Software Design Document

Date: 12/6/2023			
HongGang Wen	20213802014	Huaen Lu	20213803005
Xinzhu Li	20213803006	Qing Fang	20213803003

CATALOGUE

1 Introduction

This project plans to develop an application based on mobile terminal to show users traditional Chinese food. Chinese cuisine is an important part of Chinese excellent traditional culture, which should be inherited, protected and developed.

The purpose of this project is to protect and inherit traditional Chinese food, and the goal is to let users understand the essence of traditional Chinese food culture, and stimulate users' love for food and exploration spirit. By providing a comprehensive, unique and convenient platform for food culture exchange, users can use the app to understand and discover more food culture, whether they are making dishes at home or ordering food when dining out.

The overall scope of application of gastronomy includes the following aspects:

- Application user group: The target users of MSHZ application are those who are interested in food, love food, and want to know more about traditional Chinese food culture.
- Application scenarios: The application scenarios include referring to recipes when making dishes at home, learning about food culture when dining out, and interacting with other food lovers.
- The geographical scope of application: The geographical scope of MSHZ application is global, but it focuses on the introduction of traditional Chinese food culture.
- Application development direction: The development direction of MSHZ application is to develop in a more professional, comprehensive and international direction, and continue to expand the influence of the application and user groups.

2 Requirements Specification

2.1 User requirements

2.1.1 Users' demand for traditional Chinese food culture

About 93% of people have a specific need for traditional food culture, either to satisfy taste buds or to follow traditional memories. As the old saying goes, food is the most important thing in people's life, and Chinese food culture has formed distinct national characteristics. Nowadays, with the progress of science and technology, the taste left in my heart is increasingly dim, and the inheritance of Chinese food culture is imminent. People's awareness of protecting traditional food culture has been strengthened, not only focusing on the food itself, but also paying more attention to the cultural heritage behind the food.

2.1.2 User interface requirements

- Simplicity: Users need a simple, beautiful and easy-to-use interface that makes it easy to browse, search, share and save food information.
- Consistency: The structure of the interface must be clear and the terminology used must be

consistent, the style must be consistent with the content, and the tone of the interface font must be consistent.

- From the perspective of users: User Centered Design should be adopted in interface design to let users really participate in interface design and reflect users' ideas in the final interface design.

2.1.3 User interaction needs

Users need to interact with other users, share their experience and experience in food production, and also need to be able to provide feedback and opinions to the MSHZ app.

2.2 Functional requirements

- Food culture introduction: to provide a detailed food introduction, including ingredients, practices, tastes, history, cultural background and other aspects of the content.
- Food recipe sharing: provide a recipe sharing function, so that users can share their own recipes, production methods and experiences, and can also browse recipes shared by other users.
- Food search function: to provide a search function to facilitate users to find interested food information according to keywords, regions, tastes and other conditions.
- Food collection function: to provide a collection function, so that users can conveniently save their interest in food information.
- User interaction functions: to provide user interaction functions, including private messages, comments, likes and other functions between users, as well as the function of users to provide feedback and opinions to the application.
- Recommendation algorithm: to develop recommendation algorithms to recommend personalized food information to users based on their preferences and historical behavior.

3 Overall Design

The MSHZ application contains four pages —— login registration page, home page, community page and personal setup page. As shown in the following figure:

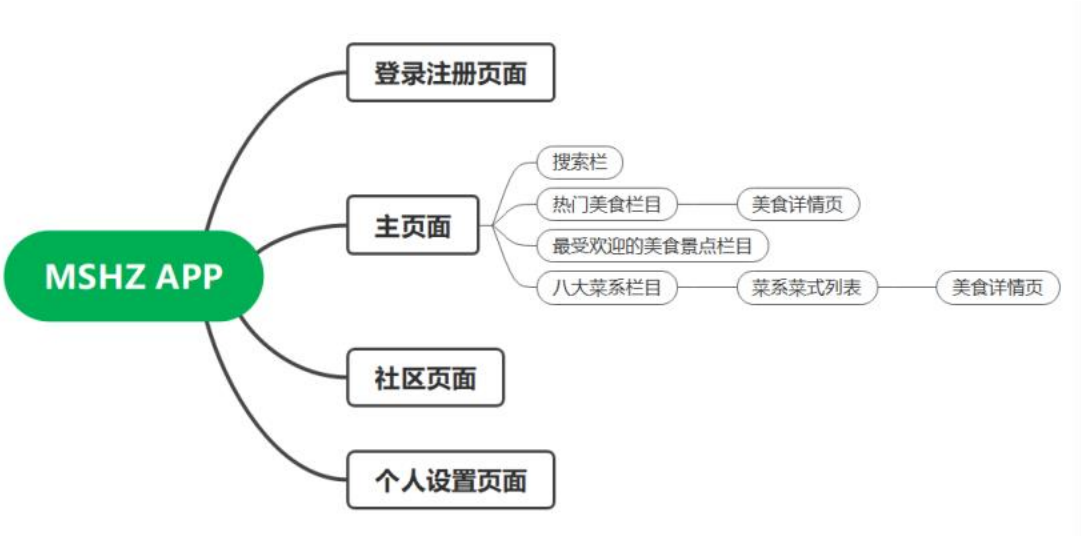

Figure 3-1

3.1 Log in and register page

On this page, users use their personal account to log in. If the account is still not registered, they need to register at the registration interface. After registration, they can enter the corresponding account password on the login page to log in.

3.2 Main page

In the main page, users can use the search bar to search for specific dishes. At the same time, the page includes popular food columns, the most popular food columns and eight cuisine columns. If users do not have specific search objects, they can browse and understand various food dishes in these three columns.

3.2.1 Dish details page

The dish details page provides users with detailed introduction of a specific dish. Users can know the introduction of a dish, the history, the ingredients and steps of the recipe. At the end of the page, the application will recommend four dishes of the same cuisine, convenient for users to browse more dishes. At the same time, if the user likes the detailed information of the dish very much and will browse again in the future, you can click the love key on the upper right side of the page to realize the collection function. Later, you can find the page of the dish in the collection column of the personal setting page.

3.3 Community page

In this page, the app will recommend posts from users, who can view all the users of the program, and give them likes, comments and other operations. In addition to looking at the recommended posts of the system, users can also search the posts of interested dishes to view.

3.4 Personal Settings Page

In this page, it includes account column, thumb up column, collection column, dynamic column and setting column. Users can set profile pictures on this page, manage account personal information, view the dishes and posts that have been liked and browsed, view the dynamic and make other routine Settings.

4 User Interface Design

4.1 Log in and register page

Opens the app, that is the login registration page. The color of the page is mainly red and white, including the components of the input user name, password and login button. The background picture shows the feeling of intangible cultural heritage and antique, and contains the characteristic elements of Chinese food.

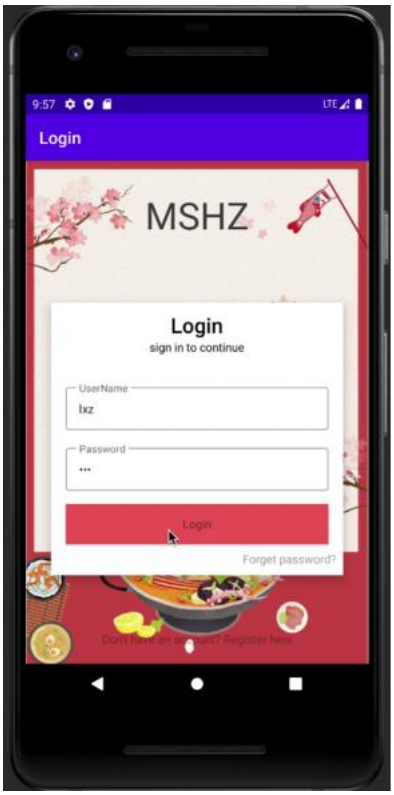

Figure 3-2

4.2 Main page

In the main page, the top of the page is horizontally scrolling and uses HorizontalScrollView for layout. Scroll below the picture is the search bar, the search button is the red color of the page, to make the page more harmonious and beautiful. At the bottom of the search bar is the popular food

recommendation bar and the most popular food attractions column, also using the left and right swipe layout, allowing users to view as much information as possible by swiping on the screen. At the bottom is the eight cuisine columns, using similar layout of Polaroid to display pictures and text, set the border corner degree to beautify the page, weaken the more impact visual effect. While the jump of the main page, the community page and the personal setting page is made up with the fixed navigation bar at the bottom of the three pages, the navigation bar is mainly colored in purple, white and black, which does not conflict with the color of the whole application and is in harmony.

Figure 3-3 Figure 3-4

4.2.1 Dish details page

The top of each dish details page is a highly blurry picture of the dish, and there is a favorite button in the top right corner. Next, the name of the dish and the category and characteristics of the dish they belong to. We use a format similar to the label to show it under the name of the dish, in a simple gray display. Next, each subheading, such as "introduction", "historical origin".... is framed with red rounded boxes consistent with the main color to make the page layout more neat. The subtitle "step x" in the step introduction of the recipe is also represented by a rounded border to make the representation of the steps clearer. Finally, more recommended dishes of the same cuisine are shown in the form of four palace grids, and more pictures of the recommended Chinese cuisine are specially designed to make the whole page more beautiful.

Figure 3-5

Figure 3-6

Figure 3-7

4.3 Community page

In the community page, the top is the search bar, the layout of the bottom post display area is a line of two recommendation posts, each post shows the cover picture set by the user, the user's image and nickname and the number of likes, the page layout is clean, and the important information is simple and neatly displayed on the page.

Figure 3-8

4.4 Personal Settings Page

In the personal setting page, we set up another red and white tone background picture, which once again reflects our Chinese food culture has a long history, broad and profound feeling. The head picture is presented in a circle at the top of the page layout, and the chart of each column below is shown in red, which is consistent with our theme color.

Figure 3-9

5 Key Technologies

5.1 Primary technologies

5.1.1 SQLite

To accomplish the function of registering and remembering the user's username and password, database is used. The database we choose is SQLite.

SQLite is a very lightweight relational database system. SQLite does not require installation settings or processes to start, stop, or configure before use, while most other SQL database engines act as a separate server process and are communicated by the program using some internal process (typically TCP/IP) to complete the task of sending requests to the server and receiving query results. SQLite does not adopt this working method. When using SQLite, the program accessing the database reads and writes directly from the database files on the disk, without any intermediate server processes. To use SQLite, you usually only need to bring a DLL to use its full functionality.

In our application, a kotlin file is generated to create a database and a table is created to store user login information, including two entities: username and password. On the registration page, the registration button is set with a click event, and clicking this button will store the user's input data in the database. The login button is also set with a click event. When the button is clicked, the table storing user data will be traversed. If there is matching data, the login is successful, and vice versa.

5.1.2 Animation

In our application, three types of animation are used, including translating animation, scale animation and fade animation. The animations are stored in the specific directory which name is “anim”. The essence of animation is the switching of different forms of objects over a period of time. Firstly, create two corresponding XML files for animation effects in the “anim” folder, and then run the overridePendingTransition method in the kotlin file to run the animation.

We used translating animation and fade animation in the page switching. We have set up a scaling animation on the button, and clicking the button can achieve scaling, which looks like clicking on a ball.

5.1.3 Fragment

In Android Studio, Fragment is a reusable UI component that represents a portion of the interface within an Activity. It is similar to an Activity, but can be added, deleted, and replaced, and can be combined with other Fragments to form a more complex UI interface.

In our application, we used it to achieve navigation bar page redirection. First, create the corresponding fragment in a folder called Fragment, and then rewrite the onSupportNavigateUp() method in the activity's kotlin file to achieve page redirection. Another thing that worked was called mobile_ The XML file for navigation is stored in a folder called navigation. In this XML file, the fragments and activities that need to be redirected are listed. It is this file that plays a role in the activity's kotlin file.

5.1.4 Adapter

We use adapter to accomplish the personal page and cuisine list page. The principle of implementing the adapter is that the data model M is responsible for storing data and displaying it on the corresponding view V through the controller C.

Two steps to accomplish the adapter:

- Create adapter class: Create an adapter class that inherits from RecyclerView. Adapter. The adapter class is responsible for binding the data model to the list item layout and displaying the list items in RecyclerView. In the adapter class, implement necessary methods such as onCreateViewHolder, onBindViewHolder, and getItemCount.
- Initialize RecyclerView: In the activity class of the settings page, use the findViewById method to obtain a reference to RecyclerView and create an adapter instance. Then, set the adapter to

RecyclerView to display the setting options.

A framework is built in the adapter class, and then use specific data references in the activity's kotlin file.

5.2 Technical challenges

5.2.1 Search bar

We encountered some difficulties in the combination of searchview and listview. Initially, I tried to create a search box by searching for information online, but the results were not satisfactory and could not present a qualified search box. At the same time, the search box should have a memory function. During the development process, we attempted to create a search box with memory function using an adapter, but it was never possible to achieve it.

5.2.2 Integration of third-party libraries and APIs

During the development process, it may be necessary to use third-party libraries and APIs to implement specific functions or services. Integrating these libraries and APIs may involve dependency management, permission configuration, and appropriate usage methods, which require careful reading of the documentation and practice.

6 Testing and User Experience Analysis

6.1 Testing

- Functional uptime: During testing, individual functions and features are verified and work properly. Whether it is user interface interaction, data processing, network communication, or other functional modules, they all operate as expected and produce correct results.

- Data Consistency and Accuracy: The data in the project is consistent across modules and the data is processed and converted accurately. Data read, storage, and update operations are performed correctly, with no data loss, corruption, or errors.

- Good responsiveness: The project has good responsiveness under user operation. User-interactive operations, such as clicking buttons, scrolling lists, etc., are able to respond in a timely manner and provide a smooth user experience without significant delays or lagging.

Stability and robustness: The tested project performs stably under different usage situations and can handle various exceptions. The project did not crash, flash back or have serious errors, and was able to handle and recover from abnormal situations correctly to ensure the safety of users' data and operations.

- Compatibility and Adaptability: The project was tested on different devices and OS versions and was able to adapt to various screen sizes and resolutions. Whether on different Android devices or

on different Android versions, the project can run properly and demonstrate good adaptability.

- Security: During testing, the security of the project was verified. The project has no obvious security vulnerabilities, such as unauthorized access, data leakage, illegal operations, etc., and complies with the relevant security standards and requirements.

6.2 User Experience Analysis

- User interface friendliness: User feedback indicates that the user interface of the Android Studio project is designed to be intuitive, clear and easy to use. The layout and organization of the various functional and operational elements make it easy for users to find the tools and options they need, improving productivity.

- Feature richness and flexibility: Users rated the Android Studio project positively for its functionality and features. They found that the project provided a rich feature set that could meet a variety of development needs. Also, users appreciated the project's flexibility to adapt to different development scenarios and needs.

- Responsiveness and performance: Users observed that the Android Studio project is fast and responsive when operating and executing tasks. The performance of the project was good, able to handle large scale code and resources and remain efficient during compilation, build and run.

- Debugging and Troubleshooting Capabilities: Users found Android Studio to provide powerful debugging and troubleshooting capabilities. The project effectively helps users locate and resolve problems in their code, providing useful debugging information and tools to make the development process more efficient and smooth.

- Documentation and Support Resources: Users appreciated the Android Studio project's documentation and support resources. They found that the project provided detailed documentation, tutorials, and sample code to help users better understand and apply the various features and technologies.

- Community support and update cycle: Users noted that the Android Studio project has a large developer community with access to a wide range of support and help. In addition, they think the update cycle of the project is reasonable, and it can introduce new features and fix bugs in time to keep up with the latest technology.

In summary, the results of the user perception analysis show that the Android Studio project performs well in terms of user experience. Users rated the user interface friendliness, feature richness, responsiveness and performance, debugging capabilities, documentation and support resources, and community support and update cycles positively. These results reflect the success of the project in meeting user needs and providing a good development environment.

7 Conclusion

In Conclusion, the objectives of the app design were achieved and well evaluated to assess the key aspects of the design, such as functional implementation, performance requirements, and user experience, and to assess that the design met the original intent and expectations of the project. Design decisions are justified: important decisions made during the design process of the project and the reasonableness of these decisions is evaluated. Selected specific technologies, architectures, and design patterns, and assessed that these decisions were consistent with project requirements and selected as best practices. By evaluating the app features and modules and analyzing the design and implementation of each feature and module, the application meets the expected functional requirements and demonstrates their interrelationship and collaboration. app performance and scalability: the performance under different loads and user volumes demonstrates that the project has good scalability and is able to cope with future expansion and growth needs. For Risk and Challenge Analysis: Identifies and analyzes potential risks and challenges in the project design. It provides recommendations and strategies to address these risks and challenges, and assesses their impact on project implementation and success.