**Login/Register/Login**

Normally, the user interface of login page should be shown as Figure.1. The users can input their username and password, and then login the system. In the meanwhile, when users type the username and click the “Next” button on the virtual keyboard, the cursor will automatically change to password input box. Additionally, the users can register new accounts by clicking the link “Register New Account”. This page is developed at “login.dart” which is under the library “pages/account”.

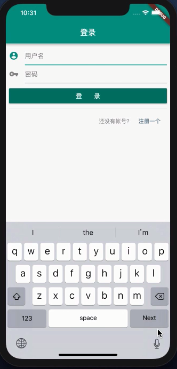
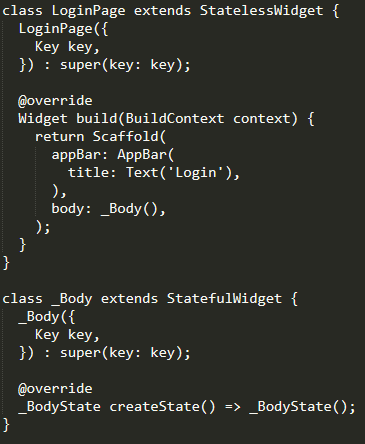
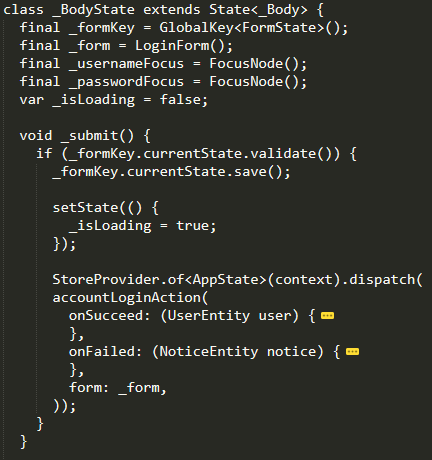


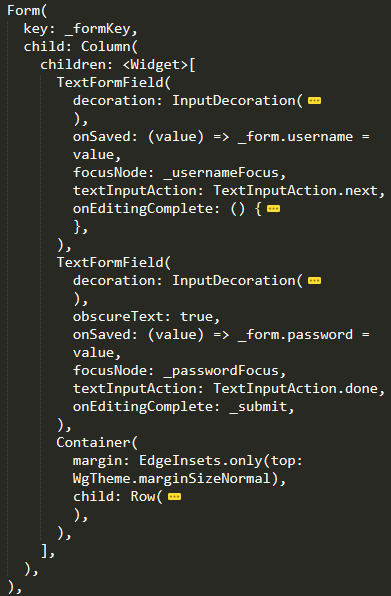
Figure.1 The user interface of Login page

Comment: This part of codes mainly defines the several basic elements of login page including title and body. The title is text “Login”, and a function \_Body() would be loaded in the body of page. In the meanwhile, the function \_Body() is the construction method of class \_Body. Additionally, the \_Body() requires a new method \_BodyState() to support.

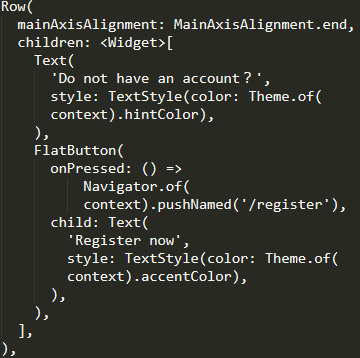




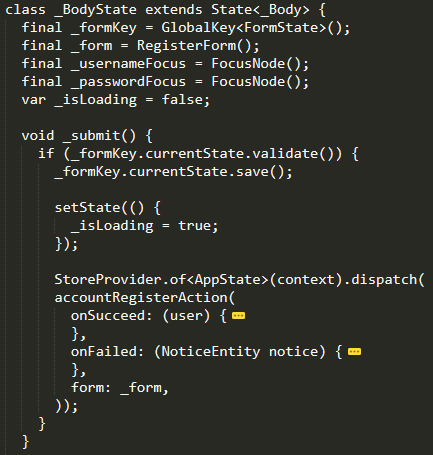
Comment: This part of codes describes the class \_BodyState. In this class, it defines some variables in order to manage the form and page. In the meanwhile, it provides a function \_submit() which set the action of form and the page. When users click the “Login” button, the form which contains username and password information will be uploading and page will display the uploading state for users.



Comment: This part of codes work for designing the input boxes and login button. The first TextFormField represents the username input box; the second TextFormField works for the password input box; the Container defines the attributes of login button. When clicking the button, the program will jump to the \_submit() function which is in the \_BodyState class.



Comment: This part of codes work for designing the register link. The Text is a message which reminds the users. The FlatButton provides a link which will jump to Register page.

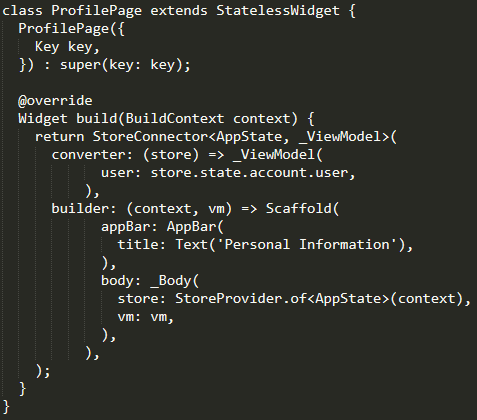
Similar to Login page, the Register page has the same user interface. The only difference is the action after clicking the Register button.

Comment: This part of codes defines the action after clicking the Register button. Compare with Login page, the action accountLoginAction() is replaced by accountRegisterAction(). Both of two action is defined in the account.dart which is under library “action”

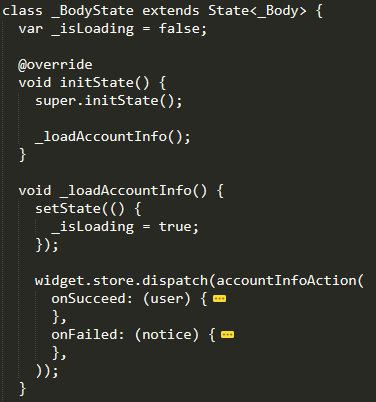
When users log in the system successfully, they can logout the system. The Logout button is displayed in the “me.dart”. After clicking it, the application will do the function accountLoginAction() which is in the file account.dart.

**User Information Edit**

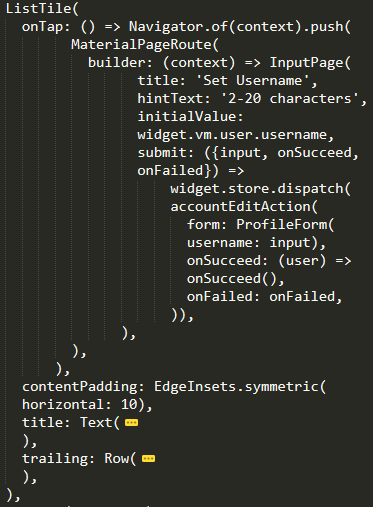
The user information has only two parts in this system which are username and phone number. It will display these information in the page “profile.dart” which is under library “pages/account”. However, this system only provides the function to edit one information once. It does not support change several things in the same time. The Figure.2 shows these three pages.



Comment: Similar to the Login and Register pages. The user information page defines several basic elements and the body of page is displayed by the function \_Body(). Similarly, the function \_Body() is the construction method of class \_Body. Additionally, the \_Body() requires a new method \_BodyState() to support.

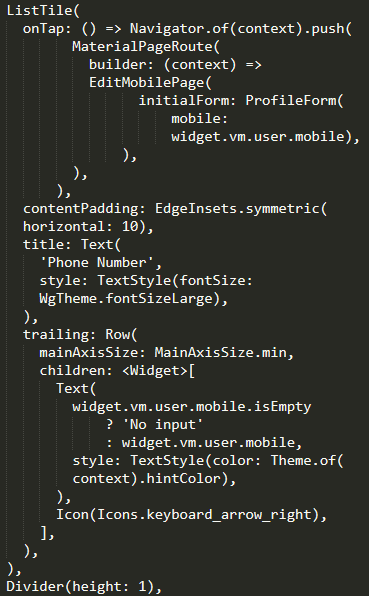


Comment: This part of codes mainly describes the class \_BodyState. When users enter the user information page, the username and phone number will be loaded through the function accountInfoAction() which is defined in the file “account.dart”.



Comment: This part of codes works for design the user interface of the user information page. In this part, it displays the username. When users want to click the username, the context will be passed to the input page which is under the library “pages/common”. The input page is a common page and its function is showing information. After that, when users enter the new username, this page will do the function accountEditAction() which is defined in the file “account.dart”.

Comment: This part of codes works for design the user interface of the user information page. In this part, it displays the phone number. However, different from changing username, the phone number will be passed to EditMobilePage which is in the file “edit\_mobile.dart” under library “pages/account”. After that, the users can input a new mobile number and the system will send verify code to the phone. This function is defined in “edit\_mobile.dart”. After that, if the verify code is correct, the number will be changed through the method accountEditAction().



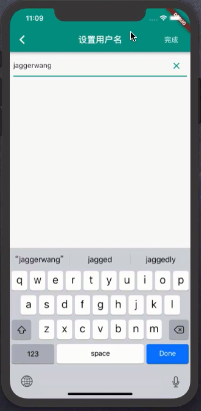
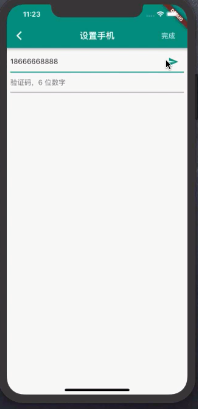
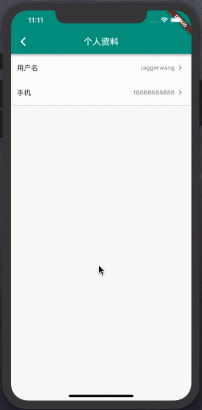


Figure.2 The user information display page and two edit information pages.