University of Washington November 7, 2018
Department of Computer Science and Engineering
CSE 490C, Autumn 2018

# CSE 490C, Programming Assignment 3, Due Wednesday, November 21, 2018, at 11:59 pm

**Designing User Interfaces (UIs) for Low-Literate Users.**

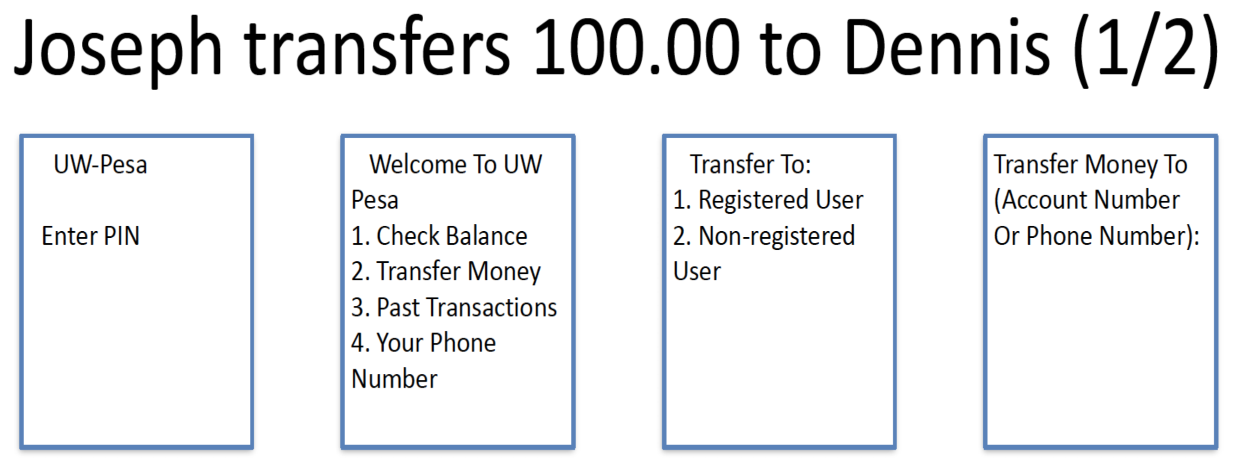
- *For this assignment, you will be creating a working prototype including user-interface for a mobile money system for low-literate users.*
- *You will be required to create the front-end design and images of the mobile money system. For the backend, you will be integrating with UW-Pesa APIs as explained by Clarice Larson in her guest lecture.*
- *This assignment has to be done individually. You will need to submit a .zip file, containing both your code and front-end designs, through Canvas.*

**Choice of Technology** This assignment will focus on design and implementation of a working prototyping, so students can use web technologies (to build a web application) or mobile technologies (like AndroidStudio or iOS/XCode) or Java-based applications to create the system.

**Designing and Drawing of Images and Icons:** Since users who are low-literate are unable to read the text, you will be required to design some User-Interface elements like Icons and images for the users to understand the menus. You can look at the Karandaaz UI Toolkit, CGAPs Design Principles for Mobile Money for inspiration or from the article assigned in Homework 5. You can download and modify many of the assets and designs available online including, but not limited to, IconFinders, FlatIcons, Icon8. Searching for free online icons and vectors will give you a lot of options. Ensure that each design is culturally appropriate to the country of your choice (From previous course assignments).

**Implementation Details:** For the assignment, you will be required to create the User-Interface menus for the Mobile Money application. The menus that you have to build are similar to the ones discussed in class. Some examples are given 1 and 2. The items you have to implement include:

1. Create User -
2. Transfer Money - Figure 1 and 2
3. View Balance - should show balance
4. Past Transactions - should show past transactions (receiving and sending of money)


Joseph transfers 100.00 to Dennis (1/2)

UW-Pesa

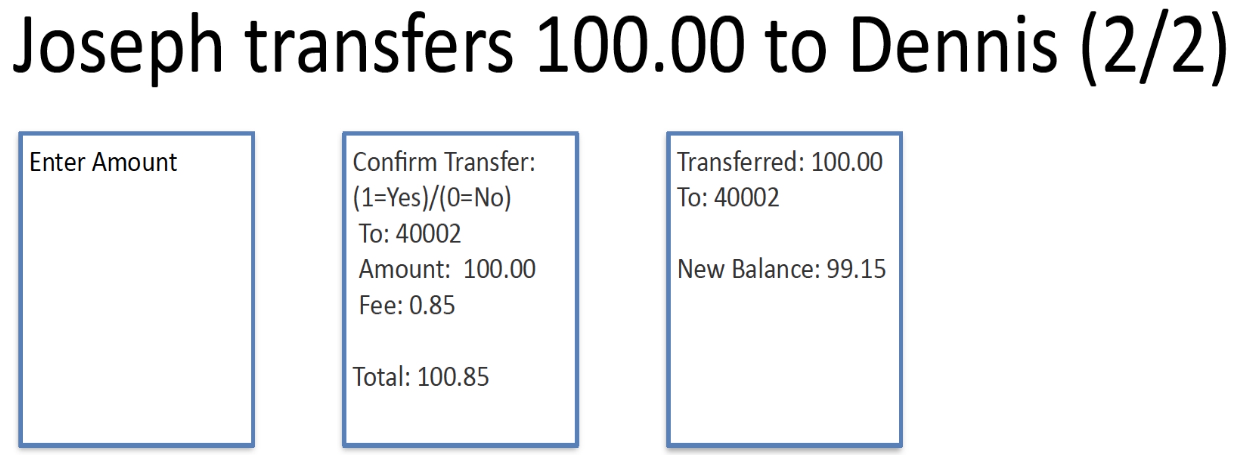
Enter PIN

Welcome To UW Pesa
1. Check Balance
2. Transfer Money
3. Past Transactions
4. Your Phone Number

Transfer To:
1. Registered User
2. Non-registered User

Transfer Money To (Account Number Or Phone Number):

Figure 1: Four screens showing the multiple steps included in the process to send money to another person (from Left to Right). Each box will be one screen in the application. (1 to 4 out of total 7 screens)


Joseph transfers 100.00 to Dennis (2/2)

Enter Amount

Confirm Transfer:
(1=Yes)/(0=No)
To: 40002
Amount: 100.00
Fee: 0.85

Total: 100.85

Transferred: 100.00
To: 40002

New Balance: 99.15

Figure 2: Three screens showing the multiple steps included in the process to send money to another person (from Left to Right). Each box will be one screen in the application. (5 to 7 out of total 7 screens).

**UW-Pesa Documentation:** For the implementation back end, you have to connect with the UW Pesa APIs.

- The UW-Pesa web portal can be accessed using its UW Pesa web portal link. You can also create users and transfer money from Agent to Users using the web portal.
- The UW-Pesa Documentation is available at at this link.
- For example, to implement Create User in the prototype application, you will have to use the Create User API of UW-Pesa. The front-end will contain the Icon and Images that you have created for Create User.

**Steps to Follow:**

1. Create a User using the UW-Pesa Sign-up Option on the main UW Pesa web portal link. Let us call this newly created user *TestUser*. Remember, you can also create a new user using the Create User API of UW-Pesa.

2. When *TestUser* is created, they have zero balance, just like you would have zero balance upon opening a bank account. You need to transfer money to a user, using an Agent login. For that follow the next step.

3. Use one of the given Agent Logins credentials to log-in to UW-Pesa as an Agent and transfer some money to the *TestUser*. This will give some balance to the *TestUser* account.

4. Once you have a user (either from the API or the web portal methods), you have to obtain an access token to use the other APIs.

5. Design the Elements and link them to the UW-Pesa APIs, and make sure all screens follow the same sequence and once a transaction is completed, the user is informed about the completion of transaction or errors, in case of errors. After that, the user should be allowed to return to the Main Menu.

6. For the purpose of the assignment, you can only implement transferring money to Registered User within the system. You do not have to implement it for the Non-Registered Users.

| **Agent ID** | **Agent Password** | **Agent ID** | **Agent Password** |
|---|---|---|---|
| 33330 | 0000 | 33331 | 1111 |
| 11115 | 5555 | 11114 | 4444 |
| 11113 | 3333 | 11112 | 2222 |
| 11111 | 1111 | 11110 | 0000 |

Table 1: Agent IDs and Agents passwords to be used on UW-Pesa Web portal to transfer (Cash-in) money to accounts for Step 3.

**Submission:** You should submit the code and the designed elements through Canvas in a .zip file (firstname_lastname.zip) by the due date. Make sure to add your name, student ID, and comments within the code to explain your implementation. If you include external APIs, header files or packages, make sure to briefly explain their need and use in the implementation.

We will inspect the code to ensure that it is original and reasonably matches up to the functionality of the system. The assignment will be graded based on the following criteria:

- Complete implementation of the Design elements

- Complete implementation of the listed features like transfer money, view balance etc.
- No serious complications or obviously missing components of the system.

**Reporting Problems:** UW-Pesa is not a production system and is in its beta version. So, while creating the applications, using its APIs or interacting with its web platform, if you face any difficulties or notice any bugs, please report to us by emailing to the TAs and instructor. Make sure your email subject mentions "UW-Pesa - Bug Reporting".

*Note: We have zero tolerance for plagiarism. Plagiarism will result in either a zero or disciplinary action. Ask the TAs or the instructor if you have any questions about collaboration. In general, asking other students how they went about the assignment is appropriate, but copying code or writing code for someone is not.*