

**Software Design Document**



2023-6-16

group 4

Name: 杜卓航、李苛垚、黄熙晋、鲍忠正

Content

[Introduction 1](#_Toc138276866)

[Requirements Specification for PetDoctor App 2](#_Toc138276867)

[User Requirements 2](#_Toc138276868)

[Functional Requirements 3](#_Toc138276869)

[Overall design 4](#_Toc138276870)

[Primary functions 4](#_Toc138276871)

[User Interface Design 5](#_Toc138276872)

[Main Pages 5](#_Toc138276873)

[Key Technology used 8](#_Toc138276874)

[Bottom navigation button 8](#_Toc138276875)

[Getting data from register page 8](#_Toc138276876)

[Bottom navigation 9](#_Toc138276877)

[Web chat room direction 11](#_Toc138276878)

[Testing 13](#_Toc138276879)

[User Experience Analysis 13](#_Toc138276880)

[Conclusion 15](#_Toc138276881)

# Introduction

Pet Doctor is designed to provide convenient and accessible veterinary care for pet owners. Through our online consultation platform, users can connect with experienced veterinarians who can provide professional advice and guidance for their pets' medical needs. Our comprehensive and reliable online medical system ensures that pet owners can receive timely and accurate diagnoses and treatment plans without the need to leave their homes This not only saves time and effort for pet owners but also ensures the well-being and health of their beloved pets.

# Requirements Specification for PetDoctor App

## User Requirements

**Registration and User Profile:**

- Users should be able to create an account by providing basic information such as name, email, and password.

- Users should have the option to create and manage profiles for multiple pets, including details such as name, species, breed, age, and medical history.

**Communication with Veterinarians:**

- Users should be able to initiate and maintain conversations with veterinarians through secure and private messaging.

- Users should have the ability to share relevant media files such as images or videos related to their pet's condition.

**Access to Pet Health Records:**

- Users should have a dedicated section to view and update their pet's medical records, including vaccinations, allergies, medications, and past consultations.

**Notifications and Reminders:**

- Users should receive timely notifications and reminders for upcoming appointments, medication schedules, and important updates from veterinarians.

## Functional Requirements

**Veterinary Profile and Availability:**

- Veterinarians should be able to create and manage their profiles, including their specialization, experience, and available time slots for consultations.

- Veterinarians should have the ability to set their availability status and update it in real-time.

**Secure Messaging System:**

- The app should provide a secure and encrypted messaging system to ensure the privacy and confidentiality of user-veterinarian conversations.

**Payment Integration:**

- The app should integrate a secure payment gateway to facilitate transactions for consultation fees and other related services.

**Feedback and Ratings:**

- Users should be able to provide feedback and ratings for veterinarians based on their consultation experience.

**Admin Dashboard:**

- An administrative dashboard should be available for system administrators to manage user accounts, veterinarian profiles, and overall system performance.

# Overall design

## Primary functions

**User registration and login:** Users can establish personal information and pet files by registering an account and logging into the application, in order to better manage and record the health status of their pets.

**Pet information management:** Users can add their own pet information, including breed, age, gender, weight, etc. They can update their pets' health records and vaccination status at any time, facilitating daily care and medical management.

**Online consultation:** Users can ask pet doctors questions and receive professional advice through the application. They can describe the symptoms or problems of pets and upload relevant photos or videos for doctors' reference. Pet doctors will respond to user inquiries as soon as possible and provide appropriate treatment suggestions.

**Main structure:**

1. Login page

2. Personal page

3. Main page

4. Navigation Page

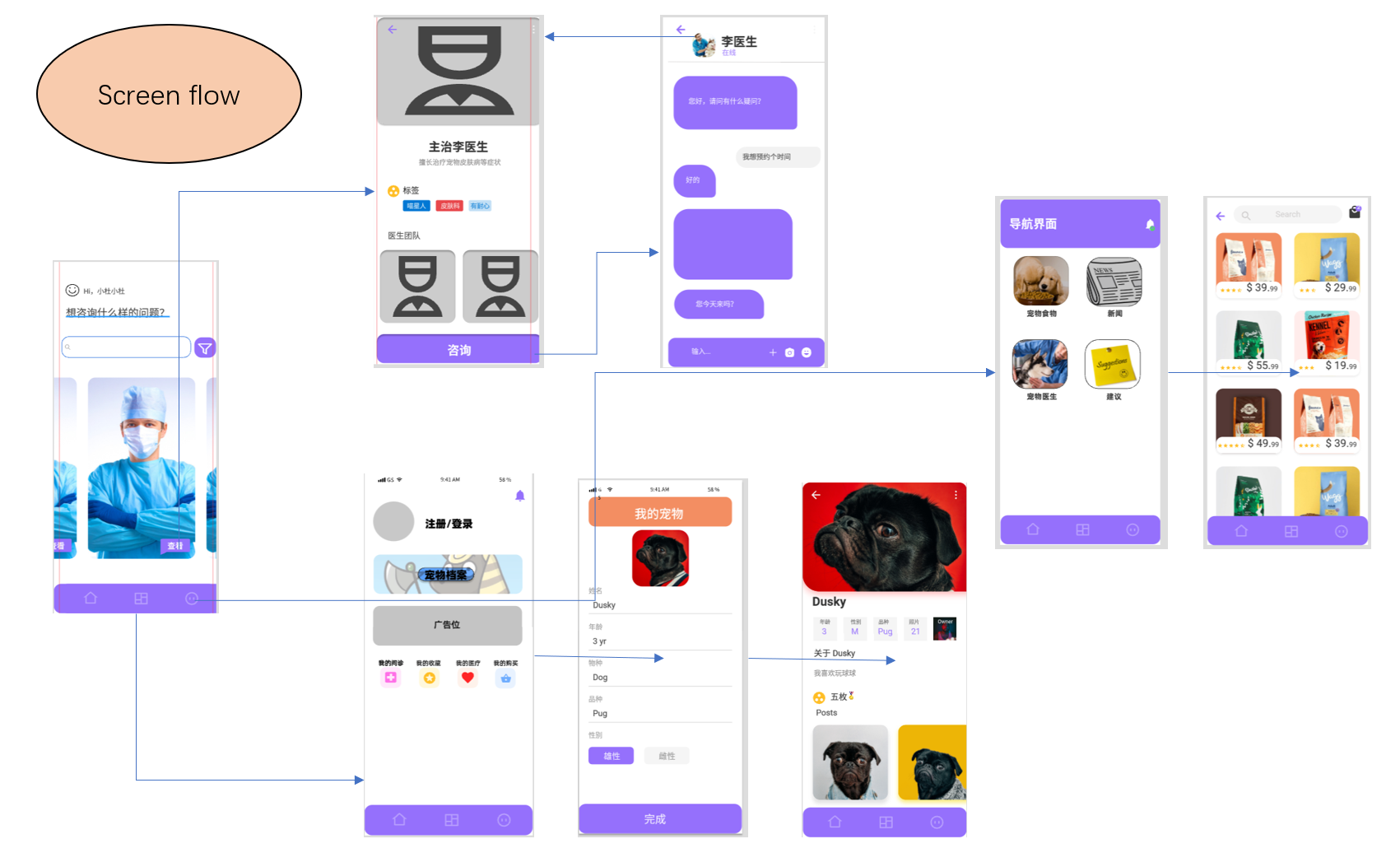
5. Doctor Page

6. Chat Page

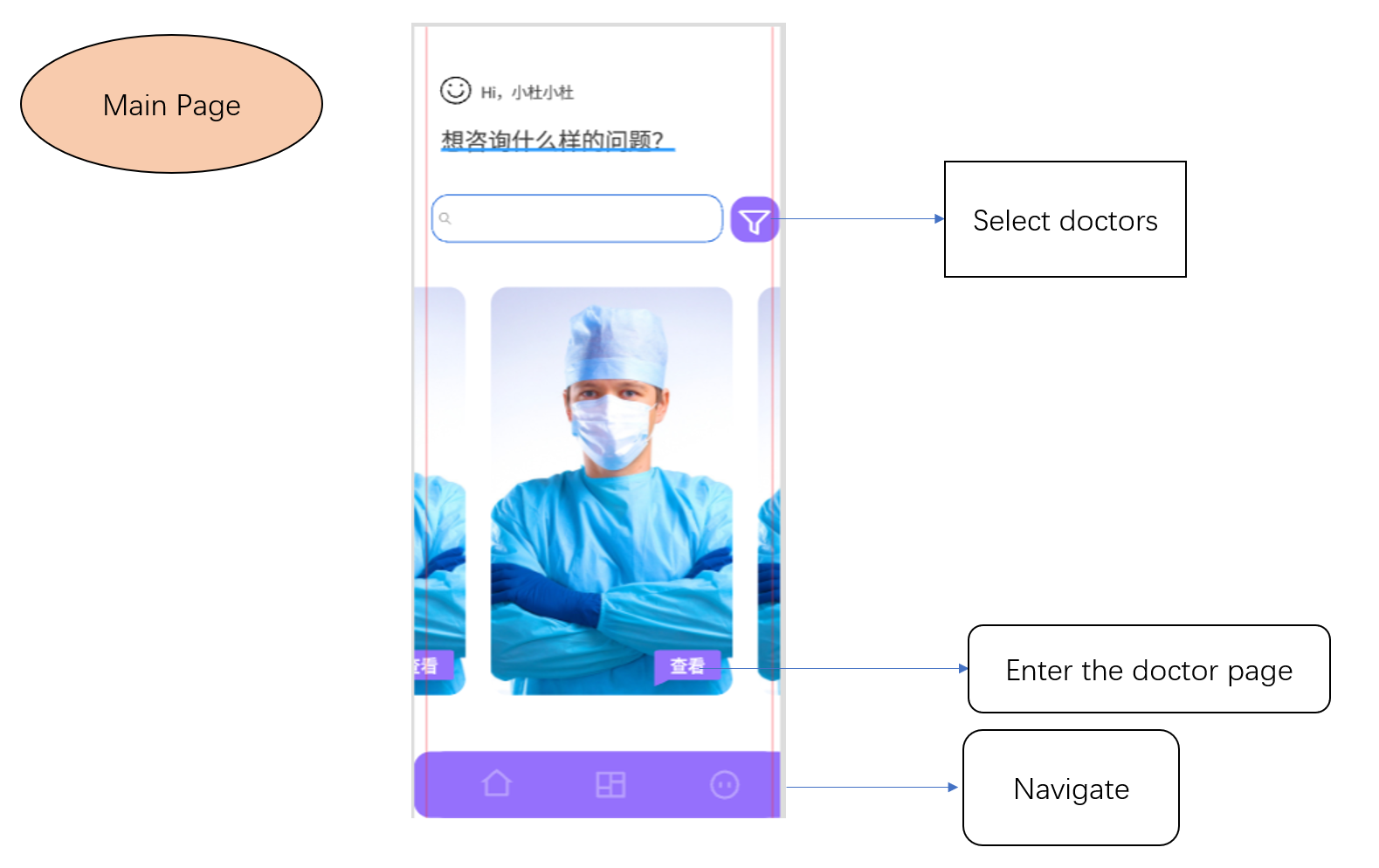
# User Interface Design

## Main Pages

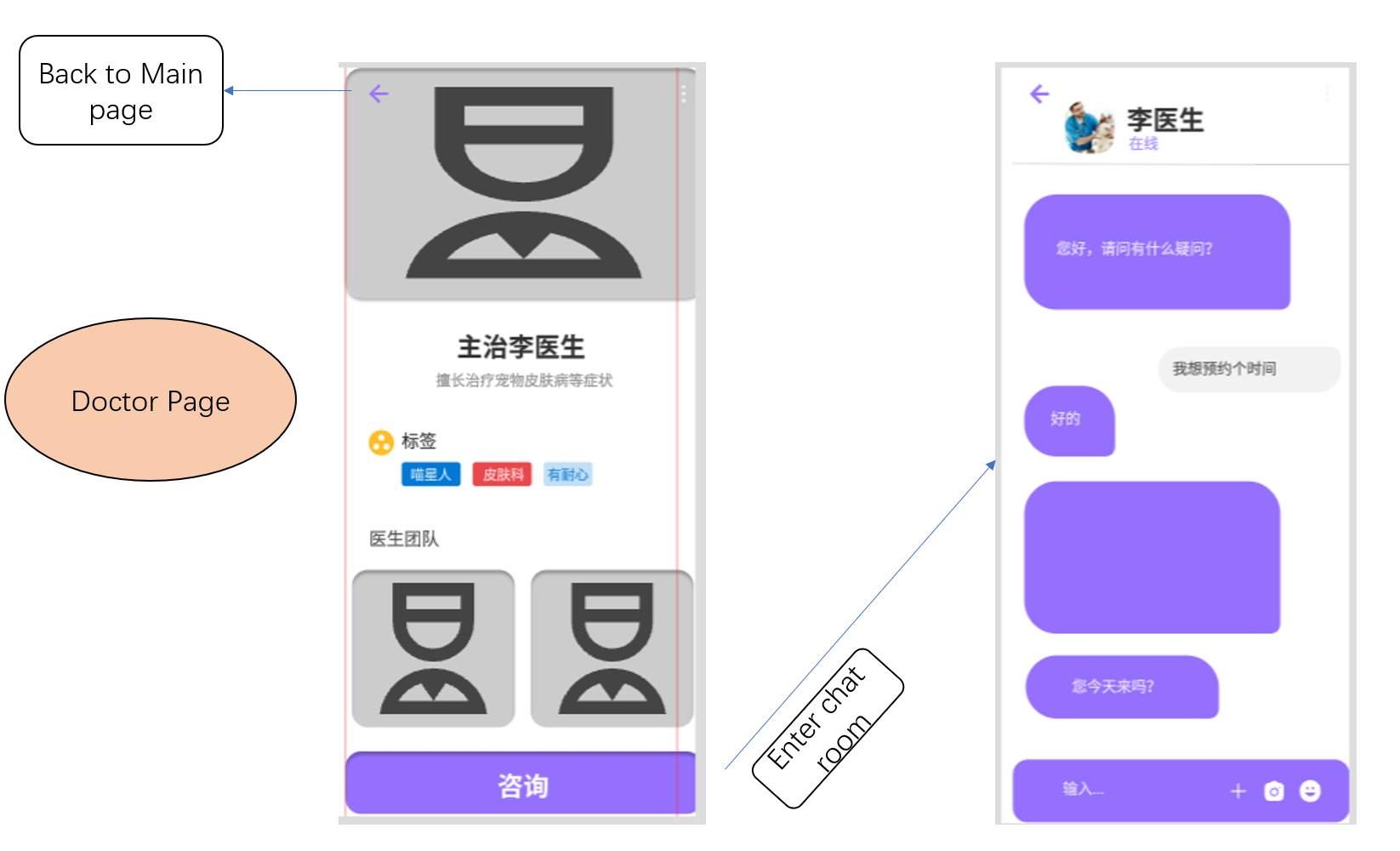
#### User flow



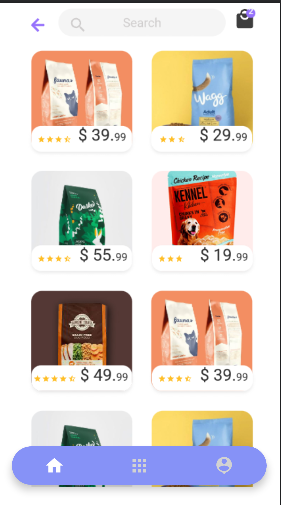
#### Main page



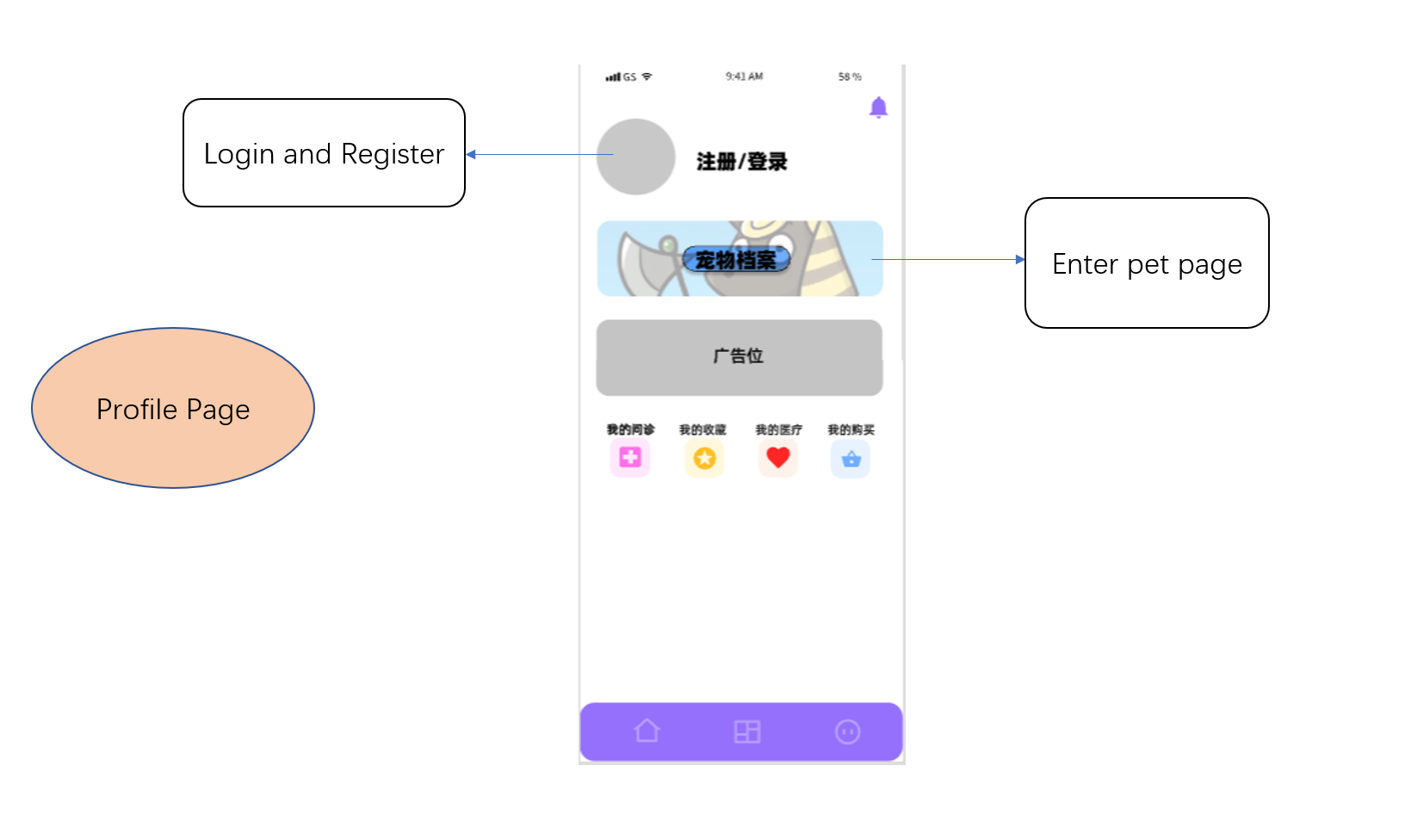
#### Doctor page



#### Purchase Page



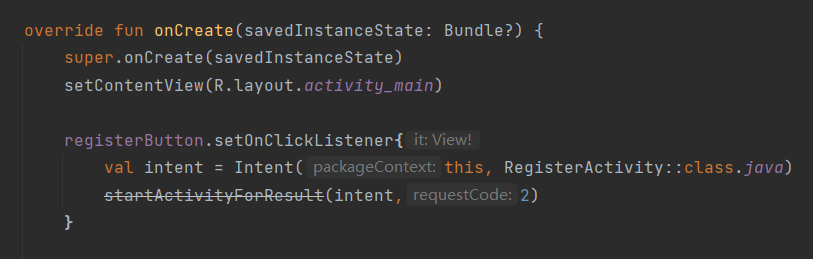
#### Profile Page



# Key Technology used

## Bottom navigation button

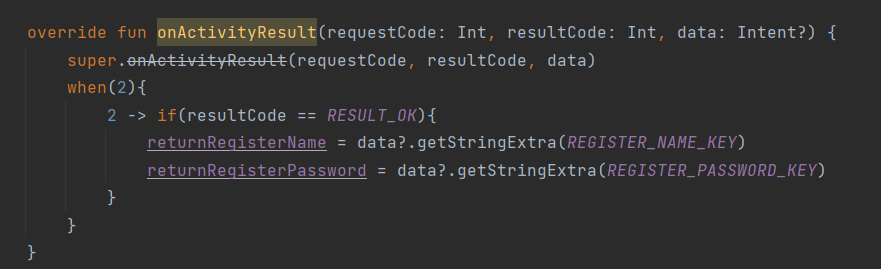
## Getting data from register page



In the Login page, you may notice that I didn’t use startActivity(intent) to jump. Instead, I use startActivityForResult(). That is because I hope to get the register username and register password transported back by register page. And the requestCode is 2 which respond to the register page.

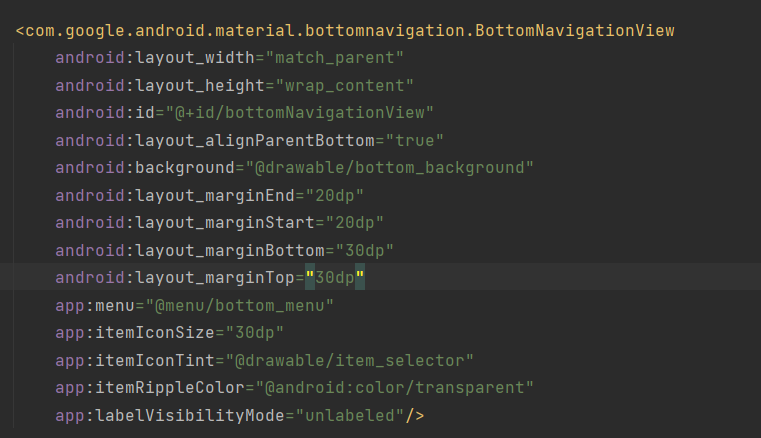


After jumping to Register page, it will start requiring the username as well as the password which user enter in the register page. In order to store the data which will be sent back to login page. The data will be stored into variable registerIntent with the help of putExtra() function.

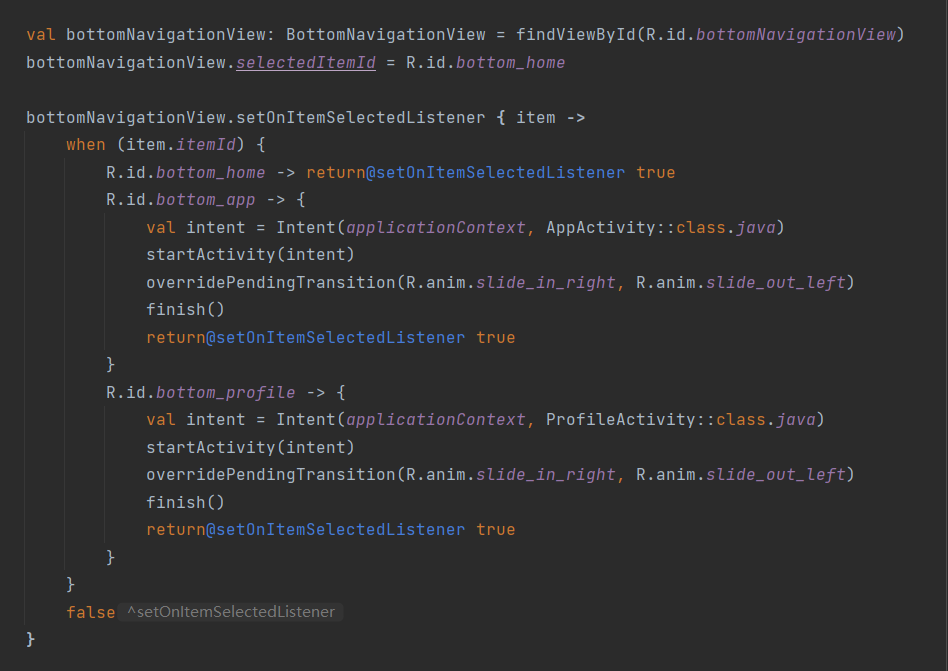


Back to Login page.Since I use startActivityforResult() to launch the Register page, it will callback onActivityResult() once the RegisterActivity was shutdown by finish() function. As a matter of fact, I need to override the onActivityResult() function to require the data from Register page. So, in this function, the requestCode is 2 which means the request is from Register page. And if the resultCode is RESULT\_OK. Then it will use getStringExtra() function to get register username and password corresponding to different keys. So that is how the login page get the username and password from register page.

## Bottom navigation

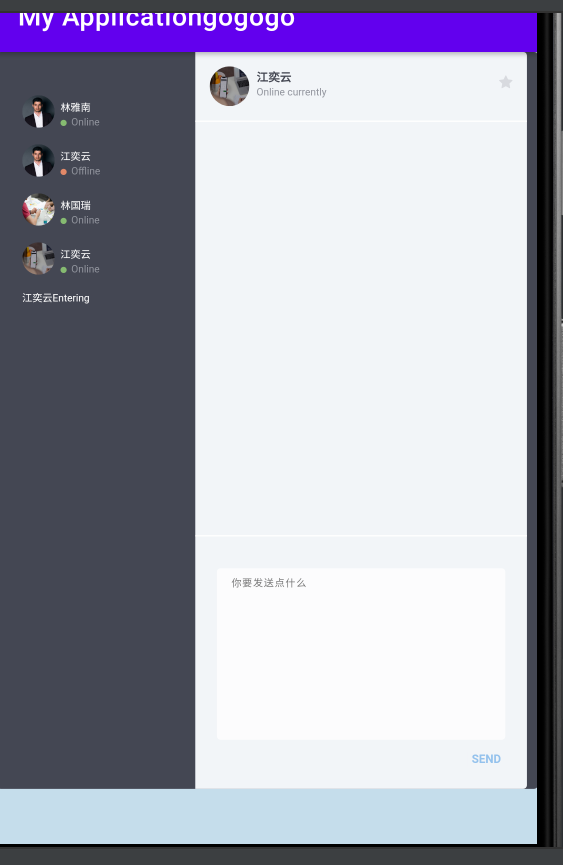


We use fragments to create a bottom navigation



How to direct to different page and how it turn bright when clicking the button

## Web chat room direction

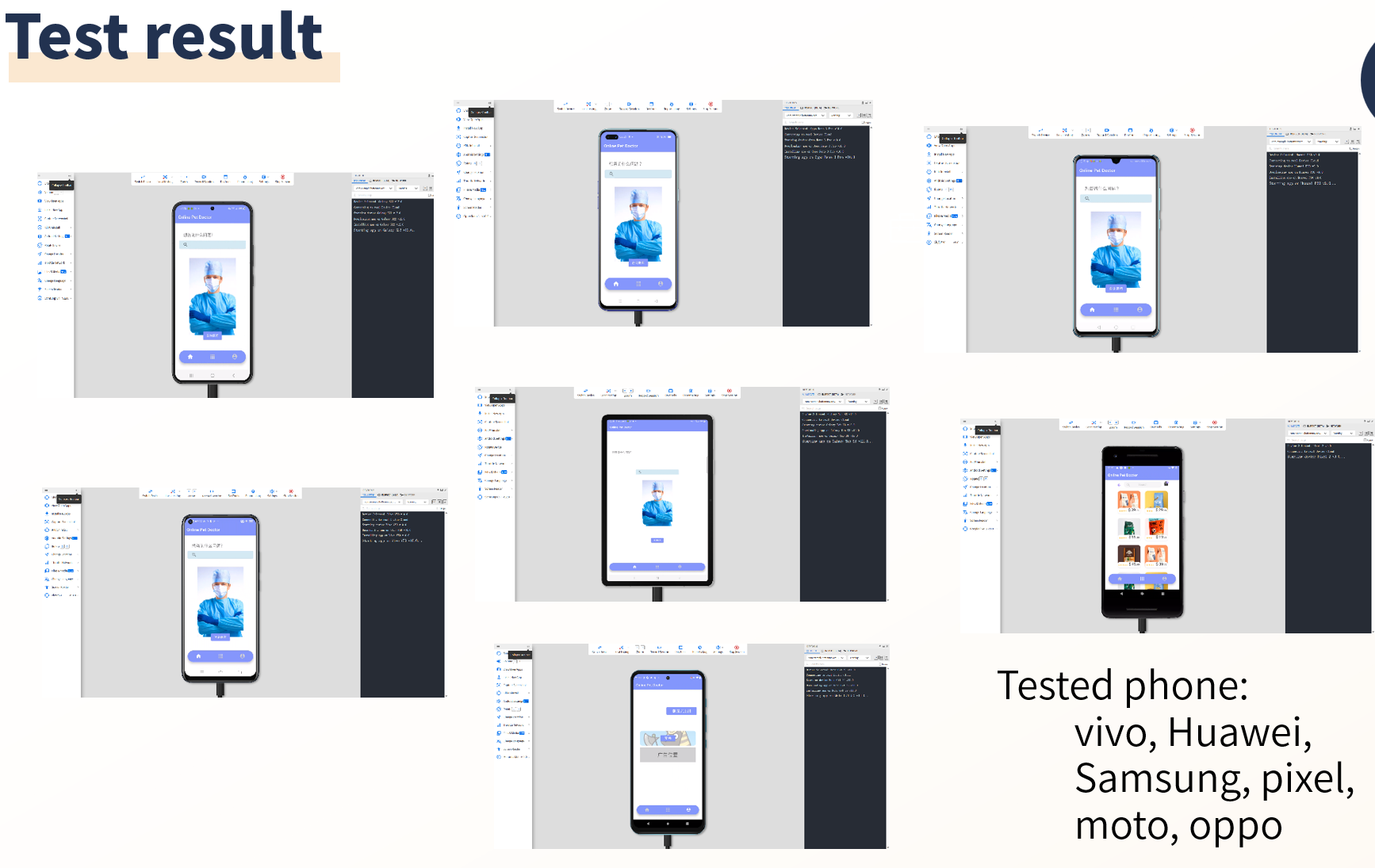


This is a chat room hosted on the local server, when clicking the chat button, it will direct to this page, and the below codes are the processes of how to reveal.



In order to get a web page, we import webkit to load the page.

# Testing

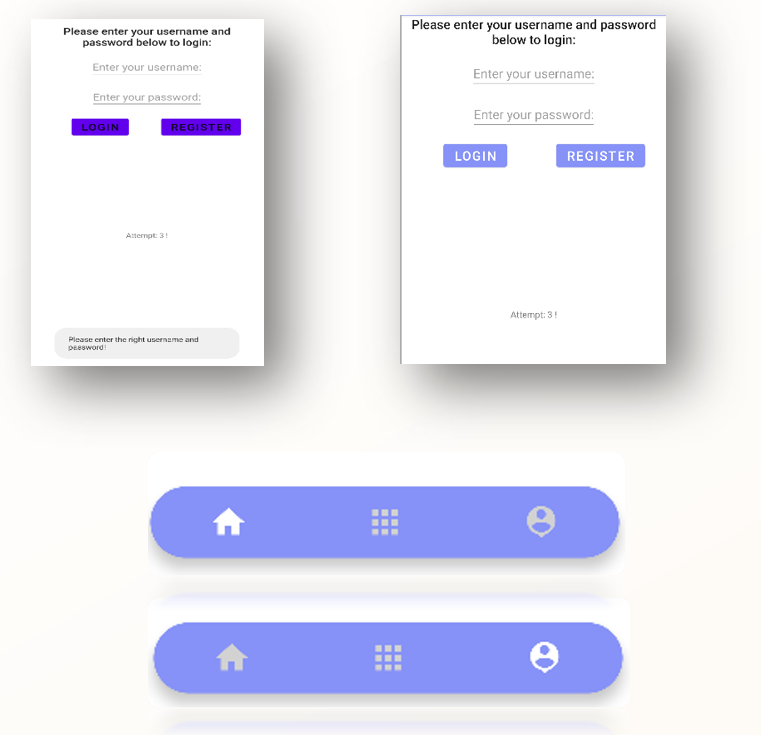


We test the app manually, most of the versions of phones of different brands have completely passed the test. We also tested the pad and mini pad. All the equipment we tested have passed the compatibility testing. But since we tested manually and the test tool is free, some latest versions of equipment didn’t available for us to test yet. Thus the test has limitation.

# User Experience Analysis

After asking for our users, we gather some of their opinions:  
1. The bottom color is too dark to see

2. We don’t have anime when clicking the bottom navigation originally.



Thus, we adjust the entire color style of our app, we set the originally dark purple to light purple color. And to address the anime problem, we use fragments to create a dynamic bottom navigation button so that when user clicking the button, it will change to different pages with a anime. Also when clicking, the button will turn from dark color to bright color.

# Conclusion

After a whole term efforts, we made our first app with the efforts of all group mates. There exist achievements as well as difficulties. But most of the problems were addressed successfully by us. In general, this is a successfully group work.

About the achievements and challenges:

Each of our group members has learnt how to use Android Studio and know not only key part of Kotlin language as well as how to use Kotlin to write a program. Although some of our group members are not good at programming at the beginning, they managed to solve the problem. And we also learnt how to make our app look better through testing and asking for our users opinion. Moreover, we know how to work as a team, without teamwork, we can’t achieve so far.

About future:

There exist some problem we failed to address

1. The card layout. We failed to put the card layout into our app. We need to learn more about this part of knowledge.
2. We still need to add more functions to make our app more useable. For example, the function for users to reserve time for doctors.
3. There exists some drawbacks of existed functions. For instance, our web chat room. Although it can run on our app with local server. It is actually a socket.io package we learnt from web application course. Without initializing the server on the local server, users may fail to visit our web page.

Finally, thanks to all the efforts of the group mates as well as the assistance of teacher, we finally put forward a useable version of app.