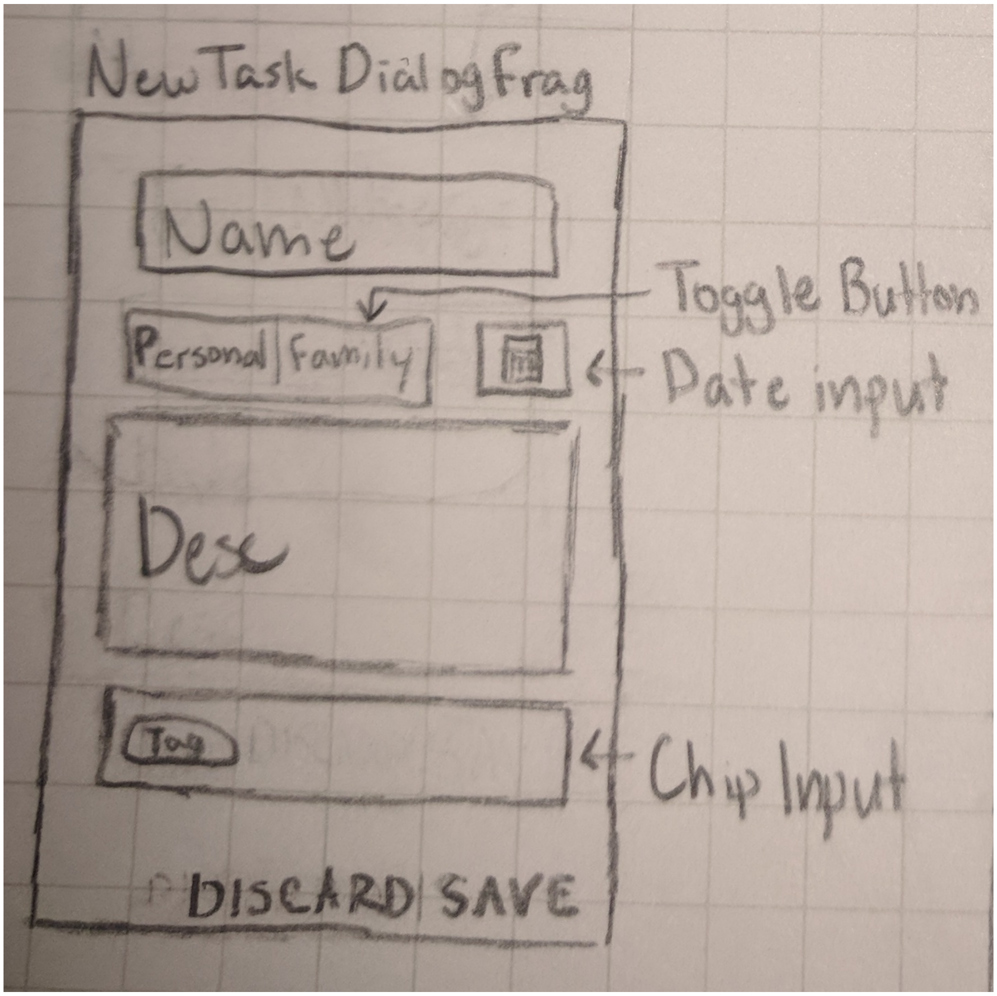
Screen 2 - Main Activity - Task Lists



Items will display in chronological order. Tag chips display below and can be scrolled left to reveal more, if there are any.

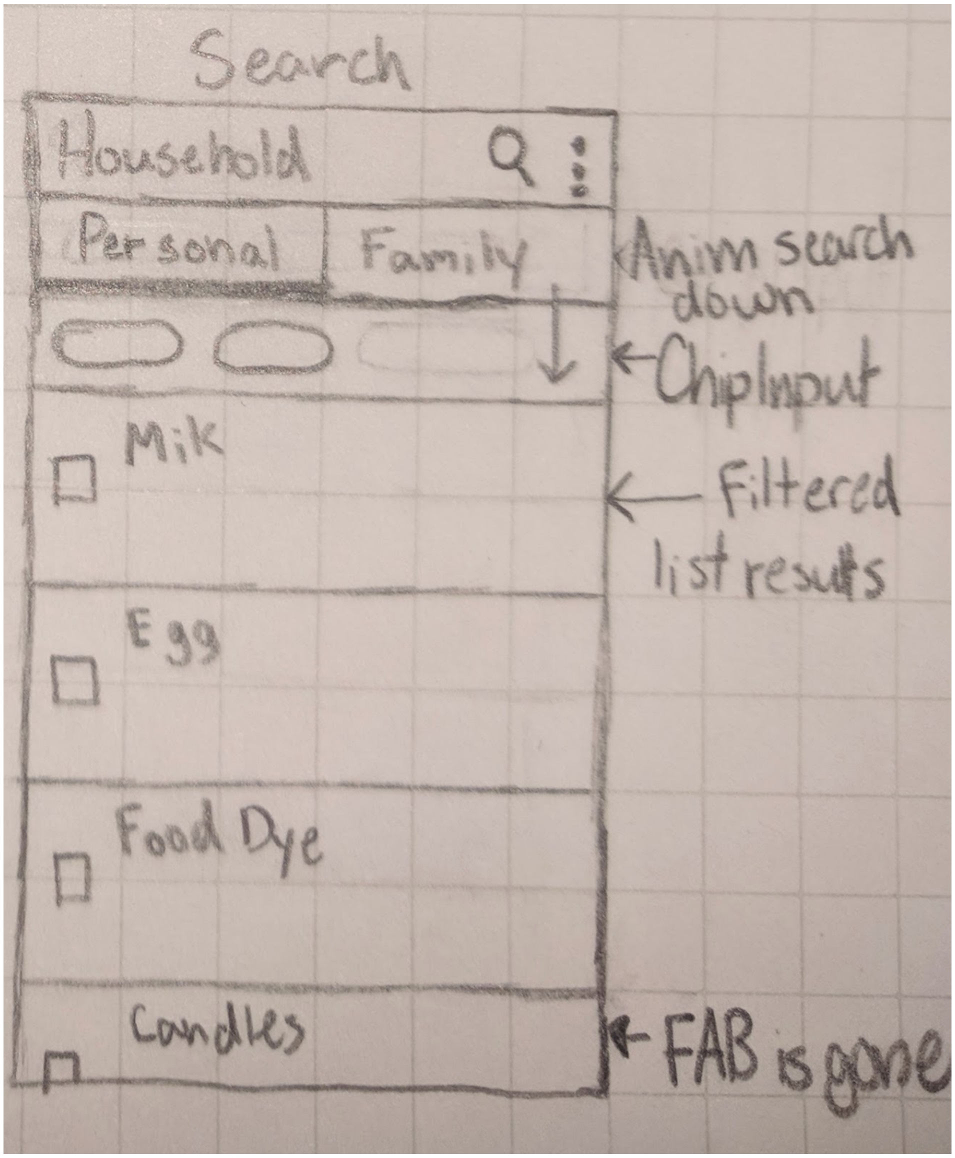
To keep things simple, the family tab items won't look different from the list item. The family tasks will have different behavior, but the data to display is the same.

Screen 3 - Main Activity - Dialog Fragment



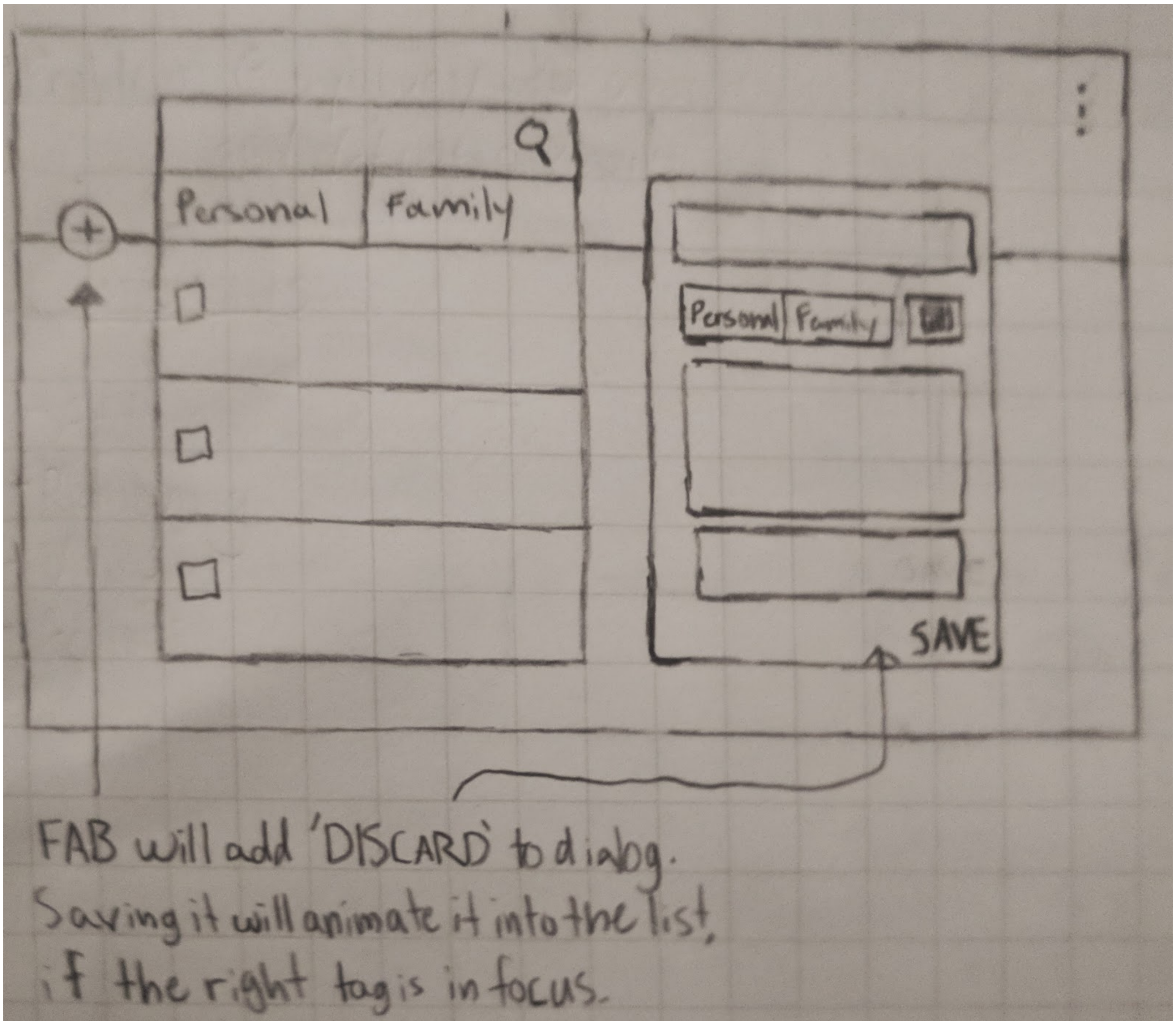
When the plus FAB is pushed, this will open. It also opens when a list item is clicked, and it will populate the fields to edit it.
If it fails (ie due to a network error), it will display a Snackbar and discard the changes.

Screen 4 - Search



Chip Input slides down to allow the user to filter the item results. Because it's a custom view, I decided to not embed it into the Toolbar.
User can still interact with the items if they click or check them.
FAB disappears so the user can't add a task while searching.
Switching between tabs will maintain search tags and will filter that list.

Screen 5 - Main Activity (Tablet)

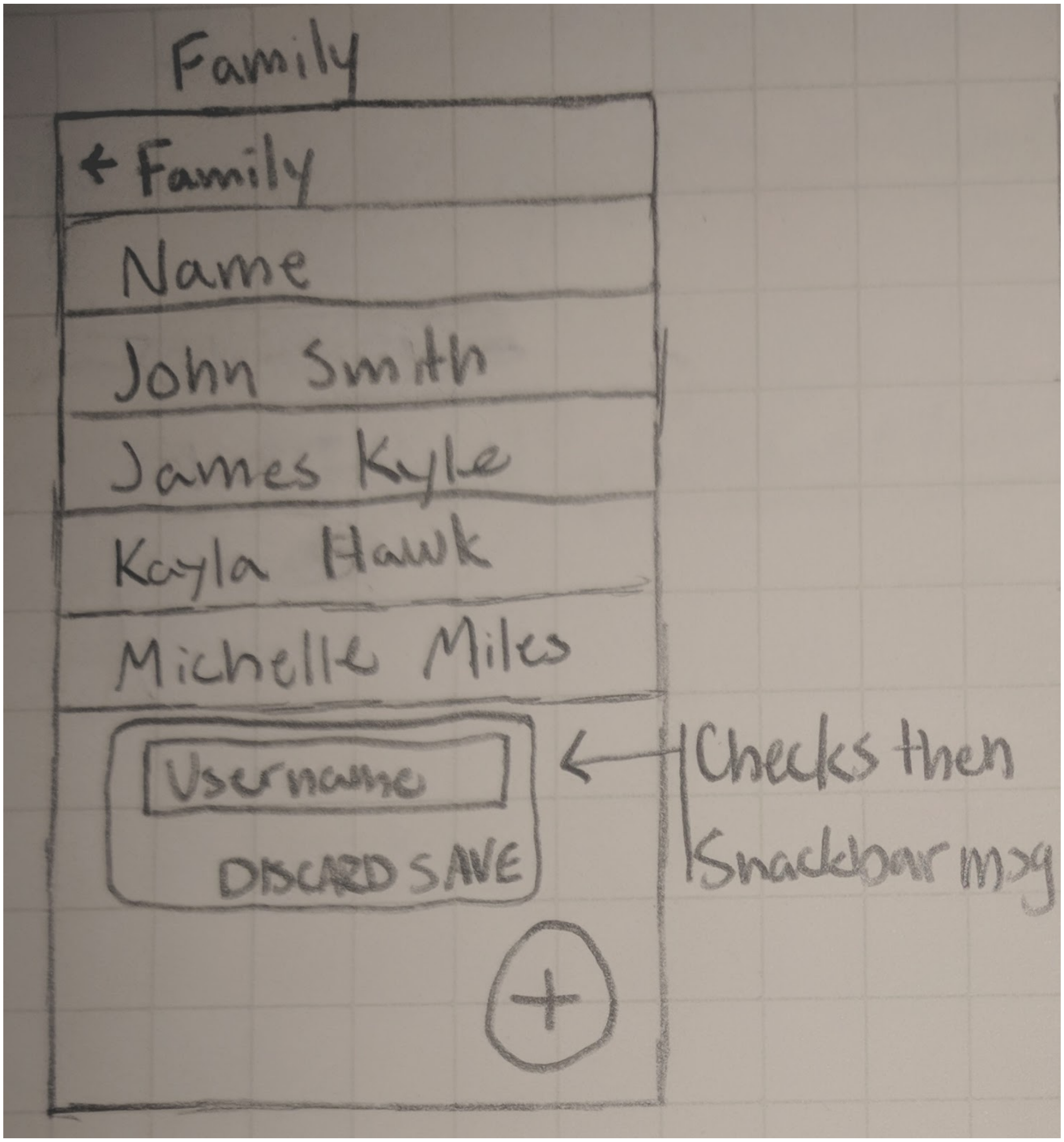

Personal
Family
Personal
Family
SAVE
FAB will add 'DISCARD' to dialog.
Saving it will animate it into the list,
if the right tag is in focus.

The tablet view will display the task list as a floating card, overlaying the Toolbar. If the width is big enough, it will display the Dialog Fragment on the right side as a permanent card. It will update the views as the user selects items on the list.

If the user wants to add to the list, the views will clear the data and add a discard button. Pushing discard will resume the user's last selected item.

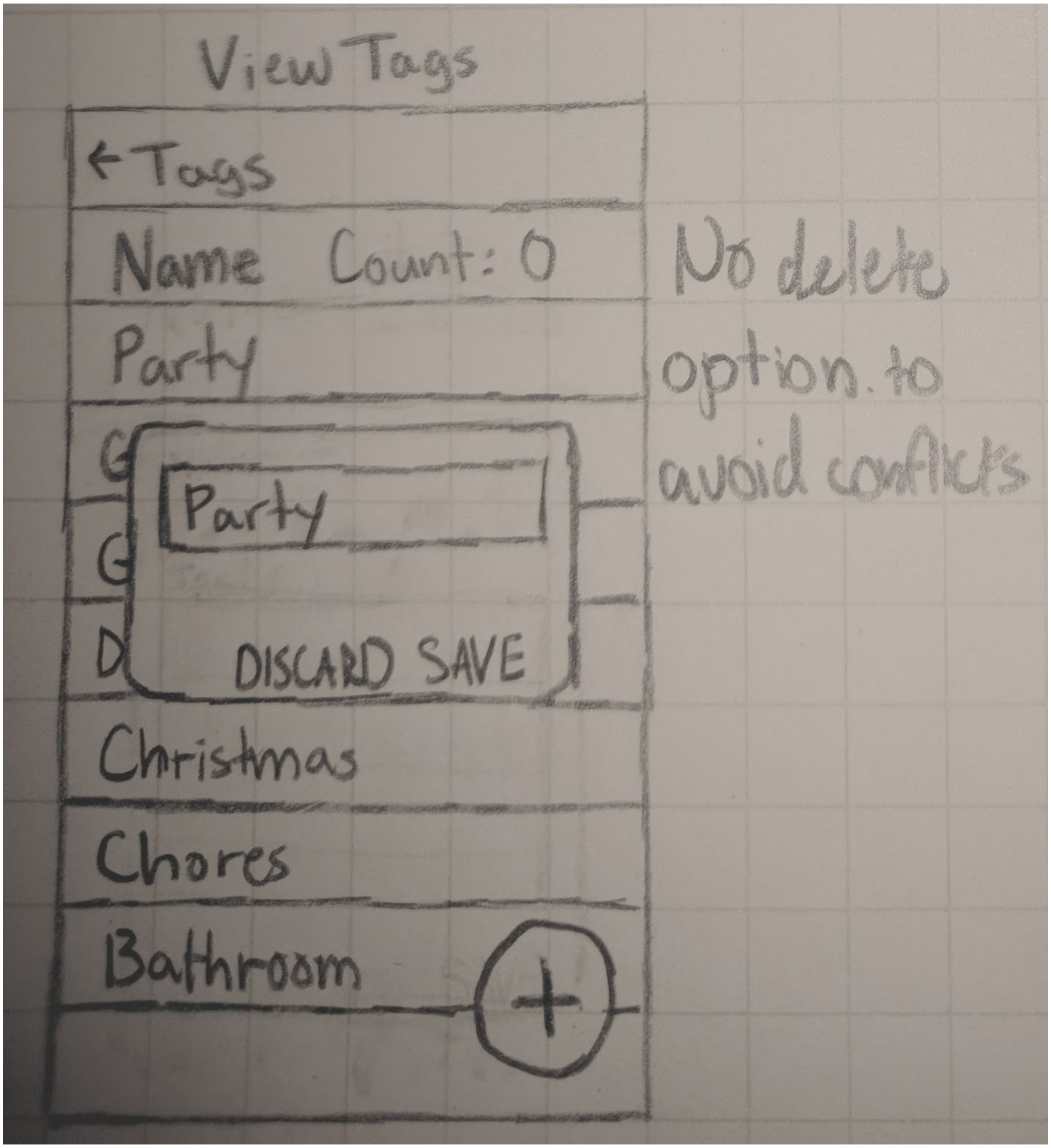
The dialog card will slide up to allow for keyboard entry as needed.

Screen 6 - Family



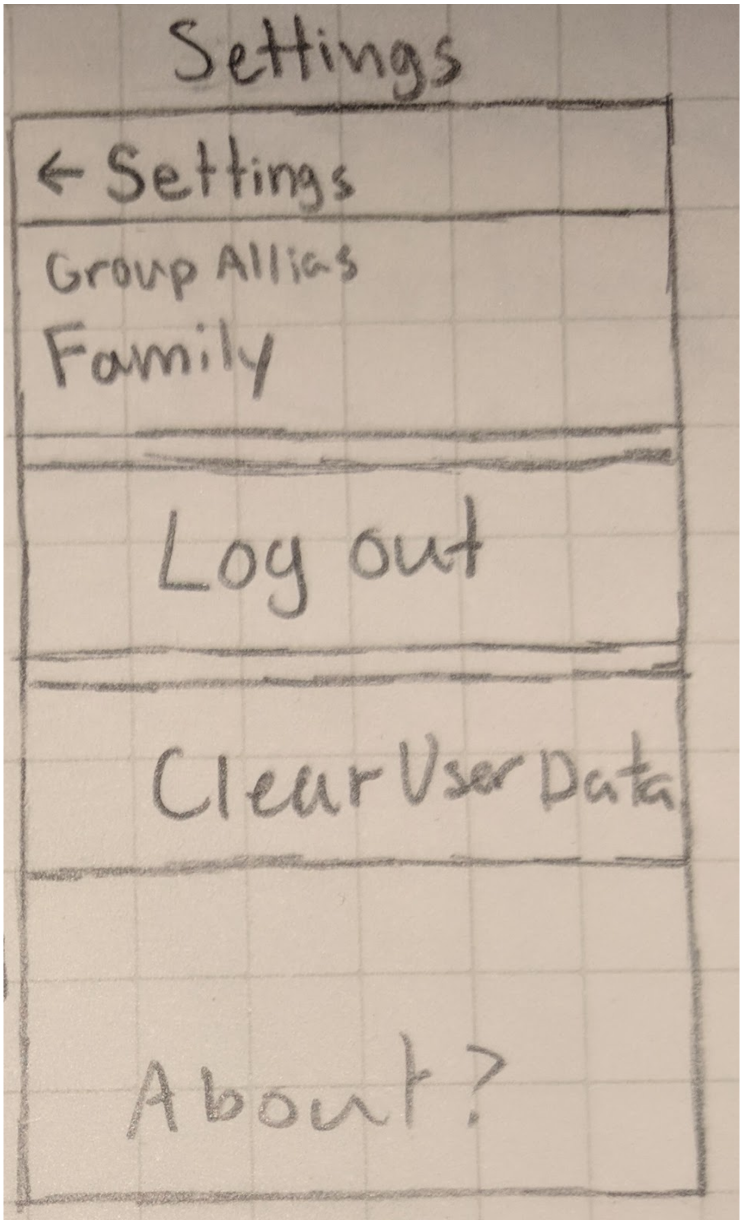
Simple list to view members. Will probably use email, to keep it unique and will pull the name on their account.
Snack bar will show if it fails.

Screen 7 - Tags



Simple list of the tags. The count is how many tasks it has uncompleted. Tags are global, so users can add them but not delete them. No restrictions or filters for this version.
Snackbar if it fails.

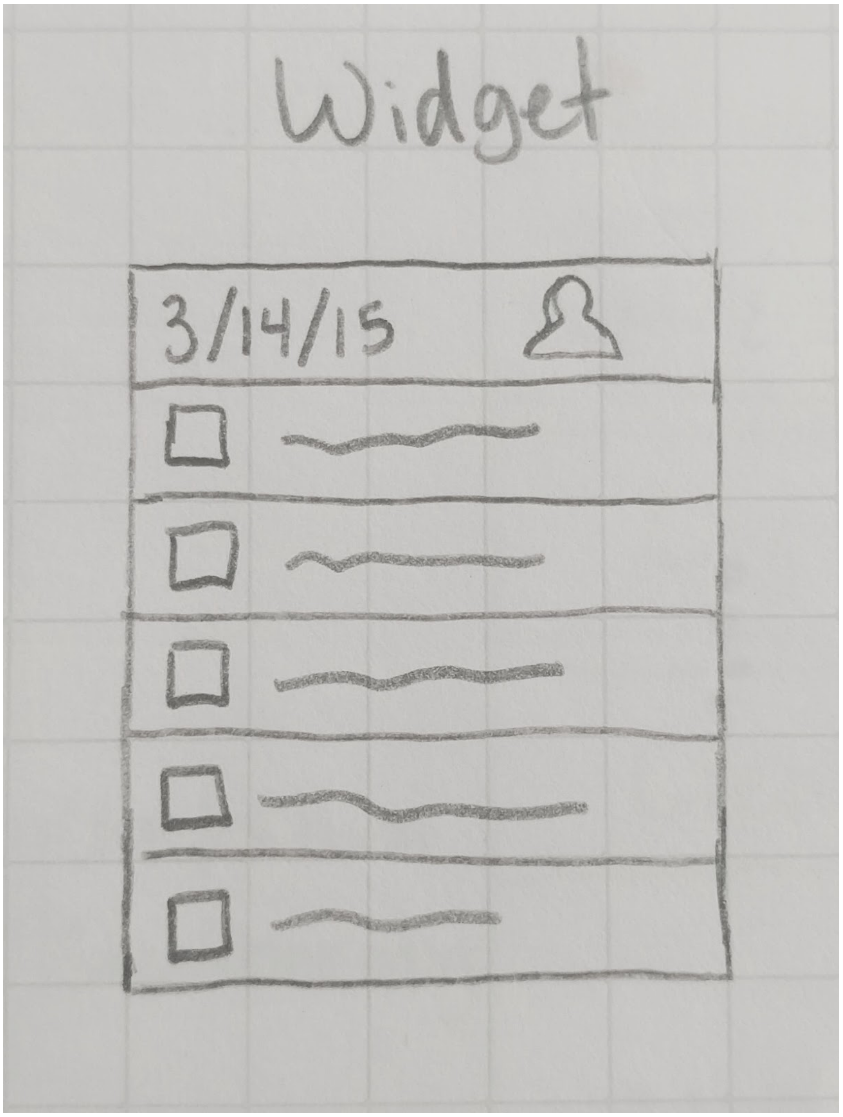
Screen 8 - Settings



A little feature I'd like to add, is let the user define the term (or terms) used to describe their group. Family doesn't feel right for some, so they can change it to something they like better.

There's also some empty space that could be filled up using some simple About Me info.

Screen 9 - Widget



A card with a list of tasks for today. A mini Toolbar will display the date and a button. The button will swap between Persian and Family.

The list is simplified. You can still check off items. Only the name is displayed.

Add as many screens as you need to portray your app’s UI flow.

Key Considerations

How will your app handle data persistence?

The app will store and communicate using the Firebase Realtime Database.

Describe any corner cases in the UX.

To edit a task, click on it and the same dialog will open to edit the details.

Firebase updates the RT DB with the last write wins by default. If a user is making an edit to a task, and someone submits a change to it, they will be notified with a Snackbar to update changes. This will discard the changes they made, or they can submit their changes to overwrite.

Firebase will keep the reference in sync, so it can stay updated without opening the app. It will notify the user with a snackbar to let them know they're offline.

Describe any libraries you'll be using and share your reasoning for including them.

Library by pchmn for displaying Chip Views and a method to input them. I incorporated it into my own Android Library that use it for tags and tagging specifically.
I also use the Firebase Realtime Database api to store and manage data.

Describe how you will implement Google Play Services.

Describe which Google Play Services you will use and how.

- Firebase Authentication
  - Use this to keep users data separate and secure
  - Family data will track user ids to link their profiles
- Analytics
  - This will keep track of some data as they use the app
    - Tasks completed
    - Size of members
    - Count of personal tasks and family tasks
    - Tag usage

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

Write out the steps you will take to setup and/or configure this project. See previous implementation guides for an example.

Subtasks:

- Configure pchmn chip library
- Setup Firebase RT Database
- Populate Database with mock data
- Setup Firebase Analytics and Identity

Firebase Realtime Database structure:
root: { tags, users, families, tasks }
tag: name: { id, count, task_ids }
user: id or email: { family_key, name }
family: id: { users, name }
tasks: user_id or family_id: { name, id, date, desc, tag_ids }

Task 2: Implement UI for Each Activity and Fragment

Subtasks:

- Build Tabs UI for MainActivity
- Add RecyclerView and adapter for both tabs
- Make a fake login screen
- Build Dialog Frag to add a task
- Search menu and ChipInput
- Implement tablet layout scheme
- Assign Accessibility features to the views
- Configure RTL layouts

Task 3: Retrieve Firebase Data

Subtasks:

- Retrieve Firebase Identity data and setup login screen
- Retrieve Firebase Database based on the identity passed via intent

Task 4: Update Task Adapter and Add Task

Subtasks:

- Assign Task data to the adapter(s)
- Implement FAB to open a DialogFrag to add a Task
- Open DialogFrag on Adapter item click to edit fields and save to the same item
- Setup offline state snackbar
- Search filters results

Task 5: Set up Family management activity

Subtasks:

- Open activity and populate member info
- Open DialogFrag when adding a member
- Check member to see if they exist
- Update Firebase with new data
- If it fails, display a Snackbar

Task 6: Add available Tags

Subtasks:

- Have an option in the menu to view tags
- If add tag fails, show Snackbar
- For now, tags will be global

Task 7: Widget

Subtasks:

- Work on layout and manifest
- Retrieve personal and family task data

- Display on list and have a button to swap to the other list

Task 8: Notifications

Subtasks:

- Have a listener for updates to the tasks
- Notify the user of the changes
    - Tasks added
    - Tasks completed
    - Family members changed

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"