

Simple Bank Application

Chuan-En Hou (1339293)

New York Institute of Technology - Vancouver

CSCI-503 Computer Programming II

Prof. Xiaojun Di

November 22, 2023

Table of Content

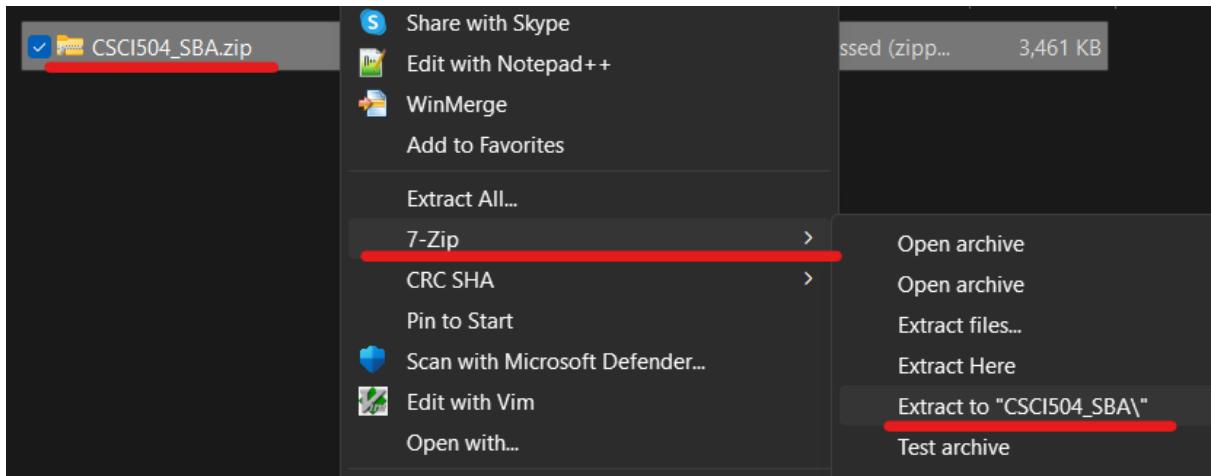
Manual.....	3
How to Run.....	3
How to Use.....	4
Class Diagram.....	8
Page Switch Mechanism.....	9
Description.....	9
User Information Storage.....	11
Description.....	11
Page Design.....	12
Sign-In Page.....	12
Description.....	12
Flowchart.....	13
Sign-Up Page.....	14
Description.....	14
Flowchart.....	15
Overview Page.....	16
Description.....	16
Flowchart.....	17
Transfer Page.....	18
Description.....	18
FlowChart.....	19
Loan Page.....	20
Description.....	20
Flowchart.....	21

Manual

This section briefly explains how to run and use the 'Simple Bank Application' (SBA).

How to Run

1. Unzip CSCI504_SBA.zip.



2. Open cmd window and navigate to CSCI504_SBA >> code.

```
(kali㉿kali)-[~] $ cd /home/kali/Desktop/CSCI504_SBA/code/
(kali㉿kali)-[~/Desktop/CSCI504_SBA/code]
```

3. Compile Entry.java.

```
(kali㉿kali)-[~/Desktop/CSCI504_SBA/code]
$ javac Entry.java
Picked up _JAVA_OPTIONS: -Dawt.useSystemAAFontSettings=on -Dswing.aatext=true
```

4. Run Entry.

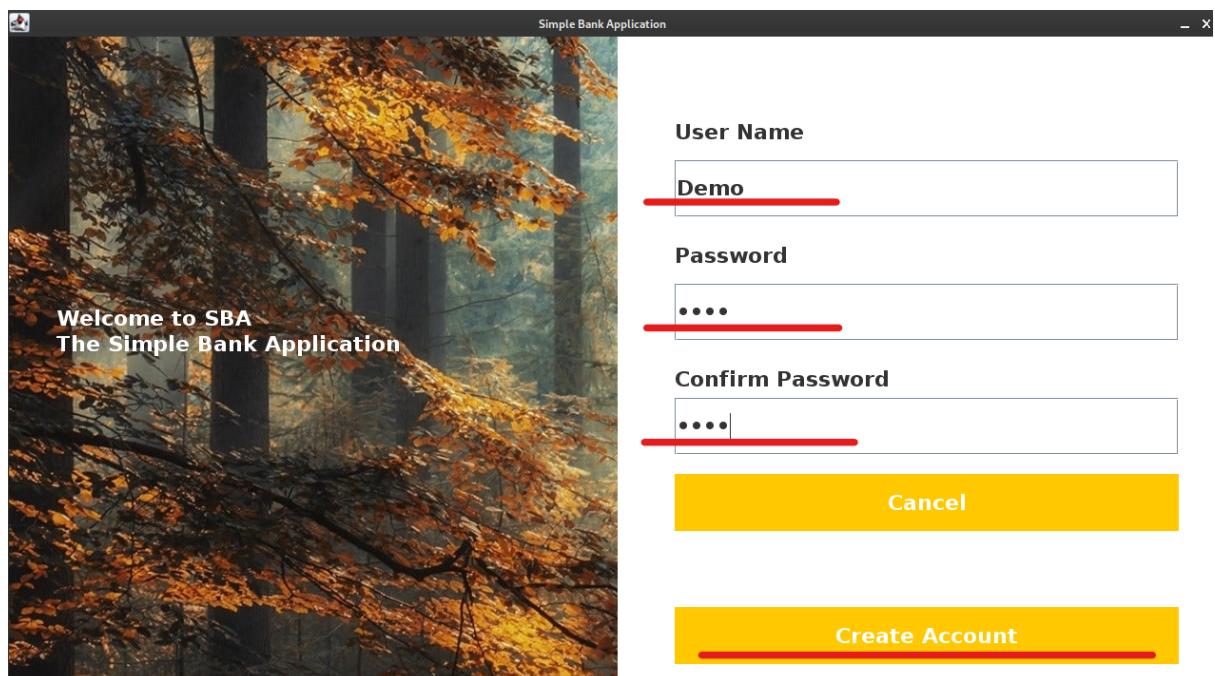
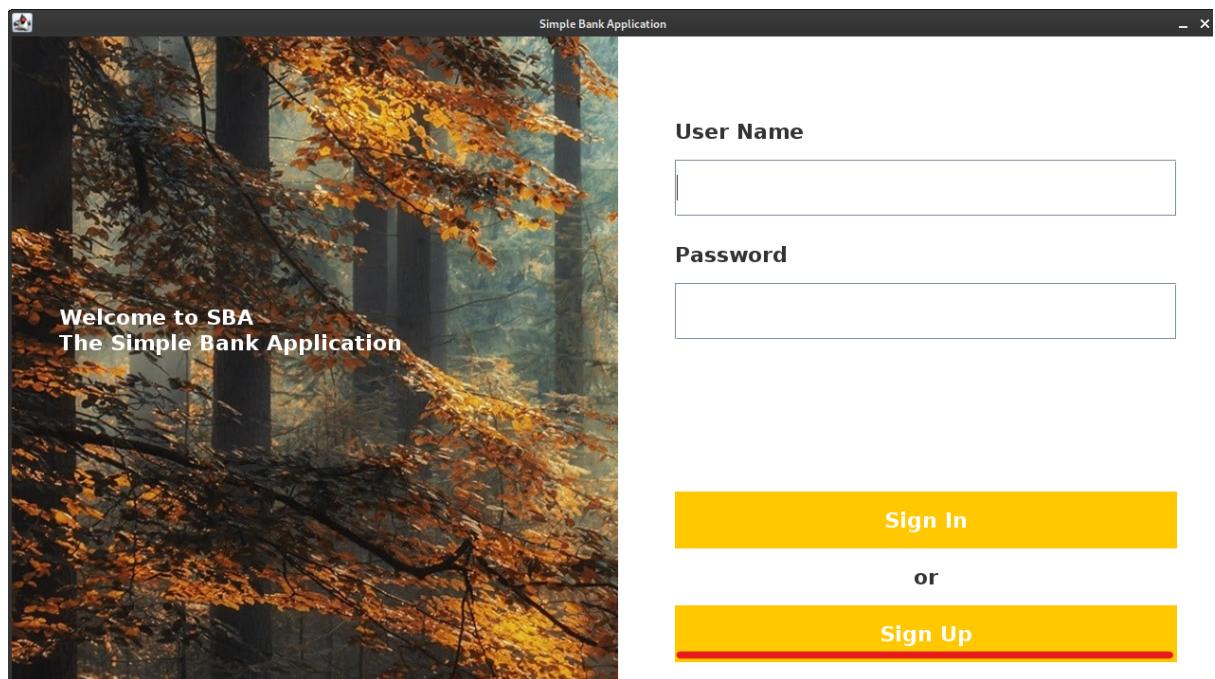
```
(kali㉿kali)-[~/Desktop/CSCI504_SBA/code]
$ java Entry
```

5. The SBA will start.

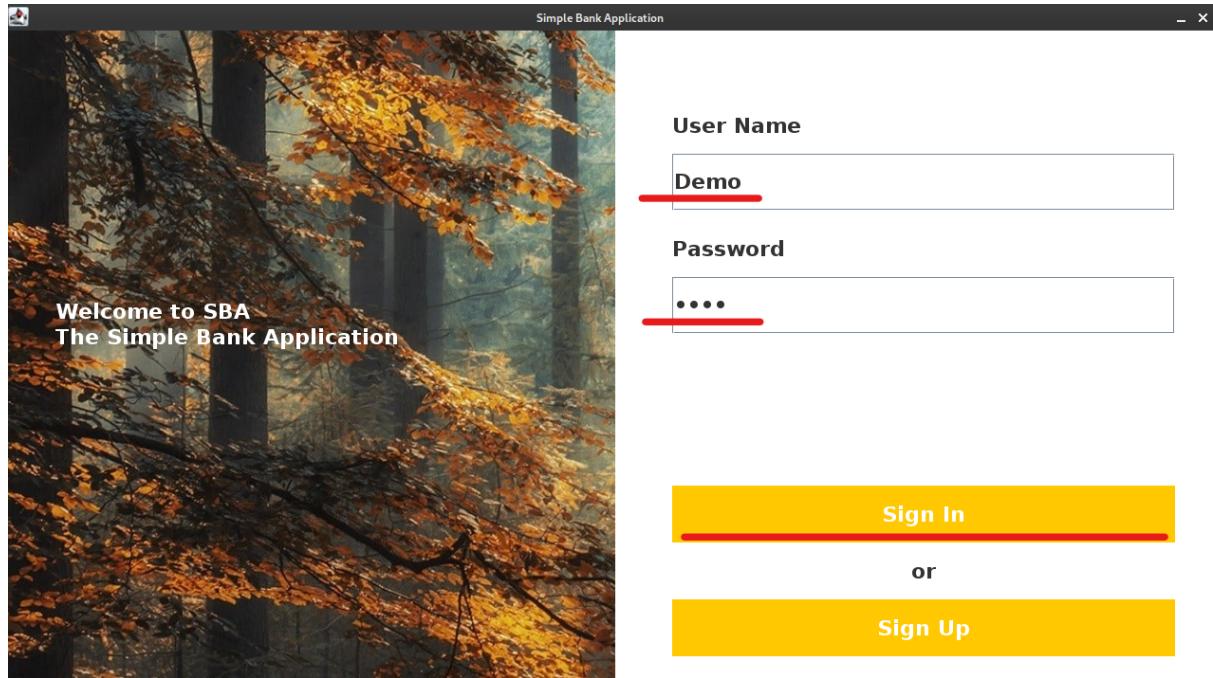
How to Use

1. Sign up for an account.

Due to an SBA banks' promotion, new users will be rewarded with a random amount of money.



- Sign in with the user's name and password.



- Users will be directed to the overview page, where various pieces of information will be displayed, including the date, current balance, current loan status, and banking information.

Click on the tabs to navigate to different pages, as indicated below:

SIMPLE BANK APPLICATION

Sign Out

Overview Transfers Loan 11 / 24 / 2023

Hello, Demo
Welcome to SBA

Account Balance	\$18041
Account Loan	- \$0

Bank Information

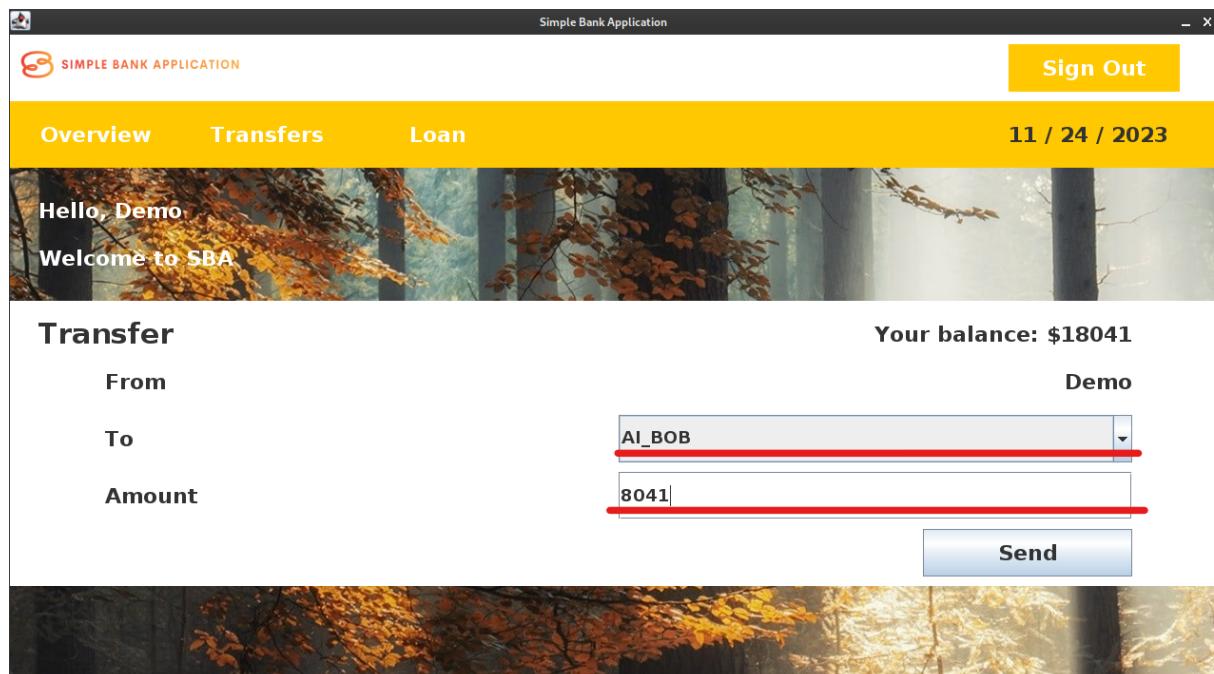
Due to recession, a 0.0% interest rate is being applied to all saving accounts.

4. Navigate to the Transfer Page, where users can transfer money to other users. By default, there are two AI users, AI_BOB and AI_JOY, to whom users can transfer money

In this case, user Demo wants to transfer \$8041 to AI_BOB. Here's how it's done.

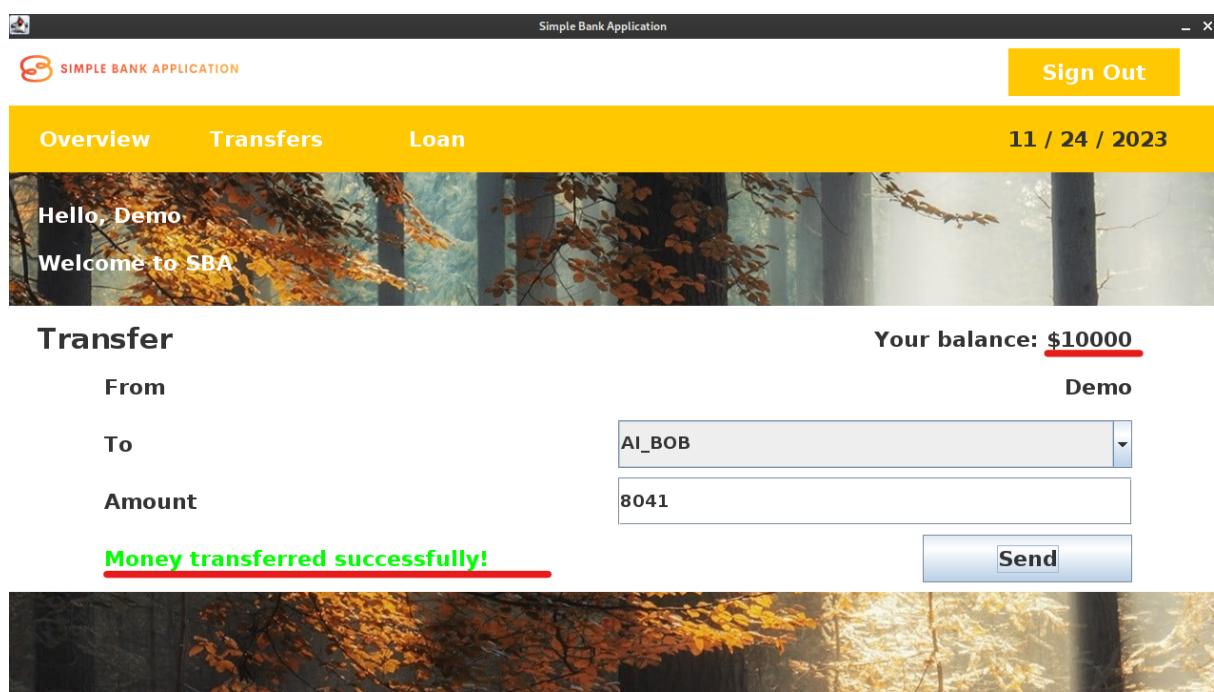
First, choose AI_BOB as the payee.

Second, enter the amount of money the user wants to transfer.



Lastly, press the 'Send' button to transfer the money.

If the transfer is successful, a confirmation message will be displayed, and the amount will be deducted from the user's account.

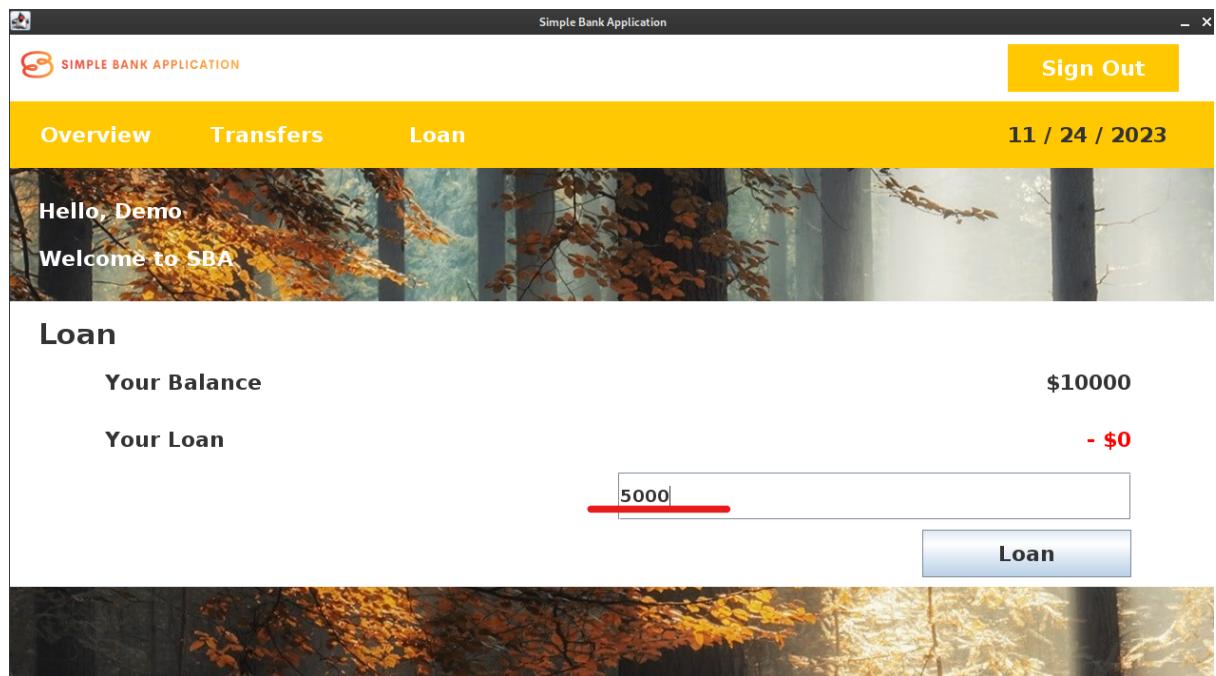


5. Navigate to the Loan Page.

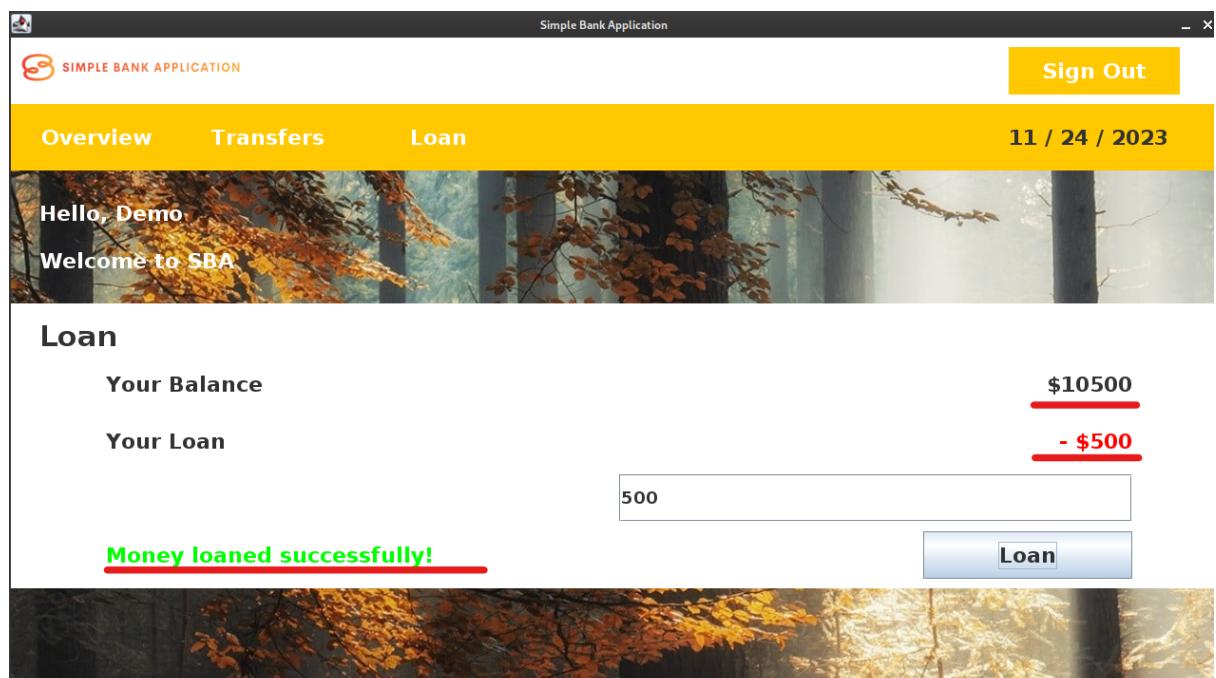
Here, the user can enter the loan amount.

Please be advised that the maximum loan limit is \$1,000.

In this example, the user Demo wants to borrow \$500 from SBA bank. Here's how it's done. First, enter the amount as \$500. Lastly, press the 'Loan' button

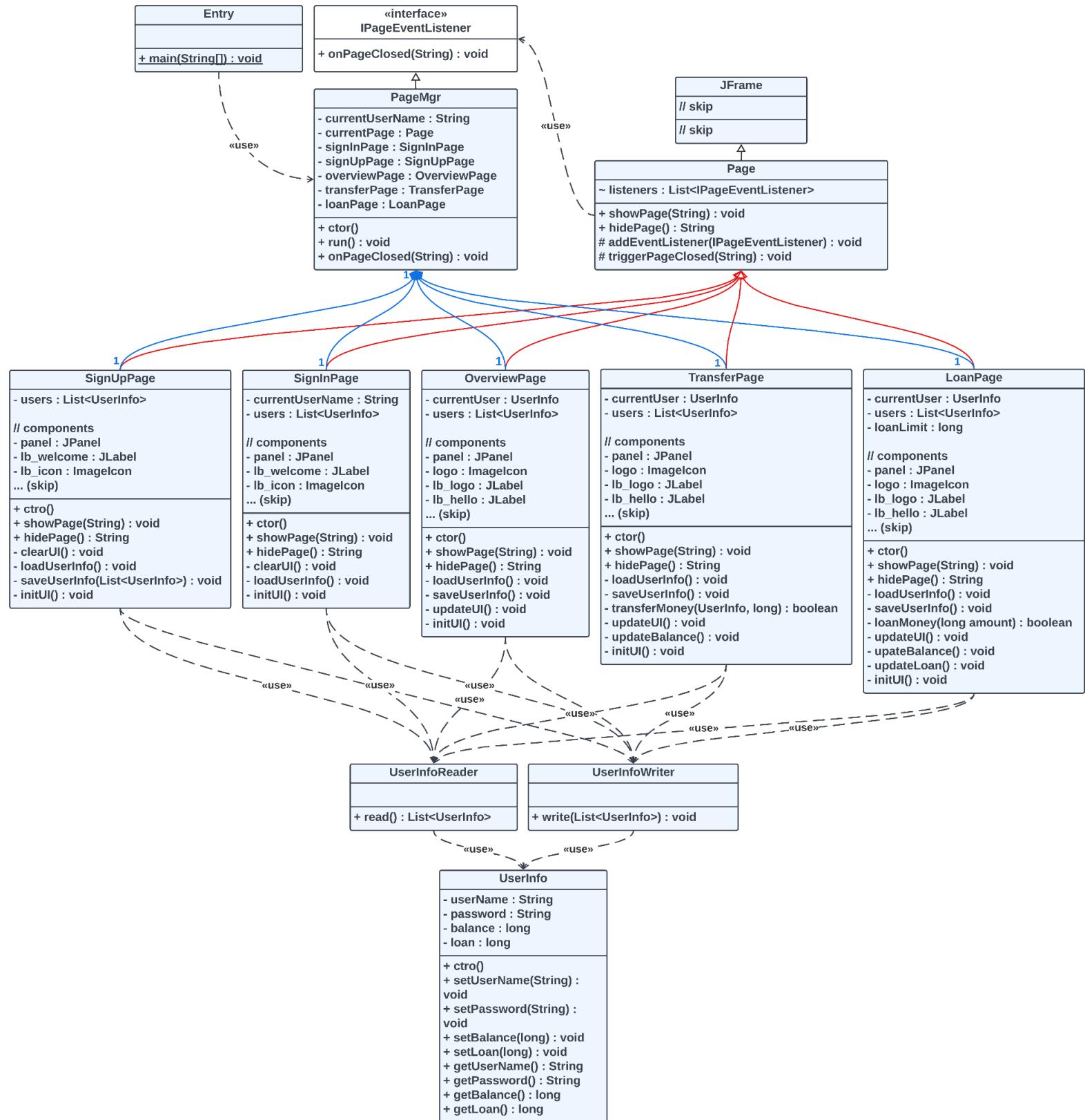


If the loan process is successful, a confirmation message will be displayed, and the borrowed amount will be added to your account along with the loan amount



Class Diagram

In this section, a class diagram will be displayed to illustrate the relationships between objects.



Page Switch Mechanism

In this section, this document will illustrate the mechanism behind page switching.

Description

This project uses PageMgr to manage the behavior of showing and hiding pages.

The rationale for utilizing PageMgr is to ensure reasonable relationships between objects.

For example, if page2 is supposed to be shown after page1 is closed, the improper way to do so is indicated in the following pseudo code:

```
class Page1 {  
    own a page2 object; // Improper: Page1 should not own a page2 object  
    if page1 is closed:  
        show page2;  
}
```

The reason this approach is improper is that page1 knows and owns page2, which is unnecessary. Page1 can function perfectly without needing to know about page2.

Another drawback is that if page1 owns the object of page2, page1 would be able to manipulate all the public fields provided by page2.

Therefore, to avoid improper relationships between objects, this project implements PageMgr, which serves as a page manager responsible for deciding when a page should be opened and closed.

Here is a pseudo code snippet demonstrating the basic concept of how it works

```
class PageMgr {  
    own a page1 object;
```

```
own a page2 object;

subscribe to page1 closed event;
subscribe to page2 closed event;

onPageClosed(nextPageName):
    close the active page;
    if nextPageName is page1:
        show page1;
    else if nextPageName is page2:
        show page2;
}

class Page1 {
    trigger onPageClosed event ( name of page2 );
}

class Page2 {
    trigger onPageClosed event ( name of page1 );
}
```

PageMgr owns page1 and page2, so page1 and page2 won't need to know each other, which makes more sense. PageMgr subscribes to the onPageClosed event of both pages to determine the timing of when to close the active page and what to show on the next page.

User Information Storage

In this section, this document will briefly explain how the information of all users are stored.

Description

There should be a place to store user information, such as name, password, balance, and loan. The choice of local storage is not considered, as closing the SBA application releases all local storage, including the user's information.

To maintain all user information after the SBA application is closed, this project has decided to store it in an .xml file. The decision to use the .xml format is based on two reasons:

1. Java has a generic XML file reader and writer, eliminating the need to devise a custom file format and logic.
2. XML files are much easier to maintain due to their well-structured tags

The structure of the XML file is as follows:

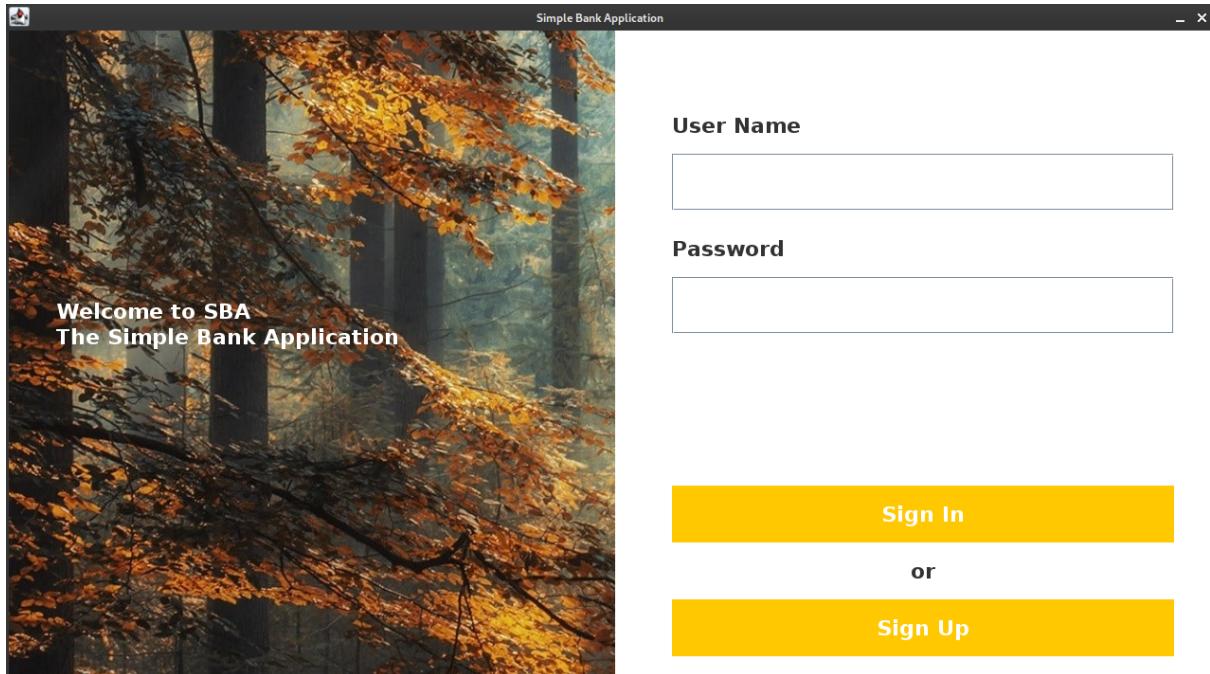
CSCI504_SBA\assets\user\userInfo.xml

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<data>
    <user>
        <name>AI_BOB</name>
        <password>bob</password>
        <balance>4000</balance>
        <loan>0</loan>
    </user>
    <!-- second user -->
    <!-- third user -->
</data>
```

Page Design

In this section, an image, along with a technical description and a flowchart, will be provided to illustrate the functionality of each of the following pages.

Sign-In Page



Description

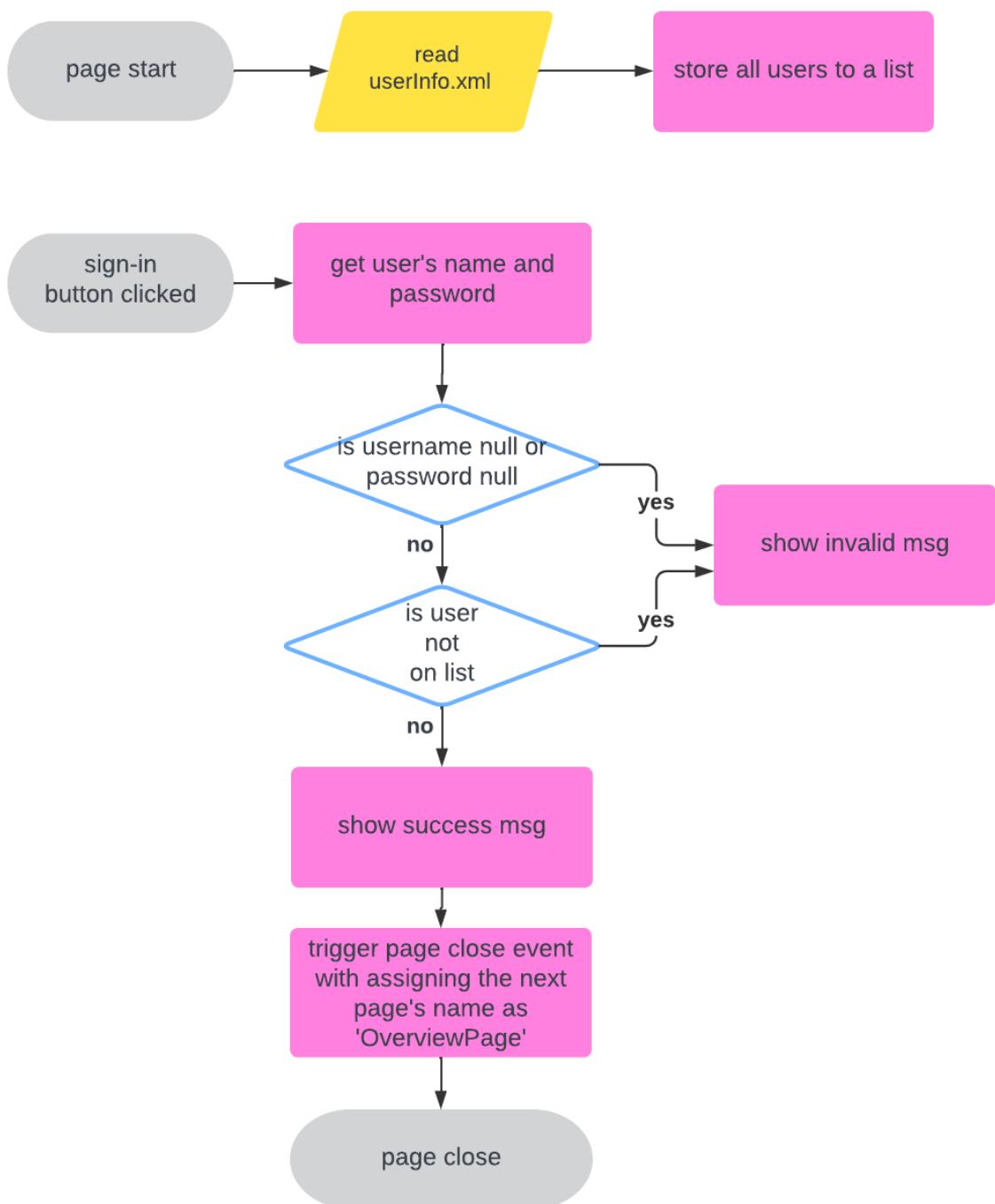
This is a simple sign-in page. The UI design is as follows: First, create a new object. Second, set the object's font size. Lastly, set the location and display area using the `setBounds` method, like the following image:

```
lb_password = new JLabel("Password");
lb_password.setFont(lb_password.getFont().deriveFont(Font.BOLD, 22));
lb_password.setBounds(700, 200, 530, 60);
```

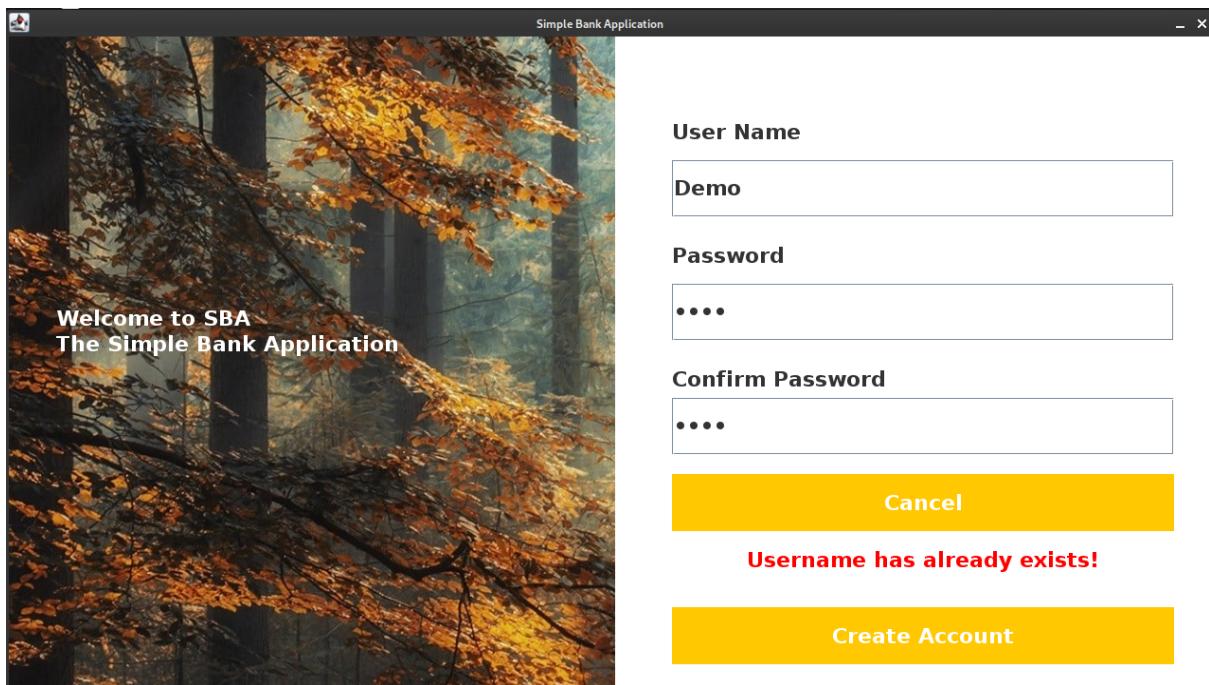
Notably, to protect the user's password and keep it hidden while typing, a `JPasswordField` object was specially used. Other than that, everything is straightforward.

Once the 'Sign In' button is clicked, the user's name and password will be retrieved and compared with a user list. If the entered user matches one of the users on the list, then sign-in will be granted.

Flowchart



Sign-Up Page



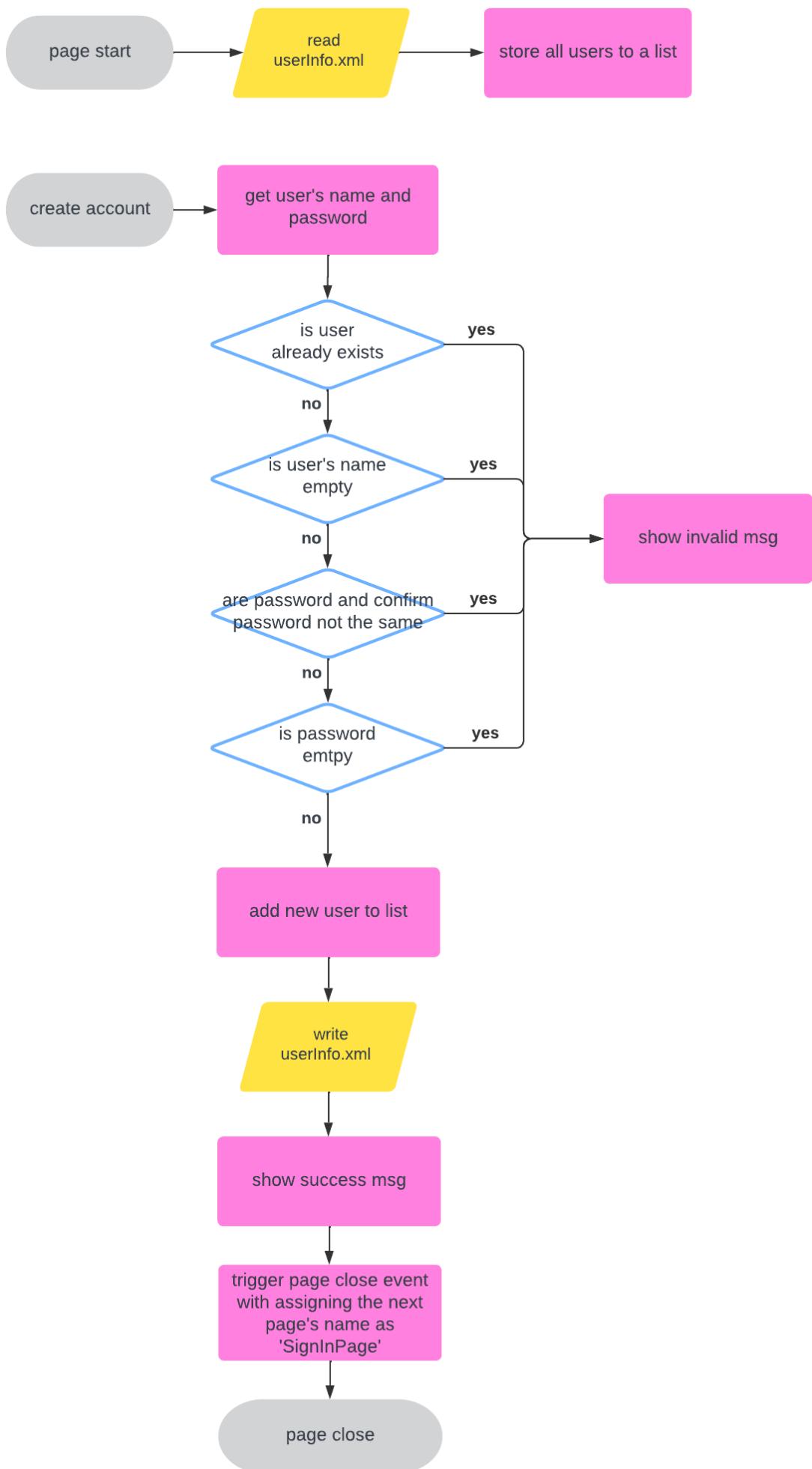
Description

The UI design is the same as the Sign-In Page, following the same principle: First, create the UI component. Second, set the font size using the `setFont` method. Lastly, set the location using the `setBounds` method.

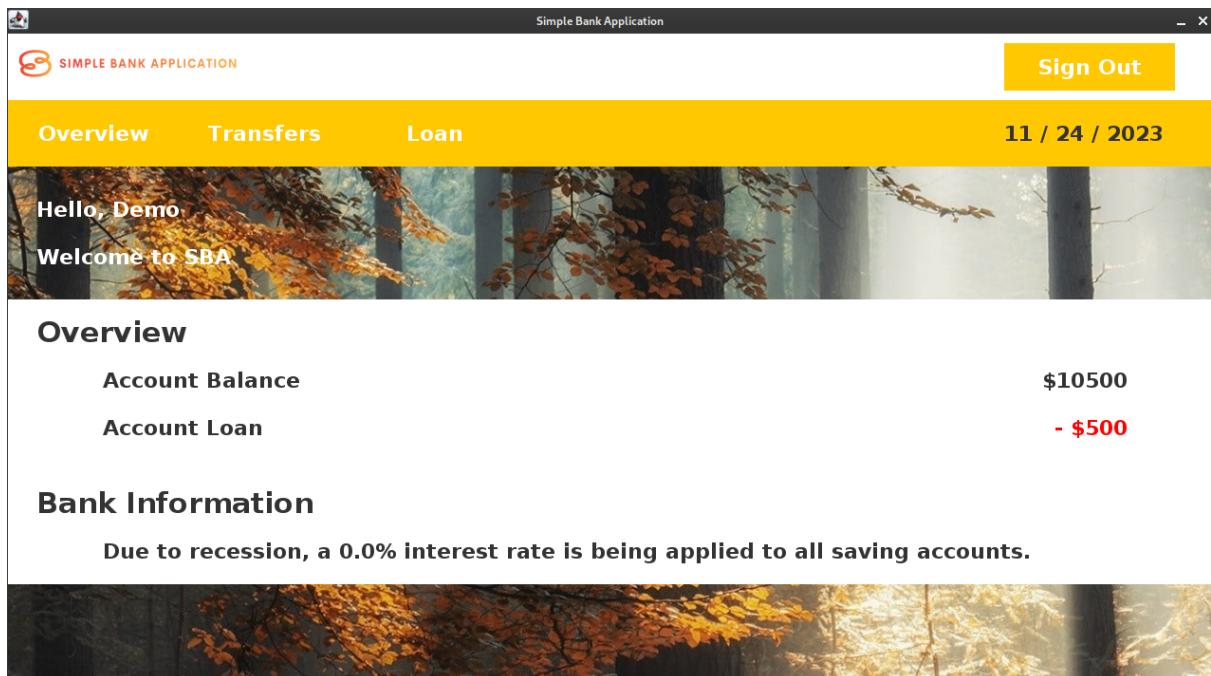
Apart from that is the process of creating an account. There are some rules for setting up an account. For example, the user to be created should not already exist, and the name of the new user should not be empty. More details will be illustrated in the flowchart.

If the cancel button is clicked, the page will switch back to the Sign-In Page without any modifications. It is a good way to cancel creating an account when users realize they don't want to create one.

Flowchart



Overview Page



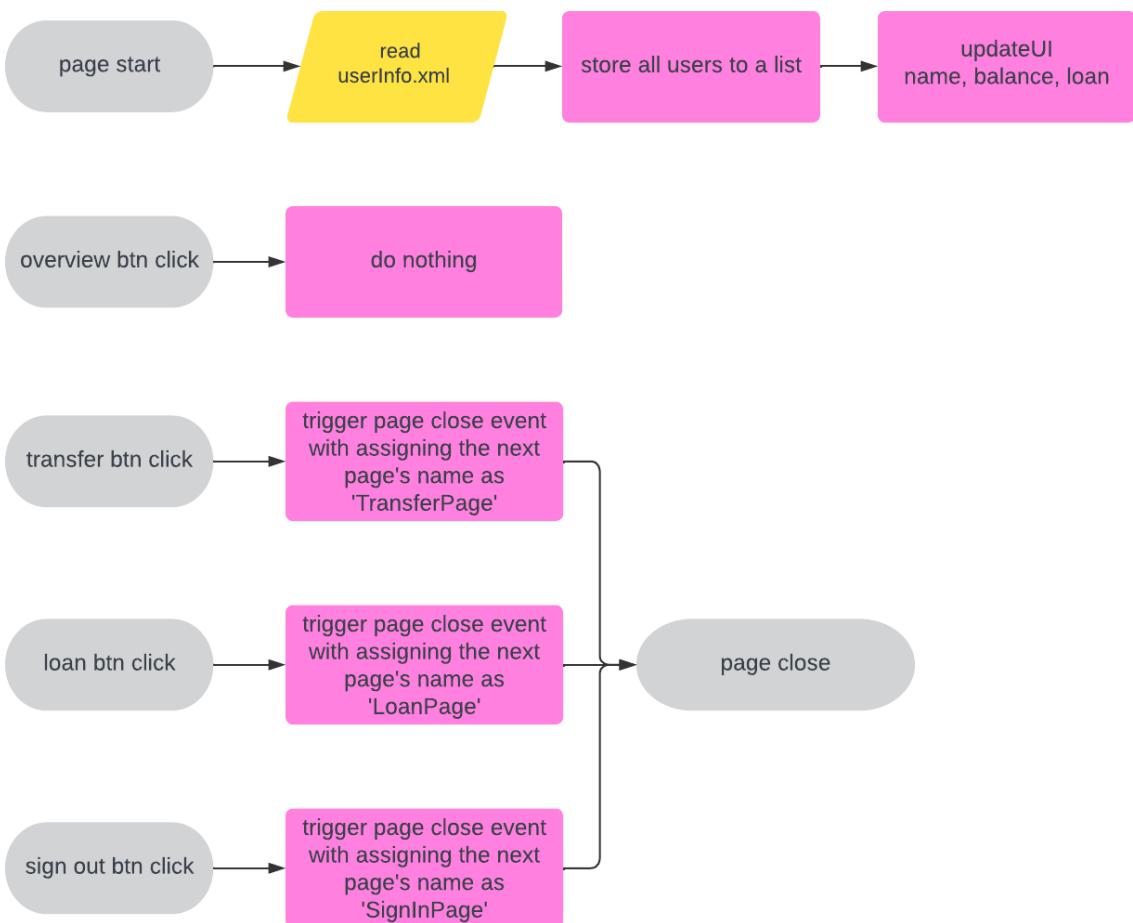
Description

Since this page is an overview page, it is mostly composed of JLabel components. What may be worth noticing is that the update time of some components may be different. When this page is instantiated, most of the labels and their associated words are defined. However, some labels, such as the user's name, balance, and loan, won't be obtained until the user logs in. Hence, the second update timing is implemented. When the user has signed in successfully and the page starts to show, the user's name, balance, and loan will be updated at that point.

In this page, there are four buttons: Sign Out, Overview, Transfers, and Loan buttons on the top of the page:

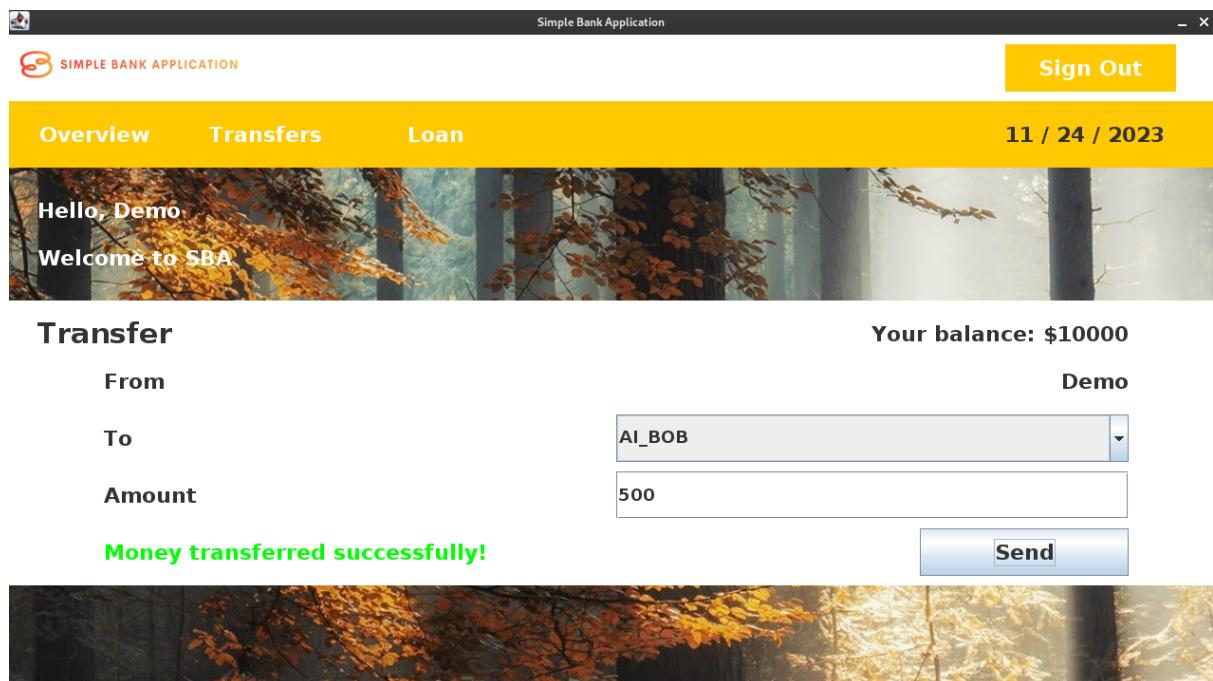
- Clicking the Sign Out button will switch the page back to the Sign-In Page.
- Clicking the Overview button will not change the page since the user is already on the overview page.
- Clicking the Transfers button will bring the user to the Transfer Page, as mentioned later in this document.
- Last but not least, clicking the Loan button will bring the user to the Loan Page, where the user will be able to loan some money.

Flowchart



Please note that overview, transfer, loan, and sign-out buttons have appeared on many pages. In order to avoid duplication in the flowchart, this document will only present their flowchart here. Their functionality is simple: whichever page's button is clicked, the user will be redirected to that specific page.

Transfer Page

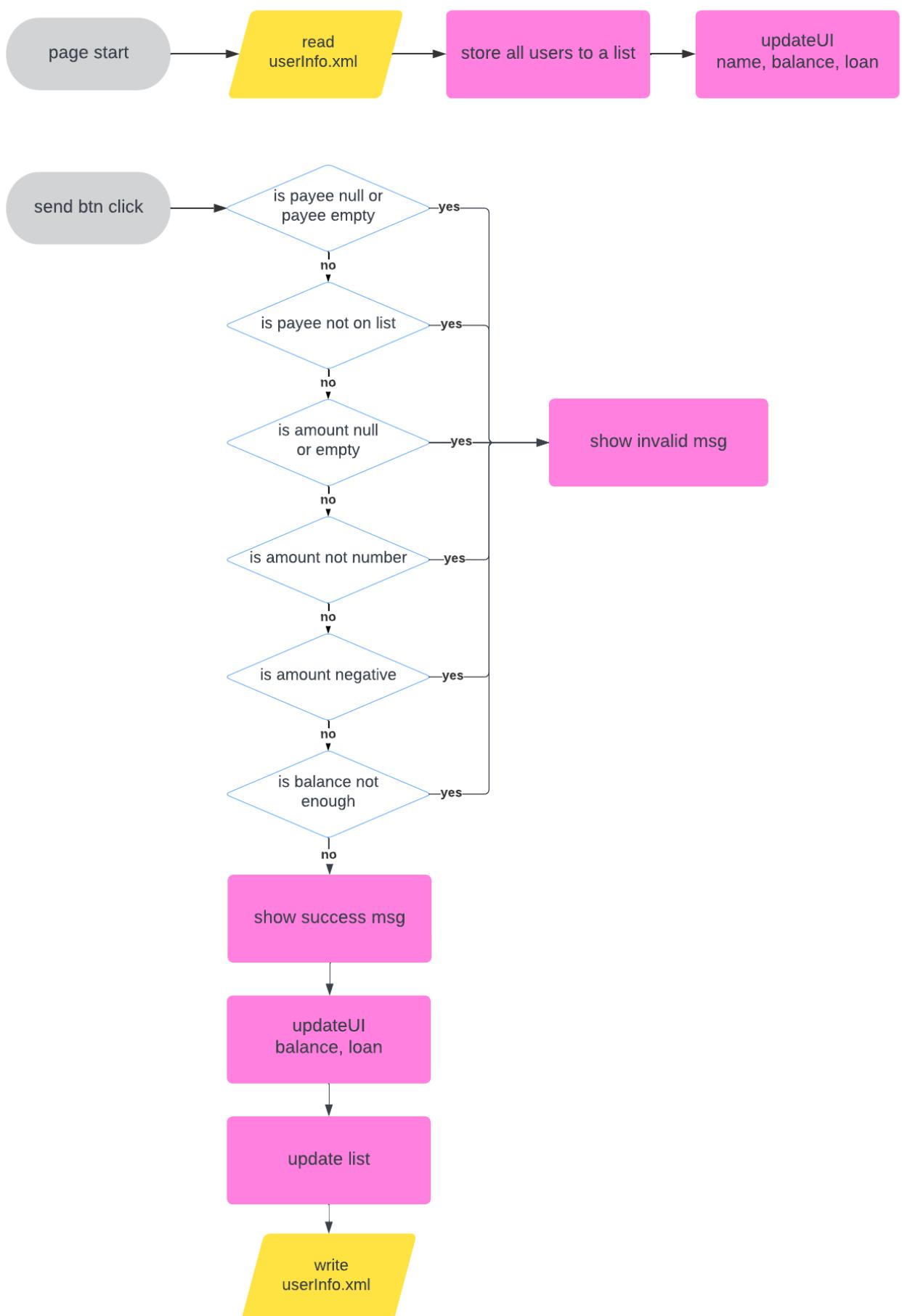


Description

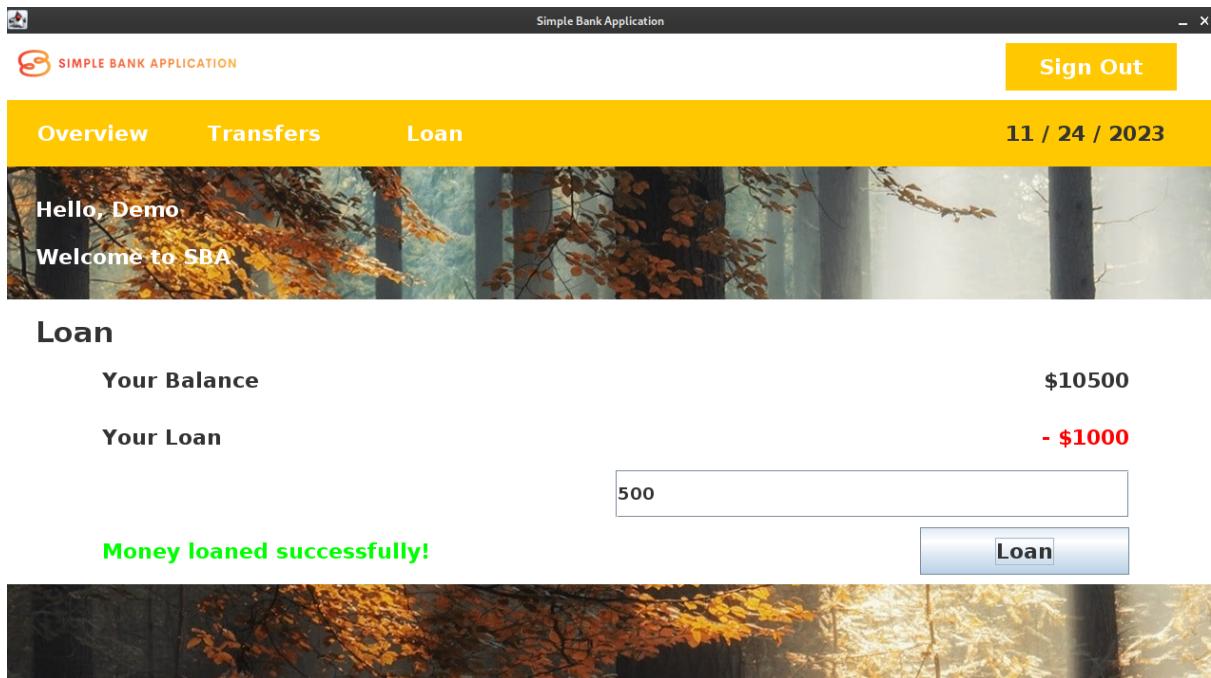
Users can perform money transfers on this page.

Here, the user can see some information about the transaction. First, the name of the payer, which is the current user, will be displayed in the first row. Second, on the second row, there is a JComboBox component used to allow the user to select one of their payees, who will be other SBA users. By default, there will be two AI users. Lastly, there is a JTextField in which the user can enter the amount of money he or she wants to transfer. Please note that the user is only allowed to transfer an amount of money that they have in their account. Once the SBA system detects that the amount is not sufficient, the transaction will be denied. Please refer to the following flowchart for detailed regulations.

FlowChart



Loan Page



Description

On this page, the user is allowed to borrow money up to a certain amount if they feel the need to do so. Please note that the maximum amount of money allowed for a loan is \$1,000. Any request amount that exceeds this limit will be denied. Here are two examples of how the limitation works:

Scenario 1: If the user has already loaned \$1,000, they won't be able to loan any more money afterward.

Scenario 2: If the user has already loaned \$500, they won't be able to loan more than \$500 next time, even if they haven't reached the limit yet.

For more details on how the loan process works, please refer to the flowchart in the following section.

Flowchart

