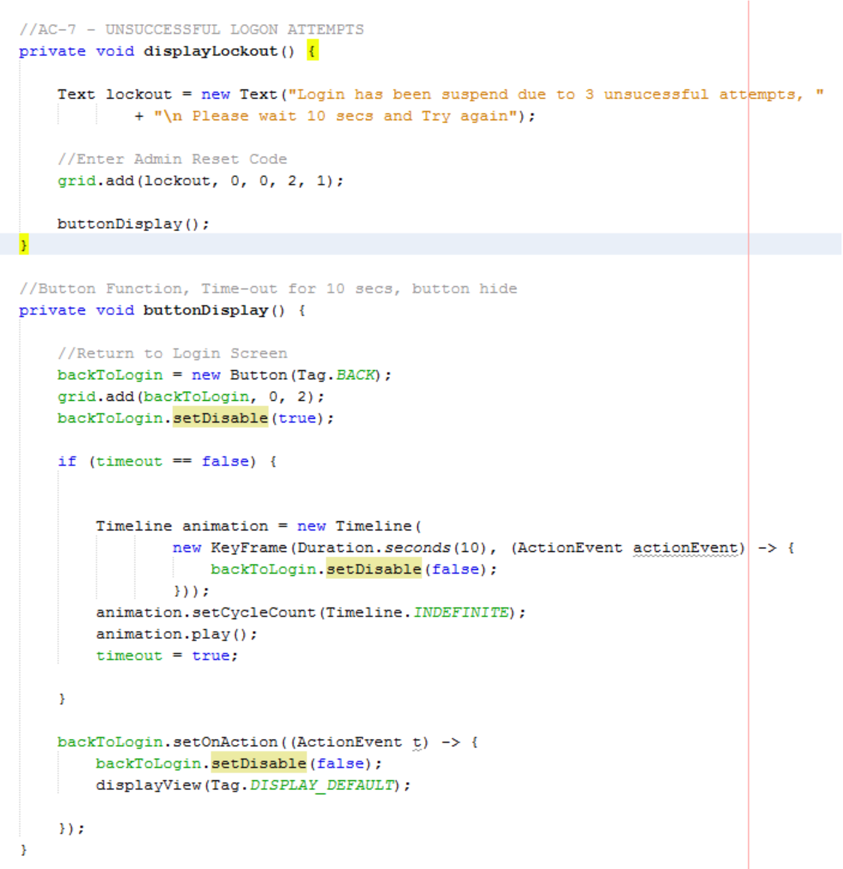
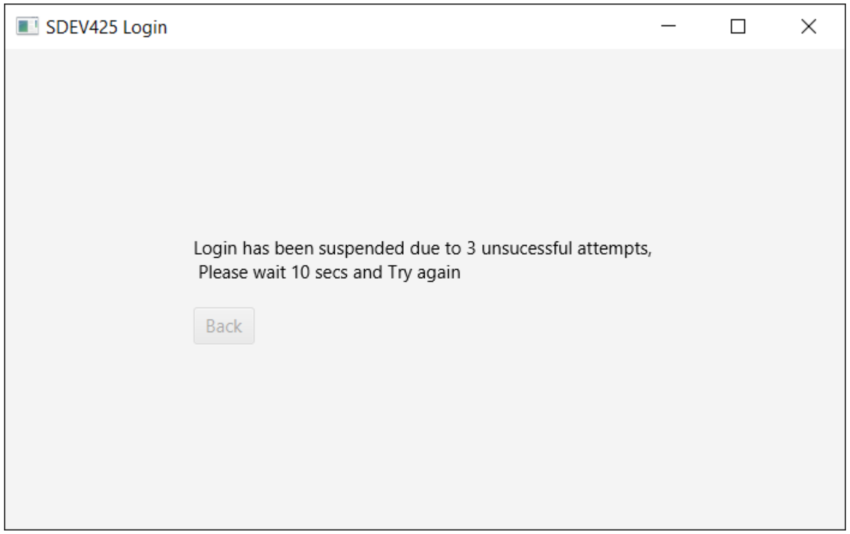
Homework 2

AC-7 - UNSUCCESSFUL LOGON ATTEMPTS:

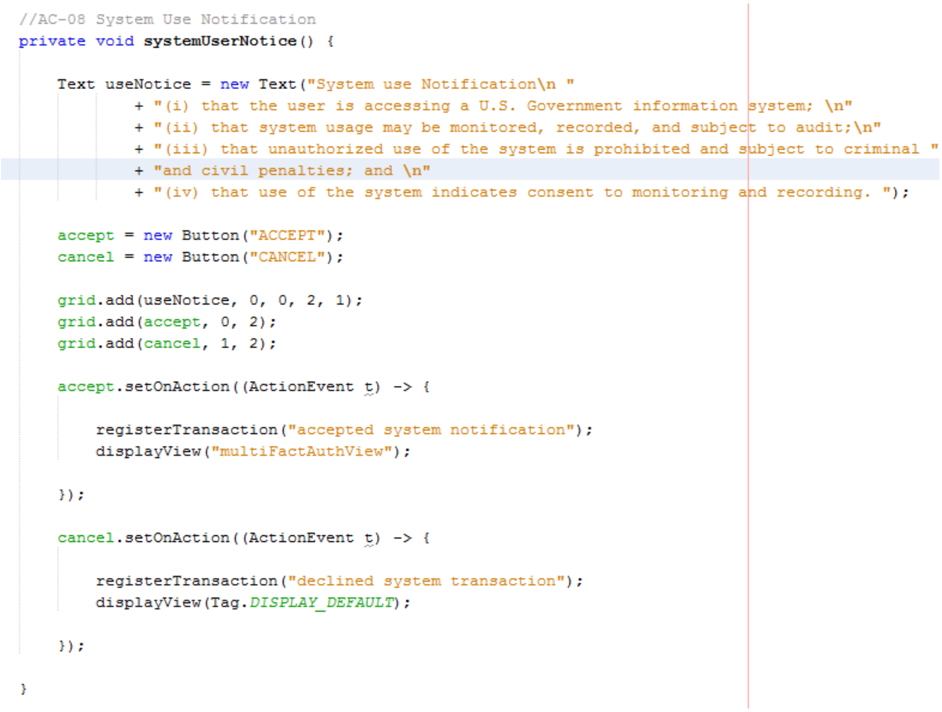


```
//AC-7 - UNSUCCESSFUL LOGON ATTEMPTS
private void displayLockout() {

    Text lockout = new Text("Login has been suspend due to 3 unsucessful attempts, "
            + "\n Please wait 10 secs and Try again");

    //Enter Admin Reset Code
    grid.add(lockout, 0, 0, 2, 1);

    buttonDisplay();
}

//Button Function, Time-out for 10 secs, button hide
private void buttonDisplay() {

    //Return to Login Screen
    backToLogin = new Button(Tag.BACK);
    grid.add(backToLogin, 0, 2);
    backToLogin.setDisable(true);

    if (timeout == false) {


        Timeline animation = new Timeline(
                new KeyFrame(Duration.seconds(10), (ActionEvent actionEvent) -> {
                    backToLogin.setDisable(false);
                }));
        animation.setCycleCount(Timeline.INDEFINITE);
        animation.play();
        timeout = true;

    }

    backToLogin.setOnAction((ActionEvent t) -> {
        backToLogin.setDisable(false);
        displayView(Tag.DISPLAY_DEFAULT);

    });
}
```

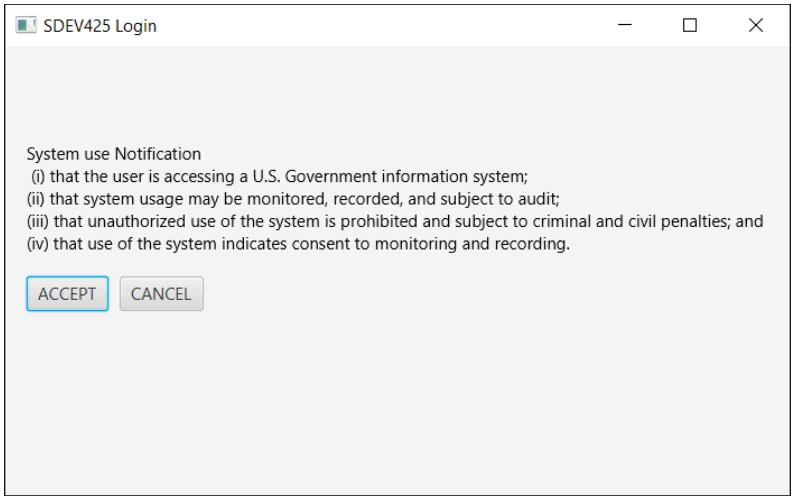


Login has been suspended due to 3 unsucessful attempts,
Please wait 10 secs and Try again

AC-8 - SYSTEM USE NOTIFICATION

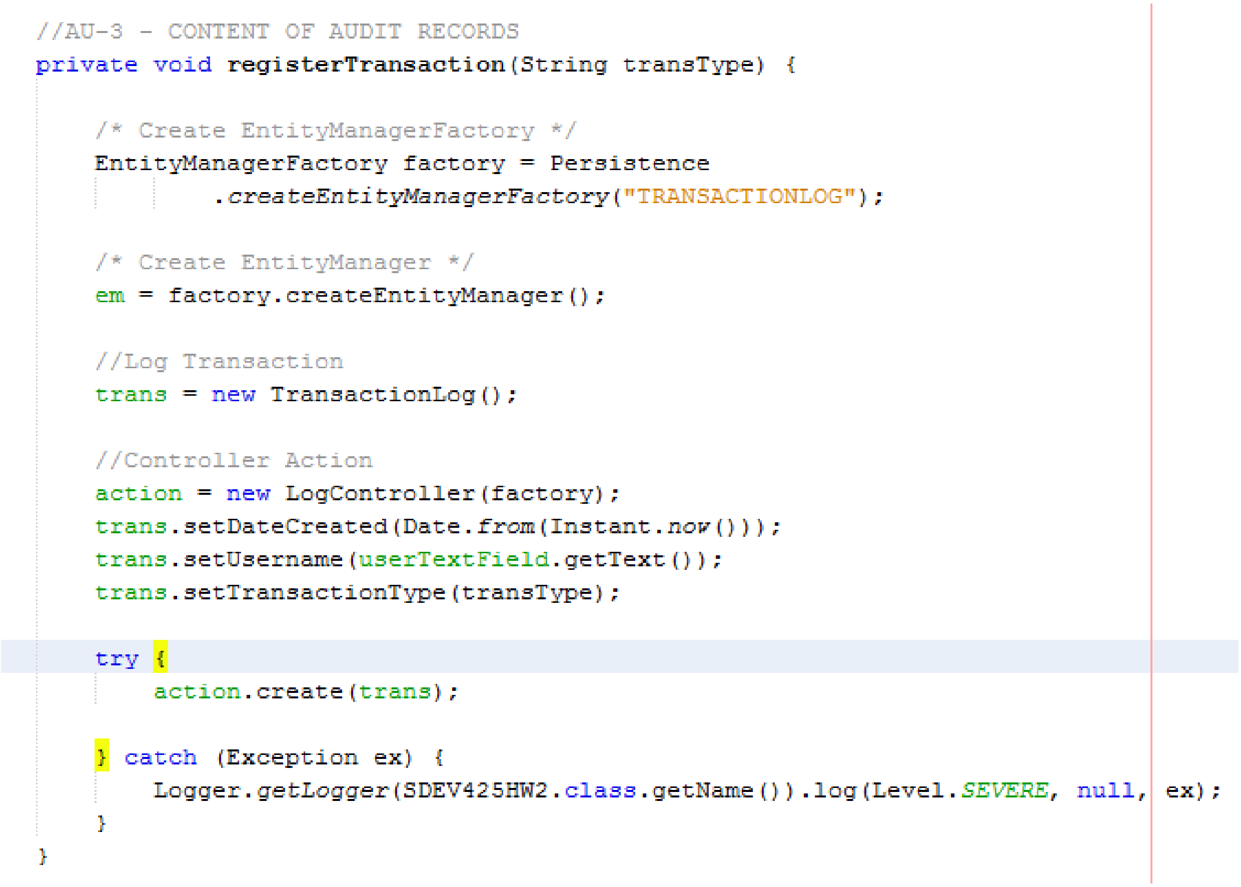


```java
//AC-08 System Use Notification
private void systemUserNotice() {

    Text useNotice = new Text("System use Notification\n "
            + "(i) that the user is accessing a U.S. Government information system; \n"
            + "(ii) that system usage may be monitored, recorded, and subject to audit;\n"
            + "(iii) that unauthorized use of the system is prohibited and subject to criminal "
            + "and civil penalties; and \n"
            + "(iv) that use of the system indicates consent to monitoring and recording. ");

    accept = new Button("ACCEPT");
    cancel = new Button("CANCEL");

    grid.add(useNotice, 0, 0, 2, 1);
    grid.add(accept, 0, 2);
    grid.add(cancel, 1, 2);

    accept.setOnAction((ActionEvent t) -> {

        registerTransaction("accepted system notification");
        displayView("multiFactAuthView");

    });

    cancel.setOnAction((ActionEvent t) -> {

        registerTransaction("declined system transaction");
        displayView(Tag.DISPLAY_DEFAULT);

    });

}
```



System use Notification
 (i) that the user is accessing a U.S. Government information system;
(ii) that system usage may be monitored, recorded, and subject to audit;
(iii) that unauthorized use of the system is prohibited and subject to criminal and civil penalties; and
(iv) that use of the system indicates consent to monitoring and recording.

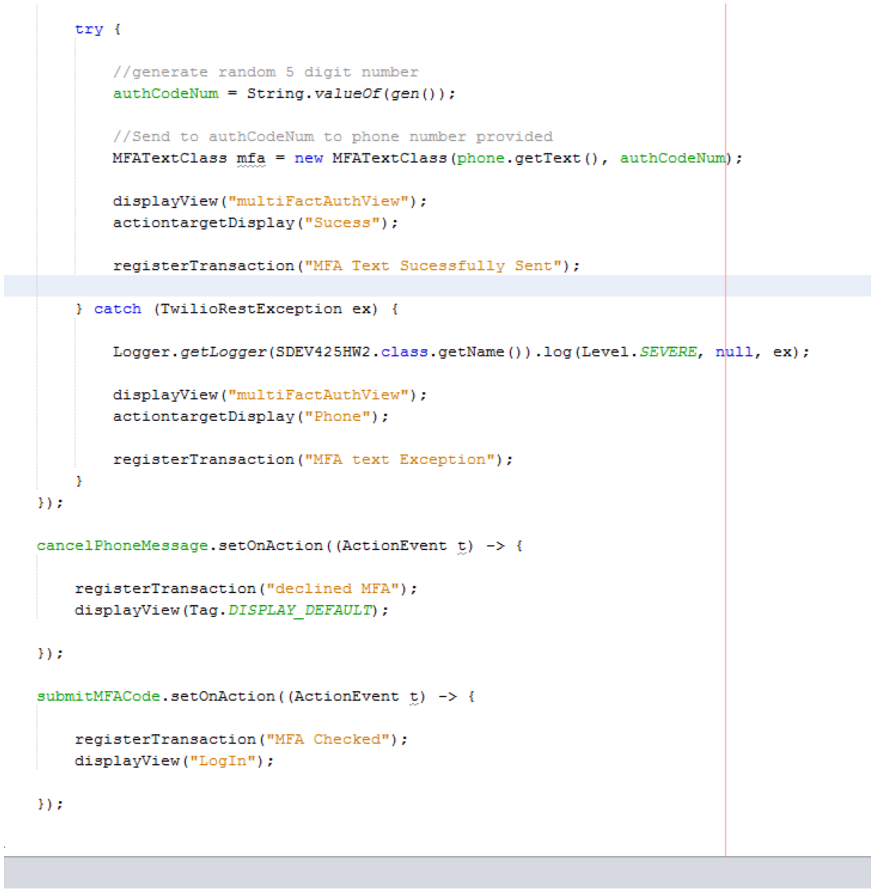
AU-3 - CONTENT OF AUDIT RECORDS & AU-8 - TIME STAMPS:



```
//AU-3 - CONTENT OF AUDIT RECORDS
private void registerTransaction(String transType) {

    /* Create EntityManagerFactory */
    EntityManagerFactory factory = Persistence
            .createEntityManagerFactory("TRANSACTIONLOG");

    /* Create EntityManager */
    em = factory.createEntityManager();

    //Log Transaction
    trans = new TransactionLog();

    //Controller Action
    action = new LogController(factory);
    trans.setDateCreated(Date.from(Instant.now()));
    trans.setUsername(userTextField.getText());
    trans.setTransactionType(transType);

    try {
        action.create(trans);

    } catch (Exception ex) {
        Logger.getLogger(SDEV425HW2.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

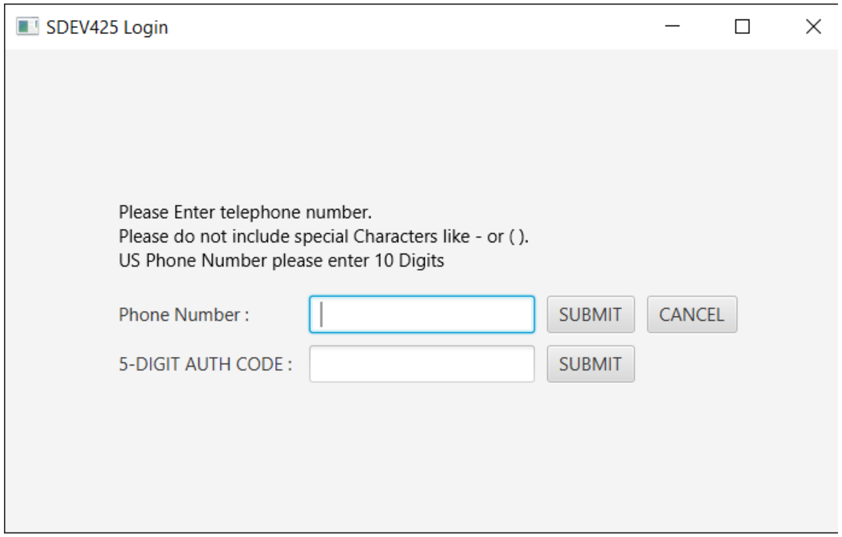
This is method stores audit records into a database table. Information can later be retrieved via an excel sheet. The method is added throughout the program to record user's interaction with the system.

IA-2(1) IDENTIFICATION AND AUTHENTICATION (ORGANIZATIONAL USERS) | NETWORK ACCESS TO PRIVILEGED ACCOUNTS



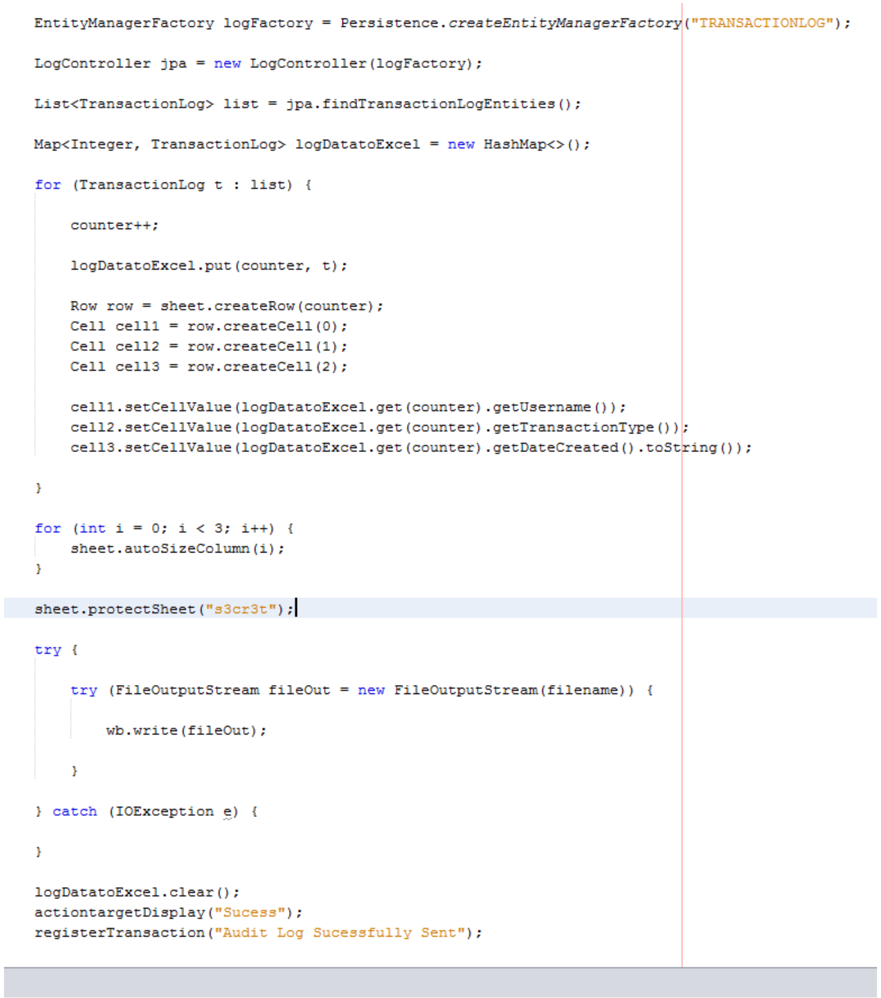
```
    try {

        //generate random 5 digit number
        authCodeNum = String.valueOf(gen());

        //Send to authCodeNum to phone number provided
        MFATextClass mfa = new MFATextClass(phone.getText(), authCodeNum);

        displayView("multiFactAuthView");
        actiontargetDisplay("Sucess");

        registerTransaction("MFA Text Sucessfully Sent");

    } catch (TwilioRestException ex) {

        Logger.getLogger(SDEV425HW2.class.getName()).log(Level.SEVERE, null, ex);

        displayView("multiFactAuthView");
        actiontargetDisplay("Phone");

        registerTransaction("MFA text Exception");
    }
});

cancelPhoneMessage.setOnAction((ActionEvent t) -> {

    registerTransaction("declined MFA");
    displayView(Tag.DISPLAY_DEFAULT);

});

submitMFACode.setOnAction((ActionEvent t) -> {

    registerTransaction("MFA Checked");
    displayView("LogIn");

});
```

Twilio API used to send text message to user with authorization code.



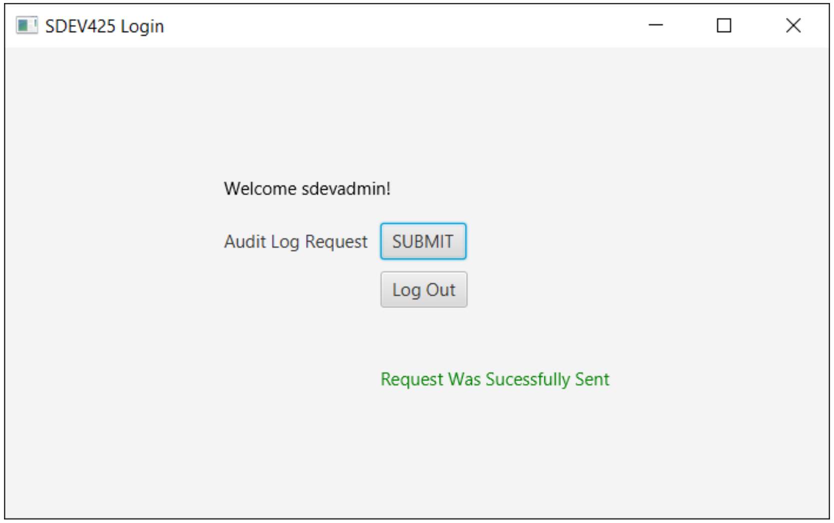
Please Enter telephone number.
Please do not include special Characters like - or ( ).
US Phone Number please enter 10 Digits

AU-9(6) - PROTECTION OF AUDIT INFORMATION | READ ONLY ACCESS:



```
EntityManagerFactory logFactory = Persistence.createEntityManagerFactory("TRANSACTIONLOG");

LogController jpa = new LogController(logFactory);

List<TransactionLog> list = jpa.findTransactionLogEntities();

Map<Integer, TransactionLog> logDatatoExcel = new HashMap<>();

for (TransactionLog t : list) {

    counter++;

    logDatatoExcel.put(counter, t);

    Row row = sheet.createRow(counter);
    Cell cell1 = row.createCell(0);
    Cell cell2 = row.createCell(1);
    Cell cell3 = row.createCell(2);

    cell1.setCellValue(logDatatoExcel.get(counter).getUsername());
    cell2.setCellValue(logDatatoExcel.get(counter).getTransactionType());
    cell3.setCellValue(logDatatoExcel.get(counter).getDateCreated().toString());

}

for (int i = 0; i < 3; i++) {
    sheet.autoSizeColumn(i);
}

sheet.protectSheet("s3cr3t");

try {

    try (FileOutputStream fileOut = new FileOutputStream(filename)) {

        wb.write(fileOut);

    }

} catch (IOException e) {

}

logDatatoExcel.clear();
actiontargetDisplay("Sucess");
registerTransaction("Audit Log Sucessfully Sent");
```

Database records are outputted to a protected excel file.



Welcome sdevadmin!

Request Was Sucessfully Sent