

1. **Demo video:**

<https://umssystem.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=29cc4250-1c6f-475f-bbbf-afea0037eed2>

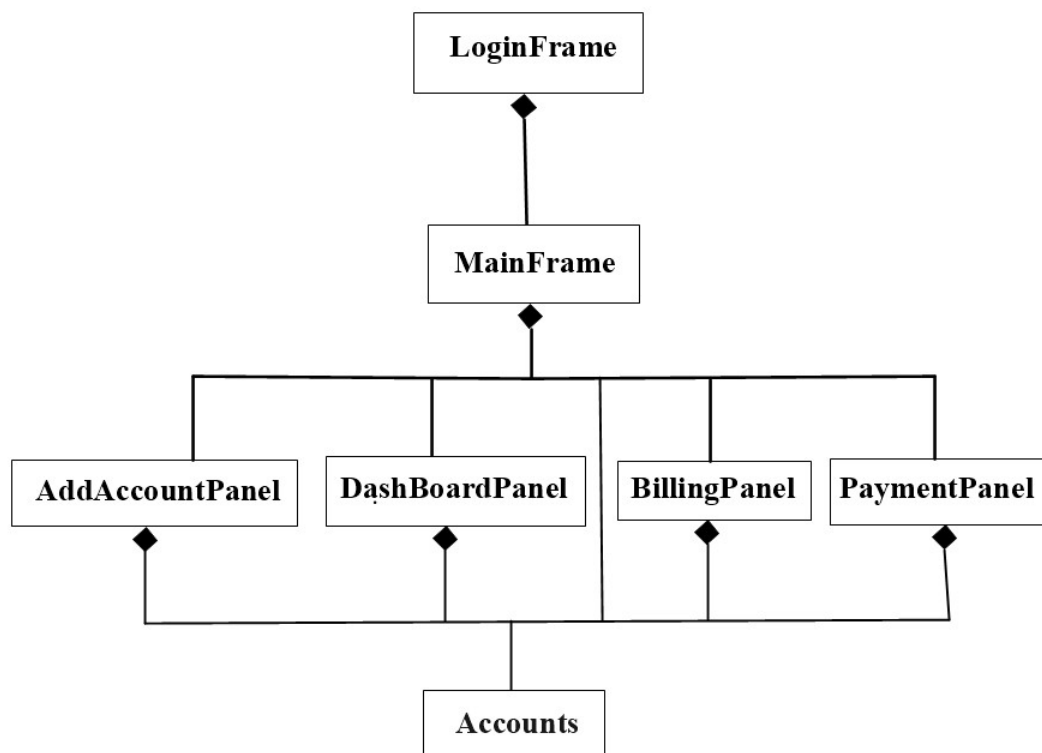
I have made some changes, since this record video, mostly for the visual of the application. The code has been changed quite bit, but overall, the main structure still maintained.

2. **Login credential:**

Username: (your MySQL database user's username. Ex: root)

Password: (your MySQL database user's password)

3. **Main classes diagram:**



Some main classes' UML details:

LoginFrame
- loginLabel : JLabel
- userNameField : JTextField
- passwordField : JPasswordField
- loginPanel : JPanel
- loginButton : JButton
- ButtonListener : class
+ LoginFrame()

MainFrame
<ul style="list-style-type: none"> - northPanel : JPanel - centerPanel : JPanel - westPanel : JPanel - southPanel : JPanel - accountLabel : JLabel - addAccountLabel : JLabel - modifyAccountLabel : JLabel - dashBoardLabel : JLabel - invoiceLabel : JLabel - paymentLabel : JLabel - addAccountPanel : AddAccountPanel - dashBoardPanel : DashBoardPanel - billingPanel : BillingPanel - paymentPanel : PaymentPanel - getField : JTextField - getAccountButton : JButton - searchAccountButton : JButton - resetButton : JButton - deleteAccountButton : JButton - helpButton : JButton - connection : Connection - statement : Statement - account : Accounts - helpMode : boolean = false - resetLabelsColor() : void - paintClickedLabel(label : JLabel) : void - showPanel(panel : JPanel) : void - ExplainTextEnable(enable : boolean) : void - ActionListener : class - mouseListener : class
<ul style="list-style-type: none"> + MainFrame() + round(num : double) : double + setExplainText(cp : Jcomponent, text : String, width : int) : void

AddAccountPanel
<ul style="list-style-type: none"> - questions : JTextArea - field1 : JTextField - field2 : JTextField - field2_1 : JTextField - field2_2 : JTextField - field2_3 : JTextField - field3 : JTextField - field4 : JTextField - field5 : JTextField - field6 : JTextField - centerPanel : Jpanel - eastPanel : JPanel - southPanel : Jpanel - westPanel : Jpanel - addAccountButton : JButton - modifyAccountButton : JButton - applyChangesButton : JButton - clearButton : JButton - plansComboBox : JComboBox<String> - account : Accounts - statement : Statement - resultSet : ResultSet - messagePanel : Jpanel - actionListener : class
<ul style="list-style-type: none"> + AddAccountPanel(statement : Statement, messagePanel : JPanel) + showAccountInfo(acc : Accounts) : void + reset() : void

DashBoardPanel
<ul style="list-style-type: none"> - northPanel: Jpanel - centerPanel : Jpanel - innerCenterPanel : Jpanel - eastPanel : Jpanel - northText : JTextArea - eastText: JTextArea - currentPeriodLabel : JLabel - lastPaymentLabel : JLabel - remainingDueLabel : JLabel - account : Accounts - price1 : double - price2 : double - price3 : double - updateInnerCenterPanel() : void - updateEastTextArea() : void
<ul style="list-style-type: none"> + DashBoardPanel() + showAccountInfo(account : Accounts) : void

BillingPanel
<ul style="list-style-type: none"> - getInvoiceButton : JButton - invoiceLabel : JLabel - account : Accounts - currentBill : double - totalDue : double - price1 : double - price2 : double - price3 : double - statement : Statement - messagePanel : JPanel - getkwh(type : String, totalUsage : double) : double - buildPDF(cost1 : double, cost2 : double, cost3 : double) : void - generateImageFromPDF() : void - ActionListener : class
<ul style="list-style-type: none"> + BillingPanel(stm : Statement, messagePanel : JPanel) + setAccount(acc : Accounts) : void + reset() : void

PaymentPanel
<ul style="list-style-type: none"> - remainingDueField : JTextField - newPayAmountField : JTextField - centerPanel : JPanel - northPanel : JPanel - southPanel : JPanel - accountInfoLabel : JLabel - lastPaymentLabel : JLabel - remainingDueLabel : JLabel - newPayLabel : JLabel - newPayDateLabel : JLabel - confirmLabel : JLabel - processButton : JButton - statement : Statement - account : Accounts - dateChooser : JDateChooser - ActionListener : class
<ul style="list-style-type: none"> + PaymentPanel(stm : Statement) + showAccountInfo(acc : Accounts)

Accounts

```
- accountNumber : int
- name : String
- address : String
- phoneNumber : String
- plan : String
- lastPaymentDate : String
- remainingDue : double
- lastPayment : double
- currentUsage : double
- lastUsage : double

+ Accounts(accountNumber : int, name : String, address : String,
phoneNumber : String,
plan : String)
+ Accounts(accountNumber : int, name : String,
address : String, phoneNumber : String,
plan : String, remainingDue : double,
lastPayment : double, lastPaymentDate : String,
currentUsage : double, lastUsage : double)
+ getAccountNumber() : int
+ setAccountNumber(accountNumber int) : void
+ getName() : String
+ setName(name : String) : void
+ getAddress() : String
+ setAddress(address String) : void
+ getPhoneNumber() : String
+ setPhoneNumber(phoneNumber : String) : void
+ getPlan() : String
+ setPlan(plan : String) : void
+ getLastPaymentDate() : String
+ setLastPaymentDate(lastPaymentDate : String) : void
+ getRemainingDue() : double
+ setRemainingDue(remainingDue : double) : void
+ getLastPayment() : double
+ setLastPayment(lastPayment : double) : void
+ getCurrentUsage() : double
+ setCurrentUsage(currentUsage : double) : void
+ getLastUsage() : double
+ setLastUsage(lastUsage : double) : void
```

4. Features:

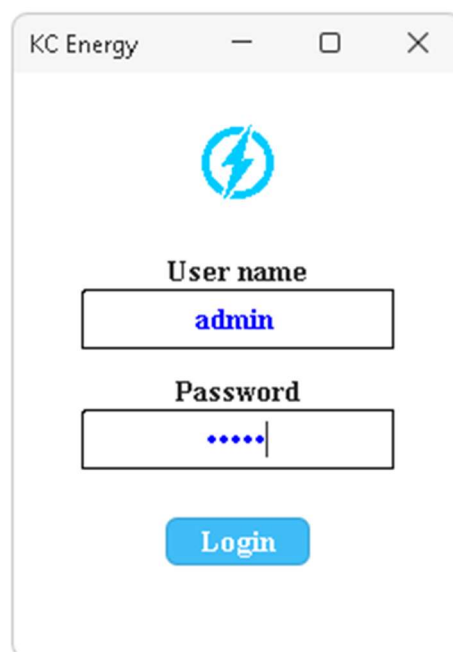
Feature	Implemented	Comments
Add new account	full	Could have unhandled errors
View/Modify customer info	full	Could have unhandled errors
Display account dash board	full	Could have unhandled errors
Generate invoice as pdf	full	Could have unhandled errors
Process new payment	full	Could have unhandled errors
Get an account by account#	full	Could have unhandled errors
Search for account	full	Could have unhandled errors
Delete account	full	Could have unhandled errors
Login window	full	Could have unhandled errors
Help Mode	full	Could have unhandled errors
Actions' status messages	full	Could have unhandled errors
Send Invoice Email	partial	Could have unhandled errors, customer email addresses are not saved in the database.

5. This explains the main idea of the MainFrame class:

The main window of the program is constructed by a MainFrame class's object. It contains 4 main panels at north, south, west, and center of the window. The North and West panels will hold buttons and labels that provide all main functions such as adding new account, retrieving an account, viewing/ modifying account, showing dashboard, generate invoice, processing payment, etc. The South panel will hold labels serve as

responding messages for any time a main button or label is clicked. Example of message are: “Account# 1000 is retrieved successfully” or “Help mode is enabled” etc. The Center will hold main panels that corresponds to each main function. For example, an add-account panel is for the add-account function, a dashboard panel is for dashboard function etc. These panels are objects of user-defined classes that extends JPanel. These special panels had buttons, fields, and every thing needed to perform that function. At first, these panels are added to the Center panel, but are not visible. After an account is retrieved using GET or SEARCH, the account information get from the database will construct an account object of Accounts class. Then this account object is passed to each special panels using their public methods so that they can have information to display and perform needed functionalities on that account object. After a change is made. The program will execute SQL command using the new information to update the actual account in the database.



6. a/ Login window:



The image shows a screenshot of a Java Swing window titled "KC Energy". The window has a standard title bar with minimize, maximize, and close buttons. Inside the window, there is a blue lightning bolt icon inside a circle. Below the icon, there are two text labels: "User name" and "Password". Under "User name" is a text field containing the text "admin". Under "Password" is a password field with five dots and a cursor. At the bottom of the window is a blue button with the text "Login".

b/ Add new account

KC Energy

Account#: **Get**  **Reset** **Delete** 

Add new

Name:

Address:


City, State, Zipcode:

Phone#:

Price first 600 kWh:

Price next 400 kWh:

Price over 1000 kWh:



Choose a plan 

Add **Clear** **Modify**

<Ready to add new Account>

c/ Get account using account #

KC Energy

Account#: 1001 **Get**  **Reset** **Delete** 

Add new

View/Modify

Dash Board

Invoice

Payment

<Account# 1001 retrieved successfully!>

d/ View/ Modify account

KC Energy

Account#: 1001 Get Reset Delete ?

Add new

View/Modify

Dash Board

Invoice

Payment

Name: Michael Jordan

Address: 1220 NW 74th St

City, State, Zipcode: Kansas City MO 64134

Phone#: 816-663-3835

Price first 600 kWh: 0.25

Price next 400 kWh: 0.31

Price over 1000 kWh: 0.35

Add Clear Modify

resident

<Account# 1001 retrieved successfully!>

e/ Account dash board

KC Energy

Account#: 1001 Get Reset Delete ?

Add new

View/Modify

Dash Board

Invoice

Payment

Full Name: Michael Jordan Address: 1220 NW 74th St, Kansas City, MO 64134

Current Period: 04/01/2023 - 04/30/2023

Total Due: \$3.19

Last payment made on 2023-04-25

Current usage: 696.2 kwh

Last month usage: 697.4 kwh

Price first 600 kwh: \$0.0/kwh

Price next 400 kwh: \$0.0/kwh

Price over 1000 kwh: \$0.0/kwh

<Account# 1001 retrieved successfully!>

f/ Generate invoice

KC Energy

Account#: 1001

KC ENERGY Contact: 816-222-333

Invoice

Full Name : Michael Jordan
Account# : 1001
Phone# : 816-663-3835
Invoice date : 2023-04

<Account# 1001 retrieved successfully!>

g/ Process new payment

KC Energy

Account#: 1001

Michael Jordan, 1220 NW 74th St, Kansas City, MO 64134

Last payment of \$1.0 received on 2023-04-25

Remaining Balance:

New payment: Date:

<Account# 1001 retrieved successfully!>

h/ Search for an account

Searching account...

Search by: Name

5 result(s) found!

Acc#	Name	Adress	Phone#
1000	James Smith	1202 NW 72nd Terrace, Kansas City, MO 64158	816-223-3342
1004	James Smith	421 Gladstone Blvd, Kansas City, MO 64124	816-533-6548
1012	Thomas Jackson	217 Gladstone Blvd, Kansas City, MO 64123	816-214-4513
1015	Paul Jackson	828 E 22nd Ave, North Kansas City, MO 64116	913-456-7894
1017	Paul Jackson	1107 N. 1st St, Kansas City, MO 64108	816-463-8353

Account# 1004 is selected! Please click OK to continue.

i/ Delete account

KC Energy

Account#:

Action can't be undone

Are you sure to delete account 1004?


<Account# 1004 retrieved successfully!>

j/ Help Mode

KC Energy


Account#: 1004

Get



Reset

Delete



Add new

View/Modify

Dash Board

Invoice

Payment

Existing account can be retrieved without account# using SEARCH.
After click the search icon: Step1: Choose a category to search.

Search by:

Name

Search

Step2: type in a few letters and click search.

Search by:

Name

Search

Step3: Choose one of the result(s) if there is any.

3 result(s) found!

Acc#	Name	Adress
1012	Thomas Jackson	217 Gladstone
1014	Daniel Thomas	829 E 21st Av
1019	Robert Thomas	4235 W Grand

Step4: Click OK to continue.

Account# 1014 is selected! Please click OK to continue.

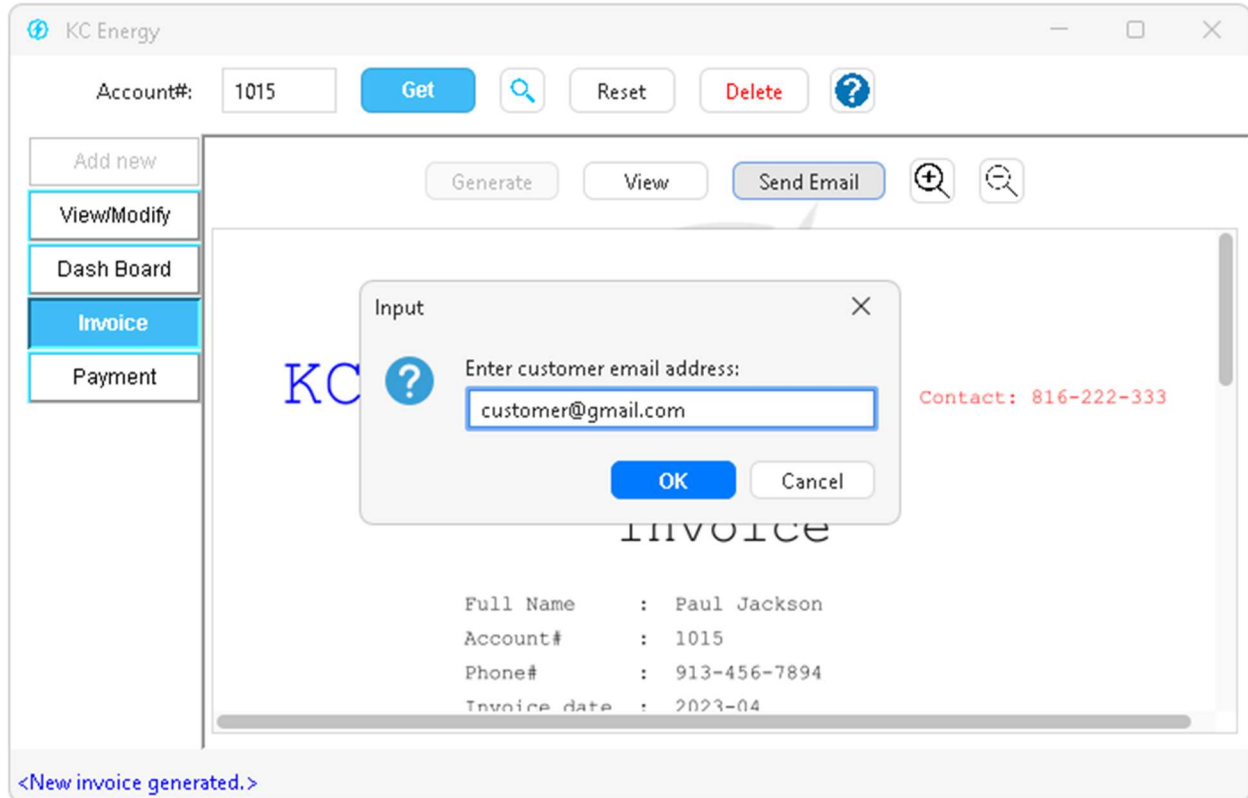
OK

<Help mode is enabled>

k/ Status messages:

<Ready to add new Account> <Account# 1001 retrieved successfully!>

1/ Send Email



7. Reference

Baeldung. (2023, January 21). Sending Emails with Attachments in Java.

<https://www.baeldung.com/java-send-emails-attachments>

Java2s. Java Tutorial.

http://www.java2s.com/Tutorial/Java/0240__Swing/publicJTableVectorrowDataVectorcolumnNames.htm

http://www.java2s.com/Tutorials/Java/Swing_How_to/Basic/Add_Background_image_to_JPanel.htm

Nilhcem. Fake SMTP Server. <https://nilhcem.com/FakeSMTP/>

Phwoelfel (2017, June 9). Multi-line tooltips in Java? Stack Overflow.

<https://stackoverflow.com/questions/868651/multi-line-tooltips-in-java>

Rose India. Showing an image in Tool Tip.

<https://www.roseindia.net/java/example/java/swing/ImageToolTip.shtml>

Sage (2013, November 19). Dynamically updating a JTable from a

vector<vector<string>>. Stack Overflow.

<https://stackoverflow.com/questions/20073362/dynamically-updating-a-jtable-from-a-vectorvectorstring>

Sarathi Technology (2022, March 16). Apache PDFBox Tutorial. Youtube.

<https://www.youtube.com/playlist?list=PLFh8wpMiEi88vWlQJj4KDzfpbIebBWIsX>

Tilman Hausherr (2014, April 27). Convert PDF files to images with PDFBox. Stack

Overflow. <https://stackoverflow.com/questions/23326562/convert-pdf-files-to-images-with-pdfbox>

Toedter. JCalendar. <https://toedter.com/jcalendar/>