

Description

Intended User

Features

User Interface Mocks

Screen 1

Screen 2

Screen 3

Screen 4

Screen 5

Screen 6

Screen 7

Screen 8(Widget)

Key Considerations

How will your app handle data persistence?

Describe any edge or 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 or other external services.

Next Steps: Required Tasks

Task 1: Project Setup

Task 2: Create a class for querying data from Cloud Firestore

Task 3: Implement UI for TestingActivity

Task 4: Implement UI for MainActivity

Task 5: Implement UI for LearningActivity, PracticeActivity, StartOrTestActivity

Task 6: Implement UI for LoginActivity

Task 7: Implement UI for menu options

Task 8: Set notifications

Task 9: Implement UI for Widget

Task 10: Make sure the app is robust

Task 11: Add free and paid flavors

Task 12: Customize and polish the app UIs

Task 13: Publish the app on Google Play

GitHub Username: [blyndiuk](#)

SanChinese 三中文

Description

The app helps users learn a big amount of Mandarin Chinese vocabulary fast and retain it efficiently.

It is very simple to use which helps you to concentrate on the main: learning the language.
It also helps you set goals and motivates you to stay on track.

Intended User

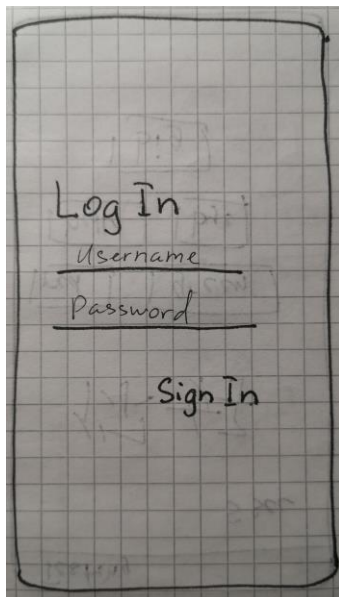
Students and adults who want to remember many words in Chinese in a short period of time.

Features

- Teaches you Chinese vocabulary using both simplified Chinese characters and pinyin and its translation to English or Russian, and provides you with correct pronunciation.
- Picks the vocabulary for you depending on your proficiency level.
- Offers you a set of exercises to memorize the vocab.
- Keeps track of your progress
- Rewards you for persistency and punishes you for lack of it by means of app tokens.
- Nags you to practice (can be changed in settings)

User Interface Mocks

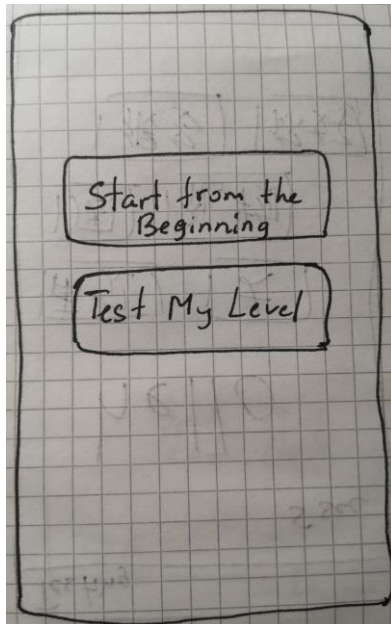
Screen 1



This is LoginActivity screen.

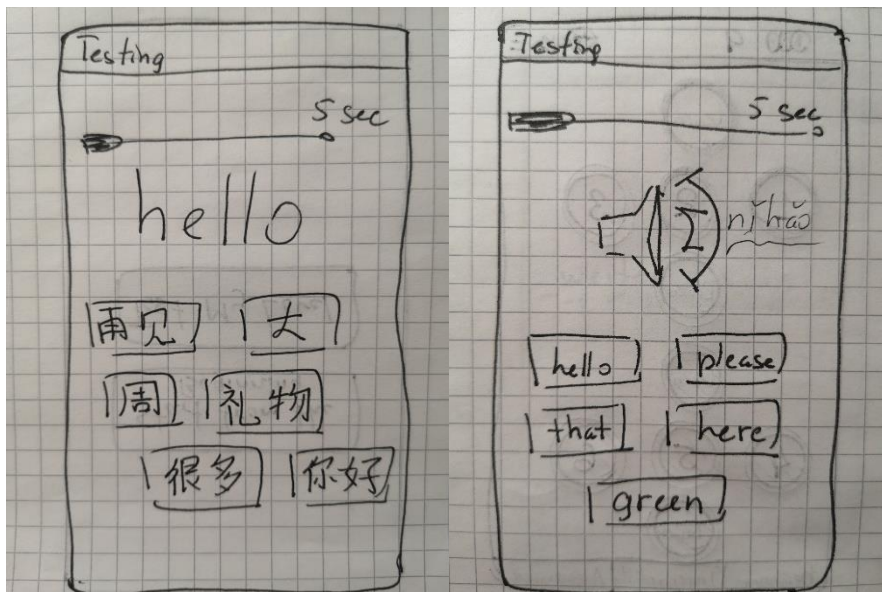
User logs in once and uses the account until logging out.

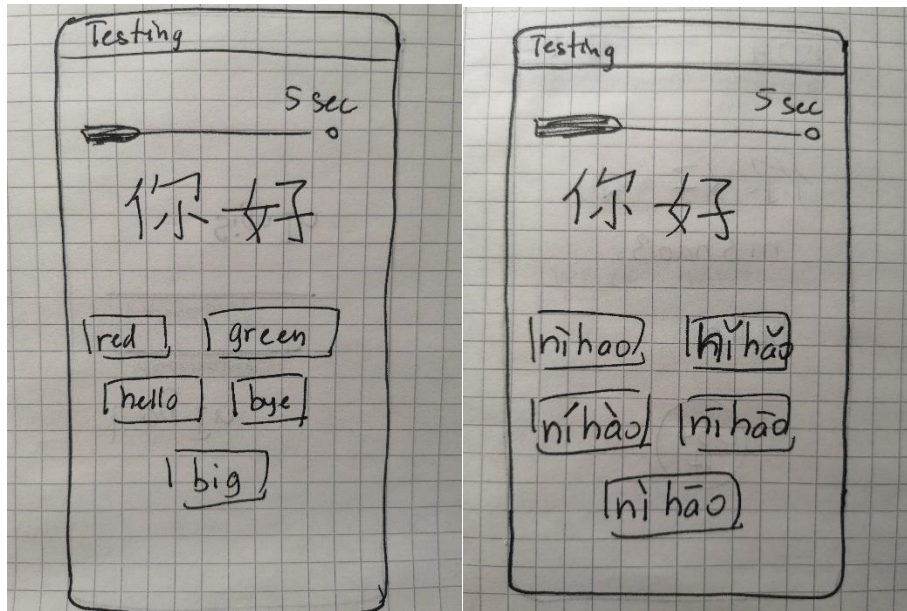
Screen 2



In StartOrTestActivity user chooses whether to start learning from the very basics or to pass the test and be moved up to the appropriate level.

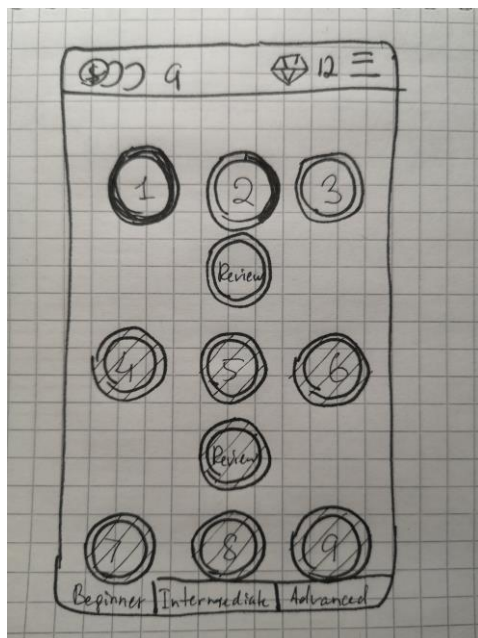
Screen 3





These are variations of TestingActivity. If you choose testing in the previous activity, this is what you see. You get 5 seconds to figure out the correct answer and can make up to 3 mistakes and/or timeouts before the activity closes and you're moved to Main Activity.

Screen 4



This is MainActivity. You see it when they open the app after logging in.

There are three levels: Beginner, Intermediate, and Advanced.

Each level has a range of sublevels each of which consists of 3 units and a review section.

Every Unit teaches 9 words and has 3 fluency levels: Bronze, Silver and Gold with corresponding colors for the icons.

The ring around every Unit's icon shows your progress.

After you've reached Bronze in all 3 units and Review, the next sublevel is activated.

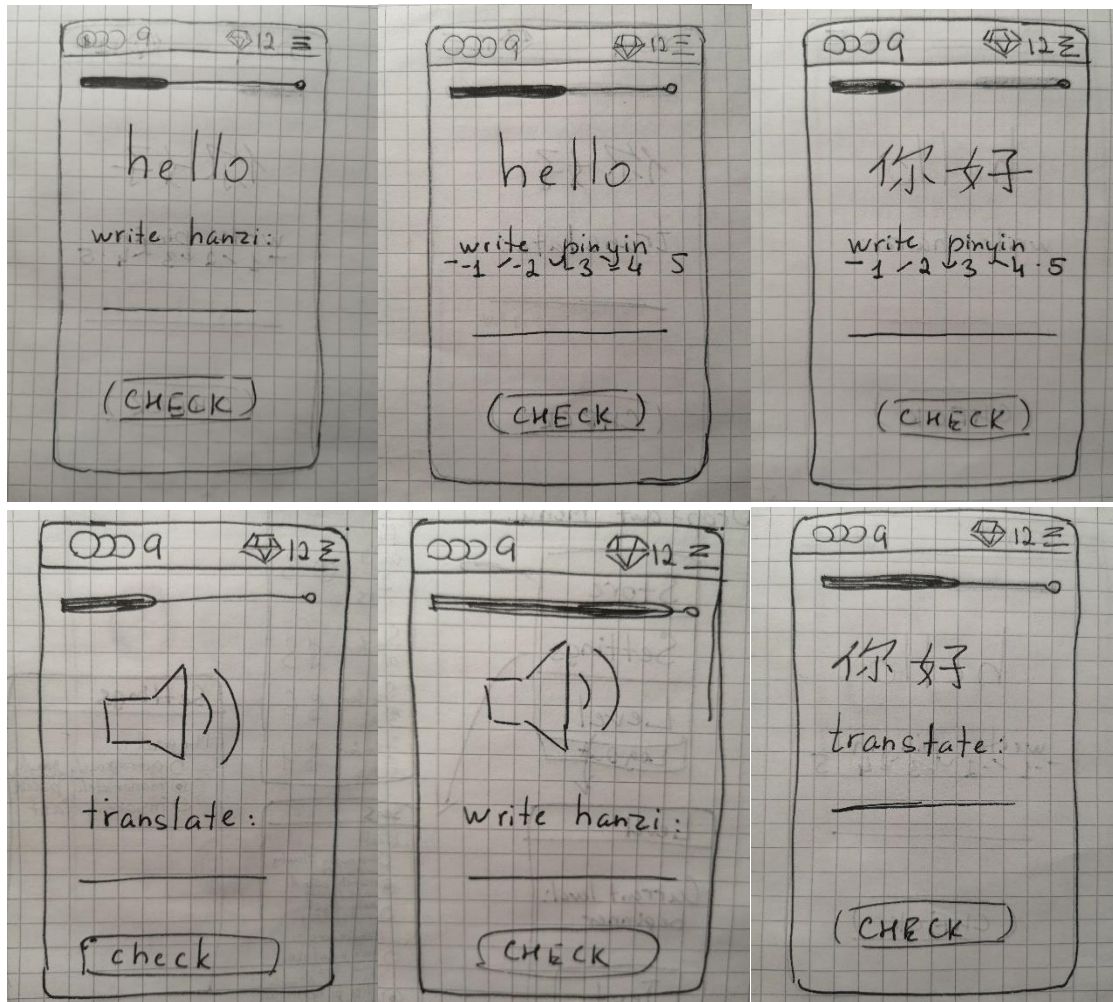
For every mastered word you get an app coin. For reaching the set goal for the day your diamond count increases by one. You can set goals in settings.

Screen 5



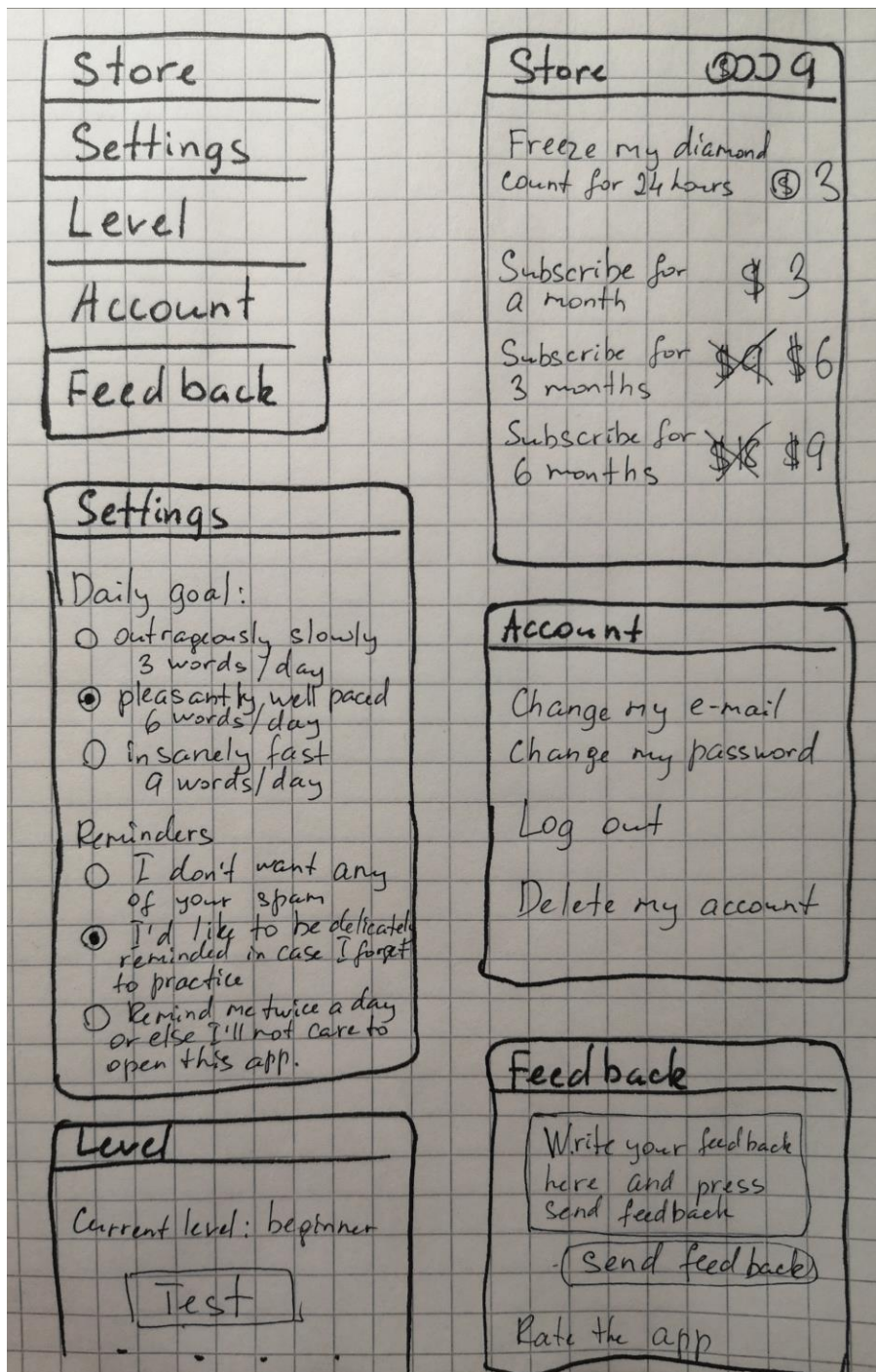
This is LearningActivity. When you enter a Unit, you're presented with 3 words to learn, one by one. You can see how each word is written in Simplified Chinese, can hear its pronunciation, can see its pinyin and translation to the language of your choice. You can take your time to familiarize yourself with the word and click Got It to move forward to the next word or Practice Activity.

Screen 6



These are variations of PracticeActivity. By being randomly presented with different tasks, you get to memorize the 3 words well and comprehensively. You type in the answer and click Next Line. If you make a mistake, you're shown Learning Activity with the word again, and get more tasks with the word you're having a difficulty with.

Screen 7



These are options in Drop-out menu.

Store

Here you can:

- Freeze your diamond count for 24 hours of inactivity, paying 3 app coins;
- Buy subscription for 1, 3 or 6 months.

Settings

In Settings you can:

- Set your daily goal: 3, 6 or 9 words;
- Set how often you want to receive reminders to practice;
- Change the language of translation to English or Russian.

Level

Here you can retest your level if you think the words you are learning are too easy for you.

Account

Here you can:

- Change your email;
- Change your password;
- Log out;
- Delete your account.

Feedback

Here you can:

- Send a feedback to the app owner.
- Be offered to rate the app on Google Play.

Screen 8(Widget)



This is the UI of the Widget. It shows you how many diamonds you've collected this far (one for every day reaching your goal) and how many hours are left before you lose your count if you don't reach your daily goal this time. It also signals whether you're in dangerous zone by color. It turns a shade of red when the time left to is less than 5 hours.

Key Considerations

How will your app handle data persistence?

I will use Cloud Firestore to handle data.

Describe any edge or corner cases in the UX.

If a user passes the whole level test, they're informed that they can use the app in order to review what they already know instead of learning new vocabulary.

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

1. Firebase libraries necessary for using Cloud Firestore for storing and querying information.
2. FlowLayout for Android, which allows child views flow to next row when there is no enough space. I need it for TestingActivity in the app, for choosing the correct word from a range of options.
3. RecyclerView library in order to use a RecyclerView to sort items in MainActivity.
4. Lottie library in order to use some cool animations to give the app more life.

5. Round Corner Progress Bar Library for Android for customizing progress bars.
6. Firebase Authentication for signing users in and providing them with a personalized experience.
7. Google Play Billing library for handling money transactions for paid version of the app.

Describe how you will implement Google Play Services or other external services.

1. Cloud Firestore – following these instructions:
<https://firebase.google.com/docs/firestore/data-model?hl=en>
<https://firebase.google.com/docs/firestore/query-data/get-data?hl=en>
2. Google Play Billing - following instructions here:
https://developer.android.com/google/play/billing/billing_overview
3. Firebase Authentication – following these instructions
<https://firebase.google.com/docs/auth/android/firebaseui?hl=en>.

Next Steps: Required Tasks

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

Task 1: Project Setup

- Create a new project.
- Configure libraries.
- Upload app audios on Cloud Firestore.
- Build a database that will include these two tables and upload them on Cloud Firestore.

vocab

id	hanzi	pinyin	english	russian
1	你好	ni3hao3	hello	здравствуйте

pronunciation

pinyin	sound
ni3	Url1
hao3	Url2

Task 2: Create a class for querying data from **Cloud Firestore**

Task 3: Implement UI for TestingActivity

Build UI for TestingActivity.

- Implement methods for querying information and displaying it;
- Implement methods for checking user answers.
- Set a Progress Bar.
- Set time limit for answering question.

Task 4: Implement UI for MainActivity

Build UI for MainActivity

- Implement methods for querying information about user progress and displaying it; keep the Units the user hasn't reached inactive and shaded gray.
- Add Bottom Navigation Bar for 3 levels of proficiency.
- Customize the app bar which displays how many app coins (words mastered) and diamonds (days of continuous reaching goals) you have.
- Add drop-out menu.
- Set circular progress bars around each Unit icon.

Task 5: Implement UI for LearningActivity, PracticeActivity, StartOrTestActivity

- Build UI for LearningActivity.
- Build UI for PracticeActivity.
- Build UI for StartOrTestActivity.

Task 6: Implement UI for LoginActivity

- Build UI for LoginActivity.
- Implement Firebase Authentication.

Task 7: Implement UI for menu options

Build UI for menu options.

- Build UI for Store option.

- Implement methods for freezing Diamond count.
- Build UI for Settings option (has 2 sets of radio buttons).
- Implement methods for adjusting setting according to User's taste.
- Build UI for Level option.
- Create an intent that takes the user back to TestingActivity when they want to test their knowledge.
- If the testing result is lower than the current level of the user, suggest a choice between staying at the current level and going down to newly tested level. If the test result is higher, automatically move the user to that level.
- Build UI for Account option.
- Implement methods for changing user credentials (username, e-mail, and password).
- Implement methods for logging out and deleting the account.
- Build UI for Feedback option.
- Enable user to send feedback straight from the app to app owner's email.
- Create an intent for rating the app on Google Play Store.

Task 8: Set notifications

- Create notifications to remind User to study.
- Make sure they can be set according to User's preferences in Settings.

Task 9: Implement UI for Widget

- Build UI for Widget.

Task 10: Make sure the app is robust

- Add a UI for when there is no Internet connection.
- Make sure the app saves state whenever the device is rotated.
- Make sure the app doesn't crash under any circumstances.

Task 11: Add free and paid flavors

- In free version, allow User to pass 3 sublevels for free. The sublevels below and above stay inactive.
- In paid version, give User unlimited access to all sublevels they have reached, but for a limited time period (1, 3 or 6 months).

Task 12: Customize and polish the app UIs

- Customize UIs according to the size of device.
- Make sure the app follows Material Design recommendations and looks pleasing.

Task 13: Publish the app on Google Play

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
 - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"