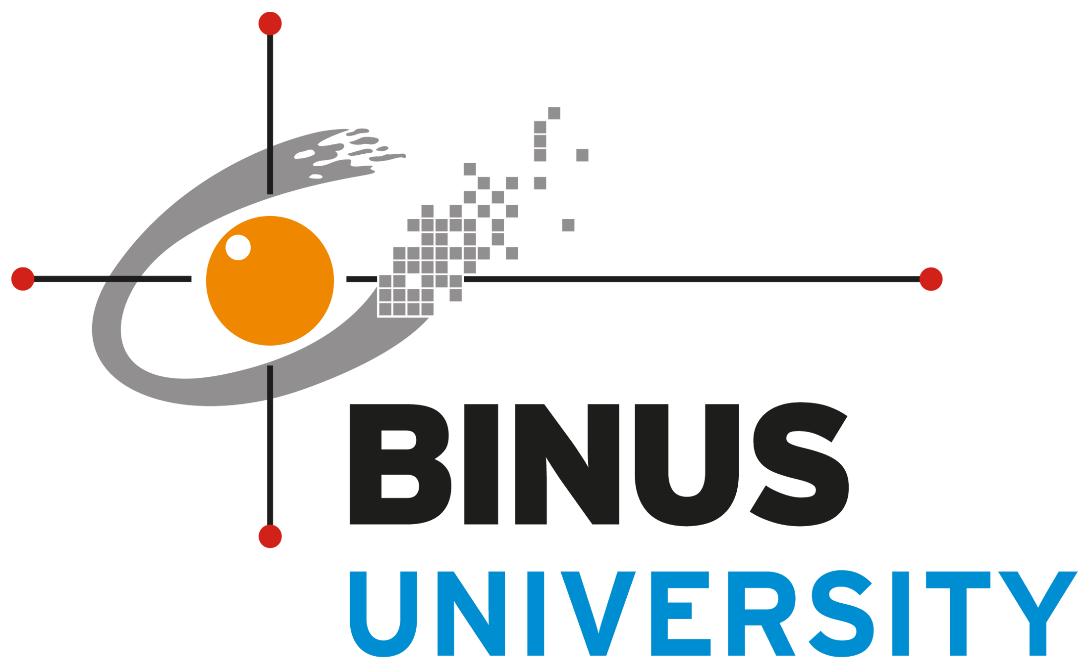
**OBJECT ORIENTED PROGRAMMING FINAL PROJECT REPORT**



**BY: ARVIN YUWONO (2502009721)**

**Project Specification**

The application is a digitalized version of *Buku Kas,* a physical book that records the income, expense and balance per each transaction made by a household or company. Its purpose is to monitor the flow of money within the household or company. This application was created to make it easier for everyone to be able to do so without going through the troublesome effort of writing formulas in Excel. The access a User has to the application depends on the level of authorization they have been granted.

**UML Class Diagram**

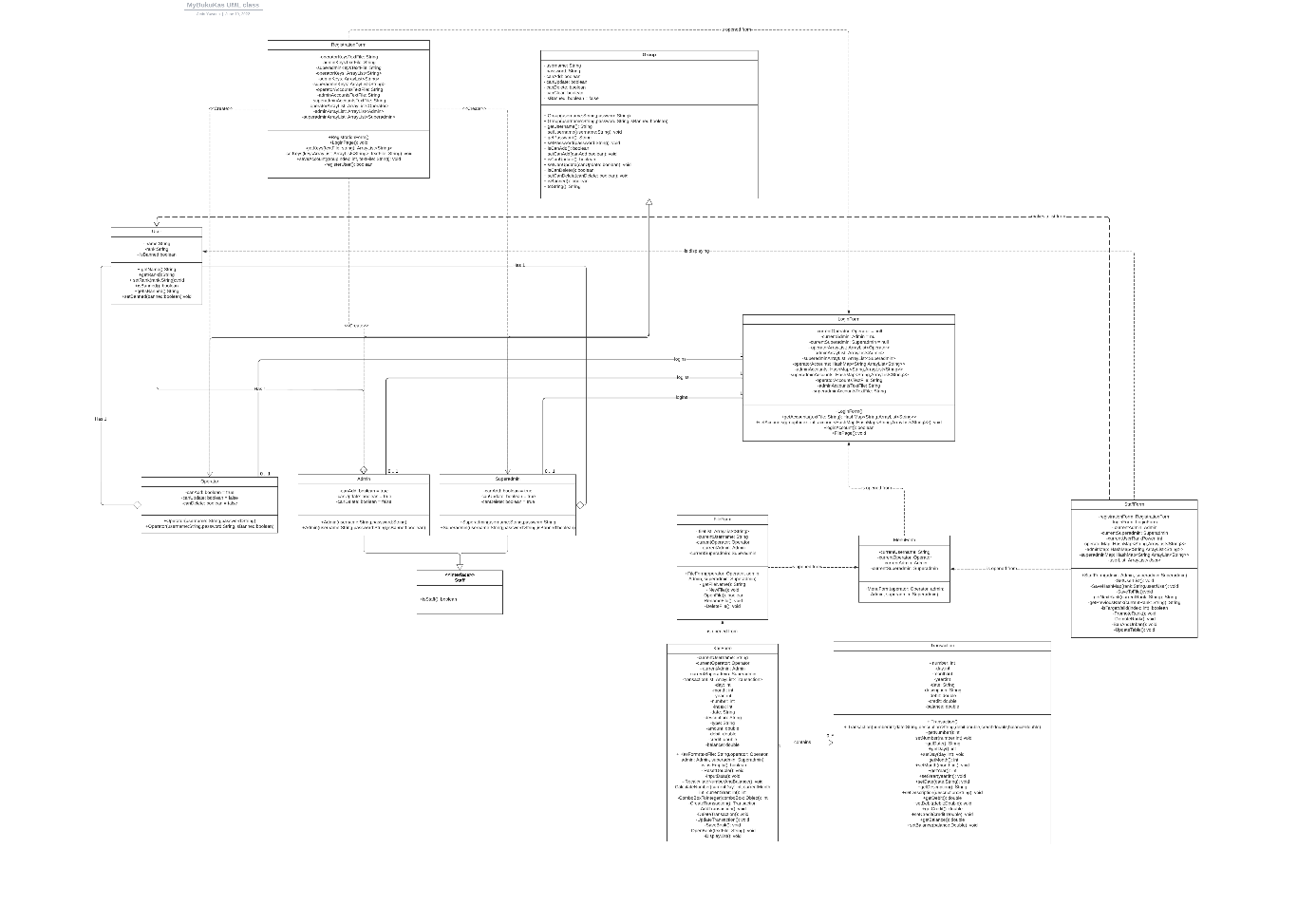


Image 1.1 UML Class Diagram

**Solution Design**

Visualization of all the forms below is made possible thanks to Swing GUI. Data storing and reading is done using Text Files.

Transaction Class

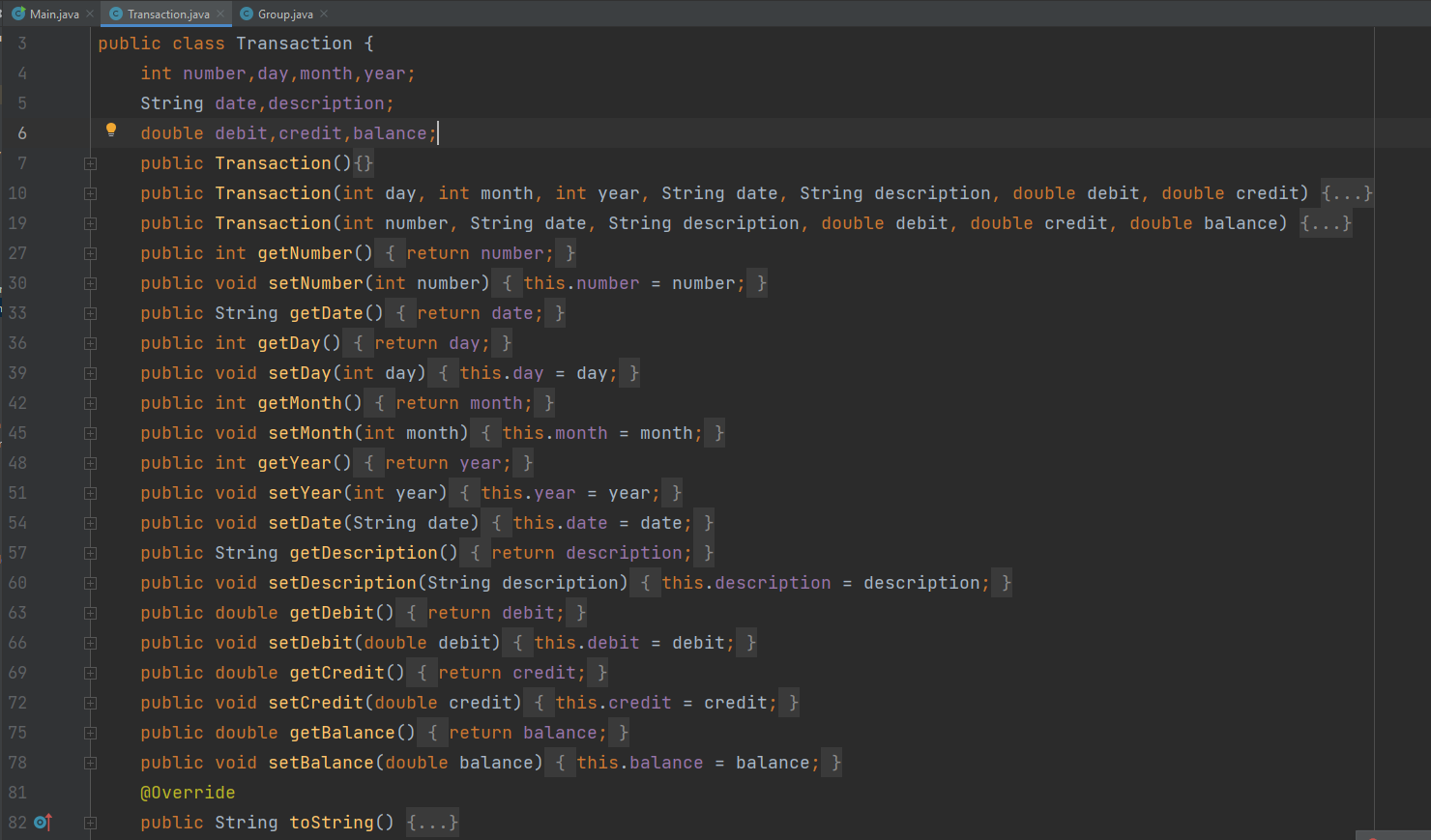


Image 2.1 Transaction Class

This class contains data types which can store details of a transaction such as the number, date, description, debit, credit, and balance. A Constructor with and without parameter is available to be used. Getter and Setter methods are also there to access the private datas. A to String method is given for testing on console.

Group Class

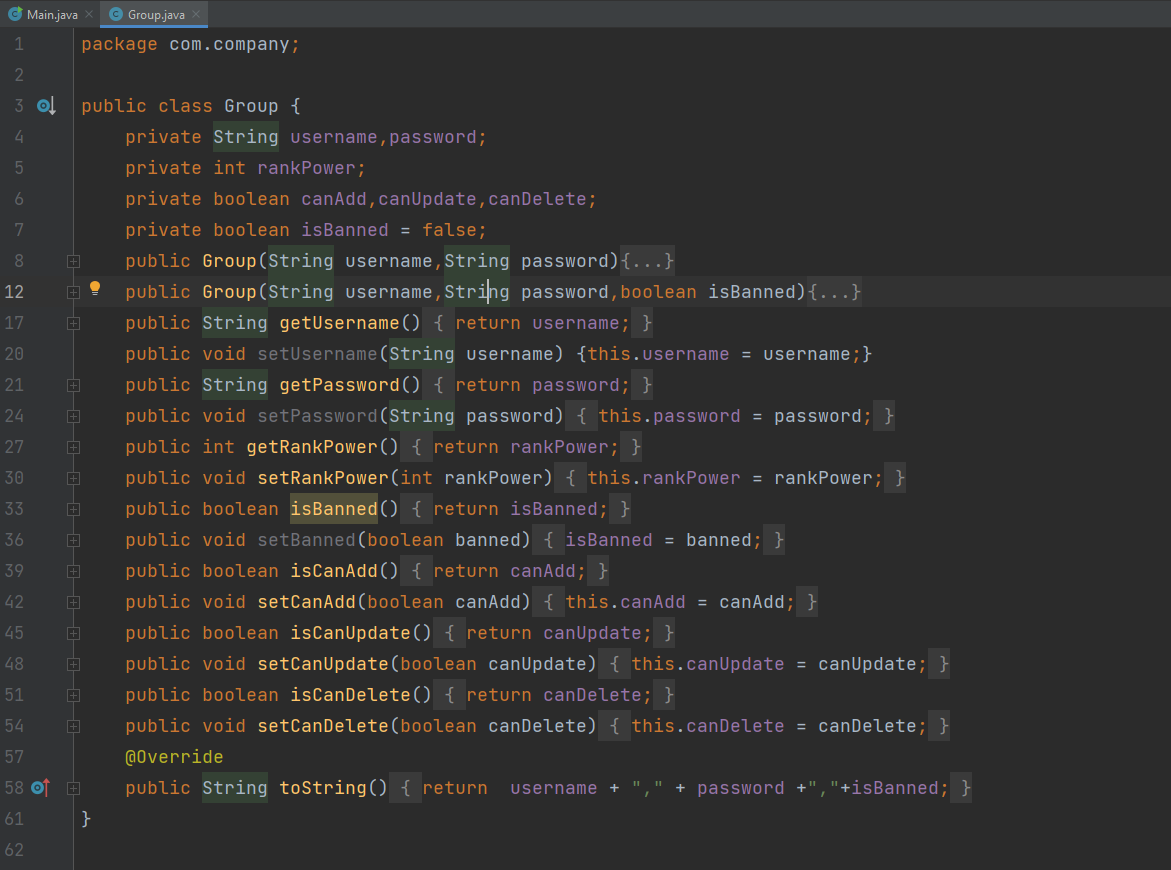


Image 2.2 Group Class

Act as the parent of actual user groups. It holds data type for the name, password, rank, right of accesses, and ban status.

Superadmin Subclass

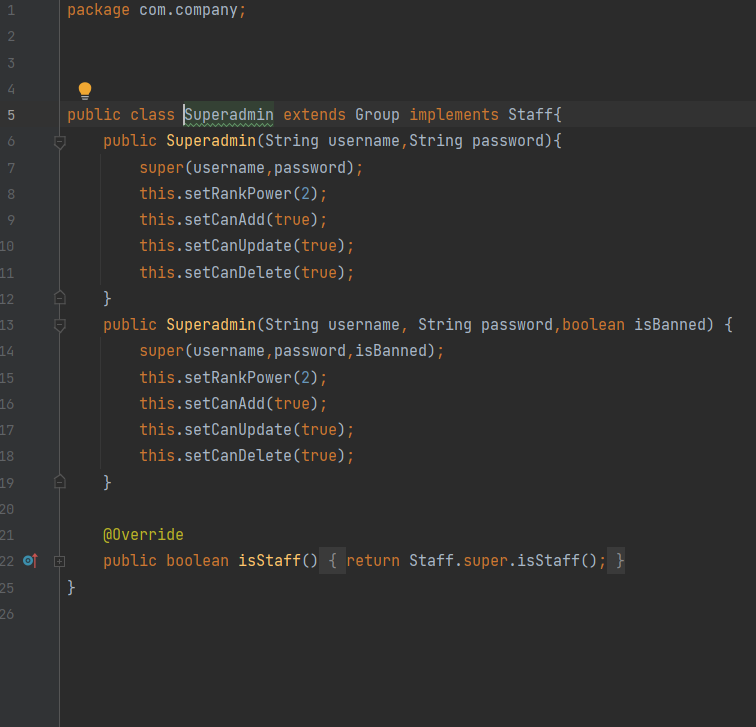


Image 2.3 Superadmin Subclass

The group’s child with the highest authority of access rights, which means it can access every form and button, which means this class can access the Staff Panel as well.

Admin Subclass

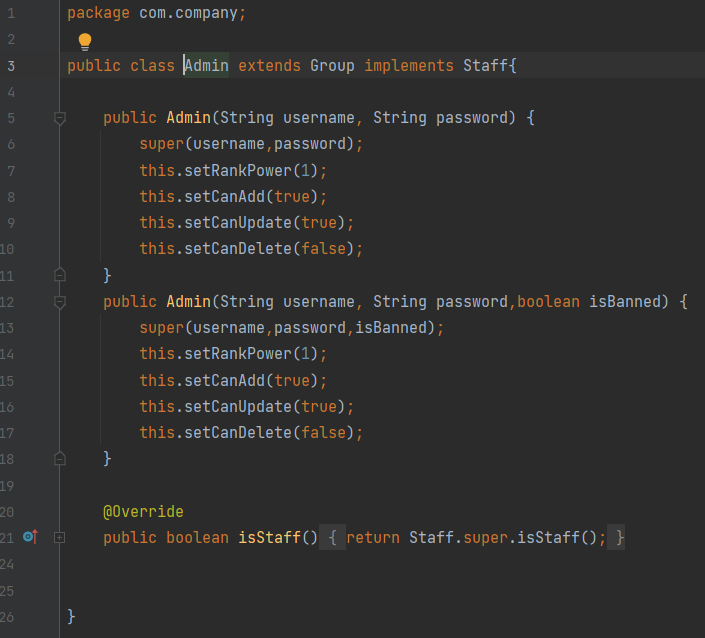


Image 2.4 Admin Subclass

The Group’s child with a medium authority of access rights, with the only restricted access of being unable to delete or clear items. Extra privileges of this class above Operator are updating transactions and accessing the Staff Panel, particularly for banning and unbanning.

Operator Subclass

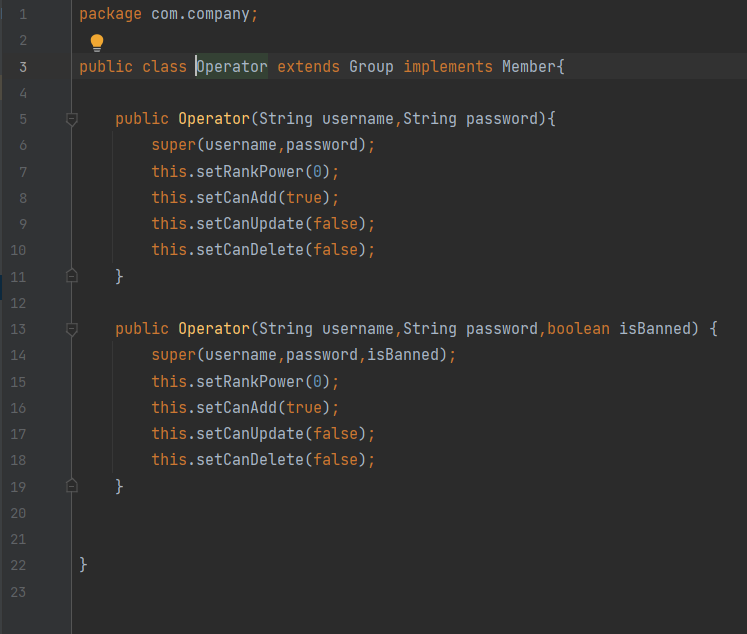


Image 2.5 Operator Subclass

Group’s child with the lowest authority of access rights. This subclass is only able to create new files and transactions.

User Class

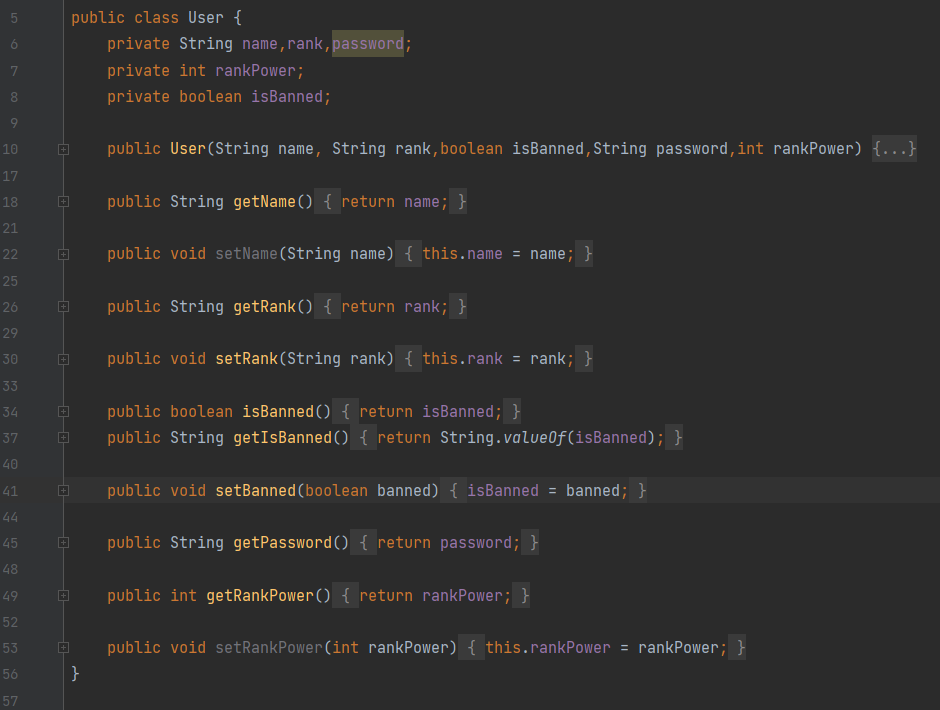


Image 2.6 User Class

Used to help group all separate user groups into a new class to be able to display all available accounts on a single table.

Staff Interface

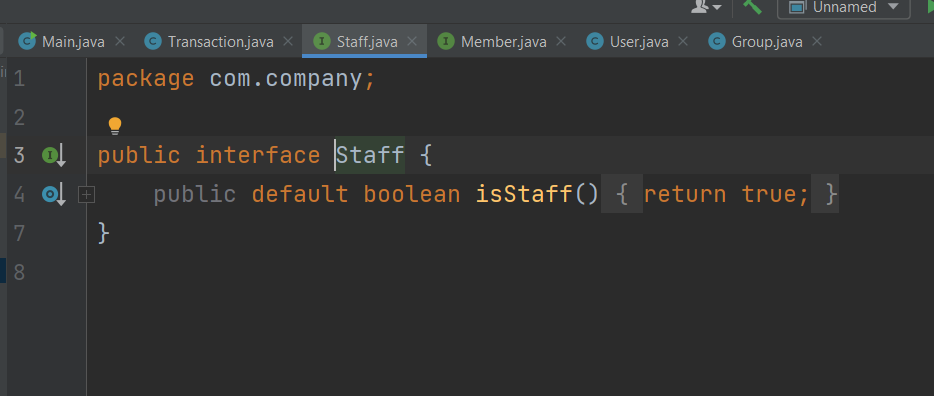


Image 2.7 Staff Interface

This interface contains a default Boolean method, which returns true. Used by subclasses such as Admin and Superadmin to recognize it is considered a staff and should be able to access the staff panel.

Member Interface

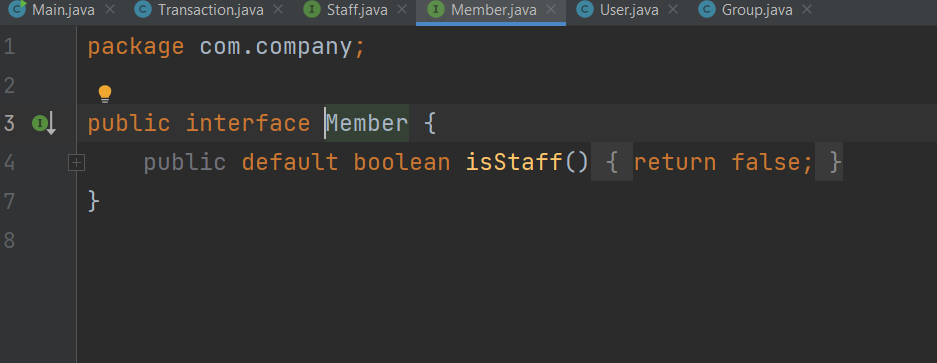


Image 2.8 Member Interface

This interface contains a default Boolean method, which returns false. Used by subclasses such as Operator to recognize it is a member and not a Staff.

Main Form

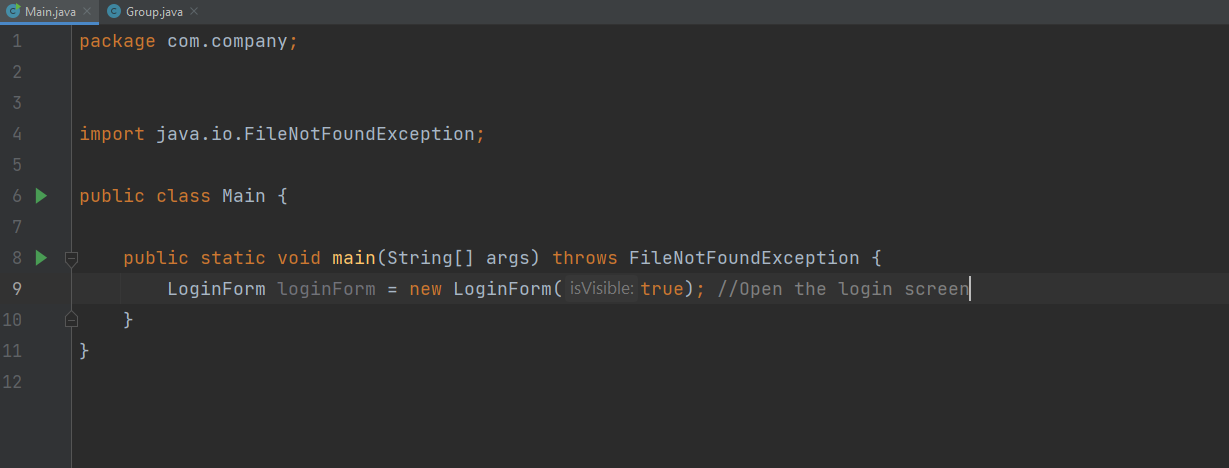


Image 2.9 Main Screen

This form contains the main function which opens the Login Form (Screen) upon execution of the application.

Registration Form

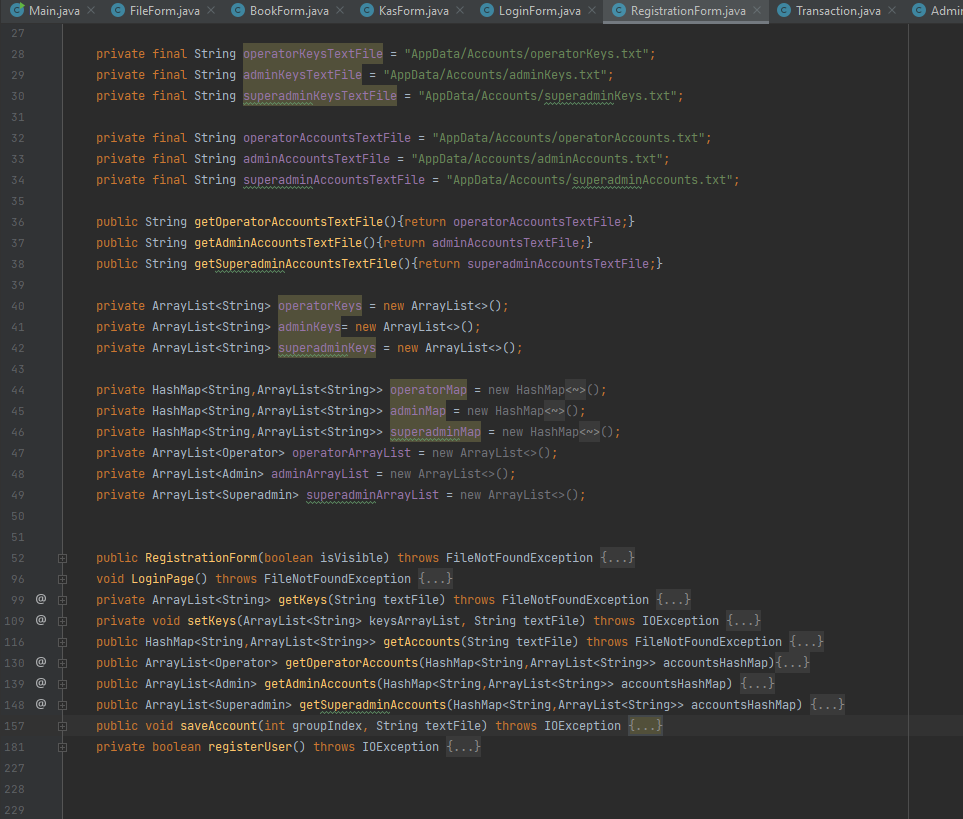


Image 2.10 Registration Screen

File and Scanner are used to read text files that contain serial keys for either Operator, Admin or Superadmin, which are then added into their respective ArrayList<String>. File Writers are then used to store username and password into respective text file based on the serial key input. The user can be returned to the Login Form by either pressing the ‘LOGIN’ button or submitting the registration.

Login Form

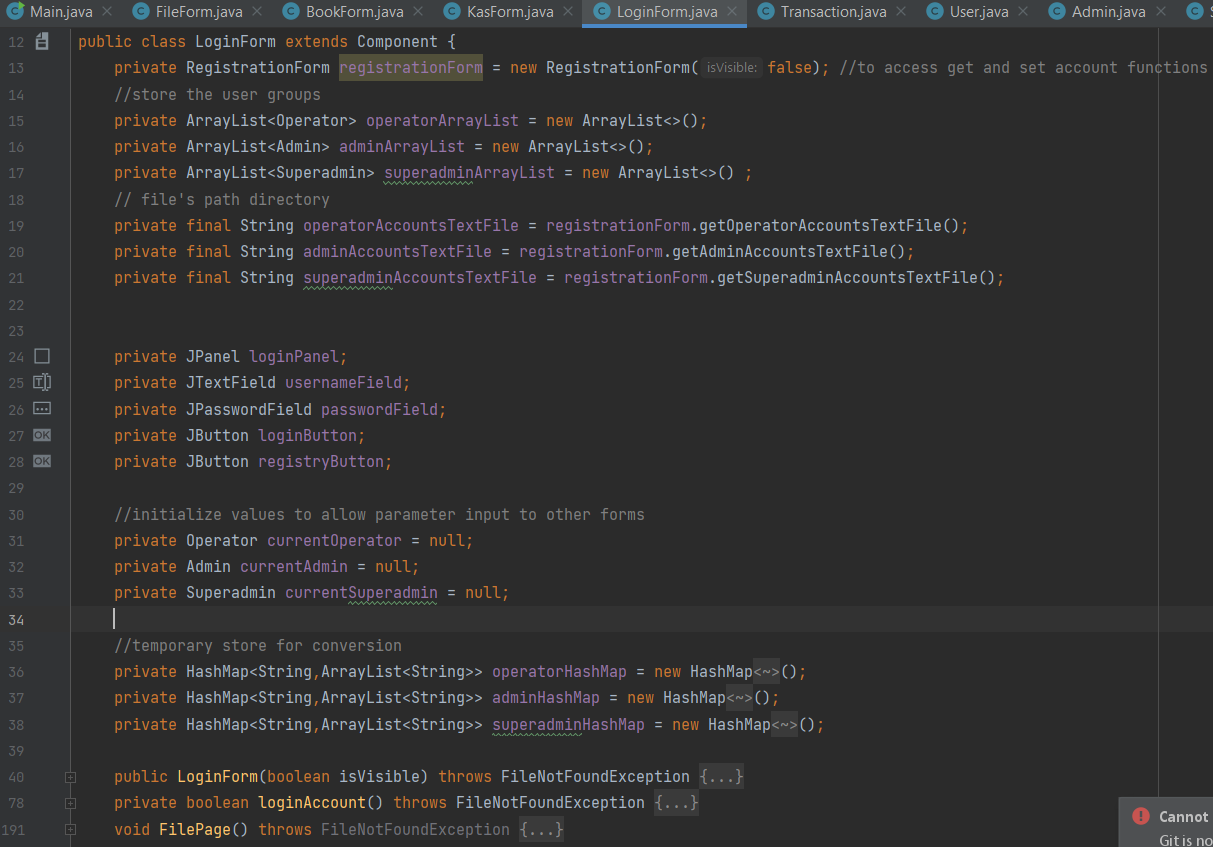


Image 2.11 Login Screen

File and Scanner are used to read text files that contain account’s username and password, which are then put into a HashMap<String, String>. An object of type Operator, Admin or Superadmin will then be created and added into their respective ArrayList. A valid login detail will allow the user to be brought to the File Form. If no account exists, the user can press the ‘REGISTRY’ button to be brought to the Registration Form.

File Form

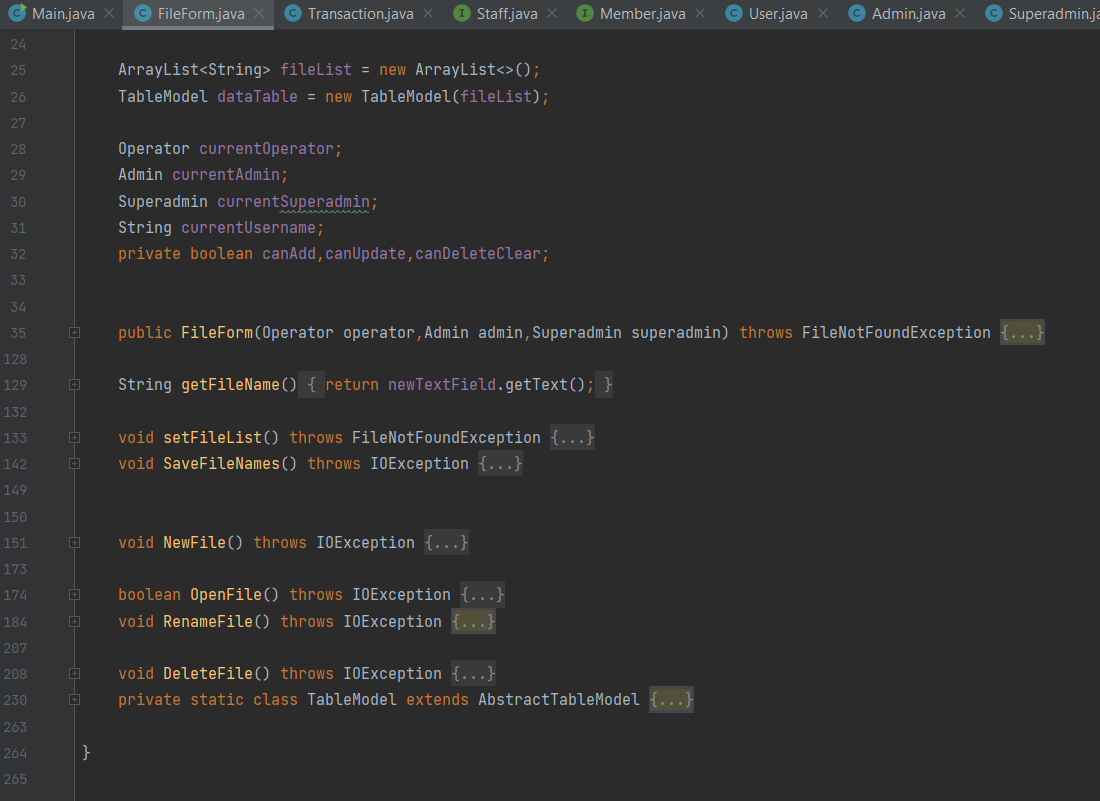


Image 2.12 File Selection Screen

Inside a Scroll Panel and Table, based off an ArrayList, all the books available to be opened will be displayed. The user can create a new book by inputting the desired name in a Text Field and pressing the ‘New’ button. To open an existing file, select a row in the table and press the ‘Open’ button, which will bring the user to the Kas Form. The first course of action is the same as opening for file renaming and deletion, except the button to be pressed differs.

Book Form

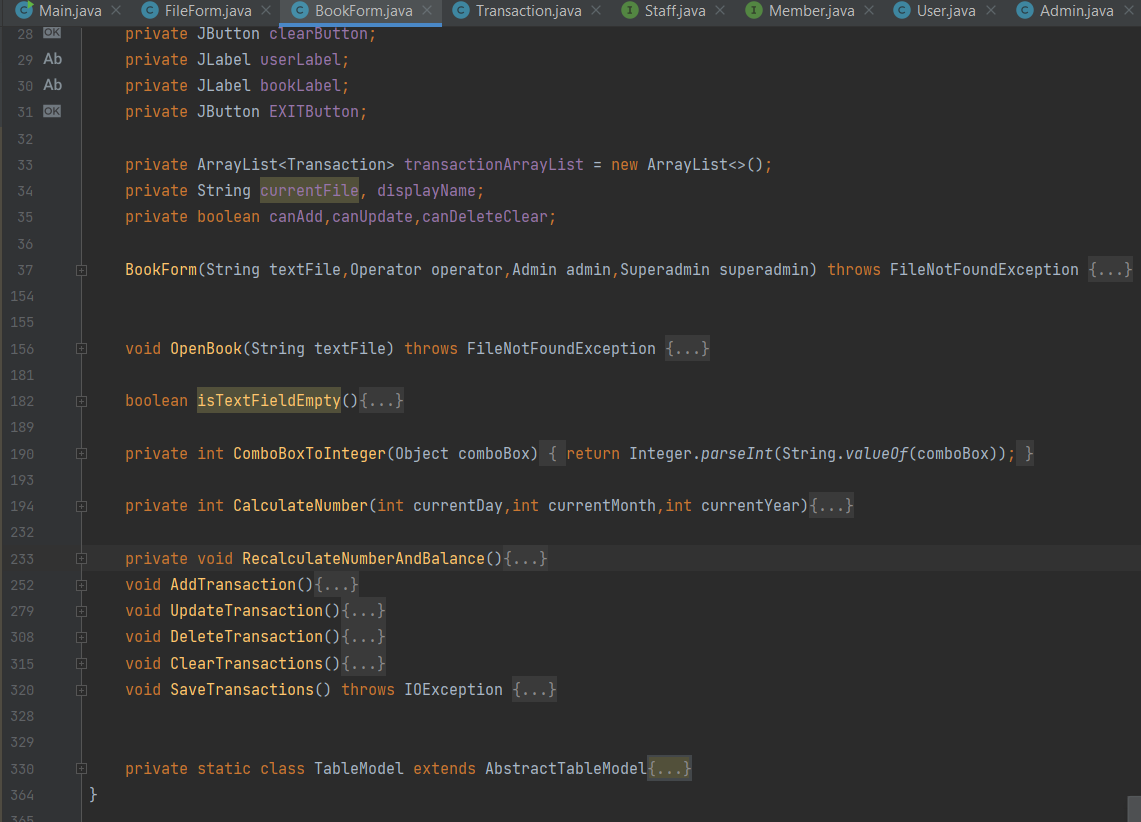


Image 2.13 *Buku Kas* Screen

The list of buttons (Add, Update, Delete, and Clear) available will depend on the current user’s rank. The date can be adjusted by using the available Combo Box, and the same goes for the type of transaction: Debit or Credit. As for money and description, they can be written in the Text Fields. When a button is pressed, it will convert the Combo Boxes’ selected item and Text Fields’ Text into the appropriate data types.

When a new transaction is to be added, it will check whether the ArrayList<Transaction> is empty first. If it is not empty, it will first calculate the appropriate number and index for it to be inserted at. Else If it is empty, it will immediately add the transaction to the ArrayList. After either course of flow, it will recalculate every transaction’s appropriate number and balance before updating the Table.

The difference between updating and deleting a transaction is that updating removes the selected existing transaction and adds a new one, while deleting a transaction only removes the selected transaction.

Staff Form

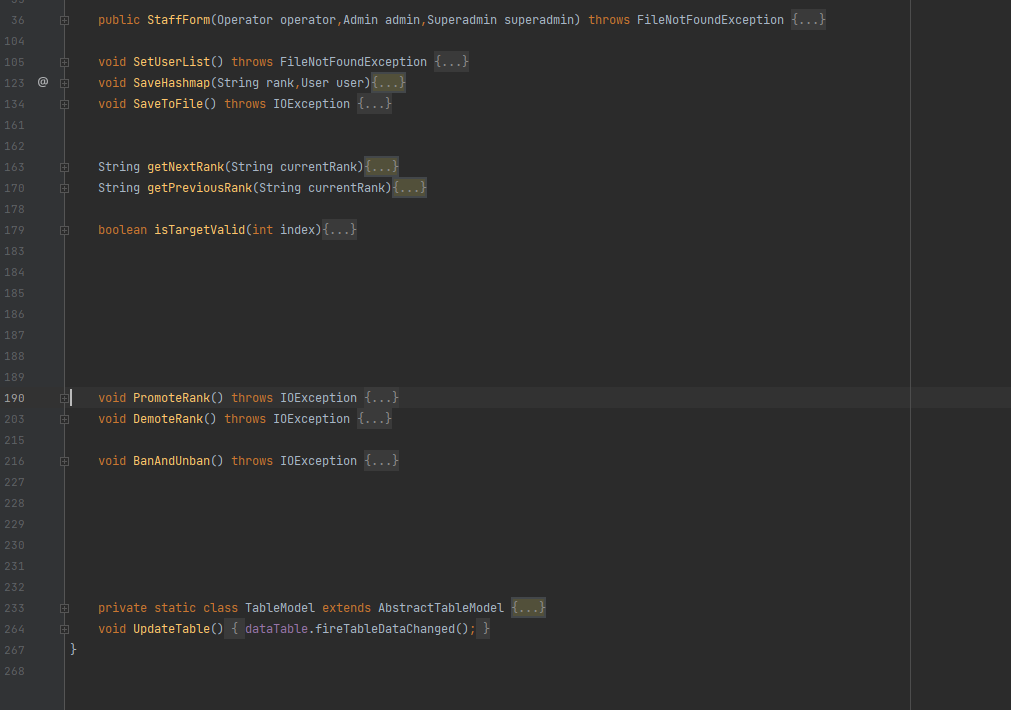


Image 2.14 Staff Panel Screen

This Form is exclusively visible to users whose rank is part of Staff (Admin and Superadmin). All users regardless of ranks will be displayed in a Table, based off ArrayList<User>, which data is obtained from the same HashMap method Login Form used and iterated using for each loop to be added into the ArrayList.

After selecting a user’s row, there are several choices to be made: Promote, Demote, Ban and Unban. Promotion and Demotion can only occur if the target’s next rank or previous rank is lower than the user’s rank. Ban and unban can be done when the target’s rank is below the user’s rank. It does not remove the account, but simply toggle a Boolean value to make login impossible.

**Evidence of Working Program**

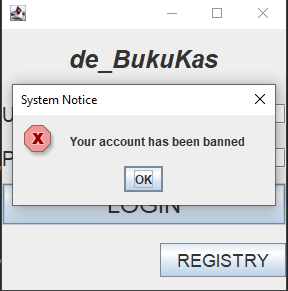


Image 3.1 Unable to login due to account being banned

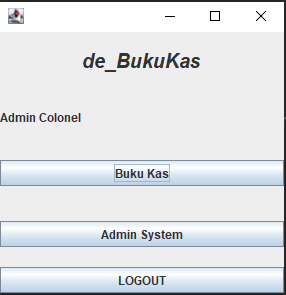


Image 3.2 Menu Screen after successful login

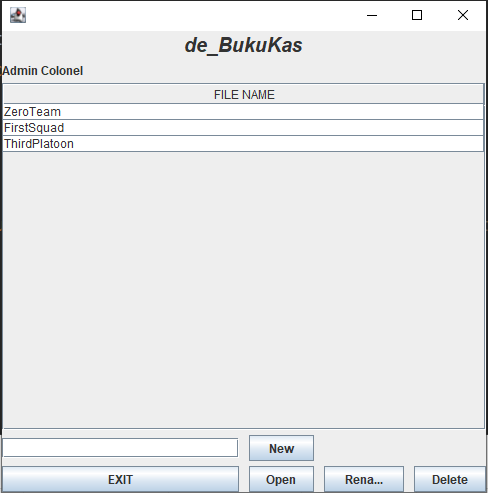


Image 3.3 File Selection Screen after choosing to go to *Buku Kas*

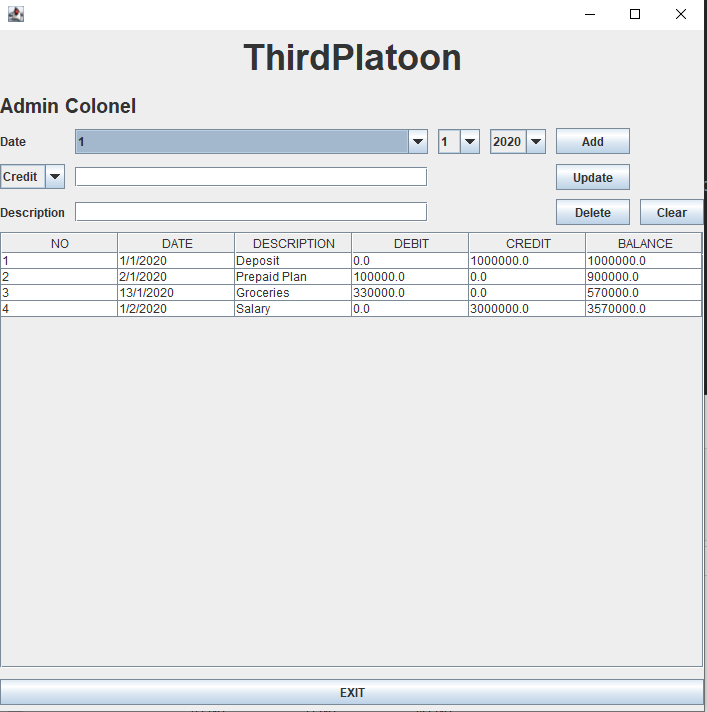


Image 3.4 A *Buku Kas* Screen

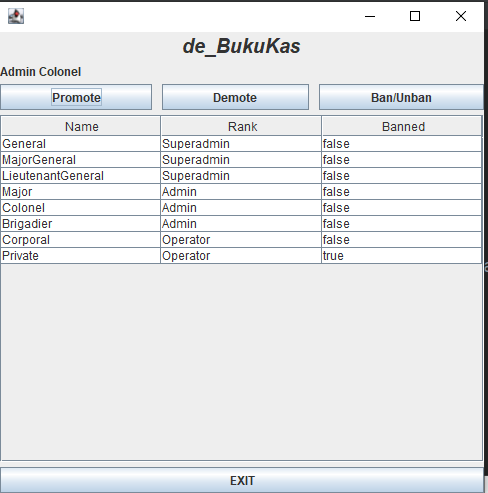


Image 3.5 The Staff Panel Screen

**Resources**

[Create Login and Registration Forms Using Java, MySQL and IntelliJ IDEA (With Source Code)](https://www.youtube.com/watch?v=gCdj9OPAeT8)

https://www.baeldung.com/java-write-to-file

<https://www.w3schools.com/java/java_hashmap.asp>

<https://stackoverflow.com/questions/10960213/how-can-i-read-comma-separated-values-from-a-text-file-in-java#:~:text=You%20may%20use%20the%20String,split(%22%2C%22)%3B>

[Java SWING #19 - How to Exit Program on Button Click in Java Netbeans](https://www.youtube.com/watch?v=ZD6626aW54M&list=LL&index=1)

https://www.geeksforgeeks.org/java-program-to-rename-a-file/