# IT Project Sprint 2

Team **ctrl**

Sep 2019

## 1 Introduction

This is a summary of what has been done during sprint 2 for team ctrl. We reflected on our progress in sprint 1 and built many more features on top of our progress. Below is a documentation of progress for sprint 2.

## 2 Client Requirement

### 2.1 User stories

We have had a comprehensive summary in sprint 1 which lists each big feature as an epic, consisting of various user stories. In sprint 2, since we are working more deeply in coding, we figured out that a story point of 8 hours is a bit too broad, and tasks involving 1 SP of work could be further broken down into smaller sections. Indeed, we encountered some complex tasks that has a few sub-tasks inside. Therefore our user story for sprint 2 will be a supplement to what we did in the previous sprint, where already finished features will not be listed here. Note that **one story point corresponds to approximately 2 hours of work.**

#### 2.1.1 Login Mechanism

- As the main user, I want to be able to log in securely with my username and password. Backend shall handle my login and store my personal information with great safety. (Therefore we shall connect our app to Firebase and verify the user upon login. This can be done via Firebase authentication functionality.) **1 or 2 SP.**
- As the main user, if I forgot the password, I need to have a mechanism of recovering and resetting my password for login. (We will implement forget password feature, where the app will send an email to the user and he can use the unique link to reset password. ) **Already implemented front-end. Backend proposed for 1 SP.**
- As the account holder, I wish account security to be properly handled and can remind me of issues when trying to create an account or perform login. (It is essential to have security reminder, e.g. password strength indicator, alerting what caused failure, and handle navigation properly.) **1 SP in all, since we already did part of the backend functionality for Firebase.**

#### 2.1.2 Upload Artefact

- As a user, when adding artefacts, I require two ways of adding: from existing family member or from totally new setting. (This requires us to have two guides on item upload, from member or from new.) **Top priority at this stage, proposed finishing time: 4-6 SP.**

- As a user, when using the guide for uploading from members, it should guide me through many stages, including choose member, choose artefact, confirm selection etc. I should quickly upload the artefact I desire as it is from existing member in the application. (We have a guide for EACH family member to add their artefact from another member.) **Front-end was almost finished, backend model still in development and we planned for approximately 4 SP, 1 for each stage.**

- As a user, when using the guide for uploading from scratch, I can first choose which media I wanna upload, including text (memorials), images or videos. For each method, a short guide will let me upload from my local file system. Each artefact has metadata that I can view later. (We have a guide for EACH family member to add their artefact from the start. User-friendly upload stages will be provided.) **High priority, planned for. 3 SP.**

#### 2.1.3 Register

- As a user, I want my artefacts to be stored in an easy-to-browse way. UI should be straightforward. (We have an artcard implementation for artefacts to be displayed.) **Most finished, working on some minor polish. 1 SP suffices.**

- As a user, I wish to share the artefacts I have to social media. (Share options to a range of social media platforms will be provided, including Facebook, Twitter, WeChat, Messenger and Instagram. To make easy sharing possible, once we direct user to sharing app, clipboard will contain the required sharing text and camera roll will contain the image from register.) **We planned for approximately 4 SP.**

#### 2.1.4 Profile

- As the account holder, I wish to freely edit my family profile, as well as editing each member's profile. (Edit page will allow users to change metadata for each member, including avatar, first name, last name and date of birth.) **Front end finished, back end in progress and should be done in 2 SP.**

- As the account holder, I need an Achievement system to reward myself and motivate me to add artefacts consistently. (There will be badges related to family milestone, community etc. And they will be unlocked once the goal is reached, for example, uploading 5 items.) **In development, this needs to be handled after most of other features are completed. We will finish it in 4-5 SP.**

#### 2.1.5 Community

- In the community, I want to share my artefacts conveniently and quickly to save my time. (There will be a button to add artefacts from your registry directly, as a form of image. Quick guide will be provided.) **First priority, haven't started yet, hope to finish in 5 SP.**

- As the client, I want the community to be simple to use and concise. Specifically, with each item in community, there should be an image, its upload location, description and number of likes it has. I can like it and make comments. I can also "cross it out", going to the next artefact. **Backend in progress, planned for 3 or 4 SP.**

## 2.2 Evidence of requirements collection

We continued using WeChat to connect with our client and had a chat on the current progress, security mechanism and how satisfied he was on features so far. The conversation was logged and can be viewed in the figures below.

# 3 Development

## 3.1 System Architecture

We have documented our system architecture for reference. They can be viewed in documentation folder in Github Repository. **Architecture.pdf** illustrates the whole structure for our application, including Front-end, backend and database communication. It is a broad and high-level image of the structure. **FamoryDBTree.pdf** is our database schema. The text info of database schema can be viewed in **design schema sketch.txt**. Essentially we used a tree structure to represent collections and fields since Firebase stores items in key-value pairs as NoSQL format. We used several collections and linked member and register with Object IDs to avoid nesting data.

## 3.2 Coding Practice

### 3.2.1 Repository

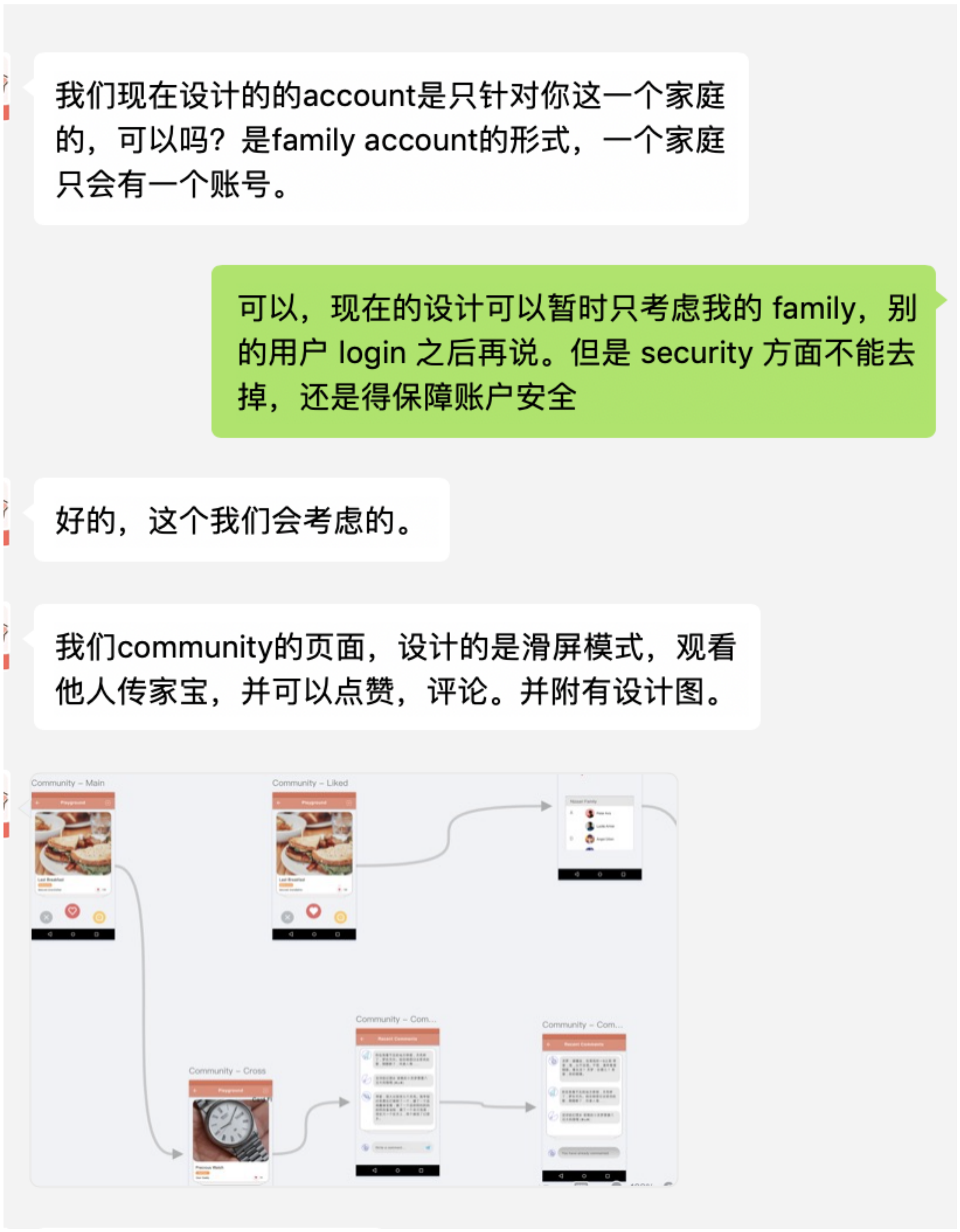
We used branching and pull requests to manage repository, so that possibility of conflicts can be minimised. The readme in repository has rules for making changes and code review. More information is listed here. Additionally, for each pull request reviewers will point out possible mistakes and it will be merged to master branch only after the errors are fixed.

### 3.2.2 Code

All important functions have commenting for others to understand the purpose of the code block. Variable and function names are also well defined. We have configuration files to define some components which need to be reused, namely colors, strings and constants. This makes code easier to understand.

## 3.3 Testing

We used both manual testing and automated testing for the application. Manual testing involved testing in development, for example, check if a certain feature (e.g. add to Firebase storage, add artefact from scratch, etc.) works. The features to check are listed in Trello and will be moved to "Done" card when finished. Automated testing involves usage of jest. We developed some lightweight testing for this. A sample test result can be viewed in figure 4.



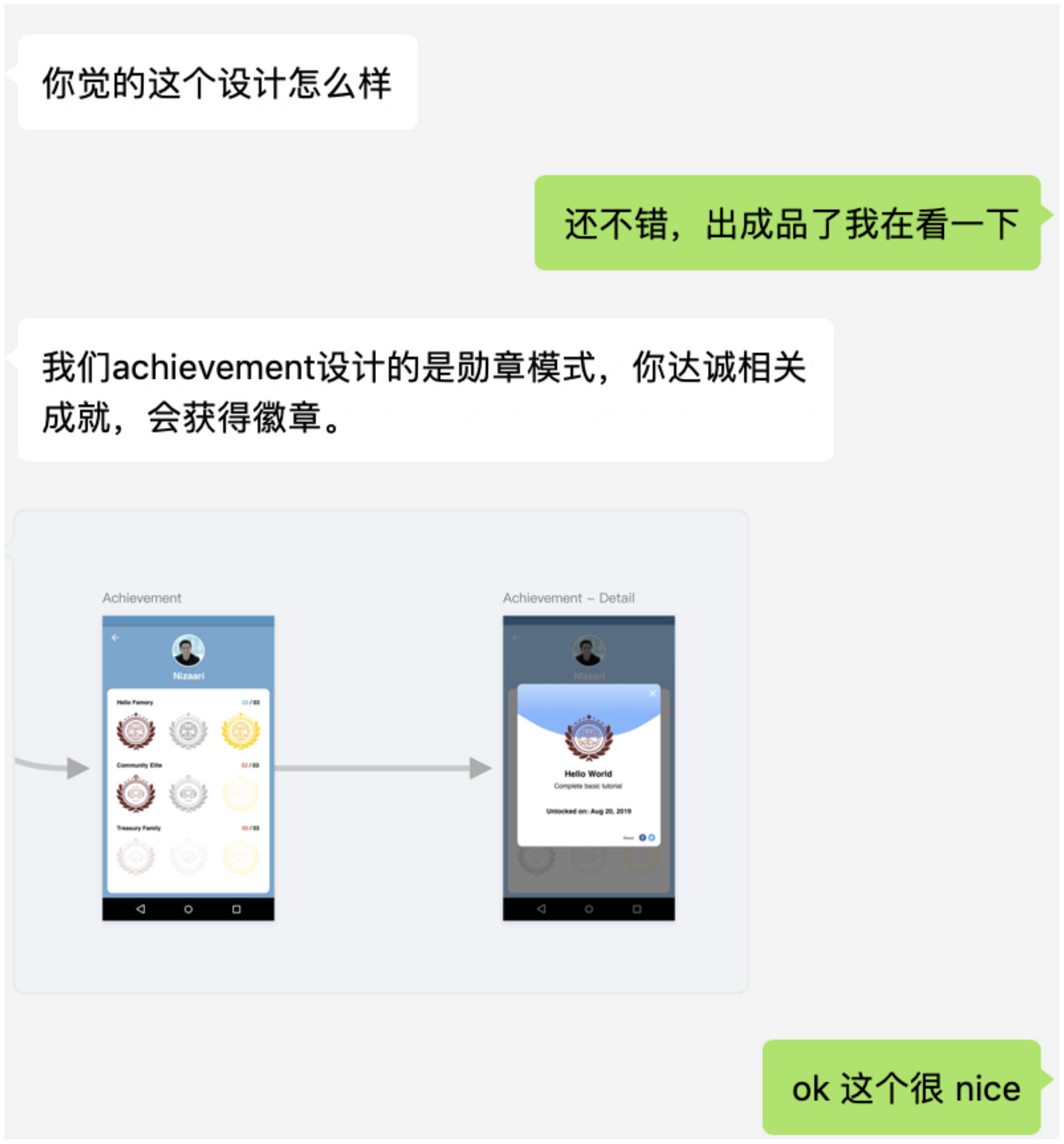
我们现在设计的的account是只针对你这一个家庭的，可以吗？是family account的形式，一个家庭只会有一个账号。

可以，现在的设计可以暂时只考虑我的 family，别的用户 login 之后再说。但是 security 方面不能去掉，还是得保障账户安全

好的，这个我们会考虑的。

我们community的页面，设计的是滑屏模式，观看他人传家宝，并可以点赞，评论。并附有设计图。

Figure 1: **Chat History**



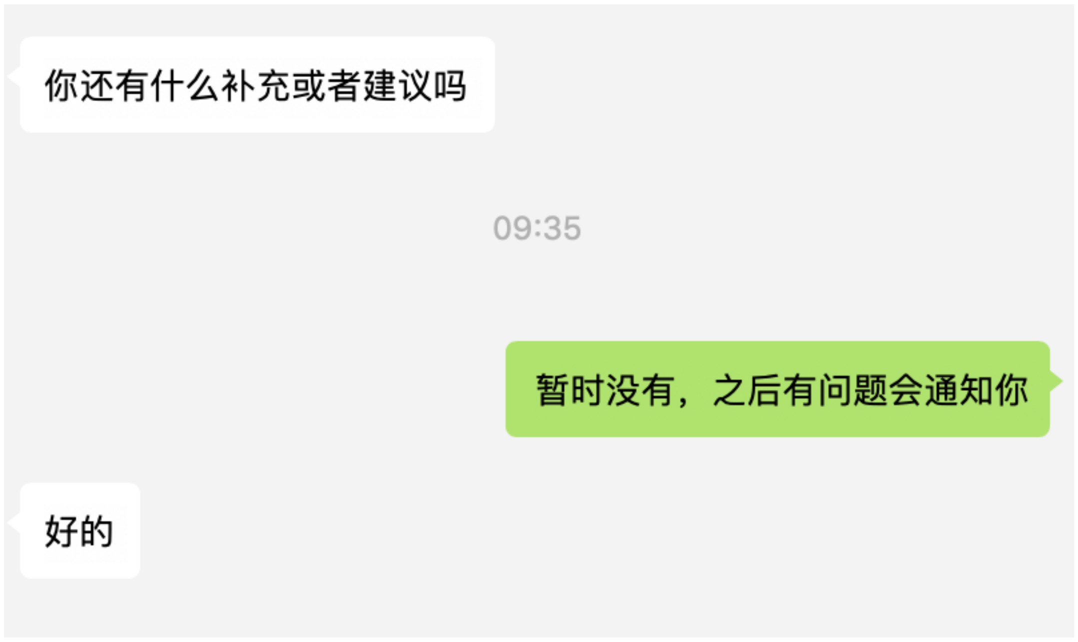
你觉的这个设计怎么样

还不错，出成品了我在看一下

我们achievement设计的是勋章模式，你达诚相关成就，会获得徽章。

ok 这个很 nice

Figure 2: **Chat History**


你还有什么补充或者建议吗

09:35

暂时没有，之后有问题会通知你

好的

Figure 3: **Chat History**

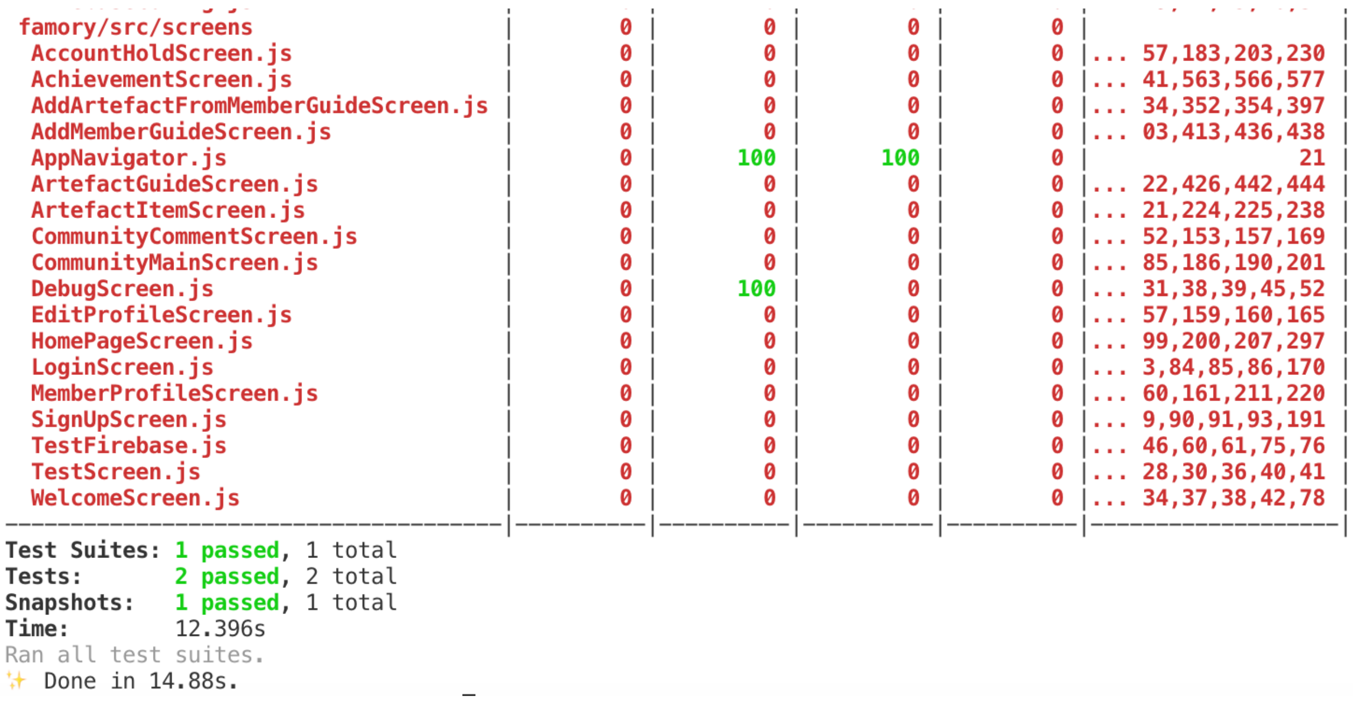

```
 famory/src/screens                    |        0 |        0 |        0 |        0 |                   |
  AccountHoldScreen.js                 |        0 |        0 |        0 |        0 |... 57,183,203,230 |
  AchievementScreen.js                 |        0 |        0 |        0 |        0 |... 41,563,566,577 |
  AddArtefactFromMemberGuideScreen.js  |        0 |        0 |        0 |        0 |... 34,352,354,397 |
  AddMemberGuideScreen.js              |        0 |        0 |        0 |        0 |... 03,413,436,438 |
  AppNavigator.js                      |        0 |      100 |      100 |        0 |                21 |
  ArtefactGuideScreen.js               |        0 |        0 |        0 |        0 |... 22,426,442,444 |
  ArtefactItemScreen.js                |        0 |        0 |        0 |        0 |... 21,224,225,238 |
  CommunityCommentScreen.js            |        0 |        0 |        0 |        0 |... 52,153,157,169 |
  CommunityMainScreen.js               |        0 |        0 |        0 |        0 |... 85,186,190,201 |
  DebugScreen.js                       |        0 |      100 |        0 |        0 |... 31,38,39,45,52 |
  EditProfileScreen.js                 |        0 |        0 |        0 |        0 |... 57,159,160,165 |
  HomePageScreen.js                    |        0 |        0 |        0 |        0 |... 99,200,207,297 |
  LoginScreen.js                       |        0 |        0 |        0 |        0 |... 3,84,85,86,170 |
  MemberProfileScreen.js               |        0 |        0 |        0 |        0 |... 60,161,211,220 |
  SignUpScreen.js                      |        0 |        0 |        0 |        0 |... 9,90,91,93,191 |
  TestFirebase.js                      |        0 |        0 |        0 |        0 |... 46,60,61,75,76 |
  TestScreen.js                        |        0 |        0 |        0 |        0 |... 28,30,36,40,41 |
  WelcomeScreen.js                     |        0 |        0 |        0 |        0 |... 34,37,38,42,78 |
---------------------------------------|----------|----------|----------|----------|-------------------|
Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   1 passed, 1 total
Time:        12.396s
Ran all test suites.
✨  Done in 14.88s.
```

Figure 4: **Automated Testing**