



The University of the West Indies, St. Augustine
COMP 2603 Object Oriented Programming 1
Assignment 2
2023/2024 Semester 2

Due Date: March 29, 2024 at 10:00 p.m.
Lockout Date: March 31, 2024 at 10:00pm

Overview: This assignment requires you to write event handling code and some supporting classes for a simple system that simulates a display of bills.

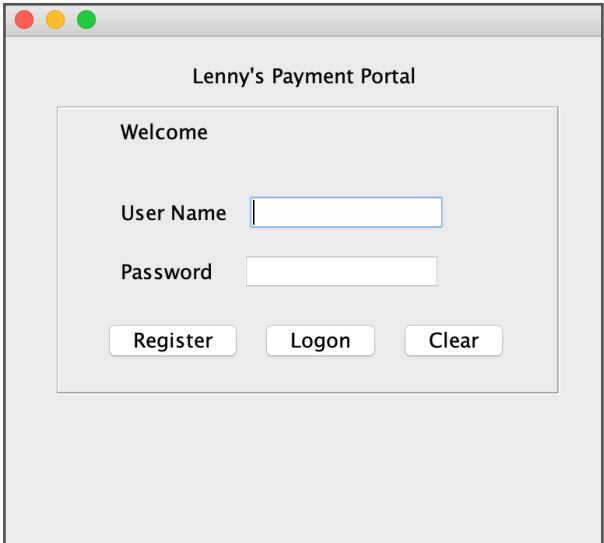


Fig 1. Logon Panel

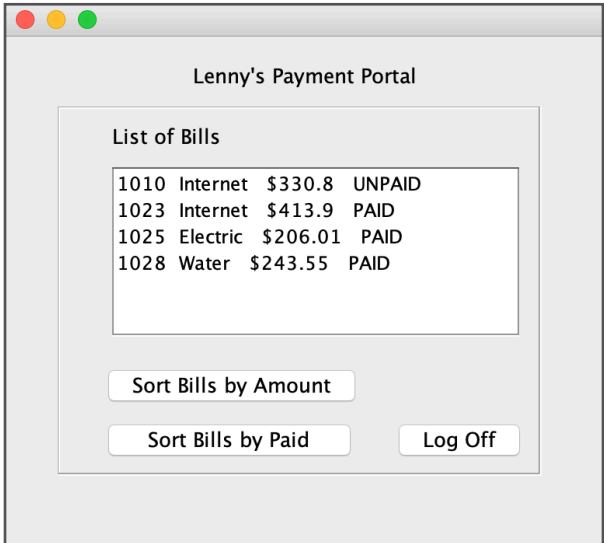


Fig 2. Bill Panel

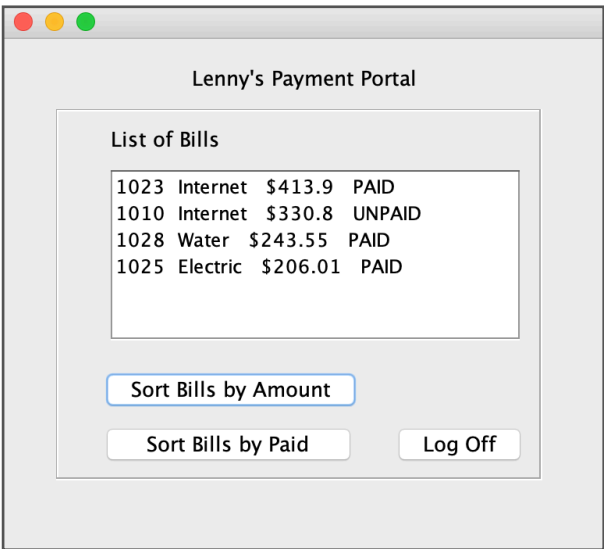


Fig 3. Bills sorted by Amount

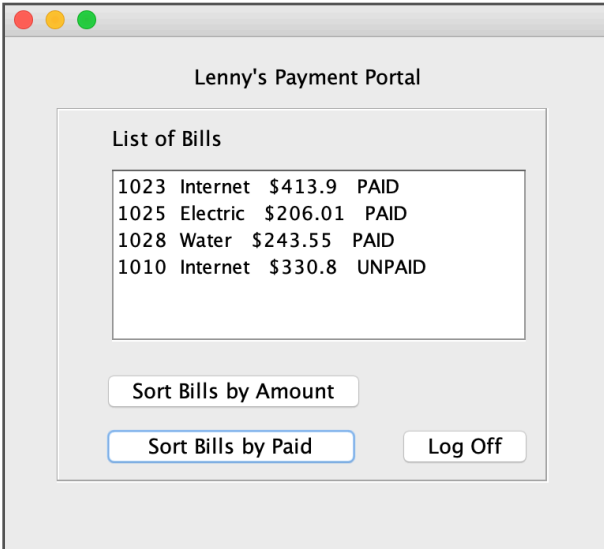


Fig 4. Bills sorted by Paid/Unpaid

The application has two main views shown in Figures 1 and 2. When a user supplies a valid username/ password combination and clicks on the Logon button, the bill panel in Figure 2 is shown.

When the user clicks on Log Off, the system reverts to the logon panel in Figure 1. The clear button empties the username and password fields of data and resets the label to Welcome as in Figure 1.

Bills are meant to be generated by the system with random data for amounts, bill type and whether paid or not. When run for the first time, the system should generate an unassigned bill dataset and then when a renter registers, a random portion of the bill dataset should be selected, removed and allocated to that renter's list of bills. This means that no two renters will share the same bill. Figure 2 shows the renter's bill list sorted initially by bill id. Clicking on the Sort Bills by Amount button in the bill panel displays the bills sorted by amount as in Figure 3. Clicking on the Sort Bills by Paid button in the bill panel displays the bills sorted by paid or unpaid as in Figure 4.

Refer to the Demo Video for illustration with more than more renter: <https://youtu.be/smsvAmgqwWs>

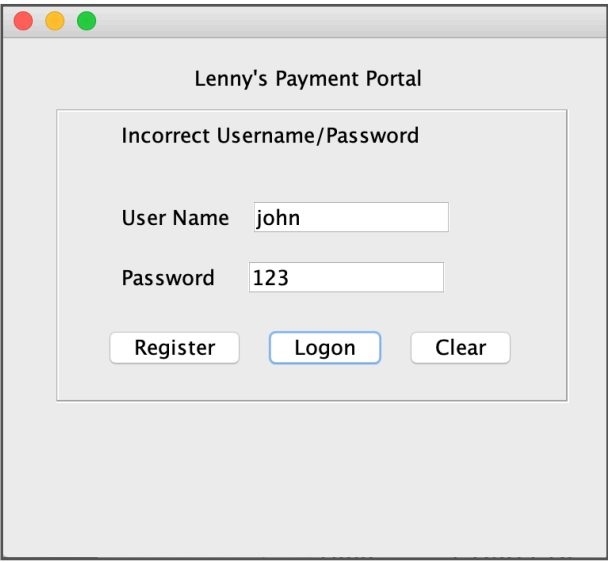


Fig 5. Invalid Login Credentials

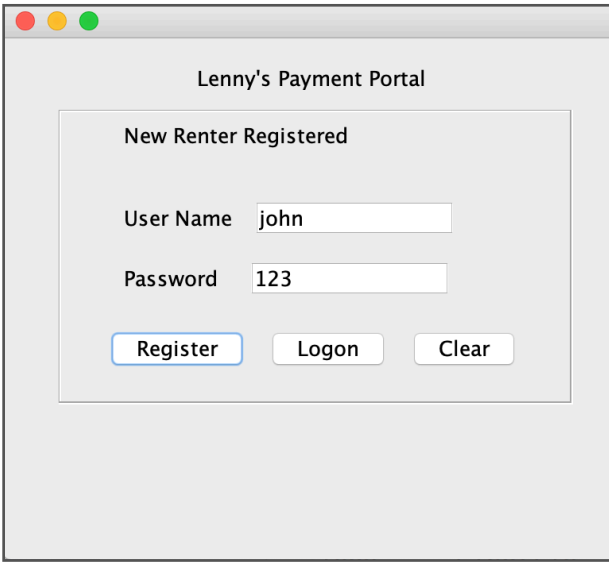


Fig 6. Successful Registration

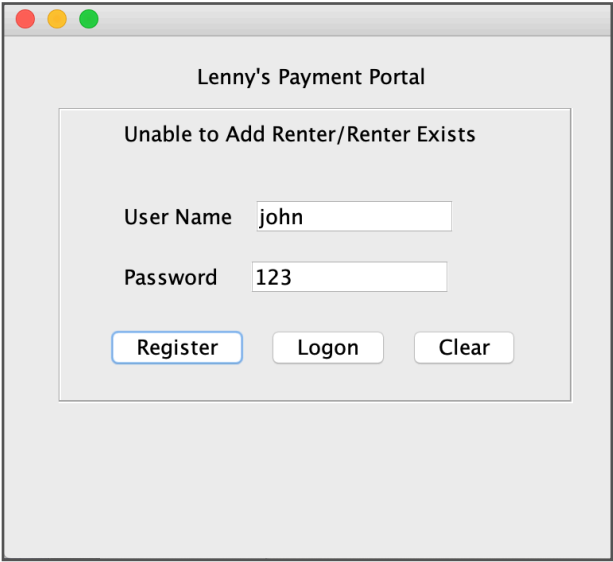


Fig 7. Unsuccessful Registration

Figure 5 shows invalid username/password results. This means that the username/ password combination was not found in the system.

Figure 6 shows the results of successfully adding a new renter to the system. Success means that the username and password combination is unique and was added to the system.

Figure 7 shows unsuccessful registration results. This means that the username/ password combination exists already in the system. The system allows blank data to be entered. The combinations matter however.

In all three cases, note the label change just under the “Lenny’s Payment Portal” title.

UML Diagram of Domain Classes

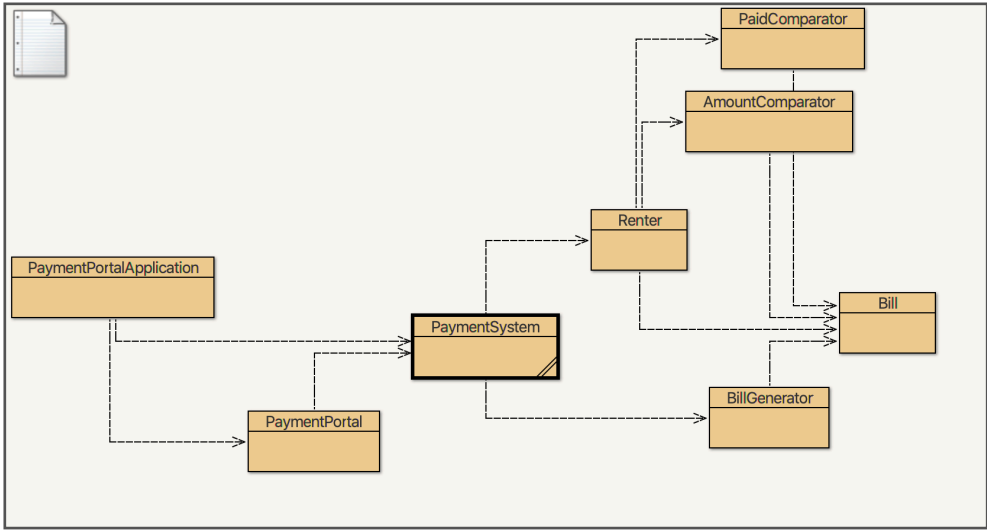


Figure 8: Interaction between classes in the system

The PaymentPortalApplication is the main class that creates the view layer (PaymentPortal) and associates a controller (PaymentSystem) with the view. All interaction with the underlying model classes (everything else) passes through the controller. You are required to write code for these classes by implementing appropriate interfaces, choosing suitable collections to manage the functionality.

Class Descriptions

The **Bill** class should have the following attributes: billNumber (String), amount (double), type (String), paid (boolean). Bill number are unique, start at 1000 and increment by 1. The constructor should accept the type, amount and paid values. All accessors are required.

The **Renter** class should have a collection of bills along with attributes for storing the user and password. The **PaidComparator** and **AmountComparator** classes implement the Comparator interface and should be used by the Renter class to perform sorting.

The **PaymentSystem** class should have a collection of renter objects and this class should generate and allocate bills to new renters. All interaction with the view layer should be limited to methods 2 to 6 in the following table. Method 1 should be private and used internally only.

The **PaymentPortal** class should have a reference to the **PaymentSystem** on creation.

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
	Renter	getRenter (Renter r)
java.lang.String[]		getRenterBills (java.lang.String userName, java.lang.String password)
java.lang.String[]		getSortedRenterBillsByAmount (java.lang.String userName, java.lang.String password)
java.lang.String[]		getSortedRenterBillsByPaid (java.lang.String userName, java.lang.String password)
boolean		registerRenter (java.lang.String userName, java.lang.String password)
boolean		validateRenter (java.lang.String userName, java.lang.String password)

All classes must adhere to the Java APIs supplied in the docs folder within the BlueJ zip file that is supplied on myeLearning.

Tips and Suggestions

- Get the simple functionality working (clear button). Panels showing or being hidden can come after.
- Start with the class with the least dependencies (no arrows leading from it).
- Set up the Model-View-Controller flow and then add on functionality in pieces.

Bill Generation Code

The following code can be used and modified to generate random bill datasets. Note: This code relies on a working Bill class existing first.

```
private ArrayList<Bill> billrepo = new ArrayList<Bill>();
private Random rand = new Random();

public void generateBills(){
    String[] types = {"Electric", "Internet", "Water"};
    int typeIndex = rand.nextInt(3);
    double amount = Math.floor((rand.nextDouble()*1000) * 100) / 100;
    int paidIndex = rand.nextInt(2);
    boolean paid = false;
    for(int i = 0; i<40; i++){
        billrepo.add(new Bill(types[typeIndex], amount, paid));
        paidIndex = rand.nextInt(2);
        if(paidIndex == 0) paid = false;
        else paid = true;
        typeIndex =rand.nextInt(3);
        amount = Math.floor((rand.nextDouble()*1000) * 100) / 100;
    }
    java.util.Iterator<Bill> iter = billrepo.iterator();
    while(iter.hasNext()){
        System.out.println(iter.next());
    }
}
```

Refer to myeLearning for the source files that you need to use for this assignment. Only two classes are provided: PaymentPortalApplication and PaymentPortal.

Submission Instructions

- Refactor the existing Java classes and add the required new classes (see Fig 8) to the application using BlueJ.
- Document your student ID at the top of each file within a comment block.
- Upload a ZIP file of your compiled project source and class files to the myElearning course page by the deadline. Submissions that do not compile will receive 0 marks. Empty submission files or faulty zip files will receive 0 marks. Check your submitted files for completeness and correctness.
- Name your ZIP file as follows: **FirstName_LastName_ID_A2.zip**. Marks will be deducted for submissions that do not conform to this naming convention. Ensure that the unzipped folder has the same name as the zipped file.
- Sign and submit the University Plagiarism declaration confirming that you are submitting your own original work and that you have not copied or collaborated with other students.
- Early submission points can be earned if you submit before the due date. No penalties will be applied but no early points will be earned if you submit by the lockout date.

Important Information Regarding Academic Conduct

- This is an individual assignment. You must attempt this assignment by yourself without any help from others, aside from the course lecturer, tutor or marker.
- You may use legitimate resources on the Internet, in books, or from the course notes to assist (unless prohibited by a question). Copying or modifying such content does not make it yours. Indicate on your submission any or all sources used in your answers within comments at the end of your files. Identify code snippets that you did not write yourself but reused from some source.
- You are not allowed to communicate, share or disclose any aspect of your solutions/work in any form to anyone except for the course lecturer, tutors, markers, and examiners.
- You are not allowed to assist others with this assignment.
- University plagiarism and academic misconduct policies apply fully.
- No part of your submission should be made publicly available even after the due date without permission from the course lecturer.
- The use of Generative AI Tools, Chegg.com, CourseHero.com or any tutoring or homework assistance services, online or not, are prohibited.
- If you are faced with extenuating circumstances, contact your lecturer right away. Concessions may be arranged on a case-by-case basis. Be honest with your claims and provide evidence where possible.