KeepSake
Documentation

Supervisor
Frank Wang

Team Members
Nina Averill - Backend Developer
Kevin Ngoh - Frontend developer/Designer
Kirentheren Sanmugam - Backend Developer
Marvel Jonathan Sulindro - Full-stack Developer

Team Name
404s&Heartbreak

Client
Michelle Gan

Table of Contents

1. Background

The things we care about have stories.

A simple grandfather clock that adorned the mantel of a childhood home — with all its earned nicks and scratches - is an heirloom. A new clock just purchased and in perfect condition, remains — at least for now — mostly just a clock.

The difference is the story behind it which has a trail of memories.

But stories often disappear as clock - and many of the things we care about - change hands and become just one more thing stored in an attic, displayed in an antique store - or sent to the garbage dump.

Its story matters.

Historical preservation and conservation are probably the most important reasons. The more one knows about an item's history — its provenance — the more one is likely to respect, appreciate and care for it. Currently, there is no mainstream mobile or web-based application dedicated to recording and preserving of family artefacts.

KeepSake is an Android application exists to help an individual protect and preserve the stories of the items they care about by documenting, organizing and sharing family artefacts. Using our application to register an item is an accessible and user-friendly way to ensure its story is safely preserved and shared with future generations.

2. Client

Client Profile

Name:
Michelle Gan

Age:
57 years old

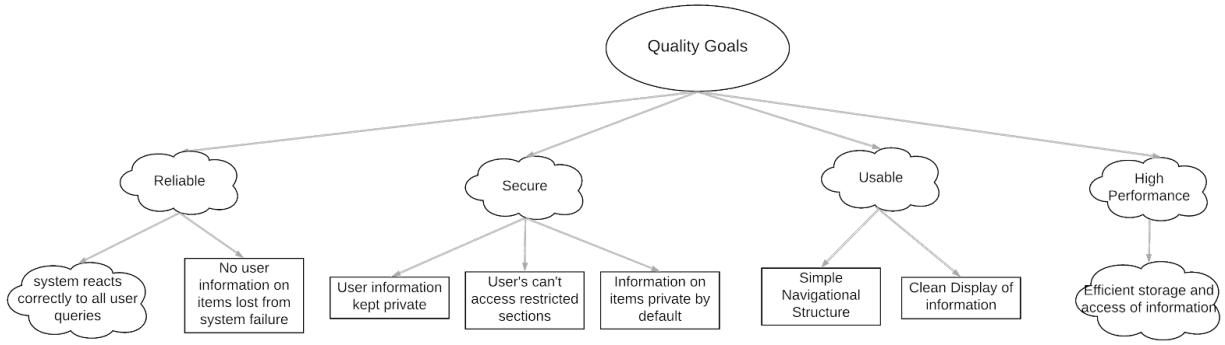
Background:
Single mother of three children, who are studying abroad. Michelle and her children have stayed in their current house for 14 years and have a lot of sentimental attachment to most of the items in the house.

Michelle is considering downsizing to a smaller house, and because of her decision she needs to part ways with a lot of sentimental items in her possession whilst retaining the memories of all the items she's getting rid of. (KonMari Method by Marie Kondo)

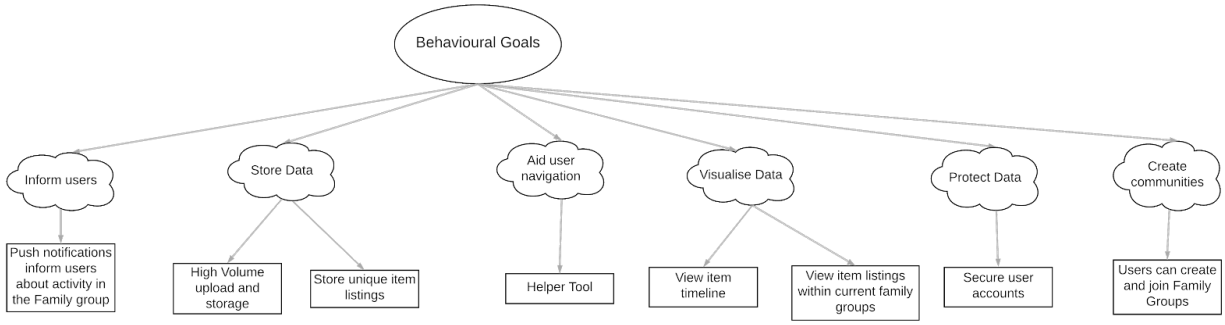
Request:
She approached us and asked if we could build an heirloom registry mobile app that she can use to retain her memories on her handheld device.

3. Requirements

3.1. Quality Goals



3.2. Behavioural Goals

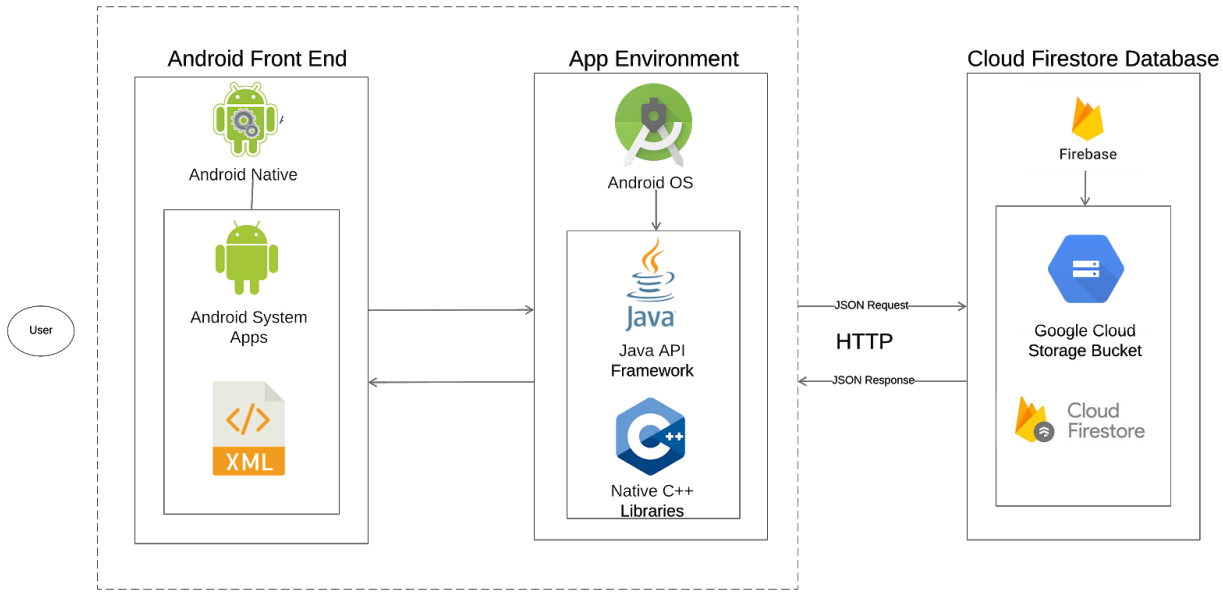


3.3. User Stories

Theme	Epic	As a	I Want to	So That	Acceptance Criteria			Priority
					Given That	When	Then	
Store/Visualise Data	Upload Items	All Persona	Create a unique record of an item I own	I can access the item online and record information about it	A user account has been created;	Users are logged into the account	A unique item is created in the system	High
					The user is assigned to a family group		Ownership of the item is assigned to the uploading user	
	Transfer ownership of items	Current owner of a listed item	Transfer ownership of that item to another user	I can read/write/access rights to another user	The user is the current owner of the item	Users are logged into the account	The unique ID of the item is transferred to the other user, who then gains read/write/access rights to the item	Medium
					The other user is a verified account			
	View Items	All personas	View item details	View the memories /history attached to that item	A listing has previously been created	Users are logged into the account	Information for the item is shown either as a summary (current information) or a timeline (history of the item)	High
					The user is the current owner of the item or is in the same family group as the owner			
					The user has read permission			
Create communities	Create Family Groups	All Personas	Create a family group	my family can record items in a secure, enclosed group	A user account has been created	User login details have been verified	A unique family ID is created and assigned to the new Family Group	High
	Join Family Groups	All personas	Join an existing family group	I can record items that my extended family group can view	A user account has been created	The user is logged in	A request is sent to the family group for acceptance	High
					The family ID is valid and exists			
Inform Users	Push notifications	All personas	Be notified when there is notable activity in a family group	I can stay updated on activity	The user is a member of the family group	The user is logged in	The user will receive notification when there is activity	Low

4. Technology Stack

4.1. Architecture Design Diagram



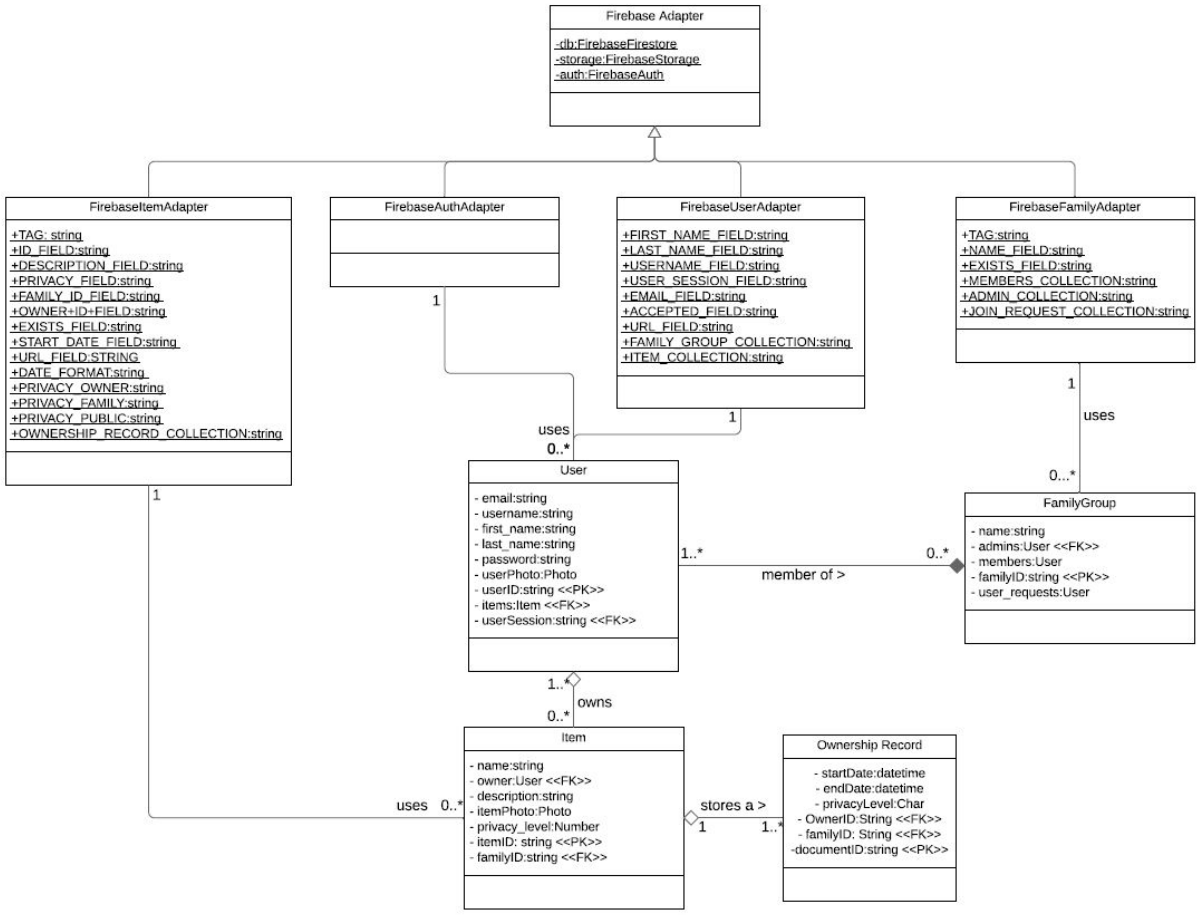
Description

- The app is primarily developed for the Android platform, and was therefore built primarily using the Android native development environment
- The frontend is developed using built-in UI packages and XML
- Used to model documents and collections
- A No-SQL server hosted on Firebase Firestore is used to store persistent data
- The Google Cloud Storage Bucket is used to store images

Justification

- We chose an Android Studio environment because the client uses and Android device and we've worked with the Java language before.
- We chose Firebase because it provides an API for developers, that lets us synchronize data and store it in the Firebase cloud. The real time database (RTDB) syncing function provides the client with all the missing files after the connectivity recovering which makes it very valuable for stable work of your app

4.2. Database Entity Relationship Diagram



Description

Data

User Model:

- Contains information on the user profile including unique identification such as username and email.
- Includes the user's first and last name, as well as their profile picture.
- Stores password for user authentication
- Stores a list of family groups that the user is in
- Stores a list of all the items the user currently owns (self-upload or transfer ownership)

Item Model:

- Contains information on the item class such as name, description, privacy, photo, and current family group.

Family Group Model:
- Contains information on the family group such as name and unique family ID.
- Stores a list of members of the group
- Stores information on the admin of the group.
- Stores a list of join requests for the group.

Ownership Record Model:
- Contains the timestamps a User owns the particular item. (Start date and end date, if exist)

Adapters Model:
- The main adapter for the application is Firebase Adapter.
- It keeps variables such as authentication, storage, and firestore.
- FirebaseItemAdapter, FirebaseUserAdapter, FirebaseFamilyAdapter, and FirebaseAuthAdapter all extends the main FirebaseAdapter class.
- The 4 adapters (FirebaseItemAdapter, FirebaseUserAdapter, FirebaseFamilyAdapter, and FirebaseAuthAdapter) have their own variables and functions specific to their uses.

Relationships

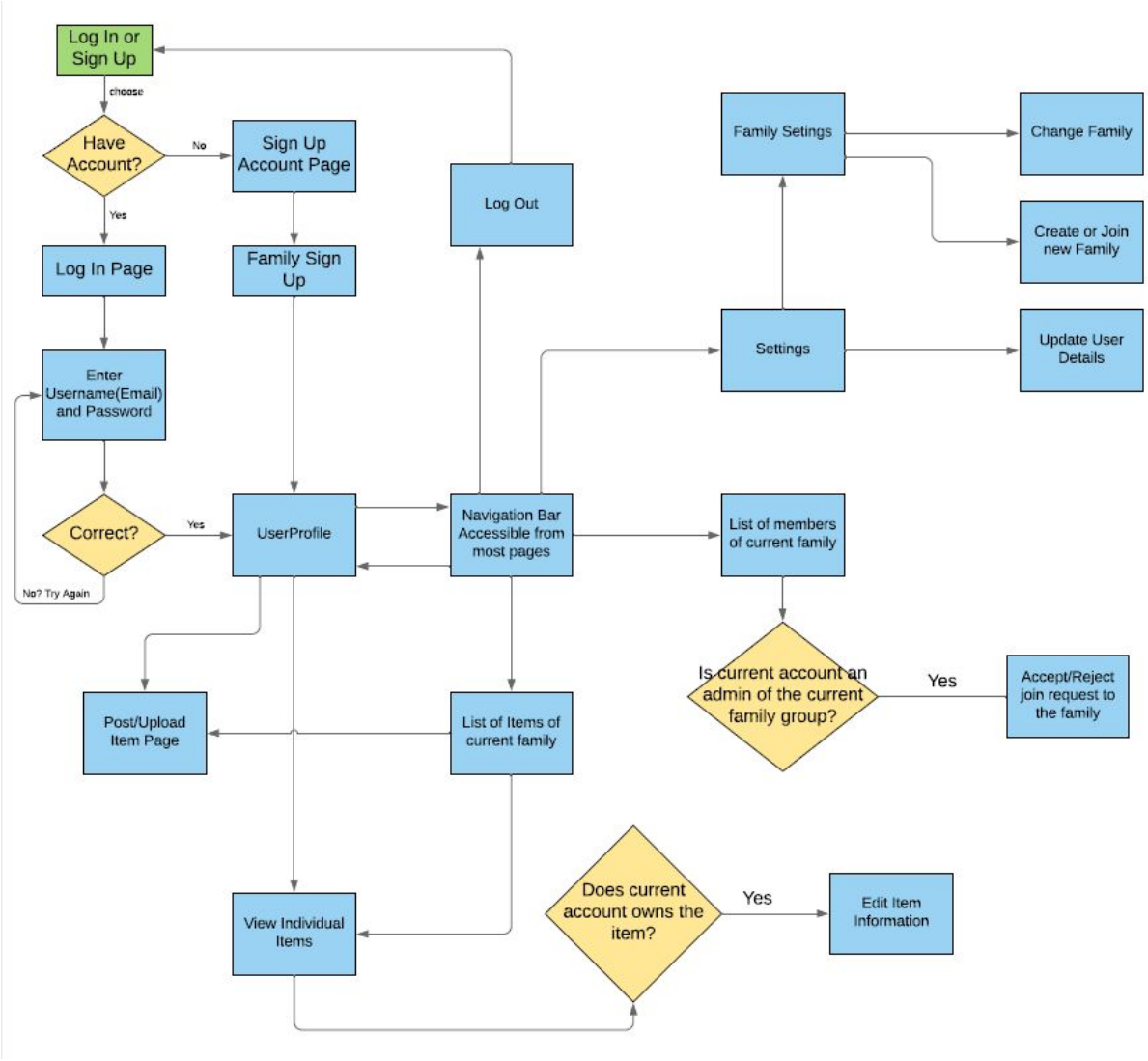
User - Family Group
- At any given time, a user could one have 1 user session.
- The user session will reflect what is shown on the 'item' page on the navigation bar when the application is running. In this item page, it will show all the items that are posted under that specific familyGroup/UserSession.
- Users can create or join existing family groups. This will allow them to post items to the specified group and view each group's items separately.

User - Item
- Users can perform CRUD functionalities on the item model. The user is also able to set its privacy and choose which family group the item belongs to.

Items - Ownership Record
- Each item model contains a list of ownership records. Every time a user that owns the item passes ownership to another user, its gonna add an ownership record to the list of ownership records.

Class - Adapters:
- Each of these adapters is used by their respective class pairings. (FirebaseUserAdapter is used by User, FirebaseFamilyAdapter is used by family group and FirebaseItemAdapter is used by Item).
- FirebaseAuthAdapter is also used by Users for authentication purposes.
- These adapters ensure consistency of access and variables between each instance of the class. This is how we ensure consistent access to the database through the app.

4.3. User Flow Journey Diagram



Notes:

1. The navigation bar is accessible from pages such as User Profile, List of members of current family, List of items of current family, Viewing individual items, Settings page, Changing Family, and update user details page.
2. The navigation bar includes access to User profile, List of members of current family, list of items of current family, settings and logout button.
3. For the 2 conditions: “is current account an admin of the current family group” and “does current account owns the item” if the answer is no then they won’t be able to proceed with the additional features. (“edit item information” and “accept/reject join request to the family”)
4. Finishing an action like “uploading an item” and “editing item information” will automatically redirect the user to the user profile page which is our “main page”.
5. Any action finished through the “settings” page will be redirected to the “settings” page.

4.4. Use Case Diagram

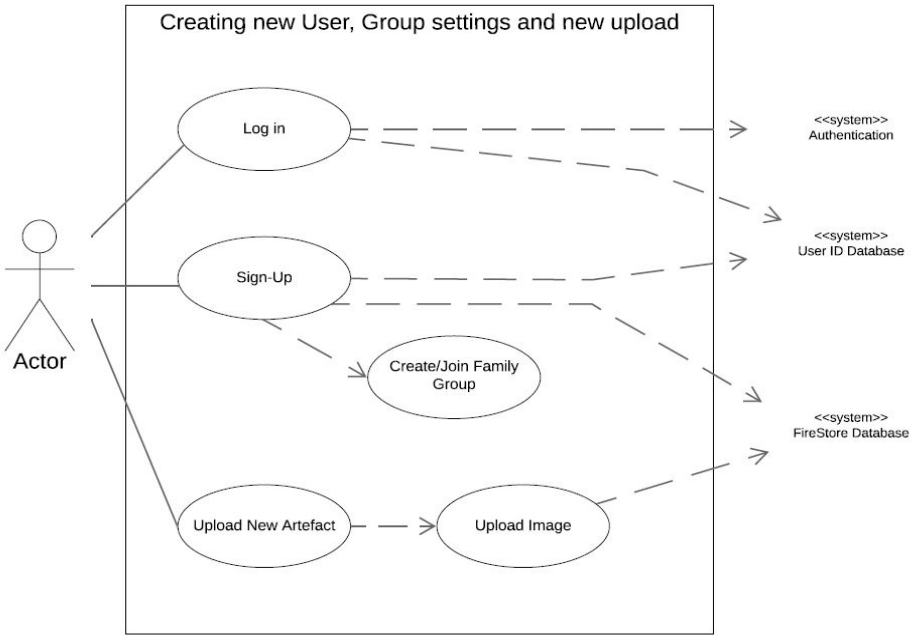


Diagram 1. Creating new User, Group Settings and new Upload

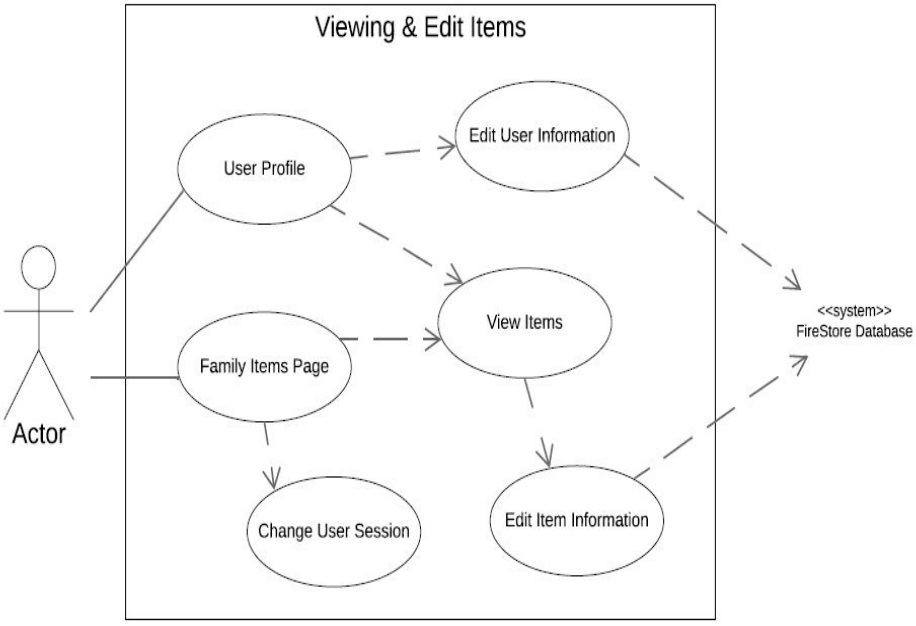


Diagram2. Viewing & Editing Items

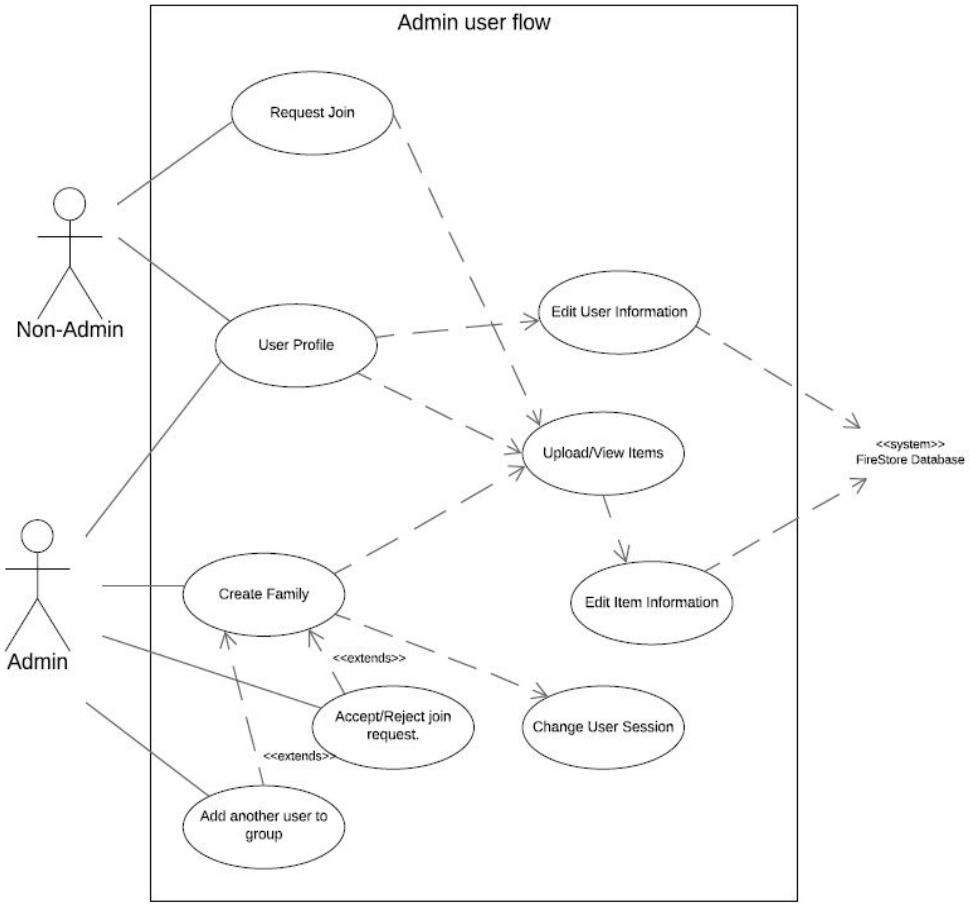
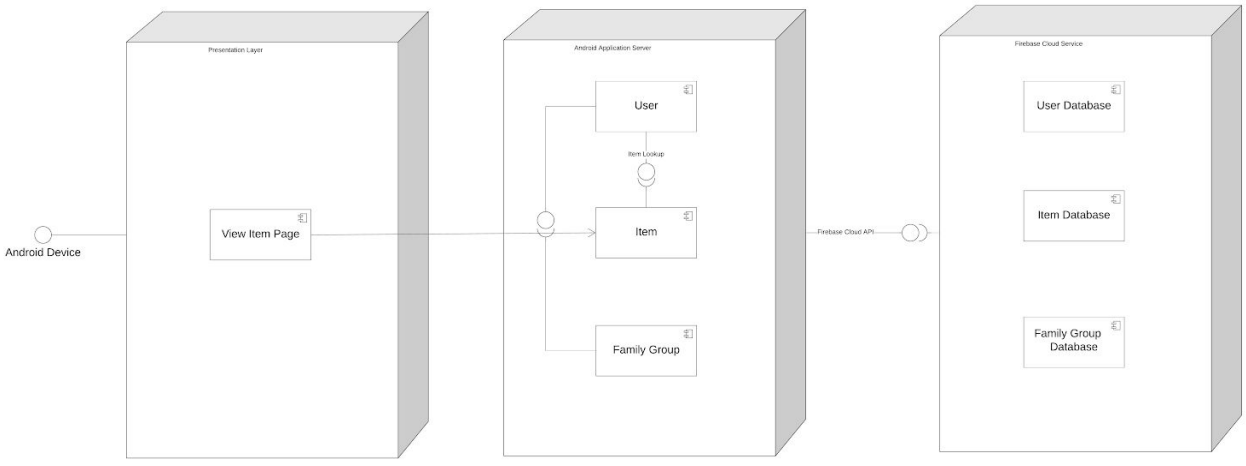


Diagram 3. Family Group access for admins vs non-admins

4.5. Component Diagram



5. Sprints

5.1. Planning - (Ahead of time / On Time / Behind)

	Sprints	Due Date
Week 2	Requirements	9th August 1pm
Week 3	WireFrame + Consult Client	15th August 11am
Week 4	Paper Prototype + Client Details Design Prototype	23rd August 11.59pm 25th August 11.59pm
Week 5	GitKraken Setup Setup Firebase and Authentication Finish 5 screens (static screens) Work on the UI/Design Layout	27th August 11.59pm 27th August 11.59pm 28th August 10.00am 28th August 10.00am
Week 6	Connect BackEnd to FrontEnd Work on Authenticating User Profile	4th September 10.00am 6th September 11.59pm
Week 7	Wireflow Diagram Architecture Diagram	13th September 11.59pm 13th September 11.59pm
Week 8	Show client current progress + design	20th September 11.59pm
“MidSem Break”	Implement Privacy for items Be able to post items on the app (image, description, etc) Retrieve backend info and post on the frontend	24th September 10.00am 27th September 11.59pm 27th September 11.59pm
Week 9	Convert URL to ImageView Dynamic UI Design Search feature	4th October 11.59pm 4th October 11.59pm 4th October 11.59pm
Week 10	Clean up UI design for posts Create timeline view	11th October 11.59pm 11th October 11.59pm
Week 11	Finish up all functionalities Do presentation Run Tests	18th October 11.59pm 22nd October, 11.12am
Week 12	Write Report up final report + individual essay	25th October 11.59pm

5.2 Sprint Review

We kept our sprint reviews casual. Team members would gather around a table for informal discussions, demos and describe the work they've done for that iteration. We asked each other questions, tried implementing new features and provide constructive feedback. Our meetings usually consist of our client, the full development team and the scrum master. Sharing in success is an important part of building an agile team. We had used Trello board, a web-based Kanban-style list-making application to organize our tasks and deadlines clearly to demonstrate the work we have completed over the sprint and compare it to the commitments given at the beginning of the spring. The sprint review schedule could be viewed at

https://trello.com/invite/b/U9oy9CgK/e04a409765a46e8f05cb3b7a3b9a650c/sprint-schedule.

5.3. Retrospective

Sprint Retrospective 1

Went Well	Needs Improvement	Concerns	Action Items
Good momentum throughout the team	Pay more attention to Trello board	Might lose momentum	Look at Trello board more often
Good communication between team members	Need to record more of our discussion; more detailed minutes	Losing time to other commitments outside of this subject	Refine backlog
Great team attendance/participation	More detailed documentation of artifacts	Ensure we continuously communicate with our client regularly	Assign a minutes taker at the start of the meeting; potentially team member adding to the documentation
Continuously asking for feedback from client and supervisor to ensure we're on track	Take deadlines more seriously	Overcomplexity of idea within the timeframe of the project	Continuously update the project management document

Sprint Retrospective 2

Went Well	Needs Improvement	Concerns	Action Items
More group working sessions	Bring up motivation	Slow integration	Prepare all artifacts before the tutorial
Attempted to create progress every week despite multiple assignments	Time Management	Not prepared for tutorial	Communicate more in slack
Productive working sessions every week	Individual working sessions have been less effective as group sessions	Need to pick up the pace for major features	Consistent working sessions on Mondays
Most of UI completed	People need to take ownership of their tasks	Losing momentum	Monitor trello board and update items as they progress
User sign-in flow completed		Trying to juggle too many tasks at once and some important tasks are being forgotten	Arrange more group discussions and communicate tasks
Front-end team ready to start supporting the back-end team			Plan out meetings in the week ahead
Good at sharing information/resources and communicating between the group			Taking Initiative to help other members

Sprint Retrospective 3

Went Well	Needs Improvement	Concerns	Action Items
Meeting a lot more often	Punctuality to meetings	Timeline til demo	Refactor firebase data
Had very productive working sessions	Coding conventions not consistent	Database consistency	Making a consistent database format for all classes/entries
Great pace of development	Communicating features that people are working on so there's not duplicate work	People having other assignments to worry about	Dedicate time to step through the app and ensure everything's connected
Communicating a lot more on slack and sharing knowledge/helping each other out		Most features are complete but the user flow needs a lot of work	Structured testing process (delete all database items and see what happens to consider edge cases, modify information and check the behaviour of the app)
Working to the project development timeline we set (creating all basic functionalities we planned to develop)		UI needs to be cleaned up	User acceptance tests
All very comfortable with the development environment and all tackling more challenging features		We might be missing some small element	
		Not considering edge cases	