**Assignment: Computer Science (CSCI 120) Project Submission  
Instructor: Dr. Sajid Hussain   
Due date: 31st of May, 2022  
Project Title: ATM Demo System  
Creator: Oladejo Oluwatamilore Pelumi**

***Background***

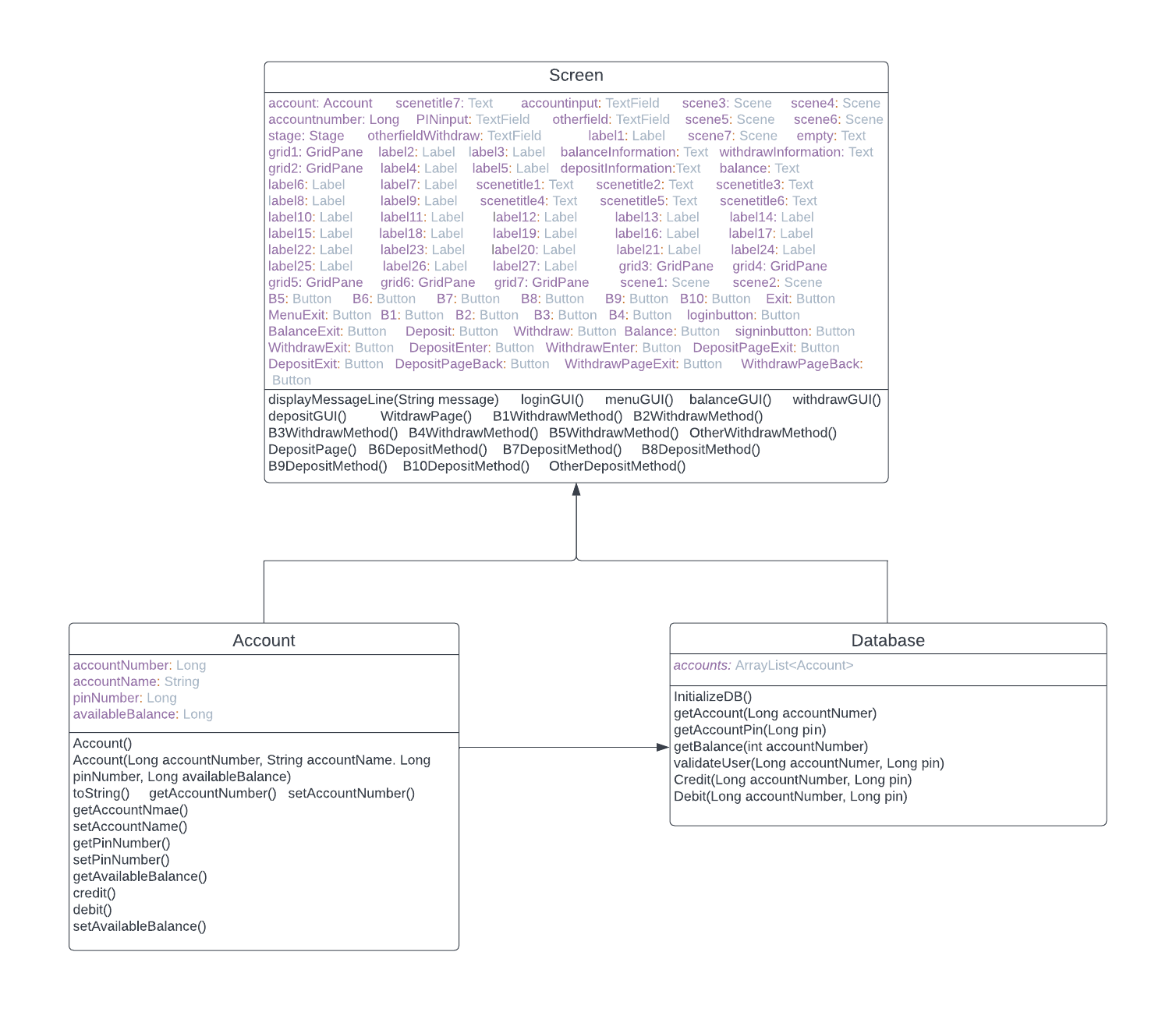
The purpose behind this project was to create an application in java programming language of at least 1500 lines of code (including comments) with a readme file and a working Graphical User Interface. While depositing a sum of at the bank, the idea suddenly hit me to develop a program that performs the basic functions of an ATM (Automated Teller Machine).

***Project Summary***

This project documentary explains the basic functionality of the program. The application was written in JAVA programming Language with compatible JAVAFX Interface (GUI). Once initiated the program asks the user for his Account number and PIN at the login page and cross references it from a CSV of authenticated users. If the details match the user is taken to a menu page where he can decide either what action to be carried out. 1 – Check user’s Available balance, 2 – Make a withdraw, 3 – Make a Deposit, and 4 – Exit the program (Sign out). I created an Account class for each object with data members: Account number, Account name, PIN, and Available Balance. I also separated each action and stage into functions for easy styling of my project.

***UML Diagram***

The figure below is a Unified Modeling Language (UML) diagram that visually represents architecture, design and implementation of this software. The respective Java classes and methods used are represented in the diagram below.



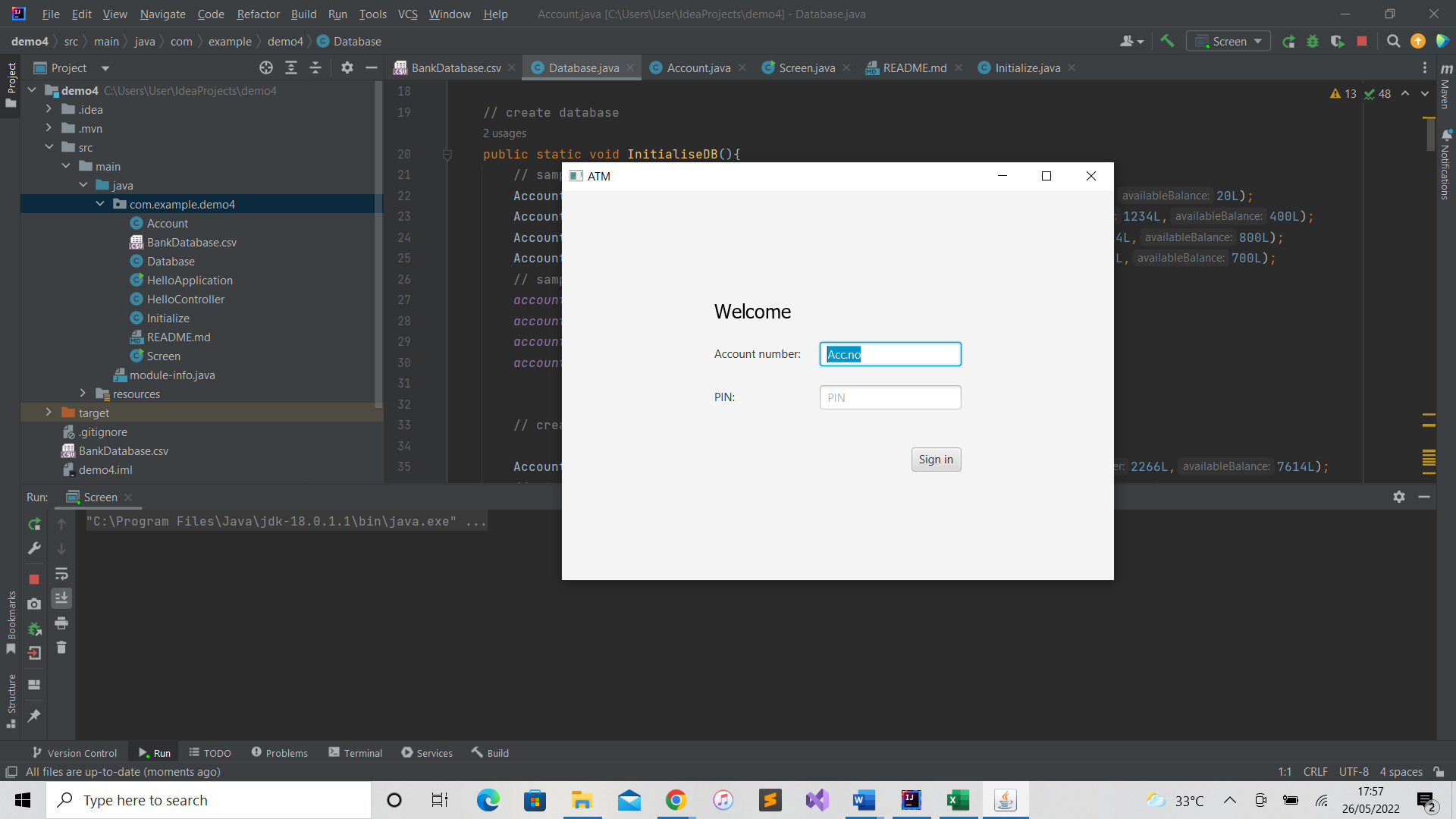
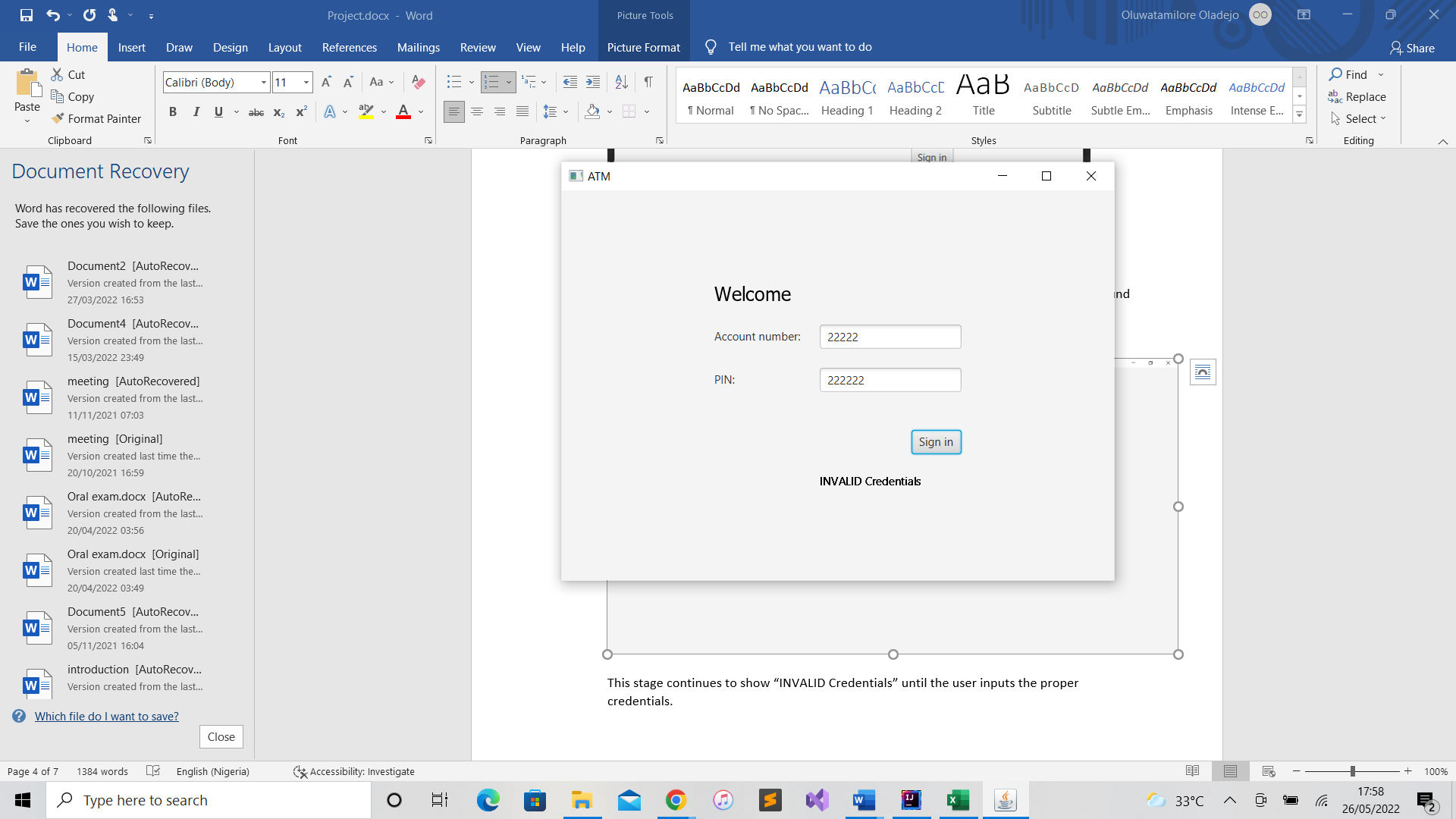
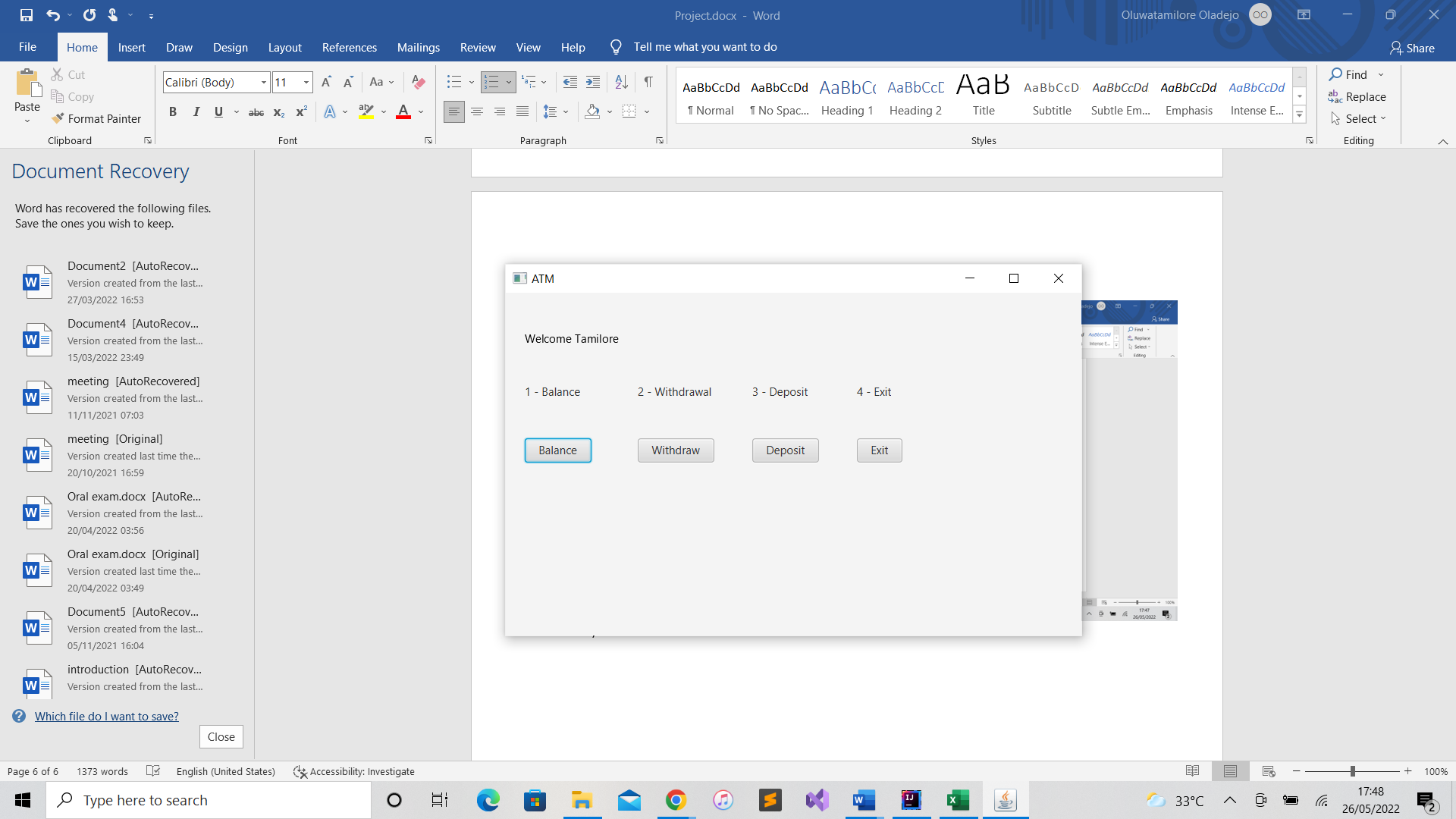
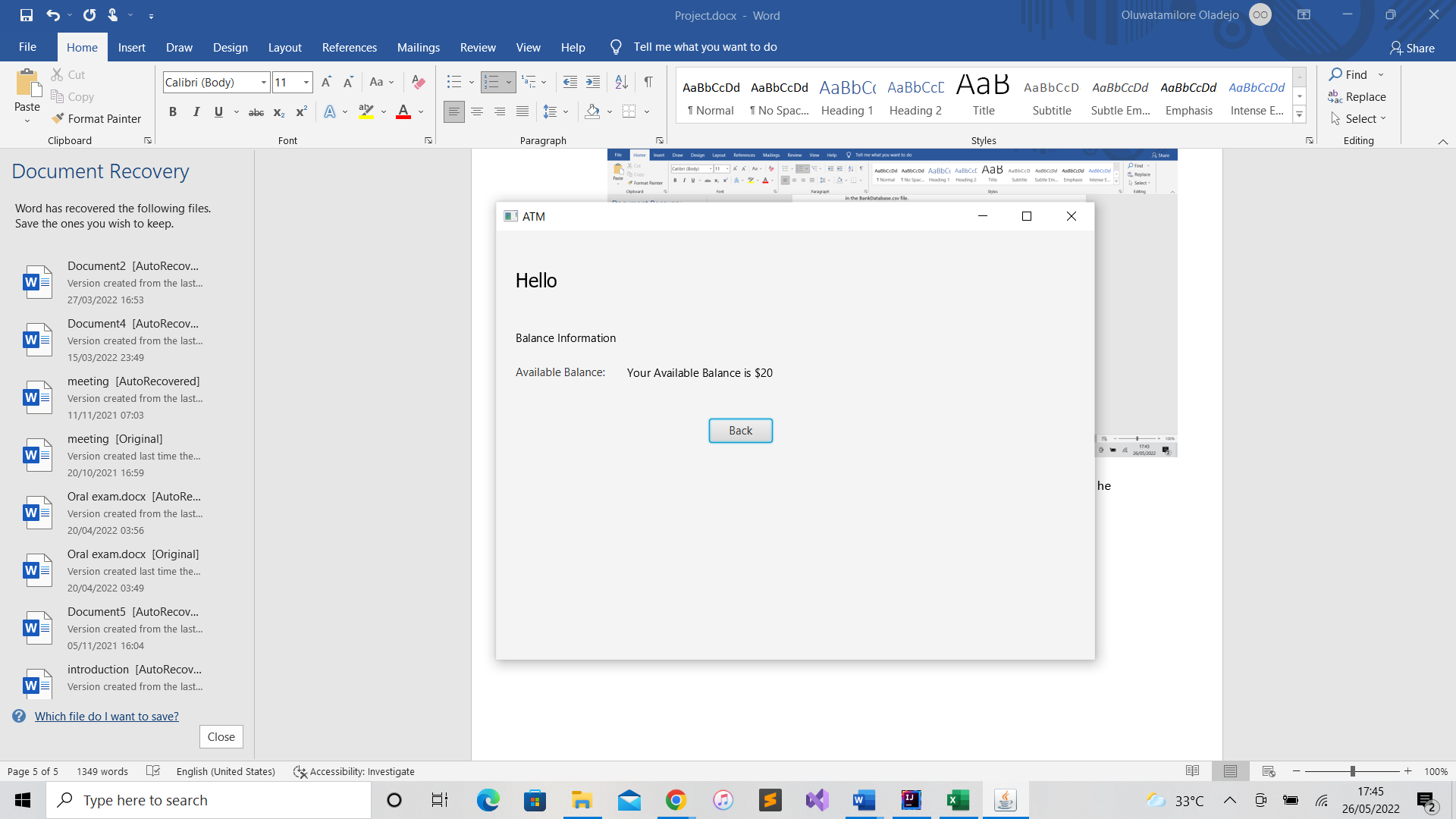
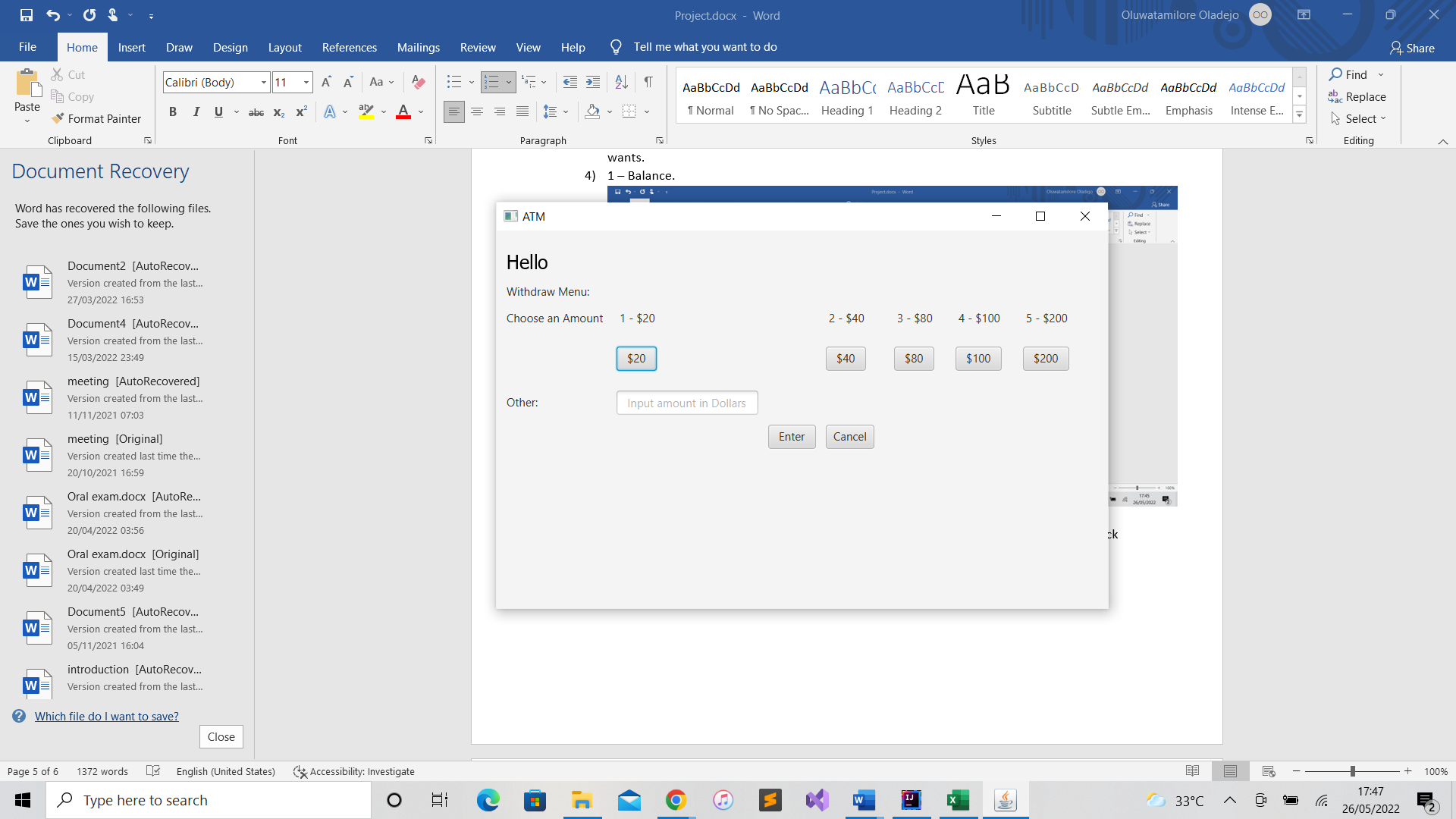
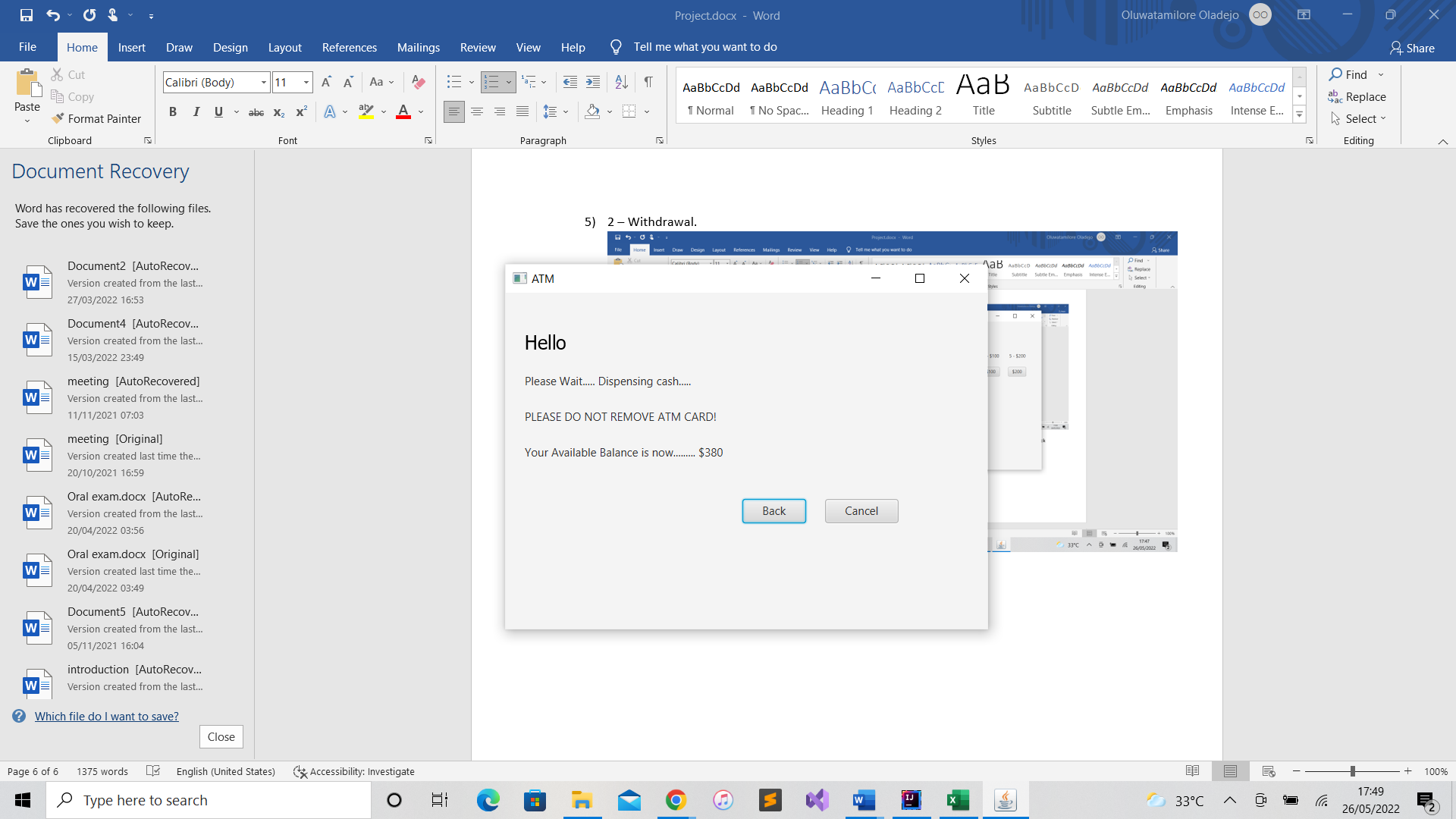
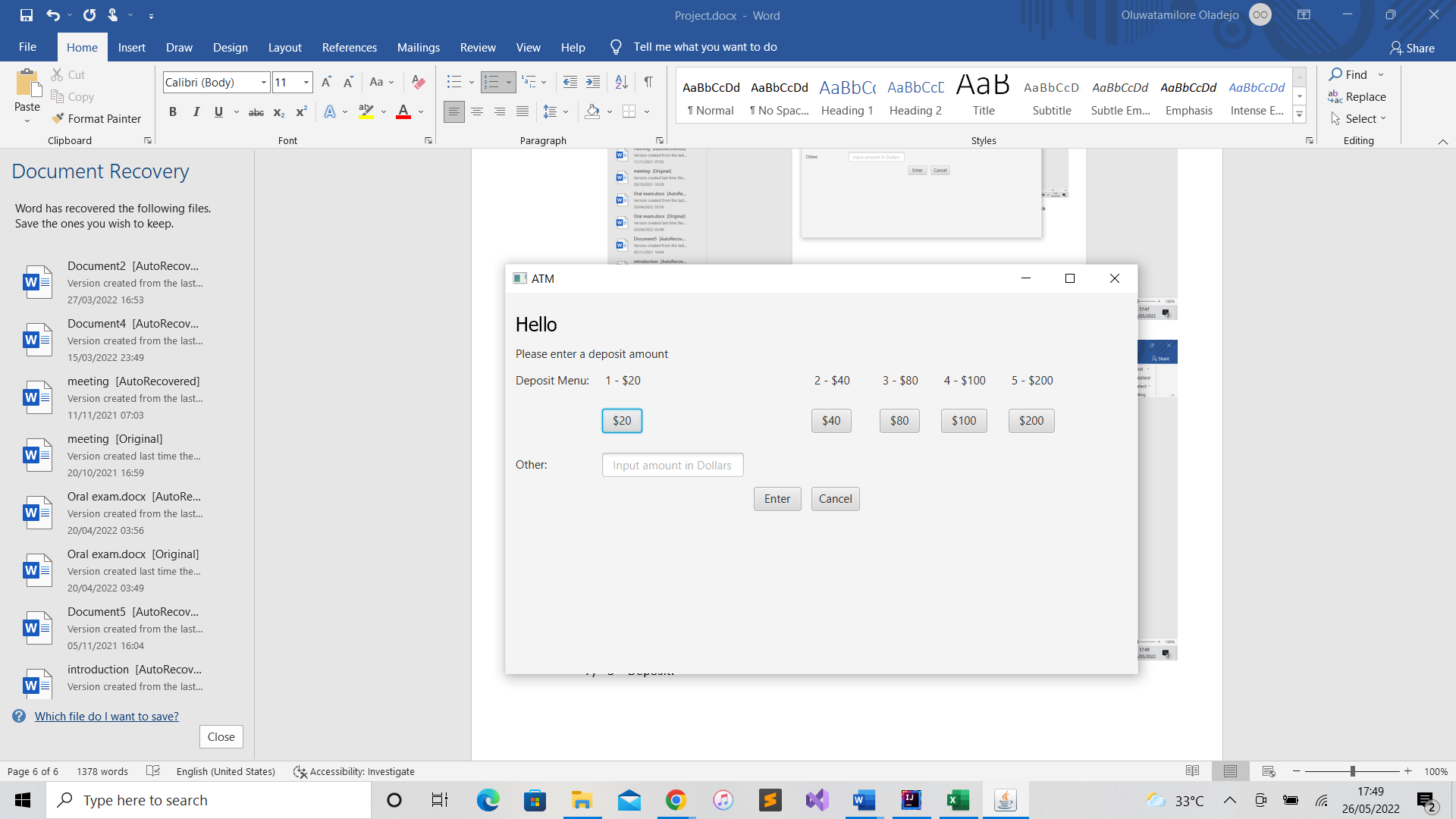
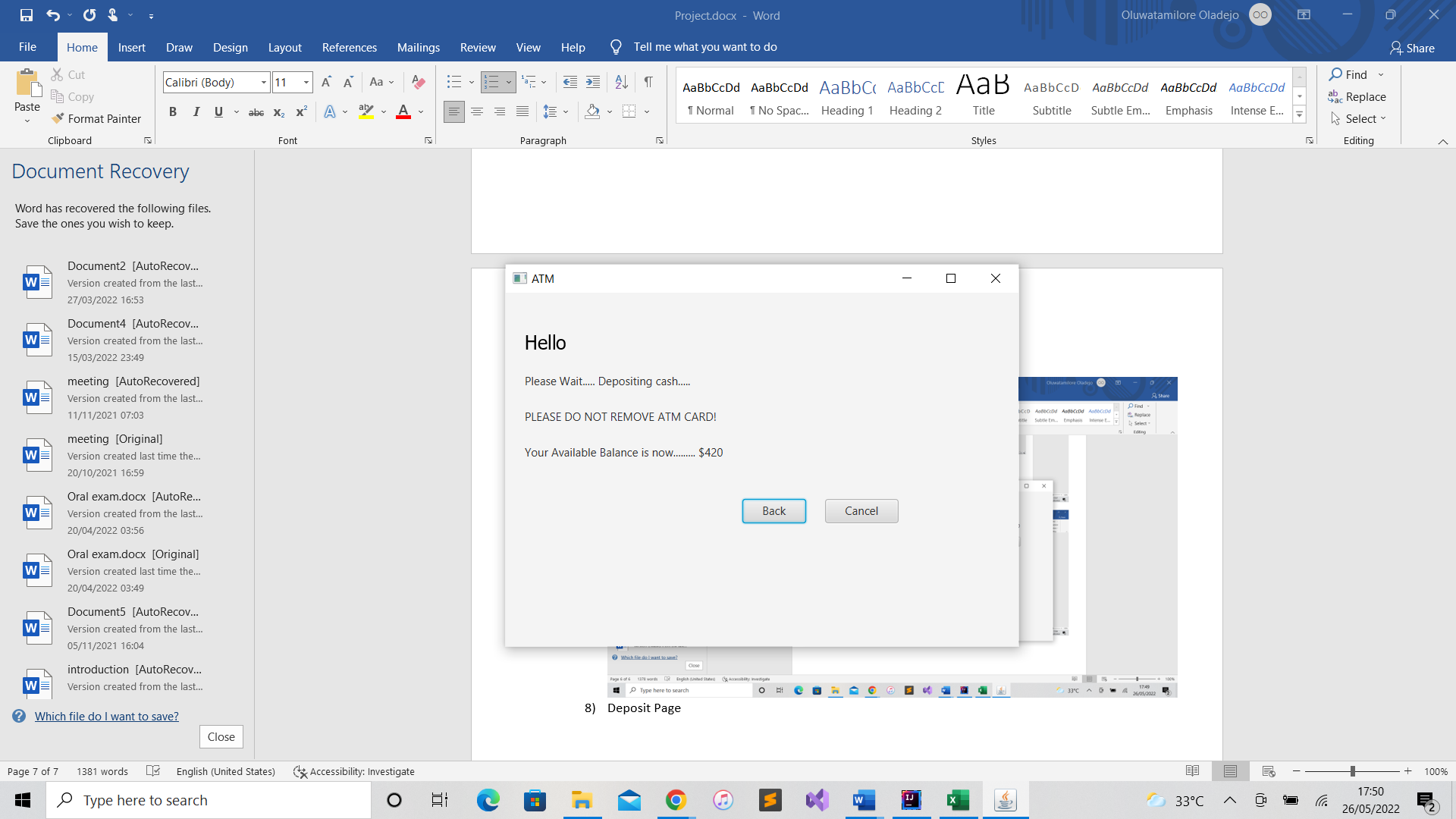
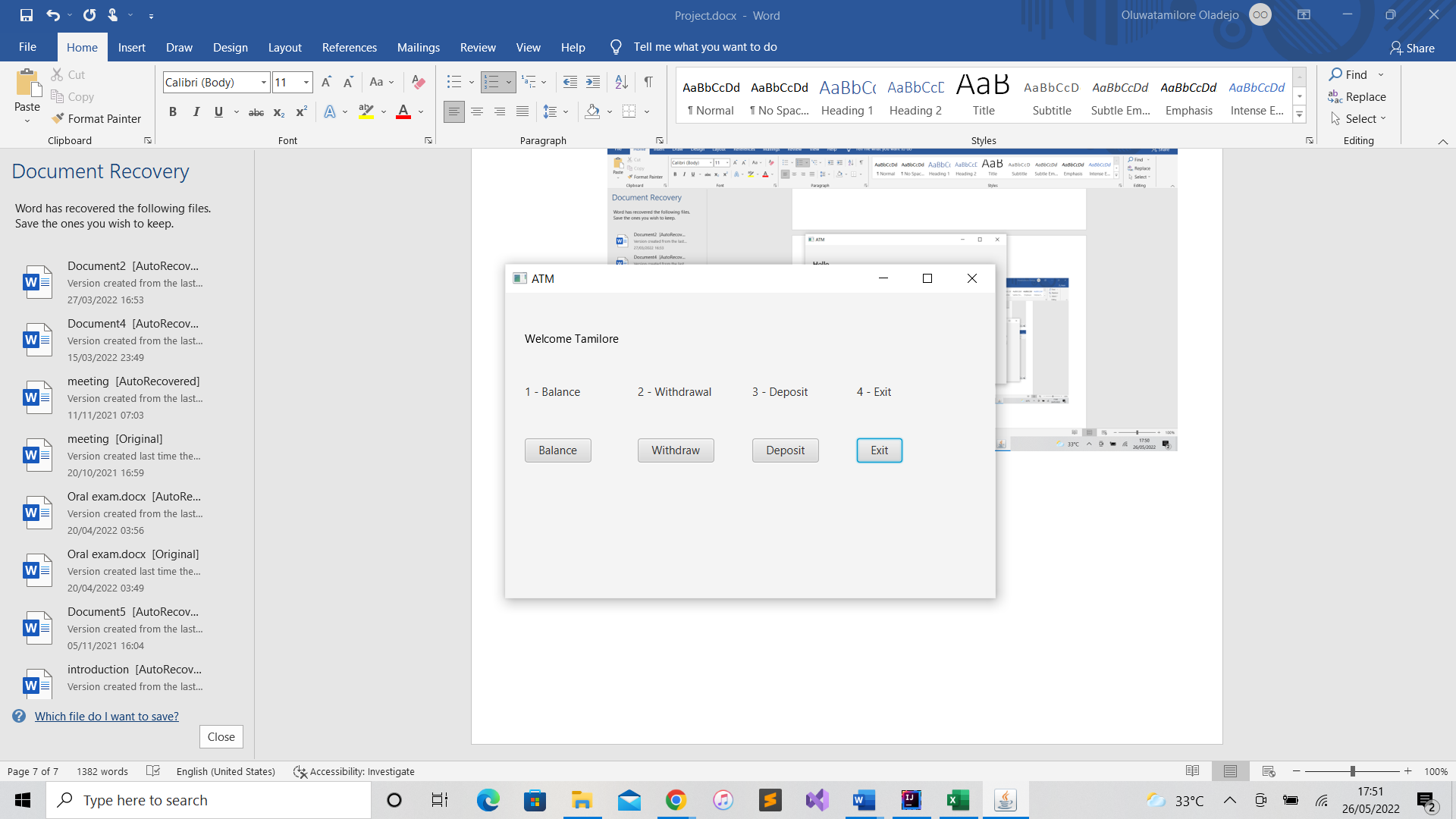
***Functionalities***

Below are brief descriptions of the classes and methods used in the creation of this demo.

1. Account class: This class was used as a format on how data would be stored in the database. The data members are as follows Account number, Account name, PIN, and Available Balance.
   1. Account(): This serves as a constructor for the class. Which means it helps store the values and attributes of the class.
   2. toString(): This function helps in returning Account objects. It converts the data members to String for it to be printed on the console.
   3. getAccountNumber(): it looks through the database and returns the account number of the current user.
   4. setAccountNumber(accountNumber): it assigns the current user an account number.
   5. getAccountName(): This method looks through the database and returns the Account name of the current user.
   6. setAccountName(): This method assigns an account name to a user.
   7. getPinNumber(): This method searches the database to return the PIN of the current user.
   8. setPinNumber(PIN): This method assigns the user a PIN.
   9. getAvailabelBalance(): This method searches through the database to return the user’s available balance.
   10. setAvailableBalance(availableBalance): It assigns the user a particular Available Balance.
   11. credit(amount): This method credits a user a particular amount.
   12. debit(amount): This method debits the user a particular amount.
2. Database class: This class was used to generate and store the valid account details in the database. It also contained functions to help in ensuring program functionality.
   1. InitializeDB(): This method creates account objects and groups them in an Array List to be stored in the database.
   2. getAccount(accountnumber): It pulls out all of the user’s information from the database using the inputted account number.
   3. getAccountPin(Pin): It pulls out all of the user’s information from the database using the inputted PIN.
   4. authenticateUser(accountnumber, pin): This method returns true if the information retrieved from getAccount() and getAccountPin match, and returns false if they don’t.
   5. validateUser(accountnumber, pin): This method checks the database using the inputted Account number and Pin to validate if the user exists or not.
   6. getBalance(accountnumber): This method returns the user’s balance using the account number inputted.
3. Screen class: This class generates the GUI with every stage and action of the program. The stages have also been put into functions for better style and functionality.
   1. dislayMessage(message): This method helps the user print a particular message on the console.
   2. displayMessageLine(message): This method helps the user print a particular message on a single line of the console.
   3. displayAmount(amount): This method prints a particular amount on the console.
   4. loginGUI(): This is the start page that prompts the user to input his Account and PIN numbers and validates it that means it checks the database if the credentials are of a valid user or not
   5. menuGUI(): This is the second page of the program after the login page. Once the user has been verified, he moves to the menu page where he can choose what action to perform. He can choose 1 – Check his balance, 2 – make a withdraw, 3 – make a deposit, 4 – sign out.
   6. balanceGUI(): This is the third page of the program after the menu page. If in the menu page the user decides to check his balance the program moves him to this page where other background functions run in order to print out the user’s current available balance on the GUI screen and console.
   7. withdrawGUI(): This is the fourth page of the program after the menu page. If in the menu page the user decides to make a withdraw the program moves him to this page where other background functions run prompting him to choose a particular amount to withdraw from the user’s account.
   8. B1WithdrawMethod(): This method is called in the withdrawal stage. It uses the debit function from the Account class to debit the user $20.
   9. B2WithdrawMethod(): This method is called in the withdrawal stage. It uses the debit function from the Account class to debit the user $40.
   10. B3WithdrawMethod(): This method is called in the withdrawal stage. It uses the debit function from the Account class to debit the user $80.
   11. B4WithdrawMethod(): This method is called in the withdrawal stage. It uses the debit function from the Account class to debit the user $100.
   12. B5WithdrawMethod(): This method is called in the withdrawal stage. It uses the debit function from the Account class to debit the user $200.
   13. OtherWithdrawMethod(): This method is called in the withdrawal stage. It prompts the user to input a withdraw amount and uses the debit function from the Account class to debit the user the inputted amount.
   14. depositGUI(): This is the fifth page of the program after the menu page. If in the menu page the user decides to make a deposit the program moves him to this page where other background functions run prompting him to choose a particular amount to deposit into the user’s account.
   15. B6DepositMethod(): This method is called in the Deposit stage. It uses the credit function from the Account class to credit the user $20.
   16. B7DepositMethod(): This method is called in the deposit stage. It uses the credit function from the Account class to credit the user $40.
   17. B8DepositMethod(): This method is called in the deposit stage. It uses the credit function from the Account class to credit the user $80.
   18. B9DepositMethod(): This method is called in the deposit stage. It uses the credit function from the Account class to credit the user $100.
   19. B10DepositMethod(): This method is called in the deposit stage. It uses the credit function from the Account class to credit the user $200.
   20. OtherDepositMethod(): This method is called in the deposit stage. It prompts the user to input a deposit amount and uses the credit function from the Account class to credit the user the inputted amount.

***Test Run***

This will show a step-by-step work functionality of my program.

1. This is the first stage that occurs once the user runs the Screen.java class file.  
   At this point the user is prompted to input an account number and a PIN which can be found in the BankDatabase.csv file.  
     
   
2. This is what happens if the person inputs the wrong credentials.  
   This stage continues to show “INVALID Credentials” until the user inputs the proper credentials.  
     
   
3. Once the user has been verified, he moves to scene 3.  
   At this point the user has been proven valid by the system and can perform any action he wants.   
     
   
4. 1 – Balance.  
   This scene displays the user’s current balance and presses the “Back” button to move to the previous scene (menu page).  
     
   
5. 2 – Withdrawal.  
   This s second action the user can perform in which he can make a withdrawal of a certain amount of money from the sum in his account to be taken out as cash.  
     
   
6. Withdraw Page.  
   This occurs at the end of the withdrawal stage. At this point the program notifies the user that the transaction was successful and informs him about his current Available Balance.  
   
7. 3 – Deposit.  
   This is the third possible kind of transaction in which the user can decide to make a deposit into is account. The user chooses a button to make a deposit of that particular amount into his account or can decide to type the amount in the text field marked “other”. He can also use the cancel button to return to the previous stage (menu stage).  
   
8. Deposit Page  
   This occurs at the end of the deposit stage. At this point the program notifies the user that the transaction was successful and informs him about his current Available Balance.  
     
   
9. 4 – Exit.   
   This is the final button on the menu page. The user can press this button as a way to sign out after all transactions have been performed. It forces the program to come to an end reverting all values to their original value.   
     
   

***Future Enhancements***

This part of my documentation contains some innovative features that I would like to add in the upcoming future.  
1. I would like to add some pictures in order to improve user interface.

2. I would like to add a feature for someone to be able to create a new user to be added into the database.

3. I would like to add a function that notifies people when a withdrawal or deposit has been made into their account.

4. I would like to add a function that prevents people from withdrawing more money than the amount in their account.