

# Youzi: Generative AI for Social Mandarin Learning

Joanne Eunjung Jun, Victor Kuan, Jo Lu, Harris (Chun Yin) Wang, Tyler Wu

Master of Information Management and Systems (MIMS)  
Info 298A: Directed Group Work on Final Project

Dr. Kimiko Ryokai

May 1st, 2024

## Abstract

American-born Chinese (i.e. ABC's) in their early 20's aim to practice Mandarin speaking but feel anxious due to shyness or a feeling of cultural disconnect. We aim to assist them in overcoming these barriers. Our approach involves using generative AI to foster conversations and create bonding with their speaking partners, ranging from family elders, new friends, or even romantic partners. Ultimately, our goal is to empower these young adults to speak Mandarin comfortably and feel connected with their Chinese heritage, thereby strengthening community bonds among fellow Chinese speakers.

## Introduction

Contextual testimonial from Victor Kuan:

*"Hi! I'm Victor Kuan, and I am interested in collaborating with some fellow MIMS people to use UX research, innovative interaction design, and iterative product development skills to explore innovative new ways to better teach Mandarin to American-born Chinese (ABC) heritage speakers.*

*Growing up as an American-born Chinese person, I have noticed that there is a significant population of ABC kids who are encouraged by their parents to learn Mandarin writing, reading, speaking. Plenty of these kids express dismay, annoyance, and unhappiness when facing these expectations. Paradoxically, as they grow up into young adults, they often express regret for not spending more time and energy studying Mandarin. I am passionate about addressing some of these problems, because I think it is important to inspire kids and young adults to love learning Mandarin, since it's quite useful to know how to communicate fluently in multiple languages!"*

As mentioned in the testimonial above, with this capstone, we aim to:

1. Study this phenomenon
2. Present some potential solutions and ideas which aim to address the issues associated with this phenomenon
3. Create some mockups visualizing and detailing the most promising ideas
4. Iteratively develop our favorite proposed solutions

To provide supplemental research for this capstone project, we created a directed group study class, supervised by Professor Marti Hearst, called "Rethinking Chinese Language Education for ABCs". Within the context of that course, we came up with a report showcasing findings from existing research literature on the successes and failures of Mandarin language education for

ABC's in addition to a report detailing our survey and user research findings that explain how second-generation Bay Area ABC young adults feel about their experience, both from their childhoods and also their recent lives, learning Mandarin in America.

To briefly summarize our findings from the directed group study course, we learned the following:

- The aspects of Mandarin language education, which seem to be working well for ABC's:
  - Immersion is extra impactful; ABC's learn more quickly and effectively when they work or live in a significantly Mandarin-speaking environment, where they will be forced to chat in Chinese until they eventually become comfortable with it.
  - Mandarin lessons, which employ very visually exciting, memorable, and weird Chinese cultural elements are more impactful.
  - Idioms and slangs are cool; Chinese idioms come with exciting explanations, contexts, and stories, and ABC's tend to learn a lot of memorable vocabulary, phrases, and cultural concepts when learning a Chinese idiom.
  - Memes are extra impactful; ABC's learn a lot when sharing funny Chinese memes or cultural viral phenomena with each other on social media or in person.
- Aspects of Mandarin language education which seem challenging for ABC's:
  - It is still very difficult for most ABC's to effectively practice chatting about complex topics like politics, socioeconomics, or academia since these topics require the usage of really advanced vocabulary.
  - On that note, it is quite difficult for most ABC's to remember advanced vocab words because they do not use those words frequently enough.
  - Most ABC's report feeling anxious about speaking Mandarin to people who are more skilled at speaking.
  - On the same note, they report feeling horrible or insecure when trying to speak Mandarin with native speakers, who then respond in English.

So, given these findings, we decided to ideate for our capstone project with the following considerations in mind:

- We wanted to think profoundly about the things that ABCs struggle the most with:
  - Vocab learning
  - Finding the courage to speak with native speakers
  - Chatting about complex topics like politics, corporate life, or academia
- We wanted to design and ideate with the goal of addressing those struggles

- We wanted to employ lessons from some of the successful elements referenced above as we ideate:
  - Use memes
  - Use idioms and slang
  - Reference very visually interesting, shocking, funny, or weird cultural elements
  - Think about clever ways to utilize and leverage social networks

With these ideological frameworks in mind, we started to design and build Youzi.

## Product

### Product Overview

Youzi is an app designed to enhance Mandarin language learning through engaging, interactive features and personalized content. The app facilitates user engagement with various functionalities tailored to different learning preferences and levels.

### Specifications

#### 1. Onboarding Screen

- a. User skills and interests profile
  - i. Create user profiles based on interest selections and language proficiency
  - ii. Division → backend: data processing and profile creation
- b. User avatar
  - i. Generate customizable avatars based on user profiles for gamification
  - ii. Division → backend: data processing and avatar generation
- c. User account setup
  - i. Assist users in creating secure and unique accounts, including username, password, and avatar assignment
  - ii. Division → backend: data storage and security; frontend: user input management

#### 2. Vibe Selection Screen

- a. Animated transition
  - i. Enhance user experience with animations transitioning into the learning interface upon vibe selection

- ii. Division → product: finalize vibe categories; UX / UI: design and create icons / illustrations; frontend: implement animations and transitions

### 3. Prompt Screen

- a. Pinyin generation
  - i. Automatically generate Pinyin for Mandarin prompts to assist with pronunciation
  - ii. Division → backend: hanzi (i.e. Chinese character) processing; frontend: character display
- b. Pinyin mapping
  - i. Display pinyin in alignment with the corresponding hanzi
  - ii. Division → backend: mapping processing; frontend: pinyin display
- c. Mandarin text-to-speech
  - i. Provide auditory representation of Mandarin text for pronunciation help
  - ii. Division → backend: audio generation; frontend: playback management
- d. Prompt regeneration
  - i. Dynamically offer users new prompts
  - ii. Division → backend: logic and data management

### 4. Post-Prompt Screen

- a. Recording
  - i. Allow users to record their responses with functionalities like pause, resume, and marking highlights
  - ii. Division → backend: data processing and storage; frontend: interface management
- b. Smart suggestion
  - i. Provide contextually relevant conversation suggestions based on user inputs and interactions
  - ii. Division → backend: suggestion algorithms; frontend: suggestion display
- c. Review screen
  - i. Enable users to sort practice sessions by difficulty or completion time
  - ii. Division → backend: data sorting and retrieval; frontend - interface management

## Marketing

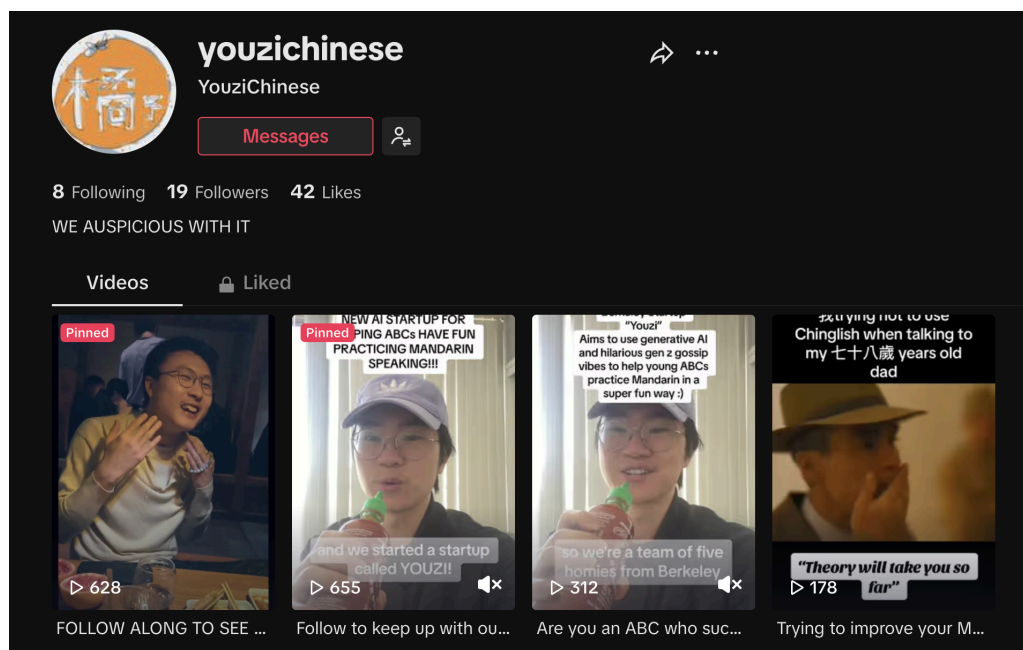
From a very early stage of our development process, we started thinking about marketing since we would like to eventually get our product into the hands of young ABC's.

So, we thought about our product marketing strategy and decided that our marketing and branding need to meet these requirements:

- Be funny and relatable for Gen-Z ABC's
- Have the potential for virality, which comes in the form of integrating humor and absurdity while considering strong SEO
- Be accessible for Gen-Z ABC's through TikTok, Instagram, and LinkedIn

Following these guidelines, we built a product marketing campaign roadmap and started creating funny TikTok videos. The TikTok videos feature our Youzi team members, funny Mandarin and English memes, skit concepts, and more.

Though we are at a very primitive stage with our marketing efforts, we are still proud to share our TikTok page here: <https://www.tiktok.com/@youzichinese>



We also spent some time developing a company web page to professionally promote our product and provide information for interested potential users and interested stakeholders:

<http://youzi.life>

## Next Steps For Marketing

We recognize that our TikTok presence at this point is rather meager, given our limited time and resources to make viral videos. We would like to continue fleshing out this marketing campaign.

Moving forward, referencing our product marketing roadmap, we would like to create the following inbound marketing videos such as:

- Educational skits, which teach our viewers about funny Mandarin slang and idioms
- Educational skits, which teach our viewers about the do's and do not's of Mandarin speaking
- Day-in-the-life style videos, which absurdly and humorously showcase the stressful and fun product development process that we go through as a team at Youzi
- Funny interview-style videos, where we interview Berkeley students (and other populations of Gen-Z) and ask them funny questions about their Mandarin language education in addition to their understanding of certain Mandarin slang

Since our marketing strategy is not designed specifically for TikTok, we can take the marketing abstractions and apply them to Instagram Reels and YouTube Shorts in order to multiply our reach.

## Timeline

Week	Date	Milestones	Key Activities	Deliverables
1	01/15/24	Initial concept discussion	<ul style="list-style-type: none"> <li>● Discussion on Victor and Tyler's directed group study conclusions</li> <li>● Preliminary ideation for app functionalities</li> </ul>	<ul style="list-style-type: none"> <li>● Initial concept validation</li> <li>● Role assignments</li> <li>● Project timeline establishment</li> </ul>
1	01/19/24	User persona & problem definition	<ul style="list-style-type: none"> <li>● Development of user personas based on preliminary research</li> <li>● Definition of main user problems</li> </ul>	<ul style="list-style-type: none"> <li>● User personas</li> <li>● Prioritized problem statements</li> </ul>

2	01/22/24	Ideation & solution selection	<ul style="list-style-type: none"> <li>Brainstorming session for potential solutions and feature ideas</li> <li>Initial selection of feasible solutions</li> </ul>	<ul style="list-style-type: none"> <li>List of potential features</li> <li>Selected solutions for further development</li> </ul>
2	01/25/24	Role delegation & infrastructure review	<ul style="list-style-type: none"> <li>Delegation of specific roles for project</li> <li>Discussion on backend infrastructure and audio / video capabilities</li> </ul>	<ul style="list-style-type: none"> <li>Role delegation confirmation</li> <li>Initial backend infrastructure plan</li> </ul>
3	01/31/24	UX design & timelines	<ul style="list-style-type: none"> <li>Discussion on low-fidelity design completion, timelines, and delegation for mid-fidelity designs</li> </ul>	<ul style="list-style-type: none"> <li>Completed low-fidelity designs</li> <li>Updated project timelines</li> <li>Delegation of tasks for mid-fidelity design phase</li> </ul>
4	02/05/24	Mid-fidelity milestone	<ul style="list-style-type: none"> <li>Shareout of current design status</li> <li>Critique and feedback session</li> </ul>	<ul style="list-style-type: none"> <li>Mid-fidelity design shareout</li> <li>Feedback incorporation into development</li> </ul>
4	02/09/24	Mid-fidelity design wrap-up	<ul style="list-style-type: none"> <li>Final discussions on mid-fidelity design adjustments</li> <li>Preparation for high-fidelity design</li> </ul>	<ul style="list-style-type: none"> <li>Finalized mid-fidelity designs</li> <li>Readiness assessment for high-fidelity phase</li> </ul>
6	02/22/24	Post-Skydeck check-In	<ul style="list-style-type: none"> <li>Review of current project status post-presentation</li> <li>Discussions on MVP and timeline adjustments</li> </ul>	<ul style="list-style-type: none"> <li>Updated project timeline</li> <li>MVP feature set confirmation</li> </ul>



7	02/28/24	UX refinement	<ul style="list-style-type: none"> <li>● UX iteration discussions</li> <li>● Finalization of high-fidelity mockups</li> </ul>	<ul style="list-style-type: none"> <li>● High-fidelity mockups</li> <li>● UX improvement plans</li> </ul>
7	03/01/24	Engineering review	<ul style="list-style-type: none"> <li>● Discussion on engineering progress</li> <li>● Backend / frontend integration</li> </ul>	<ul style="list-style-type: none"> <li>● Integrated product specifications</li> <li>● Engineering progress report</li> </ul>
8	03/06/24	UX feedback with advisor	<ul style="list-style-type: none"> <li>● Presentation of high-fidelity mockups to advisor</li> <li>● Collection of feedback on UX and product feasibility</li> </ul>	<ul style="list-style-type: none"> <li>● Advisor feedback incorporation</li> <li>● Final adjustments to UX design and engineering timeline</li> </ul>
9	03/13/24	User need re-alignment	<ul style="list-style-type: none"> <li>● Reassess and modify UX design and engineering plans based on user feedback</li> </ul>	<ul style="list-style-type: none"> <li>● Updated user need report</li> <li>● Revised UX design</li> </ul>
13	04/12/24	Final stretch planning	<ul style="list-style-type: none"> <li>● Outline final development and testing phases</li> <li>● Establish internal testing timelines</li> </ul>	<ul style="list-style-type: none"> <li>● Detailed final phase plan and internal testing schedule</li> </ul>
16	04/29/24	All hands final stretch	<ul style="list-style-type: none"> <li>● Conduct final reviews and make necessary adjustments before presentation</li> </ul>	<ul style="list-style-type: none"> <li>● Comprehensive launch package <ul style="list-style-type: none"> <li>○ Presentation deck</li> <li>○ Product website</li> <li>○ Fully-engineered demo</li> <li>○ UX prototype</li> <li>○ Poster</li> </ul> </li> </ul>

16	05/01/24	All hands submission review	<ul style="list-style-type: none"> <li>● Perform final presentation rehearsal</li> </ul>	<ul style="list-style-type: none"> <li>● Ready-to-go presentation rehearsal</li> </ul>
----	----------	-----------------------------	--	--

## User Research

### Goals

The research was conducted through in-depth qualitative interviews with participants, identifying as American-born Chinese (ABC's), who were actively learning Mandarin. This study builds upon the work of a previously conducted directed group study and aims to delve deeper by pinpointing specific user needs within ABC's. Our main focus during these interviews was to understand each participant's: relationship with the language, learning processes, and the resulting impacts on their social and familial interactions.

### Research Methodology

- Type of study: qualitative user interviews
- Setting: In-person for engaging dialogue
- Duration: approximately 1 hour per session
- Data collection tools: scripted questions recorded on Zoom to ensure consistency and allow for detailed review; slide presentation of a prototype consisting of conversation prompts
- Data analysis: focused on pinpointing specific user needs and challenges, including feedback on feasibility of prototype
- Incentives: participants received a refreshment of their choice after the interview

### Participant Demographics

Three American-born Chinese individuals from UC Berkeley's Mandarin learning program participated in the study, each with distinct backgrounds and levels of Mandarin proficiency:

- C: 19 years old; comfortably speaks and reads Mandarin; motivated by wanting to revisit Mandarin after briefly learning in childhood
- M: 22 years old; possesses basic Mandarin skills; motivated by a desire to travel
- J: 22 years old; more proficient in reading / writing Mandarin than speaking it; motivated by wanting to better engage in Mandarin-speaking communities

## Interview Questions

The following are the primary questions we posed during the interviews, complemented by more detailed, curated follow-up questions to delve deeper into each topic:

1. Can you describe your typical daily interactions in Mandarin?
2. Which topics do you find easy or challenging to discuss in Mandarin?
3. How do you practice Mandarin outside of formal educational settings?
4. What tools or resources have been most effective in aiding your Mandarin learning?
5. Could you share an experience where you successfully utilized Mandarin in a practical situation?

## Prototype: Example Prompts

After being asked the questions above, we asked each participant to answer the conversation prompts below and provide feedback.

### Part 1: Ice Breakin' & Liáo Tiān

- Mandarin: 你有没有和大学暗恋的人有趣的回忆?
- English: Do you have any funny memories with your college crush?
- Media suggestion: "My Love from the Star" for college-related crush stories

### Part 2: ABC Trauma Bonding

- Chinese: 如果你能回到大学, 你会怎么做?
- English: If you could go back to college, what would you do differently?
- Media suggestion: "Suddenly Seventeen" for a fun take on time travel and youthful regrets

### Part 3: Spill the Hàì Xiū Tea

- Chinese: 你有没有给老师起过绰号?
- English: Have you ever made up a nickname for a teacher?

- Media suggestion: "Our Shining Days" for a humorous look at the relationship between students and teachers

## Findings and Analysis

### Insights from user interviews

- Conversation initiation: higher-level Mandarin speakers are more likely to initiate conversations
- Confidence building: discussing familiar topics like family and friends in Mandarin helps build confidence
- Language integration: speaking Mandarin is usually not a part of normal life—participants have to turn on "language learning mode" to practice it
- Storytelling and conversation flows: participants appreciate guidance on how to craft coherent stories and conversations in Mandarin

### User struggles

- Complex concepts: discussing abstract ideas like unrequited love is difficult
- Humor and expression: using humor, conciseness, and exaggeration in Mandarin is a struggle
- Confidence issues: participants have a general lack of confidence in using Mandarin in various social settings
- Reading vs. speaking: participants have better speaking and listening skills, compared to reading and writing

### Opportunities for improvement

- Contextual learning: it is important to bridge the gap between textbook learning and real-life use of Mandarin
- Content presentation: content should start with easier, familiar topics and gradually introduce more complex idea
- Language tools: our users would benefit from tools, like a dictionary modal or Grammarly-style language aids
- Media integration: incorporating media recommendations would enhance engagement and learning

### Proposed product direction

- Gamification and engagement: gamification elements can make learning more interactive and less forced

- Customized learning paths: we should allow users to curate their vocab lists and choose content based on their interests, with options to select specific media for engagement
- Enhanced support: the app should provide support tools that help users decipher prompts and engage in contextual learning

## Conclusion

The feedback from participants highlights the need for Mandarin learning tools that integrate seamlessly into their daily lives, providing not only language practice but also cultural engagement. Our findings point to the conclusion that enhancing user confidence, contextual learning, and personalized content will likely significantly improve the language learning experience for ABC's.

## UX Design

### Iterations

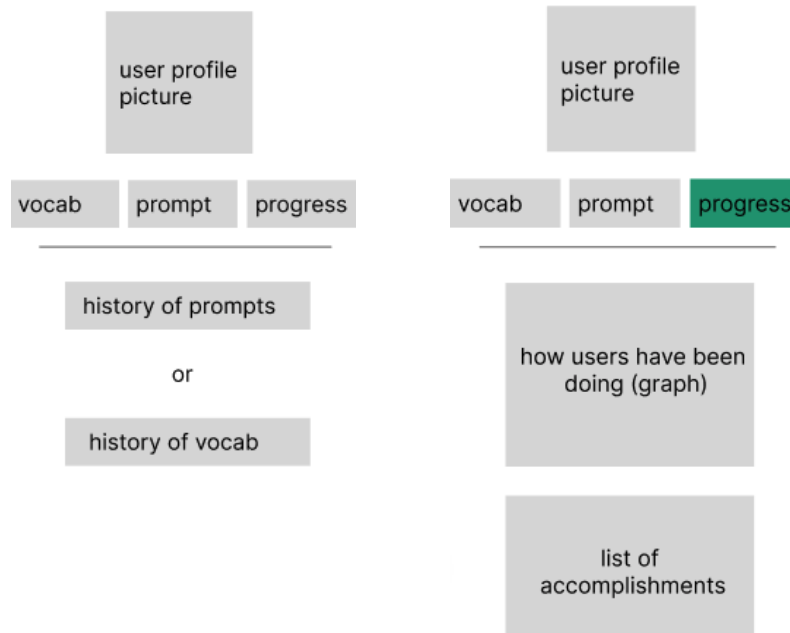
The design of Youzi went through three main stages: low, mid, and high-fidelity designs created through Figma. Before beginning work in Figma, our group first ideated by gathering sketches from every member of the group and then conducting an internal critique session to settle upon the most feasible and promising designs from the initial sketches.

## Low-Fidelity

The low-fidelity design focused on the overall architecture and flow of the app. We emphasized this by choosing to use wireframes and a greyscale color scheme. Some of the key design features that were translated from the sketches into the initial low-fidelity iteration were the conversational “vibes”, the option to audio record responses, as well as a transcript of the user’s response.



Internally, we also discussed implementing a review page for users to look at their progress over time. At this stage, we envisioned that including metrics, such as past learned vocabularies, accomplishment analytics, and achievement markers would be helpful to our users. We drafted some preliminary sketches of these features during the low-fidelity stage as shown below.



## Mid-Fidelity

The mid-fidelity design is where our group first introduced a tentative color scheme and typography. Some of the key words informing our choice in colors and fonts were: “playful”, “bubbly”, and “lightweight”.

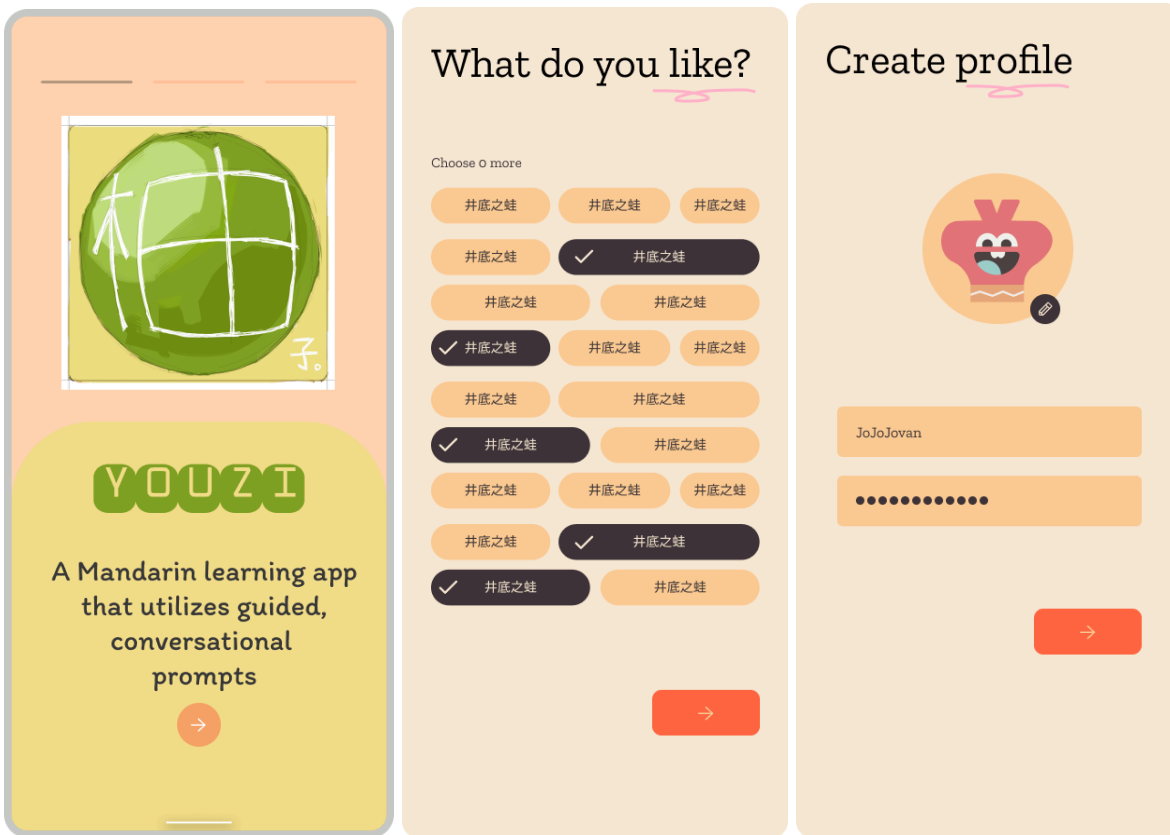


After selecting initial colors and fonts, our group also chose to divide the app into distinct phases that the user would encounter. More specifically, these can be described as: “onboarding”, “pre-prompt”, “post-prompt”, and “review”.



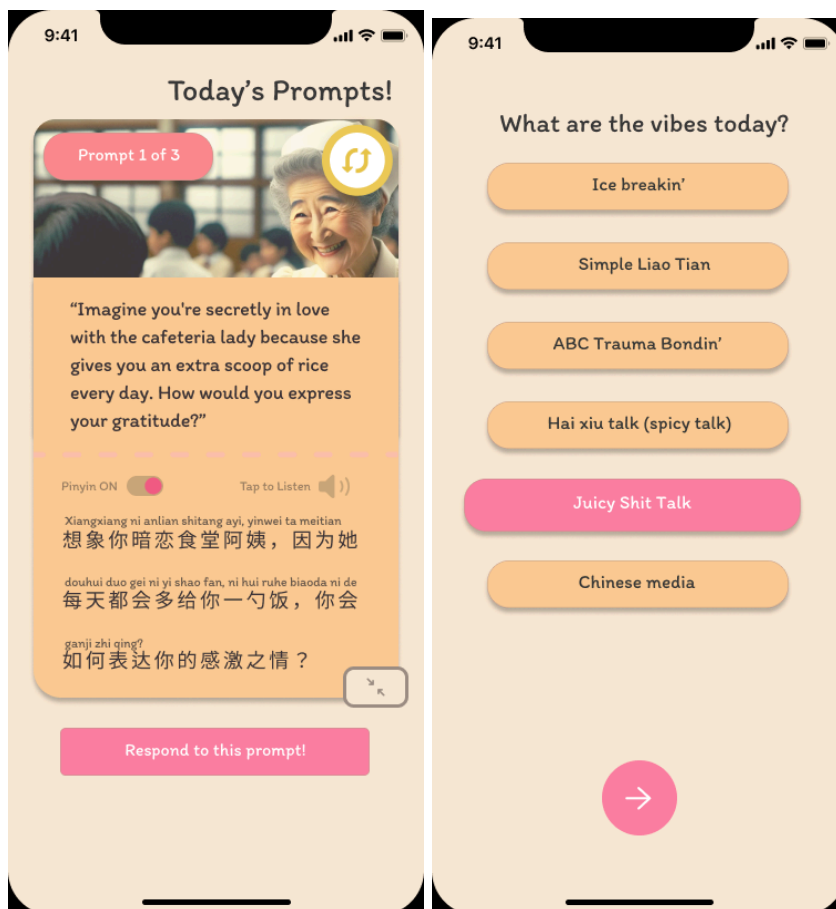
### Onboarding

Onboarding refers to when the user opens the app for the first time and is greeted by a loading screen. After the app fully loads, the user is taken through a short introduction of the app before being asked to choose some relevant interests and create a user profile.



## Pre-Prompt

The pre-prompt phase is where the user chooses what kind of conversation they would like to engage in for the day. The user is shown the randomly chosen conversational prompt, along with other relevant information, such as an accompanying translation and associated pinyin.



## Post-Prompt

The post-prompt phase is where the user can actively engage with the prompt by recording their response, learning new vocabulary related to the current prompt, or interacting with an in-app dictionary that provides more context.



Some of the key changes that we made from the low-fidelity version after conducting another internal feedback session was to increase the number of vibes to introduce more granularity and give the user ownership over their language journey.

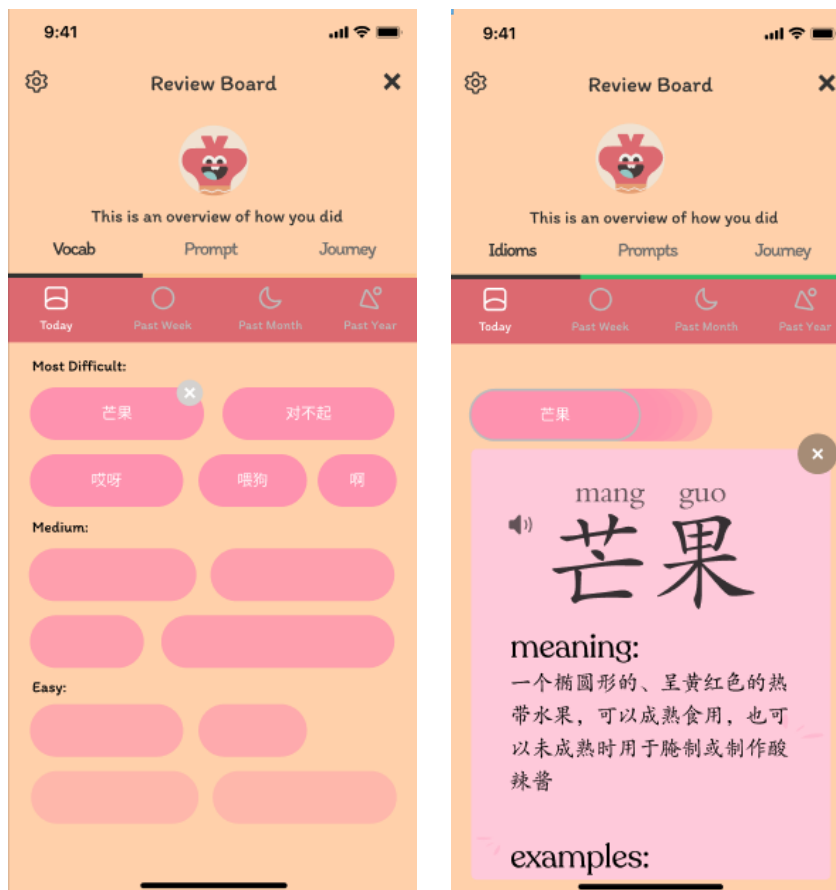
In addition, instead of having a completely separate "dictionary mode", we instead changed the design to have the dictionary feature be built directly into the post-prompt flow in order to allow users to look up words more quickly and easily, increasing usability.

## Review

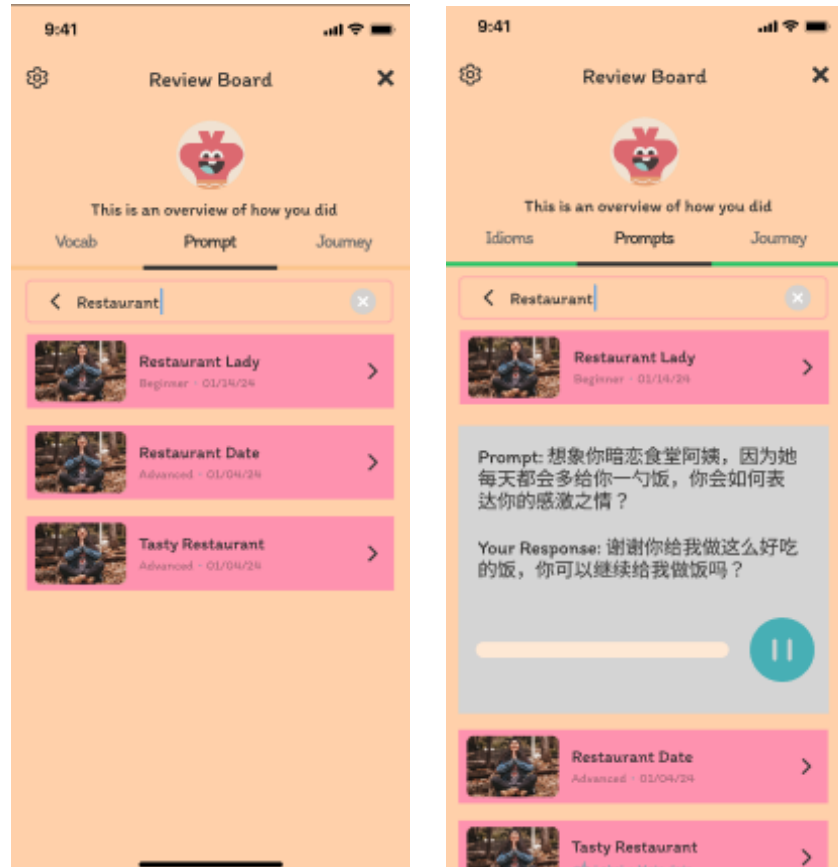
For the review phase, we decided to experiment with the design by incorporating more pink / light red into the color scheme and emphasizing the contrast between the modules, denoted with buttons and tabs, and the background. At this stage, it is crucial to note that we formed a more consolidated vision of the main features we attempted to include during the mid-fidelity stage. In general, we would like to provide the user two main review features for them to navigate through:

1. Users could view the **vocabulary** they saved in past prompts, sorted by level of difficulty. During this stage we considered additional features, such as allowing users to collapse / expand each vocabulary tab or having the option to remove a word from the list.

To make the design consistent between the prompt and review screens, we used a similar vocabulary modal as the one in the prompt screen. The modal includes an option to play the audio for the pronunciation of the vocabulary.

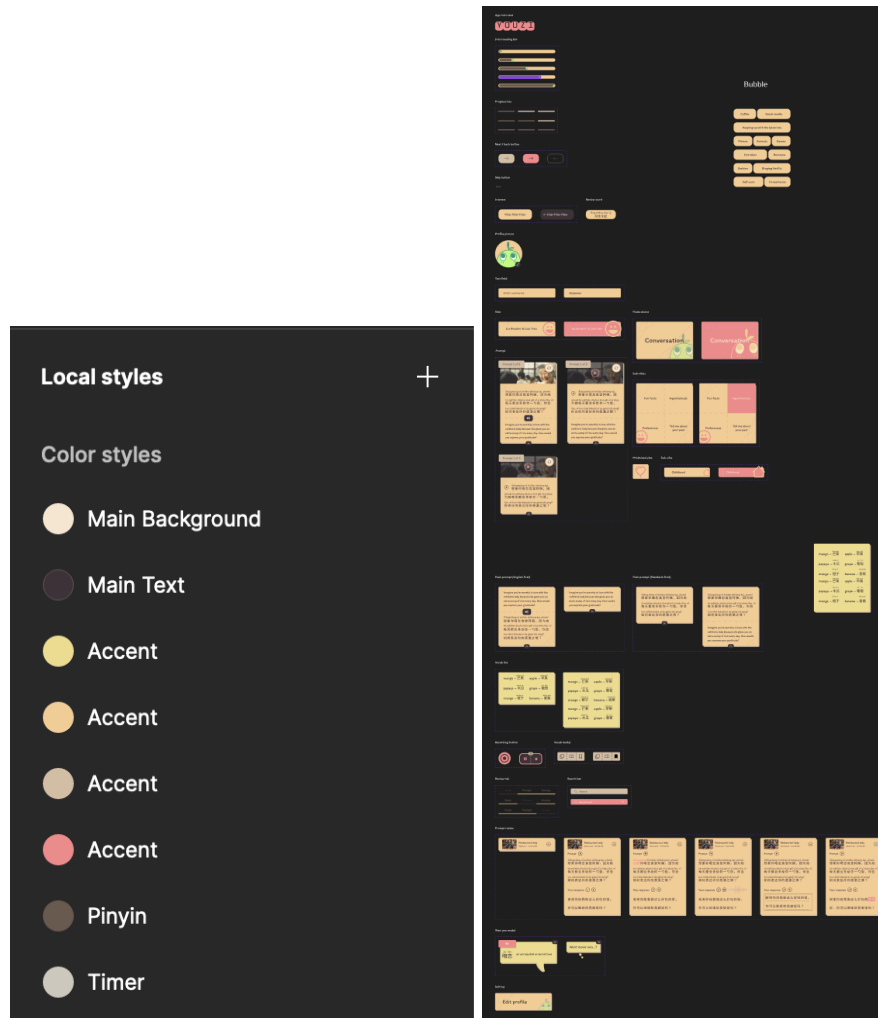


- We also wanted our app to **archive** all the past **prompts** users have completed to allow users to review their recordings and transcript. The search bar allows users to input keywords to find specific past prompts easily.



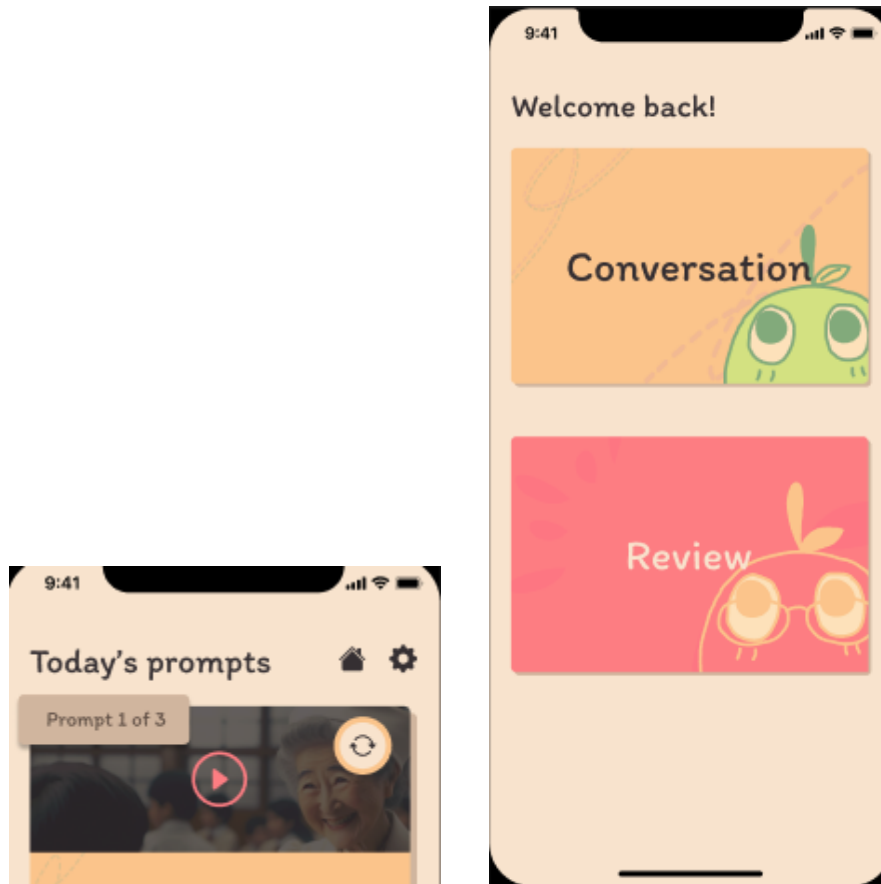
## High-Fidelity

Finally, the high-fidelity design is the stage where our group focused on creating a design system with a final color scheme, easily usable design components, and a truly interactive prototype.

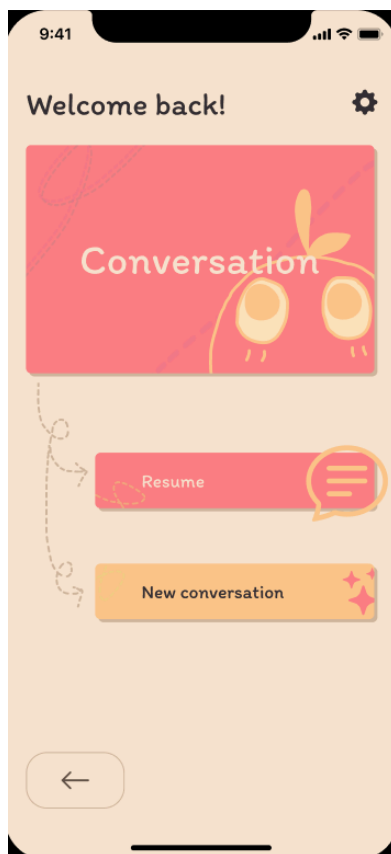


While much of the overall flow stayed the same, we made significant improvements to certain screens, namely the transition screens, vibe selection, pre-prompt, and post-prompt screens.

In our previous iterations, each module was disjointed from each other. In order to create a smooth user flow, we added a hub page, accessible through the home button in the upper right corner, where users could freely switch between conversation mode and review mode.

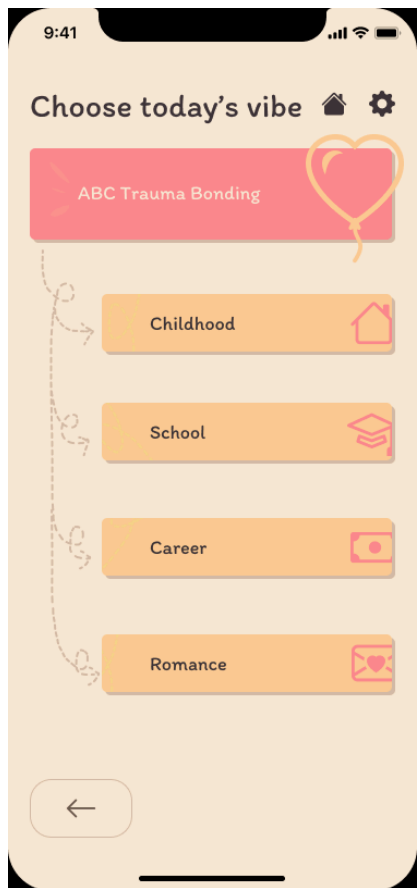


When users select conversation mode, they also have the option to either start a new prompt, or the option to resume a previous unfinished prompt.

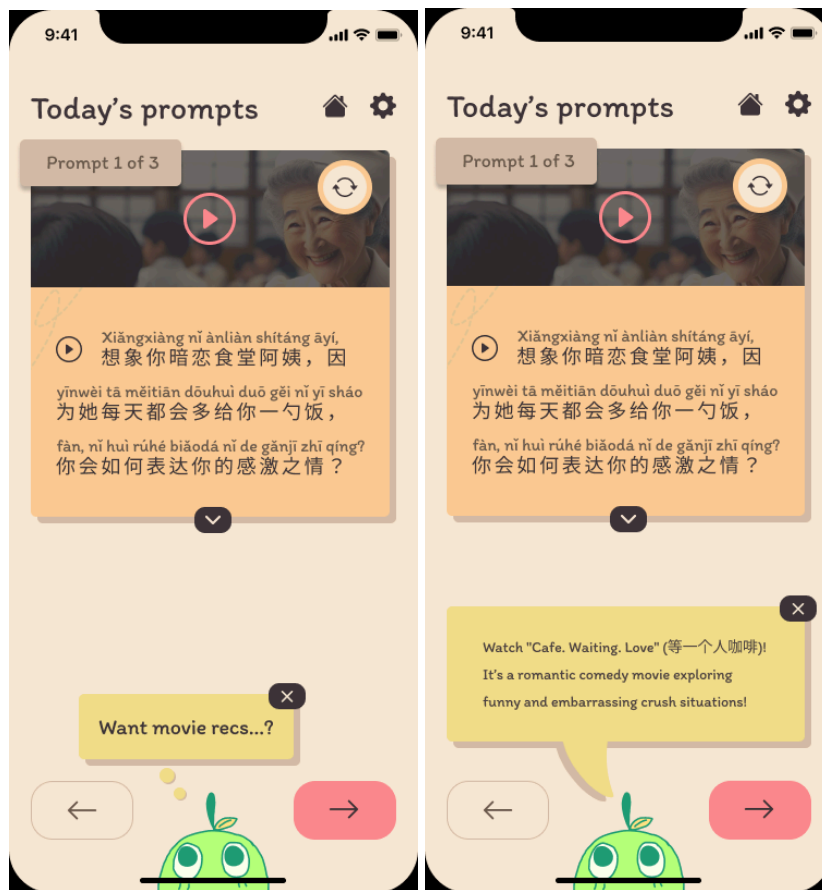




The previously discussed user interviews helped our team realize that there are some prompts that our target user group would be unwilling to discuss with certain members of their social circle, such as discussing romance with their parents. Thus, in order to give the user even more control of their experience within the app, our team made a major design change to introduce “sub-vibes”, or in other words, subcategories for each main vibe.



Next, another significant change after receiving the feedback that our app felt too safe was to introduce an interactive “Clippy”-esque mascot called Xiao-you that would live at the bottom of the app. Xiao-you’s purpose is to introduce a sense of liveliness to the app by allowing users to interact with it, such as providing media recommendations when tapped on or providing reactions to certain actions that users perform within the app.



Finally, one of the last major changes that our team decided upon was to introduce a vocabulary modal that could be triggered by long-pressing any vocabulary on the prompt screen. This feature was introduced to address the fact that users would need multiple ways to interact with a vocab, such as copying it, viewing the English meaning, or even saving it for later review. The dictionary modal was also further simplified in order to make the app feel less cluttered.



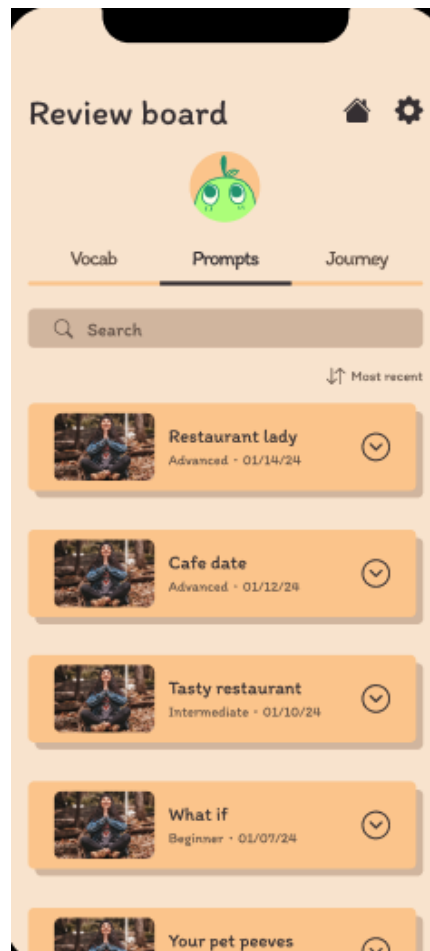
## Review

We made the following changes to the vocab and review tabs to increase usability and align overall design scheme with the other parts of the app:

1. Under the vocab tab, we added pinyin to each vocabulary word to allow users to easily recall the pronunciation without having to display the dictionary modal. We also added the word counts for each difficulty level to provide users with more information of how many words they have saved without having to press the three dots to view all the vocabulary words at once.



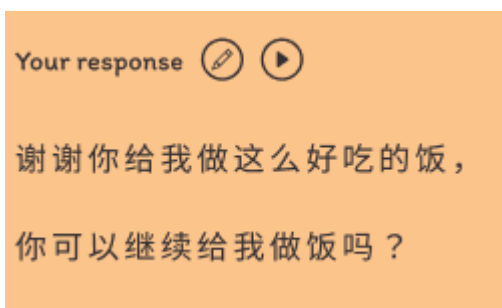
- Under the prompt tab, we made certain elements more prominent, such as changing the color of the search bar into pink and outlining the collapse / expand chevrons with circles. We also added a filtering button underneath the search bar for users to sort the content based on how recently they were completed.



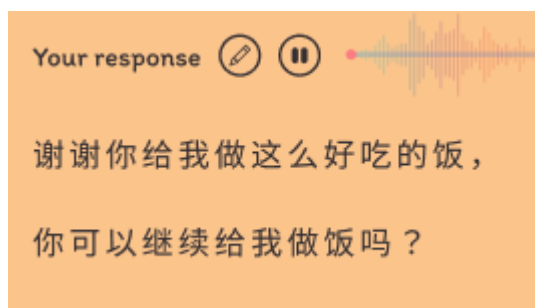
Some key upgrades were made to the expanded version of the past prompts. First, when the user plays the prompt, they can see the characters and their corresponding pinyin being highlighted in synch with the audio.



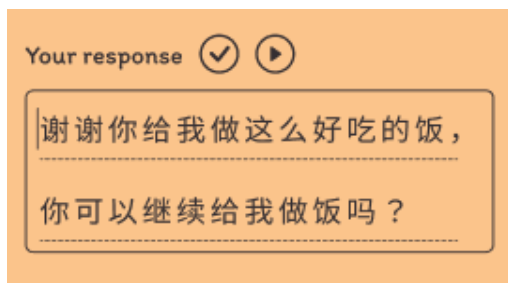
Additionally, the "Your response" feature has been enhanced to allow users to not only playback their responses but to also edit them if they identify any errors. Any user input will be highlighted in pink, allowing users to easily identify which parts of the transcript were edited.



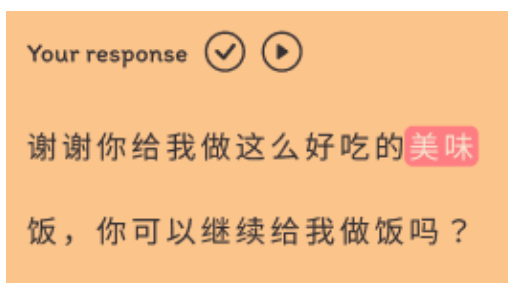
Past response



Audio recording of past response



Response editing



New input is highlighted

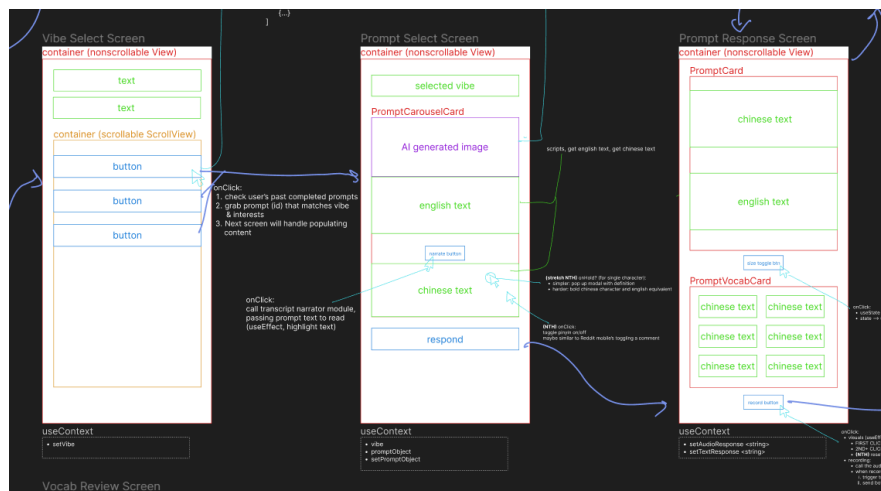
## Frontend

### Tech Stack

The frontend of the app was built using React Native and Expo-Go framework in VS code. The codebase can be found on [Github](#).

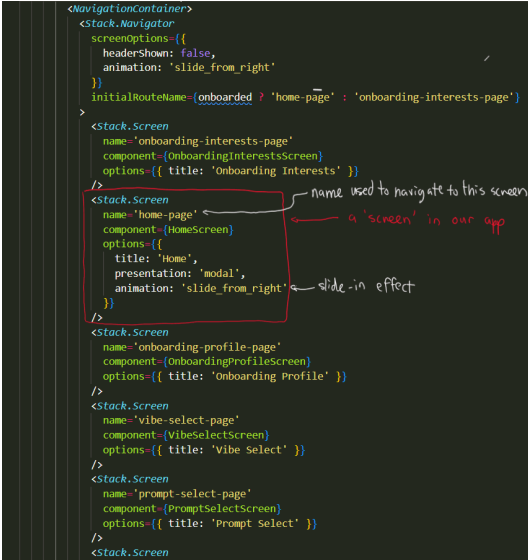
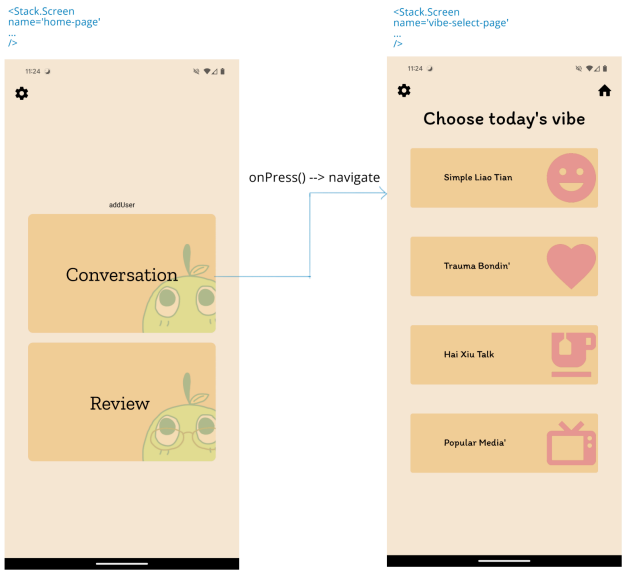
### Screens and Navigation

The first step of creating the flow of the app was constructing templates of each screen and connecting their flow using [react-native-navigation](#). We created a frontend [low-fidelity](#) design in Figma to help us map out the screens and components necessary.



**Example sequence of screens from the prompt-conversation flow**

This involved creating a navigation stack using [react-navigation/native](#) in App.js, that defined the names and properties of each screen, as well as a series of buttons for each page that defined the navigation flow.

<p>Screen (navigation-stack) structure</p>	 <pre> &lt;NavigationContainer&gt;   &lt;Stack.Navigator     screenOptions={{       headerShown: false,       animation: 'slide_from_right'     }}     initialRouteName={onboarded ? 'home-page' : 'onboarding-interests-page'}   &gt;     &lt;Stack.Screen       name="onboarding-interests-page"       component={OnboardingInterestsScreen}       options={{ title: 'Onboarding Interests' }}     /&gt;     &lt;Stack.Screen       name="home-page"       component={HomeScreen}       options={{         title: 'Home',         presentation: 'modal',         animation: 'slide_from_right'       }}     /&gt;     &lt;Stack.Screen       name="onboarding-profile-page"       component={OnboardingProfileScreen}       options={{ title: 'Onboarding Profile' }}     /&gt;     &lt;Stack.Screen       name="vibe-select-page"       component={VibeSelectScreen}       options={{ title: 'Vibe Select' }}     /&gt;     &lt;Stack.Screen       name="prompt-select-page"       component={PromptSelectScreen}       options={{ title: 'Prompt Select' }}     /&gt;     &lt;Stack.Screen   </pre>
<p>Navigation button structure + navigation logic</p>	<p>See next page</p> <div style="display: flex; justify-content: space-around;"> <div data-bbox="500 865 863 1102"> <p>Navigation Button Structure</p> <pre> &lt;TouchableOpacity   style={styles.homeButton}   onPress={() =&gt; {     navigateToMode(mode)   }}   {...props} &gt;   </pre> </div> <div data-bbox="966 865 1404 1102"> <p>Navigation logic</p> <pre> const navigation = useNavigation(); const navigateToMode = (mode) =&gt; {   if (mode == 'prompt') {     navigation.navigate('vibe-select-page');   }   else if (mode == 'review') {     navigation.navigate('review-mode-page');   } }   </pre> </div> </div>
<p>Screen &amp; button appearance</p>	



## Onboarding Flow

We did not have time to make an onboarding flow; however, it would ideally be a flow where users can create their profile, set their learning level, goals, and toggle settings (e.g. explicit prompts, traditional vs. simplified hanzi characters, and turning pinyin on or off).

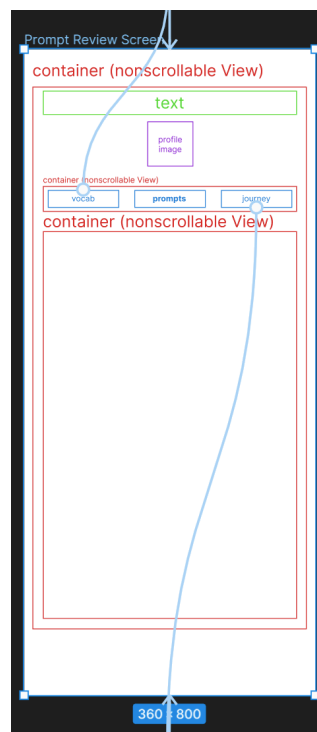
## Conversation Flow

The conversation flow is a very linear flow. The screens essentially follow a series of buttons until the user exits or completes the conversation.

## Review Flow

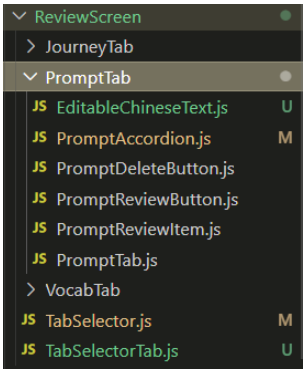
The review flow is a much more “shallow” flow since there are several modes of review, and many of the screens were actually nested within each other.

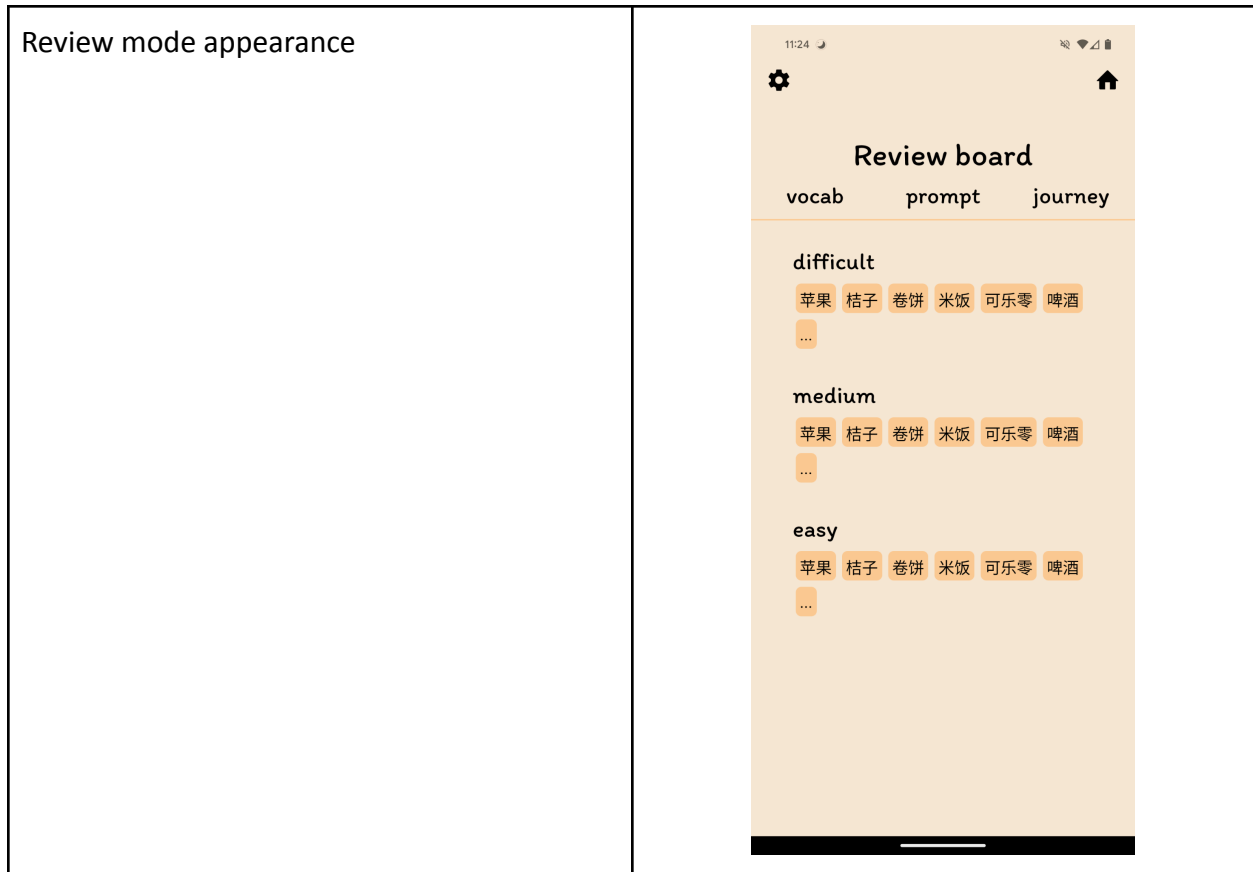
This prompted us to create a “tab/sub-screen” layout, much like the tabs in a web browser, such that pressing one of the review mode tabs would simply update the “tab” rather than navigate to an entirely separate screen.



***Design of flow between tabs***

After mapping out the design, we decided to break down the component structure so that each tab is effectively its own screen component, and the navigation is handled in TabSelector.js.

<p>Review screen folder structure</p>	
<p>Dynamic tab logic</p>	<p>Buttons for switching tabs</p> <pre data-bbox="824 783 1281 1052"> export default function TabSelector() {   const {     review_tabs,     activeTab,     setActiveTab   } = useContext(ReviewContext);    const handleTabSwitch = (page) =&gt; {     setActiveTab(page);   };    return (     &lt;View style={styles.tabSelector}&gt;       {review_tabs.slice(0, 2).map((page, i) =&gt; {         // {review_tabs.map((page, i) =&gt; {         return &lt;TabSelectorTab index={i} page={page} onPressFn={handleTabSwitch} /&gt;       })}     &lt;/View&gt;   ); } </pre> <p>Dynamic tab (renders based on activeTab value)</p> <pre data-bbox="824 1129 1287 1497"> return (   &lt;ReviewContext.Provider value={{     review_tabs,     activeTab,     setActiveTab   }}&gt;     &lt;View style={[youziStyles.centeredView, styles.reviewView]}&gt;       &lt;Text style={youziStyles.headerText}&gt;Review board&lt;/Text&gt;       { /* &lt;ProfileImage /&gt; */ }       { /* on tab click, switch current tab */ }       &lt;TabSelector /&gt;       &lt;View style={styles.tabView}&gt;         {activeTab == review_tabs[0] &amp;&amp; &lt;VocabTab /&gt;}         {activeTab == review_tabs[1] &amp;&amp; &lt;PromptTab /&gt;}         {activeTab == review_tabs[2] &amp;&amp; &lt;JourneyTab /&gt;}       &lt;/View&gt;       &lt;HomeButton /&gt;       &lt;SettingsButton /&gt;     &lt;/View &gt;   &lt;/ReviewContext.Provider&gt; ) </pre>



## Universal Navigation Buttons

For ease of use for the user, we also opted to create some universal navigation buttons: home and settings. Both are straightforward and navigate the user back to the respective page.

## Text



The text in our app provided a unique challenge in that it needed to satisfy the following properties:

1. Translation from Mandarin to English (and vice versa)
2. Read-aloud on press
3. Long-press menu with copy, dictionary, audio, and save
4. Include both toggle-able pinyin and hanzi information

The first two properties were able to be accounted for with [CC-CEDICT](#) and [expo-speech](#), respectively, and the third property with built-in react-native onPress functionality. However, the fourth property was pretty custom, and is not feasible with a basic react-native <Text> object. Thus, we decided the best thing to do was create a custom Text component.

### HanziPinyinBlock

Since a pinyin / hanzi character module was a universal requirement throughout the app, we created a `HanziPinyinBlock` component (i.e. HPBs for short).

Untoggled Pinyin	toggledPinyin (shortPress)	longPress
		To add in the future: will have the option to copy, define, translate, and save vocab

### HanziPinyinArray

Built off of the HPB component, this array mapped multiple HPBs to create an “interactive” sentence. This was necessary for parts of our app like the prompt text and user response transcription. At this point, the padding also needed to be adjusted to account for characters with longer pinyin (e.g. zhōng vs. wǒ).

nǐ yǒu méi yǒu chuān cuò  
你有没有穿错

*An array of HPBs*

### EditableHanziPinYin

Another challenge came in the review mode, where it was necessary for the HzPyBlock to display toggle-able pinyin while also being editable by the user to correct for mistakes that they

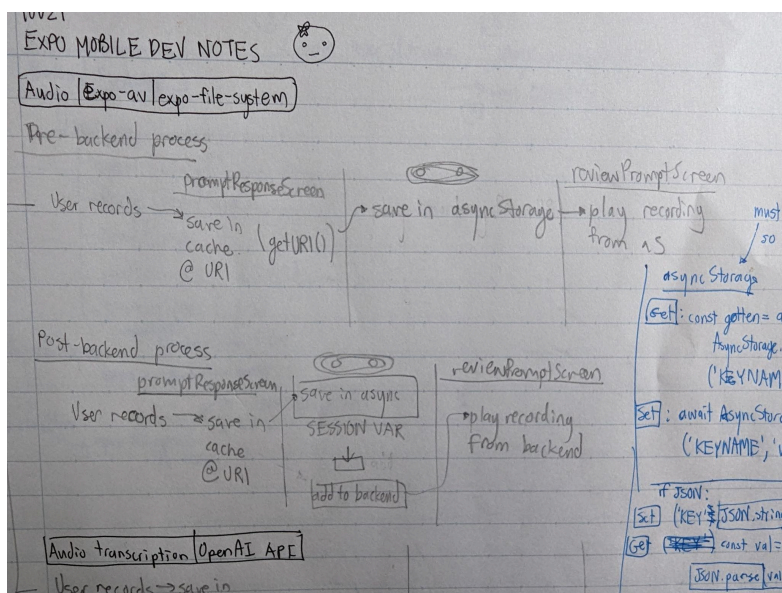
made while speaking or the speech-to-text API made in transcription.

## Media

Media is a core part of the design philosophy of our app, coming in the forms of audio, video, images, and animated components.

### Cached data (AsyncStorage)

Before delving into media, it was necessary to set up the Async Storage of the app. This is somewhat analogous to the local storage that comes with a web application that is stored in the web browser—it is data that is stored on the user's device in association with the app. This is how we imagined the initial data-flow:



We treated AsyncStorage as the main storage for storing user information (e.g. settings, language preferences, and prompt responses) during development. For example, a user's response to a prompt would look like this.

```
LOG NEW recording {
  "id": 25417,
  "title": "something that made you really happy today",
  "prompt": "今天我笑到肚子痛，你猜为什么？",
  "vocabList": [
    {
      "hanzi": "笑",
      "pinin": "xiao",
      "translation": "to laugh"
    },
    {
      "hanzi": "肚子",
      "pinin": "duzi",
      "translation": "stomach"
    },
    {
      "hanzi": "痛",
      "pinin": "tong",
      "translation": "hurt"
    }
  ],
  "uri": "file:///data/user/0/host.exp.exponent/cache/ExperienceData/%2540anonymous%252Fyouzi-mobile-27f26289-ed0a-4853-ac87-7096019ab175/Audio/recording-beea1356-d2e7-4bc2-b603-e77ffd57ddad.m4a",
  "vibe": "chatting",
  "subVibe": "chats",
  "difficulty": 0,
  "date": "5/1/2024",
  "transcription": "\"我今天看了一只猫试图跳上桌子，但它没跳上去，直接滑到地上，超级好笑，我笑喷了！\""
}
```

**Prompt response data; note: the uri is the local file path to where the m4a audio file is stored**

Creating storage like this was necessary for development so that audio files persist across different screens. The way we designed the data flow was as follows:

1. User **creates** audio file in prompt response screen
2. Audio file and meta information is **stored** in Async Storage
3. User **retrieves** audio file to play back and edit in prompt review screen

In the future, we hope to move most of the user data to the backend, and the app will interface directly with firebase using the [react-native-firebase](#) package. This will also reduce the app size for the user, which is much more ideal and practical if we ship this app.

## Audio

Audio was one of the most essential and also one of the most difficult parts of the frontend to implement. There were many potential options for tackling audio.

Functionality	Package	UI component	Screens using component
<b>Recording audio:</b> audio handled recording audio quite well, but it was still pretty limited in handling playback. iOS could handle start, pause, and play, but Android could only play from the beginning and stop	<a href="#">expo-av</a> ( <a href="#">Audio module</a> )	Record button	Prompt response screen
<b>Reading: read</b> Mandarin text out loud as audio (also did not have pausing on Android)	<a href="#">expo-speech</a>	Read-aloud button	Prompt select screen, prompt response screen, & prompt review screen
<b>Playing audio file:</b> same restrictions as above	<a href="#">expo-av</a> ( <a href="#">Audio module</a> )	Playback button	Prompt review screen
<b>Transcribing audio:</b> as a language-learning app that relies on speaking as the main mode, we needed to transcribe audio into text so that users could connect what they spoke to characters.	Open-AI's <a href="#">speech-to-text API</a>	N/A	Prompt response screen (after user stops recording)

## Video

In order to save storage costs (and make use of existing Mandarin media), we opted to embed video content from YouTube, rather than creating our own novel Mandarin media content. The displayed video was obtained by using the [Youtube API](#) to obtain the search results for the media recommendation. After searching using the media recommendation as a keyword, we grabbed the video ID of the first video which matched our parameters: under 3 minutes and not age-restricted.

Functionality	Package	UI component	Screens using component
Displaying video	<a href="#">react-native-youtuve-iframe</a>	An embedded video 	Prompt response screen

## Image

The three main types of images were:

1. Prompt-related AI-generated images
2. Iconography
3. App mascot: Xiao-you

Prompt images and iconography were straightforward inclusions, so we will only detail the Xiao-you mascot.



## Xiao-you Mascot

Implementing Xiao-you involved creating a separate component: <XiaoYou /> and associated visual components (e.g. speech bubble and thinking bubble) that could be triggered by user actions (e.g. pressing vocab). If time permits, we would like to add emotional states to Xiao-you.



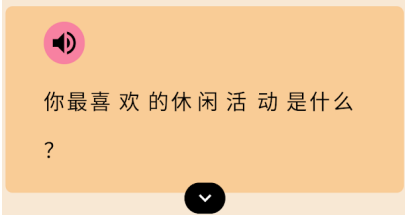

***Xiao-you pipes up when the user presses a vocab term***

## Animation

To address the “joyfulness” element of our app design, we wanted to create more motion within our app on top of the multimedia content.

### Component Animation

There were many basic components of our app (e.g. buttons, modals, drop-down menus, etc.) that required animation. These were handled with React Native’s [Animated](#) library, usually modifying a component’s numeric properties (height, opacity, etc.) by tying them to an `Animated.Value` state variable, then changing this `Animated.Value` with something like a button press. For example, for the prompt-response card, to create an animated toggle for showing the English translation, we used:

Animated.Value	<pre>const heightAnim = useRef(new Animated.Value(0)).current;</pre>
Using Animated.Value in Animated.View	<pre>&lt;Animated.View style={[styles.toggleableView, { height: heightAnim }]}&gt;   {expanded ?     &lt;View       style={[styles.englishTextView, youziStyles.englishTextView]}&gt;       &lt;Text style={youziStyles.englishPromptText}&gt;{getEnglishText(promptID)}&lt;/Text&gt;     &lt;/View&gt; : null   } &lt;/Animated.View&gt;</pre>
Triggering animation with Animated.timing(...).start()	<pre>const toggleEnglish = () =&gt; {   // console.log('toggling english')   setExpanded(!expanded);   Animated.timing(heightAnim, {     toValue: expanded ? 0 : 125, // Adjust the height value as needed     duration: 300, // Adjust the duration as needed     useNativeDriver: false,   }).start(); };</pre>
Result	<div style="display: flex; justify-content: space-around;"> <div data-bbox="574 1423 976 1692"> <p>Unexpanded</p>  </div> <div data-bbox="1003 1423 1404 1812"> <p>Expanded</p>  </div> </div>

## Xiao-you Animation

Due to time constraints, we were not able to fully animate our mascot Xia-you. Ideally, the goal was to create a lively mascot that involved stringing together keyframes created by our UX designers. If time permits, we would probably look into a package, [Lottie](#), that renders Adobe After Effects animations for use in React Native.

## Styling

Unlike traditional JS/HTML/CSS styling or React-based component styling (JS/CSS), React Native styling requires that each component's styles are included in the components JS file as a stylesheet. Universal styles, variables, typography were included in a global `youziStyles.js` file and included component styles as needed throughout the app.

```
1  'use strict'
2
3  import { StyleSheet, Dimensions } from "react-native";
4
5  > export const youziDimensions = { ...
9  };
10
11 > export const youziColors = { ...
32 };
33
34 v export const youziStyles = StyleSheet.create({
35 v   centeredView: {
36     backgroundColor: youziColors.backgroundPastelOrange,
37     alignItems: 'center',
38     justifyContent: 'center',
39     paddingTop: 100,
40     height: '100%'
41   },
42 v   horizontallyCenteredView: {
43     alignItems: 'center',
44     justifyContent: 'flex-start',
45     paddingTop: 110
46   },
47 v   headerText: {
48     fontSize: 40,
49     fontFamily: 'Ttim'
```

*A snippet of our global styles file*

## Constraints

There were many constraints for the frontend process, but they can be condensed into: manpower, knowledge, and time.

Building an app with one frontend developer within ten weeks was quite difficult. It was also very much a learn-on-the-go process, since our frontend developer's only other experience with mobile development was building a few screens for a personal app idea that they had in the past.

Although a lot translated from their background in web development, there will still be many difficulties, such as differing features and behaviors for iOS and Android (which is not as big of a problem for different web browsers). This created a lot of unexpected complications to account for compared to a web-based application.

If we were to restart this process, perhaps we would have been more cognizant of our own limitations, and perhaps we could have focused on an app concept that was more feasible to create given our constraints.

## Prompt Engineering and Generative AI

### Methodology

To get the data we needed, we decided that we would use prompt engineering techniques and generative AI. We decided that this would be necessary because it is more scalable to use generative AI than to just manually write out thousands of Chinese conversation starter prompt messages for our users. In addition, with generative AI, we are able to fine-tune the vibe and tone of each conversation starter prompt, such that they are quite appropriate and relatable for a Gen-Z American-born Chinese speaker.

### Necessary Data

Firstly, we thought about the data that we might need for each conversation starter prompt, and what sorts of formatting would be appropriate. After much discussion and user research, we found that it would make sense for each conversation starter prompt to include:

- Some sort of icebreaker question for the user, written in Mandarin
- The translation of the icebreaker question, written in English
- Some Chinese vocabulary entries that may be relevant for answering the question
- Some additional funny Chinese slang phrases or idioms, which may be relevant for answering the question, inspired by our user research that young adult ABC's are particularly intrigued and amused by funny terms

- Some Chinese media recommendations (e.g. movies, songs, shows) related to the conversation topic, from our finding that our target demographic user base is particularly interested in learning via immersive media

### Further Refinement of Data Requirements

Next, we thought to further refine these data requirements. For example, we thought about:

- Character and input limits per field: we thought about how many characters we could fit in these data types without compromising the look and feel of the frontend software. As a team, we made sure to iteratively come up with a frontend design that would be compatible with the style and quantity of data we were generating.
- Narrowing down the content topics: we considered how to limit the scope of topics that the generative AI can come up with in addition to the topics that young adult Gen-Z ABC users may care the most about.
- Language difficulty: we wanted to know what kinds of Mandarin language difficulties should be present given the skills and goals of our average user.

### OpenAI Test Tools

Then, we started playing around with ChatGPT and the OpenAI Assistant Playground in order to get an initial gauge as to what we might need to tweak going forward. Very soon, we realized that we needed to learn more about:

- Prompt engineering best practices: how do we generate much better conversation starter prompts?
- Automating this process: how can we use the OpenAI API, such that we can generate prompts in a much more scalable way?

### Learning & Applying Prompt Engineering Best Practices

Then, we sought to learn more about prompt engineering best practices so that we could generate higher quality conversation starter prompts. For example, we learned about these particular techniques and applied them in the OpenAI Playground setting:

- Retrieval Augmented Generation Prompting (RAG)
- Chain of thought prompting

- Tree of thought prompting
- Maieutic prompting

## Moving Onto Automation

Finally, we felt satisfied with our prompt engineering tests in the OpenAI Assistants Playground, and we were satisfied with our Assistant Instructions. However, we still needed to find a way to parametrize and automate the process, such that we could easily generate and store over a thousand conversation starter prompts. So, we knew that we had to code up a Python script to do this, which also included copying over our OpenAI Assistant ID and buying access for an OpenAI API key:

```
# Victor Kuan
# March 2024
import time
import random
from csv import writer
from openai import OpenAI
client = OpenAI(api_key="sk-
ASSISTANT_ID = "asst_2gl87dd
```

*Using OpenAI API and OpenAI assistant to start building out automation*

## Key Considerations While Coding

The main point of coding up this prompt generation process instead of solely relying on the OpenAI Assistants Playground is that coding lends us more efficiency when thinking about parametrization and bulk.

### Parametrization

- First, we checked with our designers and frontend engineer to get on the same page in terms of how to best categorize our data and how the users would interact with this data.
- We decided to categorize conversation starter prompts by “vibe” and “sub-vibe.” For example, users can select from a list of four vibes, including “general icebreaker,” “spicy

gossip,” “trauma bonding,” and “media discussion.” For each of these conversation vibes, we came up with four associated sub-vibes.

- To customize prompt strings (that we send to OpenAI), we then created four different methods to generate conversation starters for each different vibe.

```
generate_media_convos()
generate_icebreaker_convos()
generate_haixiu_convos()
generate_trauma_convos()
```

*Helper functions which are specifically designed to print out prompts of specific vibes*

- Within each method, and while accounting for each vibe, we also further considered the sub-vibes, language difficulty, and age appropriateness of the content.

```
list_of_haixiu_vibes = ["silly gossipy vibe", "gossipy vibe", "fun gossipy vibe"]
list_of_haixiu_subvibes = ["spill the tea", "romance", "divulge some juicy secrets", "rant club"]
list_of_haixiu_topics_0 = ["something dramatic that happened to you recently", "something that pissed you off recently",
list_of_haixiu_topics_1 = ["college romance", "college romance regrets", "college dating weird stories", "spicy things th
list_of_haixiu_topics_2 = ["weird secrets you've been holding onto", "funny lighthearted secrets you have", "profound se
list_of_haixiu_topics_3 = ["something bothering you recently that you want to rant about", "bullshit that you want to ran

def generate_haixiu_convos():
    curr_vibe = "hai xiu / spicy talk"
    for x in list_of_haixiu_vibes:
        for y in list_of_haixiu_subvibes:
            if y == "spill the tea":
                for z in list_of_haixiu_topics_0:
                    generate_convo_starter(curr_vibe, x, y, z, 0, 0)
                    generate_convo_starter(curr_vibe, x, y, z, 0, 1)
            elif y == "romance":
                for z in list_of_haixiu_topics_1:
                    generate_convo_starter(curr_vibe, x, y, z, 0, 0)
                    generate_convo_starter(curr_vibe, x, y, z, 0, 1)
            elif y == "divulge some juicy secrets":
                for z in list_of_haixiu_topics_2:
                    generate_convo_starter(curr_vibe, x, y, z, 0, 0)
                    generate_convo_starter(curr_vibe, x, y, z, 0, 1)
            elif y == "rant club":
                for z in list_of_haixiu_topics_3:
                    generate_convo_starter(curr_vibe, x, y, z, 0, 0)
                    generate_convo_starter(curr_vibe, x, y, z, 0, 1)
```

*Within each vibe, we accounted for different topics, sub-vibes, and conversational tones*

## Running the Script

Finally, using this script, we were able to iteratively print out roughly one thousand prompts in CSV format, while considering vibes, sub-vibes, difficulty ratings, and more. However, running the script proved to be quite a slow and lengthy process. We have provided a screenshot from the CSV file below:

ID	ENTIRE_PROMPT	VIBE
43985	<p>Chinese convo starter #1: 最近发生了一件很劲爆的事, 你想听吗?</p> <p>English translation #1: Something dramatic happened recently, want to hear about it?</p> <p>Related Chinese media recommendation #1: Watch "前任3: 再见前任" for dramatic love stories and unexpected events.</p> <p>One vocab word that is relevant to use in the response #1: 最近 (zújìn) - recently</p> <p>Second vocab word that is relevant to use in the response #1: 发生 (fāshēng) - happened</p> <p>Third vocab word that is relevant to use in the response #1: 事 (shì) - event</p> <p>Funny Chinese slang/phrase that can be used in the response #1: 爆料 (bàoliào) - Spill the beans</p> <p>Example response #1: "哦, 太棒了, 我最喜欢听劲爆的故事了! 发生了什么? 是关于爱情的吗? 还是其他的秘密呀? 赶快爆料吧!"</p> <p>English translation of response #1: "Oh, awesome, I love hearing dramatic stories! What happened? Is it about love, or some other secret? Spill the beans quickly!"</p> <p>Follow up Chinese convo starter #2: 嗯, 其实是关于一个意想不到的相遇。</p> <p>English translation #2: Well, it's actually about an unexpected encounter.</p> <p>Related Chinese media recommendation #2: "转角遇到爱" showcases unexpected encounters and their consequences.</p> <p>One vocab word that is relevant to use in the response #2: 意想不到 (yìxiǎngbùdào) - unexpected</p> <p>Second vocab word that is relevant to use in the response #2: 相遇 (xiāngyù) - encounter</p> <p>Third vocab word that is relevant to use in the response #2: 关于 (guānyú) - about</p> <p>Funny Chinese slang/phrase that can be used in the response #2: 碰瓷 (pèngcí) - (literally) bumping into porcelain, a metaphor for unexpected or unwelcome encounters in a humorous mood.</p> <p>Example response #2: "真的吗? 我也遇到过意想不到的情况, 细节很重要, 是不是很浪漫的相遇, 还是像电视剧那样的碰瓷?"</p> <p>English translation of response #2: "Really? I've also experienced unexpected situations; details are crucial. Was it a romantic encounter, or like a drama series kind of bumping into each other?"</p> <p>Final follow up Chinese convo starter #3: 是一种你绝对想象不到的浪漫!</p> <p>English translation #3: It was a kind of romance you could never imagine!</p> <p>Related Chinese media recommendation #3: "我不是药神" suggests unexpected events can lead to profound changes, offering a unique take on romance and life.</p> <p>One vocab word that is relevant to use in the response #3: 绝对 (juéduì) - definitely</p> <p>Second vocab word that is relevant to use in the response #3: 想象 (xiǎngxiàng) - imagine</p> <p>Third vocab word that is relevant to use in the response #3: 浪漫 (lǎngmàn) - romantic</p> <p>Funny Chinese slang/phrase that can be used in the response #3: 甜蜜的负担 (tiánmì de fùdān) - sweet burden, used humorously to describe romantic situations that are overwhelmingly sweet.</p> <p>Example response #3: "天哪, 听起来太激动人心了! 你能分享更多细节吗? 这种甜蜜的负担是每个人都梦寐以求的。"</p> <p>English translation of response #3: "Oh my, that sounds thrilling! Can you share more details? This sweet burden is what everyone dreams of."</p>	hai xiu / spicy talk
9571	<p>Chinese convo starter #1: 你最近经历了什么让人脸红心跳的事情?</p> <p>English translation #1: What recent experience made your heart race and face blush?</p> <p>Related Chinese media recommendation #1: Watch "前任3: 再见前任" for some dramatic love stories.</p> <p>One vocab word that is relevant to use in the response #1: 经历 (jīnglì) - experience</p> <p>Second vocab word that is relevant to use in the response #1: 让 (ràng) - to make</p> <p>Third vocab word that is relevant to use in the response #1: 脸红 (liǎnhóng) - blush</p> <p>Funny Chinese slang/phrase that can be used in the response #1: 尴尬癌 (gāngà ài) - second-hand embarrassment</p> <p>Example response #1: "哎呀, 上个礼拜在公司大会上, 当着所有人的面, 我不小心叫错了我的老板的名字, 我的老板笑了, 但我感觉自己就像患了尴尬癌一样, 脸红了好久。"</p> <p>English translation of response #1: "Oh well, last week during a company meeting, in front of everyone, I accidentally called my boss by the wrong name. My boss laughed, but I felt like I had second-hand embarrassment, and"</p> <p>Follow up Chinese convo starter #2: 那之后你如何处理这个尴尬的情况?</p> <p>English translation #2: How did you deal with the embarrassing situation afterward?</p> <p>Related Chinese media recommendation #2: "夏洛特烦恼" is a funny take on dealing with life's embarrassing moments.</p> <p>One vocab word that is relevant to use in the response #2: 之后 (zhī hòu) - afterward</p> <p>Second vocab word that is relevant to use in the response #2: 处理 (chǔlǐ) - to handle</p> <p>Third vocab word that is relevant to use in the response #2: 尴尬 (gāngà) - embarrassing</p> <p>Funny Chinese slang/phrase that can be used in the response #2: 土味情话 (tǔ wèi qíng huà) - cheesy pick-up lines</p> <p>Example response #2: "我决定正面处理, 第二天我给老板送了一杯咖啡, 并附上一张写有土味情话的卡片, 开玩笑说希望这能抵消我叫错名字的尴尬, 老板大笑了, 说这事就此翻篇。"</p>	hai xiu / spicy talk



## Cleaning the New Data

Upon chatting with the designers and frontend engineer, we found that our prompts were not completely properly formatted and thus were not usable in our frontend code yet. So, we thought to do some data cleaning. We considered the following:

- Adding better-labeled header names
- Further modularizing the prompt string data into more specific CSV columns, such that our frontend engineer would not have to manually clean and inefficiently parse the data every time they fetch the prompt string data
- Getting rid of unnecessary data or improperly formatted data

```
# helper function for adding column names to the header just in case the data is corrupted
def fix_column_names():
    # Load the CSV file, assuming it has no header row
    df = pd.read_csv('new_data_2.csv', header=None)

    # Assign new column names
    df.columns = ["ID", "ENTIRE_PROMPT", "VIBE", "DETAILED_VIBE_YOU_CAN_IGNORE", "SUB_VIBE", "TITLE_DESCRIPTION", "WORK_SAFETY_RATING",

    # Save the updated DataFrame back to a CSV file
    df.to_csv('updated_file.csv', index=False)
```

```
if(len(row[1]) < 10):
    continue

convo_starter_1 = find_middle_text("Chinese convo starter #1:", "English translation #1:", text)
english_translate_1 = find_middle_text("English translation #1:", "Related Chinese media recommendation #1:", text)
media_1 = find_middle_text("Related Chinese media recommendation #1:", "One vocab word that is relevant to use in the response #1:", text)
vocab_1_1 = find_middle_text("One vocab word that is relevant to use in the response #1:", "Second vocab word that is relevant to use in the response #1:", text)
vocab_2_1 = find_middle_text("Second vocab word that is relevant to use in the response #1:", "Third vocab word that is relevant to use in the response #1:", text)
vocab_3_1 = find_middle_text("Third vocab word that is relevant to use in the response #1:", "Funny Chinese slang/phrase that can be used in the response #1:", text)
funny_1 = find_middle_text("Funny Chinese slang/phrase that can be used in the response #1:", "Example response #1:", text)
example_response_1 = find_middle_text("Example response #1:", "English translation of response #1:", text)
example_response_translate_1 = find_middle_text("English translation of response #1:", "Follow up Chinese convo starter #2:", text)
```

***Writing a script to automate the data cleaning process (adding more descriptive headers and further modularizing the CSV file for engineering accessibility)***

## GitHub, the Final Step

We made sure to add all of our prompt engineering efforts and data to a GitHub repo to use version control to safely save our work. Our prompt engineering GitHub repo can be viewed here: [https://github.com/vkuan/youzi\\_prompt\\_engineering](https://github.com/vkuan/youzi_prompt_engineering)

## Next steps

### Further user testing to be done

While our team conducted user interviews over the course of the second year, we would ideally like to conduct more in the future for two main purposes: first, to validate our design decisions in the most current iteration and see which features are worth keeping or need changing; and second, to conduct in-person usability testing. The second is especially important in order to ensure that all the affordances within the app are easily understood even by a first-time user of the app.

Additionally, as our team is currently working on introducing monetization to future iterations of the app, we feel that it is important to also expand the pool of interviewees that we speak to. Rather than just speak to the target users of the app (i.e. ABC's), we would also be able to glean insightful information from others in their social circle, such as parents, or even school staff, who may wish to use the app within their classrooms.

### Features in Consideration

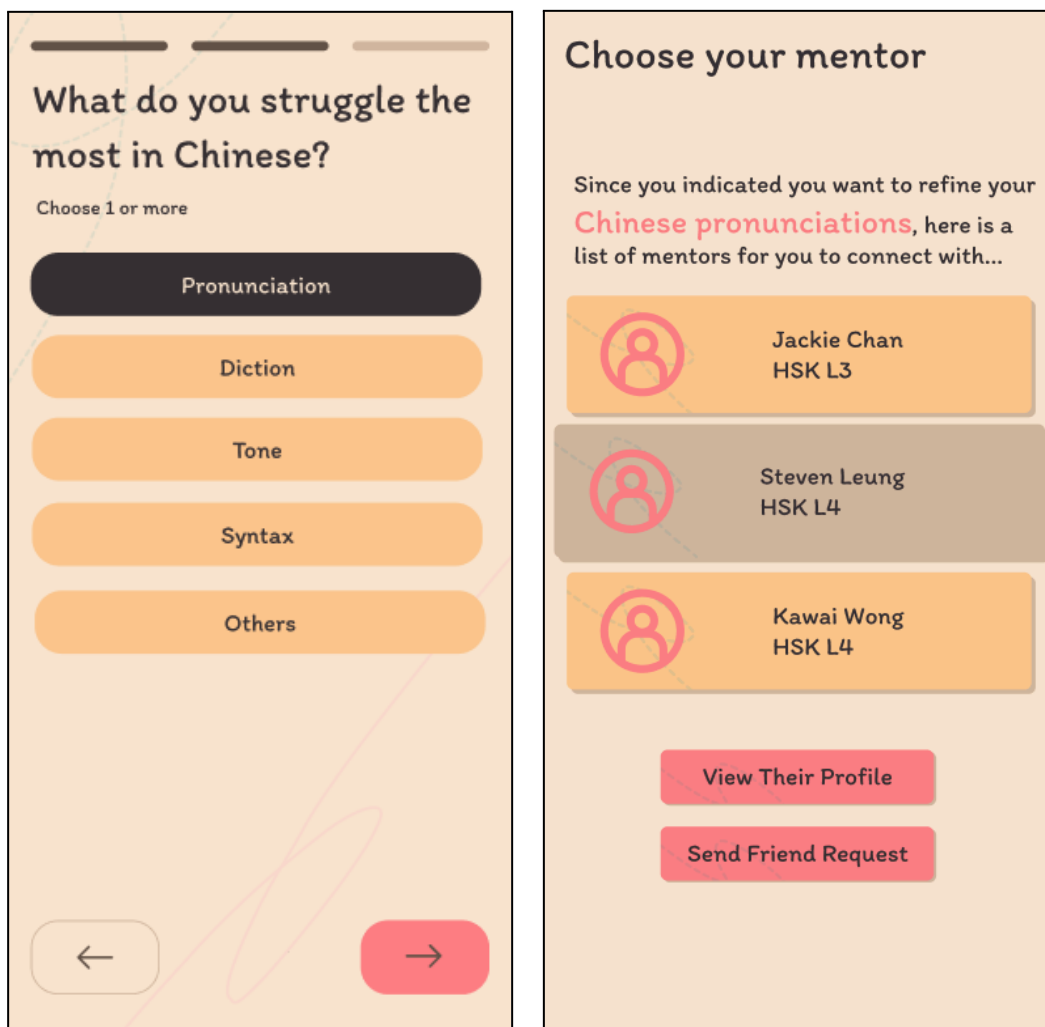
With the past three months of app development, our team successfully focused on designing and developing the signup, conversation vibe selections, conversation prompts and recordings, and review board. In the near future, our team desires to extend the development effort of Youzi and propose some features to be matured that amplify the “social learning” aspect of our product. We incorporated concepts learned from INFO 233 - Social Psychology. Currently, we have the below backlogged features as mid-fidelity design wireframes.

One social psychology theory of our app design project is to integrate Legitimate Peripheral Participation (LPP), a concept introduced by Jean Lave and Etienne Wenger in their theory of situated learning, which refers to the process where newcomers engage in learning and participation within a community of practice. In other words, newcomers start by engaging in simpler tasks on the periphery of the community and gradually move towards more complex roles as they gain experience and knowledge. In the context of our Chinese language learning app Youzi, implementing “mentorship” and “preview of content” to new users when they are about to join the app aligns with LPP principles. We aim to help new users ease into the perceptually daunting Chinese language learning tasks by allowing them to seek mentors, make friends before engaging, and learn more about the Youzi app before they join the community with confidence.

## Mentorship

The first feature we have in our roadmap is to introduce “mentorship” before prospective users joining the app. We aimed to understand the impact of mentorship on e-learners' confidence, engagement, and sense of community. In the study “Online mentoring: challenges and strategies”, Jan and Mahboob suggest that facilitating interaction and engagement via technology is a positive way to develop confidence in users. It would also generate effective relationships and a sense of community that help an e-learner who is somehow new to a topic obtain goals, security, and control during the learning process. These were all fostered by curating connections of new users with someone they could potentially look up to when they start the learning journey.

In the Youzi app, one of our design choices is to introduce mentorship with the existing users being mentors and new users being mentees, who can specify the area they want to work on.



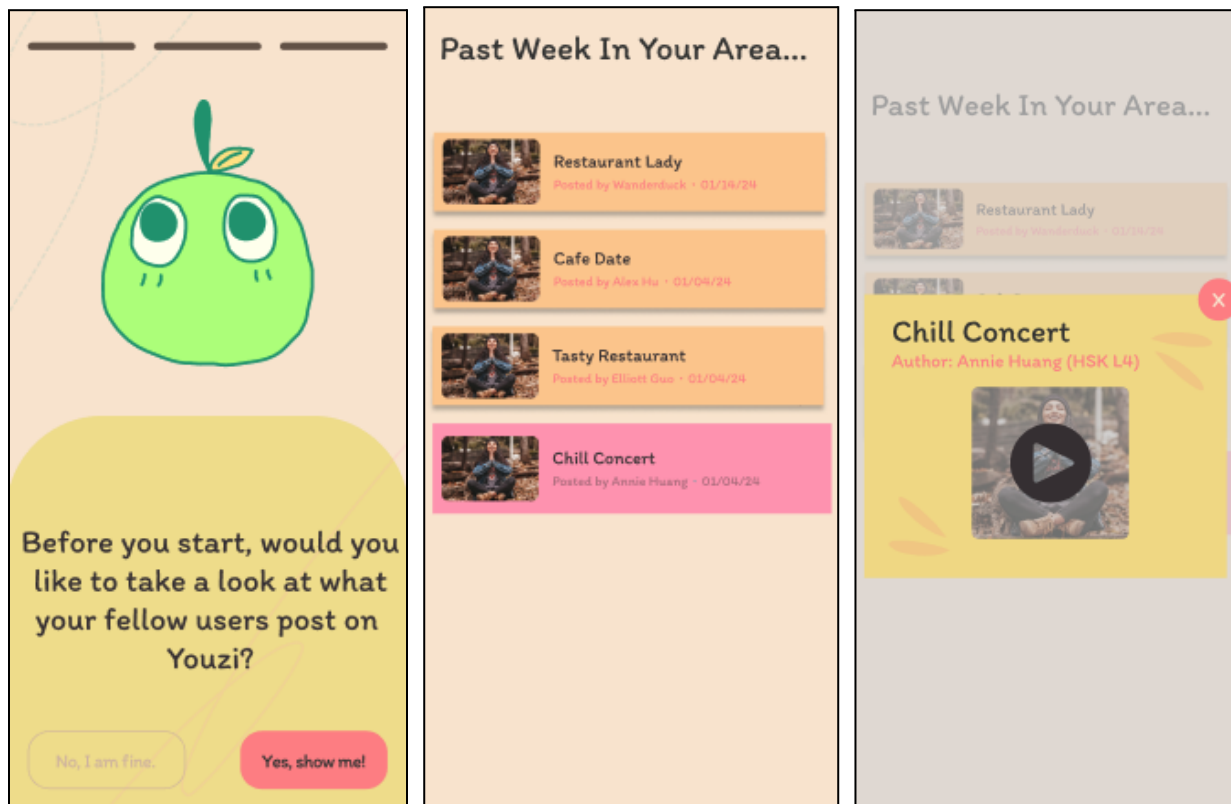
We aim to allow new users to self-evaluate what in Chinese learning they would like to focus on the most and match the existing qualified users as their mentors to help them guide their use of the app. By connecting new users (i.e. mentees) with experienced users (i.e. mentors) who have a deep understanding of the language learning process, the app facilitates access to a community of practice. This community consists of individuals who share common goals, norms, and practices related to language acquisition. Mentors serve as knowledgeable insiders who guide and support mentees in navigating this linguistic community. LPP emphasizes the idea of newcomers starting with peripheral activities and gradually moving towards more central roles within the community. In the context of Youzi, new users begin as mentees, engaging in language learning tasks and receiving guidance from mentors. Over time, as they gain knowledge and skills, they may transition into mentor roles themselves, contributing to the community and supporting other learners.

In addition, we believe that within communities of practice, socialization and identity formation play crucial roles in learning and participation. Mentors in Youzi would not only provide linguistic support but also help mentees develop a sense of belonging and identity as language learners. Through interactions with mentors and peers, mentees acquire language-related knowledge, skills, and practices, shaping their identity as legitimate participants in the language learning community.

Initially, we aimed to focus on the new users and primarily assist them to gain confidence and find their community more easily. However, there were also some surprises in our findings. While we expected that mentorship would primarily benefit mentees, we also observed positive outcomes for mentors themselves. Mentors reported increased satisfaction, a sense of fulfillment, and enhanced mastery of the subject matter through teaching and guiding others. This dual benefit of mentorship was an unexpected yet valuable discovery in our user research on the original prototype of our app.

## Preview of Content

Another design element where we applied LPP was to introduce previews of content posted by existing users before new users join. We believe that by providing new users with **previews of existing users' posted content**, our app would offer them access to the knowledge and expertise shared within the community.



This content serves as examples of how experienced members of the community engage with the platform, contribute valuable information, and interact with one another. Access to this knowledge is crucial for newcomers to understand the norms, expectations, and practices within the learning community.

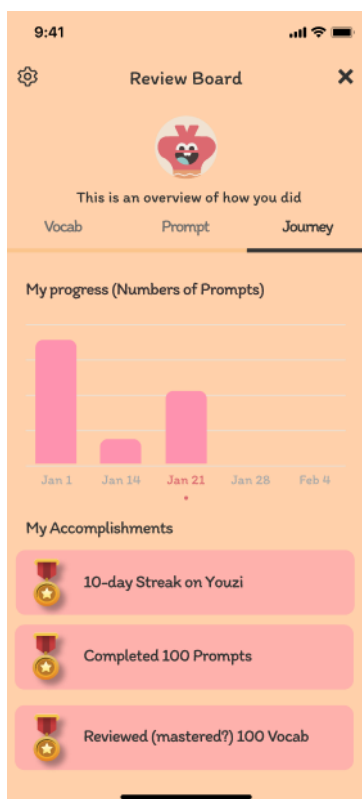
In our "preview of content" flow, when new users sign up, we aim to prompt them to explore preview content within their local user community. This approach is designed to foster a sense of proximity and connection among users by introducing them to fellow users located in their cities. By doing so, we hope to encourage users to potentially meet in person, establish friendships, and enhance their overall experience with the app. The preview content will include a curated list of posts generated by other users over the past week, offering new users a glimpse into the types of content and conversations they can expect on the platform. Additionally, new users will have the opportunity to access recorded Mandarin conversations on

specific topics, providing them with a preview of the app's functionalities and potential learning opportunities before fully engaging with it.

As LPP suggests that newcomers start with peripheral activities and gradually move towards more central roles within the community, setting a preview of existing users' content would serve as an initial step in this gradual engagement process. New users can explore the content, familiarize themselves with the community's interests and discussions, and decide how they want to participate further. This gradual exposure helps new users transition from peripheral observers to active participants. What we also foresee by implementing preview content is the existing users' content may provide new users with role models to emulate. New users can observe how experienced users structure their posts, engage with others respectfully, provide valuable insights, and contribute positively to the community. These role models shape new users' understanding of desirable behaviors and practices within the community.

## Learning Journey

Finally, the last feature we would like to add, although not related to LPP, is to add a “Journey” tab to the review board. Here, we would like to capture the total prompts a user has finished and group them by week on a bar chart to show the user their trend of the amount of prompts they learned over the past few weeks. Underneath the progress chart, we would like to show the users some potential badge they could achieve through the app. For example, we would like to encourage the users to keep logging into our app, so they could keep their momentum in learning.



## Conclusion

### Ethics & Considerations

Throughout the app design and development process, our team wanted to acknowledge some important ethical concerns we had as we aimed to create an inclusive, supportive, and safe online environment in our product:

1. **Cultural Sensitivity and Appropriation:** Ensuring that Youzi respects and accurately represents Chinese culture is crucial. There is a fine line between cultural appreciation and cultural appropriation. We wanted our product to avoid stereotyping and aimed to develop input from individuals who are culturally competent or ideally, members of the community (e.g. young American-born Chinese) it aims to serve.
2. **Data Privacy and Security:** Users will likely share a lot of personal information, including potentially sensitive data about their linguistic abilities, personal identity, and cultural background. Thus, we found it important to ensure robust data protection and clearly communicate these measures to users, which is essential to maintain trust.
3. **Bias in AI and Algorithms:** We think it is important to ensure that these systems are trained on diverse datasets that represent various accents and speech patterns. AI systems can inadvertently perpetuate bias if they are not properly calibrated.
4. **Accessibility:** One of our future goals is to ensure the app is accessible to users with disabilities, including those with visual, auditory, or cognitive impairments. This includes providing text-to-speech capabilities, subtitles for videos, and ensuring the app's interface complies with accessibility standards.
5. **Age considerations:** We understand that our content is AI-generated, and so, there are some content-related risks. As our content is prompt-engineered and could potentially mention topics related to trauma, politics, and romance, we want to make sure that children are not using our app. As such, we will write an End User License Agreement (EULA) and a privacy policy, which will both prohibit users under the age of 17 from registering for our app. To enforce this, we can explicitly ask users for their age at the point of app registration. Alternatively, we can ask users to log in with Google, so we can abstract away the age checking process to the user's Google profile.

## Limitations

Due to constrained resources, including limited time and the availability of only one frontend and one backend engineer on the team, several features could not be implemented as originally envisioned in the product specification. A notable omission is the "smart suggestion" feature, which was intended to analyze user responses and provide live suggestions. Implementing such a feature would require significantly more resources than currently available. Additionally, there is a potential risk of breaching user data privacy without a properly established Data Privacy and Security agreement.



Another limitation is the narrow demographic of our user testing group, which consisted solely of UC Berkeley students of similar ages. While our product is indeed targeted at this demographic, exposure to a broader range of users could have provided more diverse insights and highlighted potential areas for usability improvements. Due to the limited scope of our user testing pool, we have not been able to perfectly address or incorporate such varied feedback.

## References

### DRG

- Rethinking Chinese Language Education for ABCs:  
[https://docs.google.com/document/d/1PcU3AO0DXaYuySUDiPWlYJcFmZIXsgScMxc\\_KjidYaA/edit#heading=h.3emjtiymvoav](https://docs.google.com/document/d/1PcU3AO0DXaYuySUDiPWlYJcFmZIXsgScMxc_KjidYaA/edit#heading=h.3emjtiymvoav)
- Prompt Engineering, OpenAI, and Database Management in the context of building a language learning platform:  
<https://docs.google.com/document/d/1jeSJSnK8B6VCFNvxFI0X-bz-p3wccDOWp4HmFgvnPc/edit#heading=h.3emjtiymvoav>

### Figma

- <https://www.figma.com/file/uBMWCAatUvGqXXDBRdTXq4/Capstone-Design-Ideation?type=design&node-id=627%3A439&mode=design&t=dSRDShEOyDUdKhR6-1>

### GitHub

- Youzi Frontend Engineering Mobile Development Repo:  
<https://github.com/tylerwu2222/youzi-mobile/tree/main>
- Youzi Prompt Engineering Repo:  
[https://github.com/vkuan/youzi\\_prompt\\_engineering](https://github.com/vkuan/youzi_prompt_engineering)

### Slides

- [https://drive.google.com/file/d/1ZHunOlin\\_vuu13-Af\\_ONGxC8BLWkv-3l/view?usp=drive\\_link](https://drive.google.com/file/d/1ZHunOlin_vuu13-Af_ONGxC8BLWkv-3l/view?usp=drive_link)

### Website:

- Project about page:  
<http://youzi.life/>

## Contact Info

- Team email: [youziappwork@gmail.com](mailto:youziappwork@gmail.com)
- Team members:
  - Joanne Eunjung Jun: <https://www.linkedin.com/in/jun-j/>
  - Victor Kuan: <https://www.linkedin.com/in/vkuan/>
  - Jo Lu: <https://www.linkedin.com/in/jovan-lu/>
  - Harris (Chun Yin) Wang: <https://www.linkedin.com/in/harriswangchunyin/>
  - Tyler Wu: <https://www.linkedin.com/in/tyler-wu/>