SITCoin Wallet

CSC2007 Mobile Application Development Spring 2022

Creating a login screen

Fork the repo **csc2207-quiz01-2022** and inspect the code within the project.

The goal of this section is to create a login screen that accepts two fields: student number and password. The user enters the student number in the first field, and a password in the second field. The “Login” and “Clear” buttons are below the text fields

Design the layout of the `LoginActivity` similar to the given screenshot, to display a login screen. Use the provided image file in the Android project under `/res/drawable/`

Hitting the “Login” button will validate the student number, and password. Hitting the “Clear” button will clear the values of the text fields.

Ensure that the text fields have a hint value entered in them. Please ensure the IDs are correct within the XML**.**

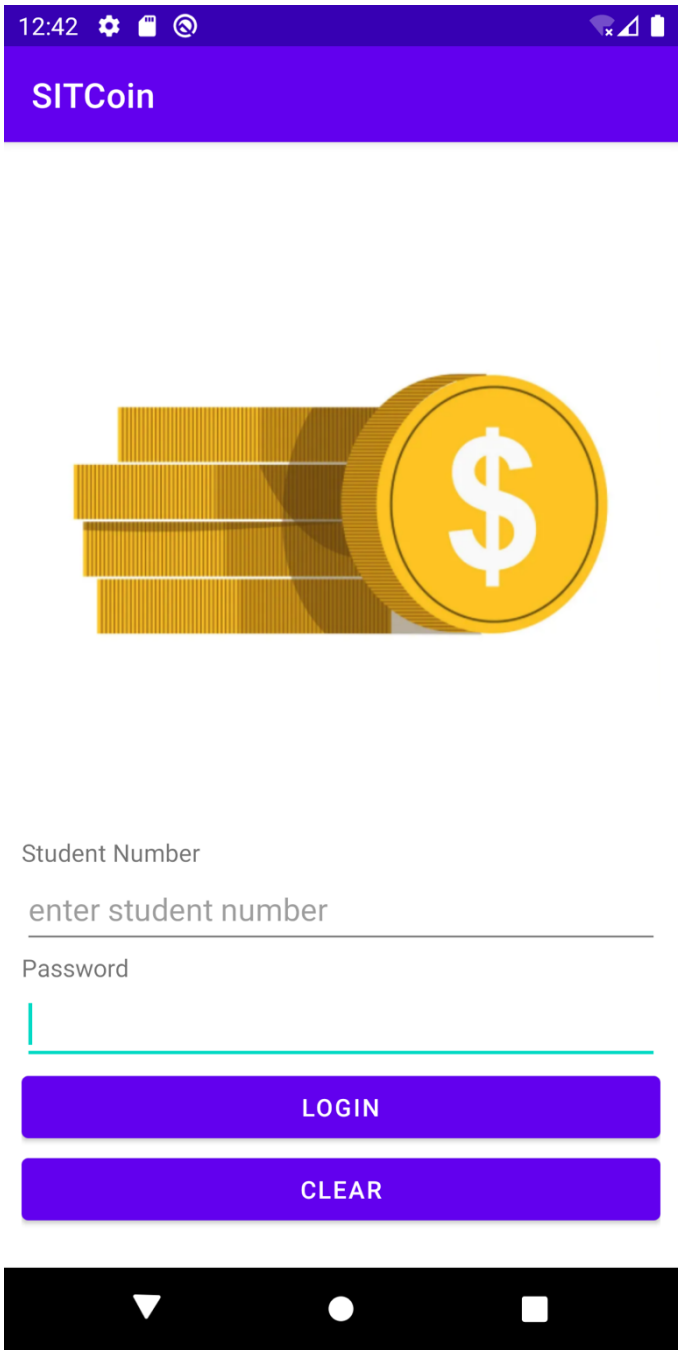
Note: The password text field should hide the entered password upon entry.

Note: Do not use the auto generator tools within Android Studio. Create the screens from scratch.

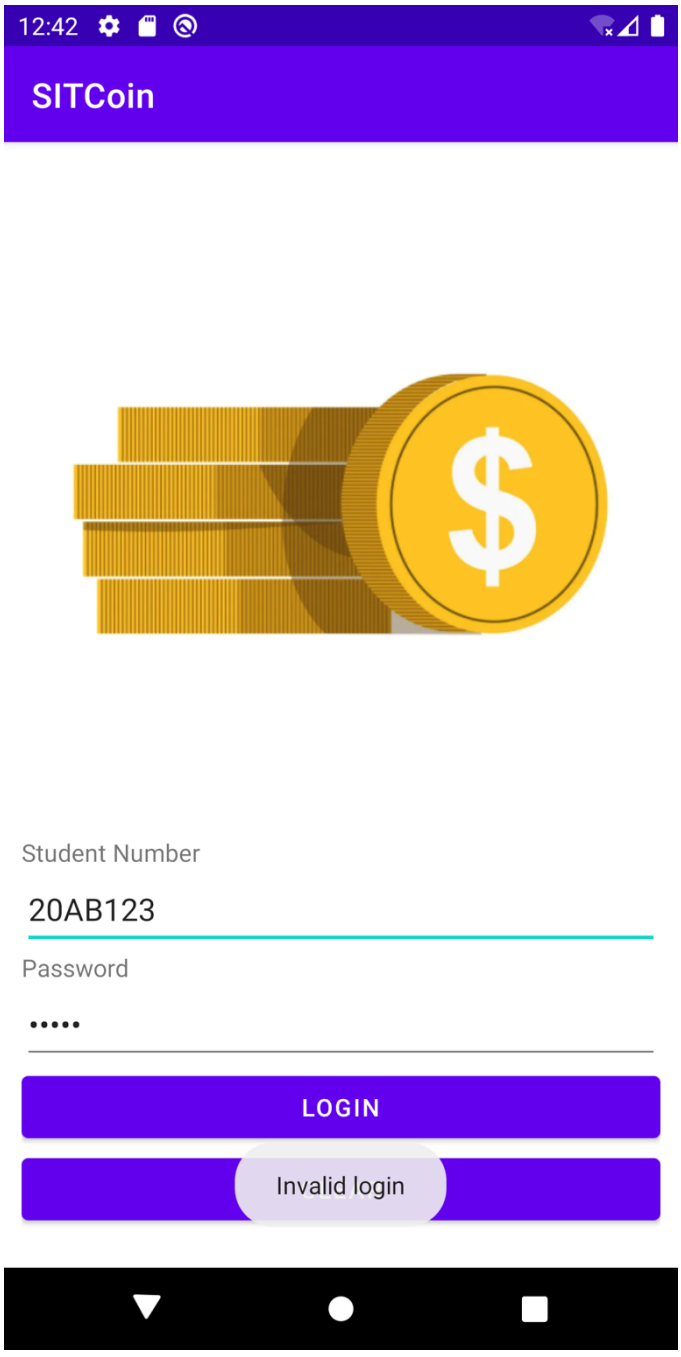
IMPORTANT: Ensure there are no spelling errors on the button text.

IMPORTANT: Ensure the following IDs within the XML:

- **id for the student number is editTextStudentNumber**
- **id for the password is editTextPassword**
- **id for the coin logo is imageViewCoin**
- **id for the login button is buttonLogin**
- **id for the clear button is buttonClear**



LoginActivity



LoginActivity with invalid login

Check the validity of the fields entered

On the login screen, hitting the “Login” button will first validate the student number. Only when the student number is valid, then is the password is checked.

If the student number is invalid, a “toast” popup message will appear with the string “Invalid Student Number”.

Implement the following logic to check for **valid student number format**:

- Exactly 7 numerical digits

The logic check should be implemented within a method in `LoginActivity` called:

```
fun isStudentNumberValid (studentNumber: String): Boolean {

}
```

The correct password is “yoquierodinero”. Implement the logic to check for the correct password. If the password is invalid, a “toast” popup message will appear with the string “Invalid login”.

This should be implemented within a method in `LoginActivity` called:

```
fun isPasswordValid (password: String): Boolean {

}
```

Examples on how to create a toast message are here:

http://developer.android.com/guide/topics/ui/notifiers/toasts.html

Note: Obviously this is insecure but for the purposes of the quiz we will ignore this for now.

Main screen on successful login

On successful login, the user is directed to the next screen where a welcome message and the student number is displayed, along with an image and a value of **10 SITC**.

The “Send SITC” and “Receive SITC” buttons are below the image.

The student number displayed is not hardcoded and will depend on what is entered on the login screen. Implement the logic to display the successful login screen.

Implement the screen `MainActivity` to display the welcome message on successful login, along with image, coin value of **10 SITC**, as depicted in the screenshots. Use the provided image files in the Android project under `/res/drawable/` to create the screen.

Use the strings provided in `strings.xml` to assist in your implementation.

IMPORTANT: Ensure the following within the XML for MainActivity:

- **id for the welcome message is textViewWelcome**
- **id for the SITCoin value is textViewValue**
- **id for the image is imageViewWallet**
- **id for the Send button is buttonSendCoin**
- **id for the Receive button is buttonReceiveCoin**
-

MainActivity on successful login

Receive coin

Design the layout of the `ReceiveCoinActivity` similar to the given screenshots.

Use the provided image files in the Android project under `/res/drawable/` and the strings provided in `/res/values/strings.xml` to create the screen.

For now the receive coin screen does nothing else.

IMPORTANT: Ensure there are no spelling errors on the all the text and the buttons.

IMPORTANT: Ensure the following within the XML for ReceiveCoinActivity:

- **id for the message is textViewReceiveCoin**
- **id for the QR Code image is imageViewQR**

ReceiveCoinActivity

Send coin

Design the layout of the `SendCoinActivity` similar to the given screenshots. Use the provided image files in the Android project under `/res/drawable/` and the strings provided in `/res/values/strings.xml` to create the screen.

For now, don't check the validity of the student number text field.

Check for invalid amounts of SITC sent:

- The amount can be integer only
- The amount sent can only be equal or less than the current wallet value

If the amount is invalid, a "toast" popup message will appear with the string "Invalid amount".

This should be implemented within a method in `SendCoinActivity` called:

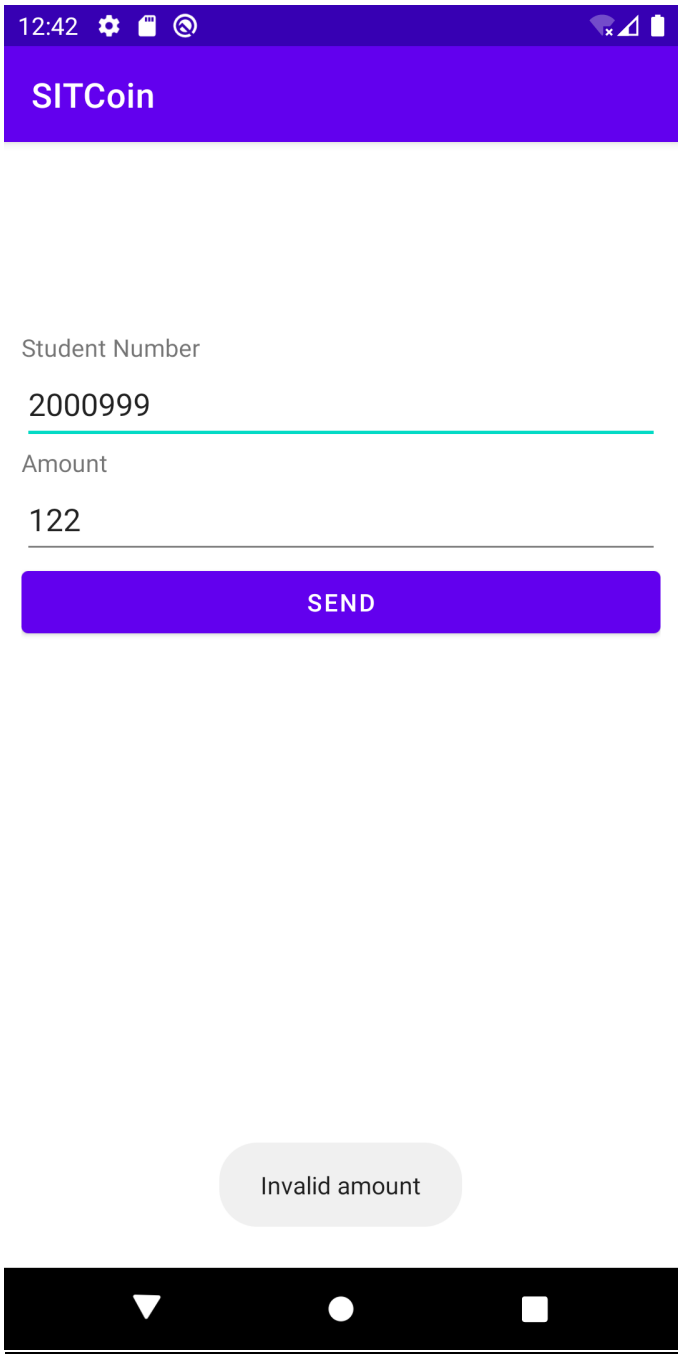
```
fun isValueValid (value: String): Boolean {

}
```

If the amount is valid, a "toast" popup message will appear with the string "Amount sent", and the **SendCoinActivity** will close, bringing the user back to the **MainActivity,**

The MainActivity will then show the remaining amount of SITC left in the wallet.

IMPORTANT: Ensure the following within the XML for SendCoinActivity:

- **id for the student number is editTextStudentNumberSend**
- **id for the amount is editTextAmount**
- **id for the Send button is buttonSend**



SendCoinActivity with invalid amount entered

MainActivity after 2 SITC was sent out

Lab Quiz 1

1. Fork the repo **csc2007-quiz01-2022.**
2. Design the layouts of each of the screens similar to the given screenshots.
3. Implement the missing tags within AndroidManifest.xml to launch the correct activities.
4. Implement the logic on the login screen to check for the validity of the following fields: student number, password. The login screen should be named **LoginActivity**.
5. Implement the Toast popups and messages for invalid entries
6. Implement the logic to clear the fields when the “Clear” button is pressed
7. Implement the logic to navigate to the successful main screen if the verification is correct, this screen should be named **MainActivity**.
8. Implement the logic to navigate to **ReceiveCoinActivity**
9. Implement the logic to navigate to **SendCoinActivity**
10. Implement the Toast popups and messages for invalid coin value
11. Implement the logic to display and subtract the coin amount after sending
12. Remember to comment and indent the code and implement it in a modular fashion, using different methods or classes if appropriate. Ensure it conforms to Kotlin coding conventions.
13. Commit and push all changes to your forked repository **csc2007-quiz01-2022.**

IMPORTANT: Do not change the activity names, method names, method signatures or package name. Ensure there are no spelling errors on all the text. Remember to commit and push to repo!

Grading Criteria

Total: 15 marks

1. **LoginActivity** layout and UI elements similar to screenshot (1 mark)
2. **MainActivity** layout and UI elements similar to screenshot (1 mark)
3. **SendCoinActivity** layout and UI elements similar to screenshot (1 mark)
4. **ReceiveCoinActivity** layout and UI elements similar to screenshot (1 mark)
5. Logic on the Clear button to clear the fields (1 mark)
6. Logic to check student number validity: Exactly 7 numerical digits (1 mark)
7. Toast if credentials are invalid (1 mark)
8. Navigate to successful main screen based on valid login (1 mark)
9. Display welcome message with student id entered from login screen (1 mark)
10. Implement the logic to display **ReceiveCoinActivity** when the “Receive SITC” button is pressed (1 mark)
11. Implement the logic to display **SendCoinActivity** when the “Send SITC” button is pressed (1 mark)
12. Toast if sending amount is invalid (1 mark)
13. Return to **MainActivity** if sending amount is valid, showing subtracted amount (1 mark)
14. Modular code design and good naming conventions for methods and variables (1 mark)
15. Well-commented code (1 mark)

END OF DOCUMENT