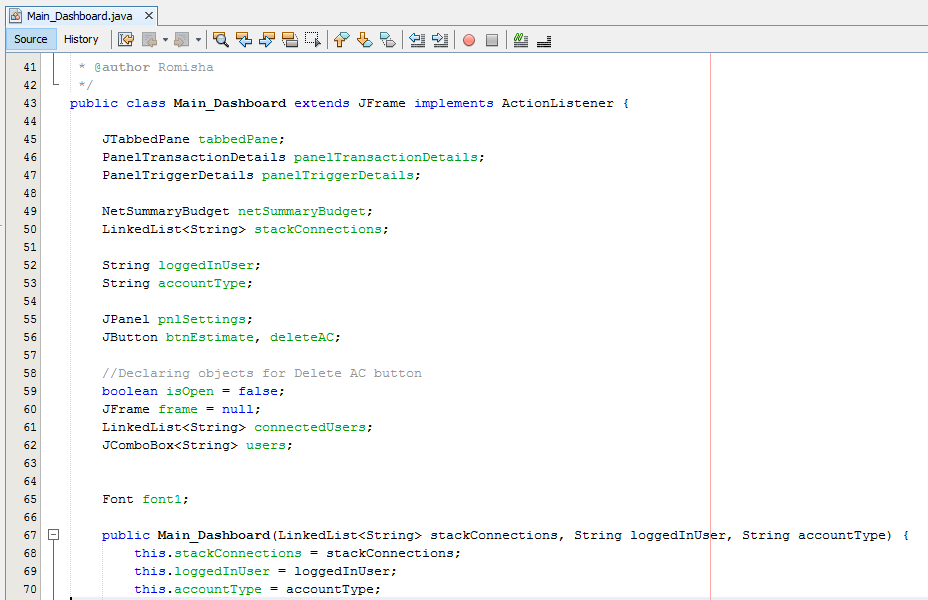
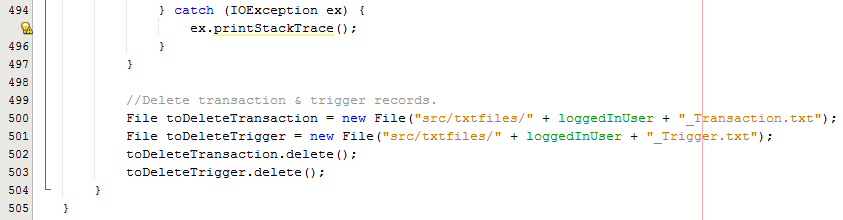
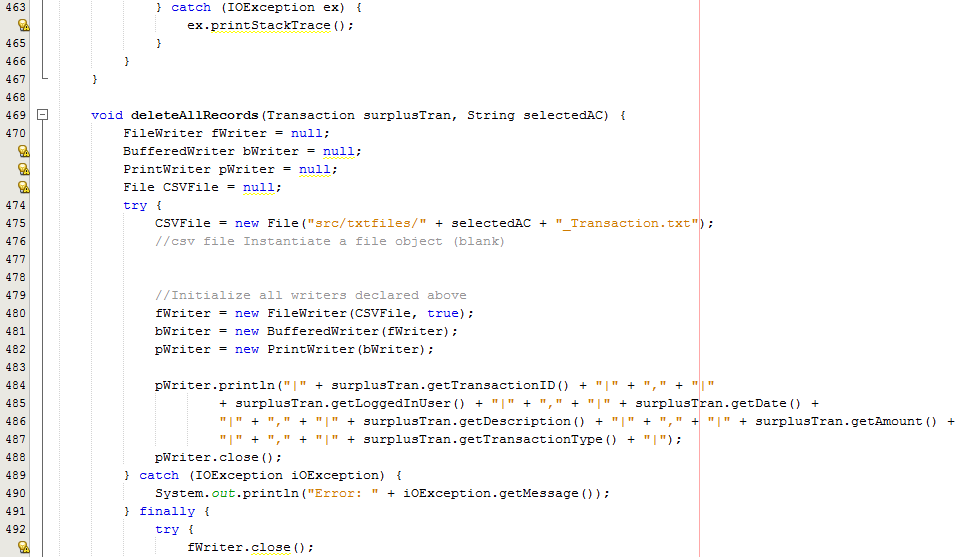
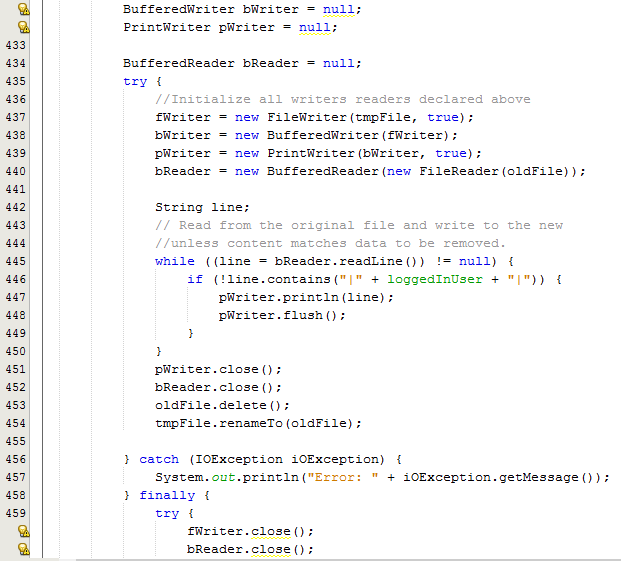
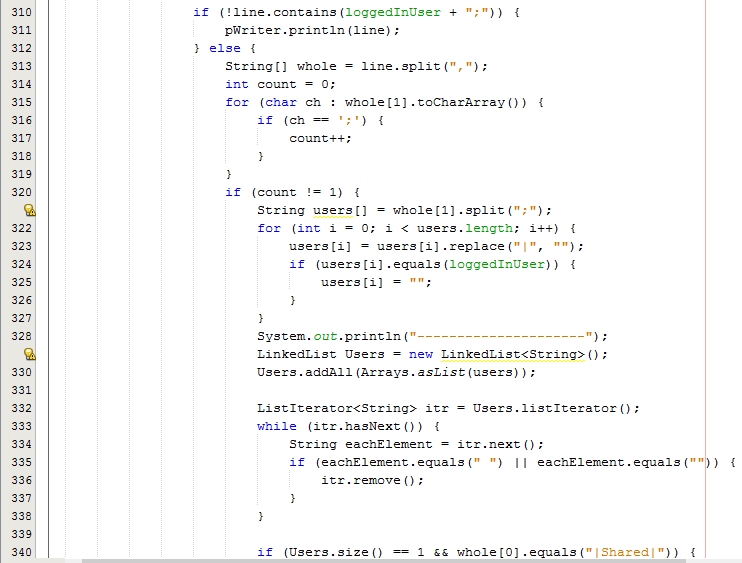
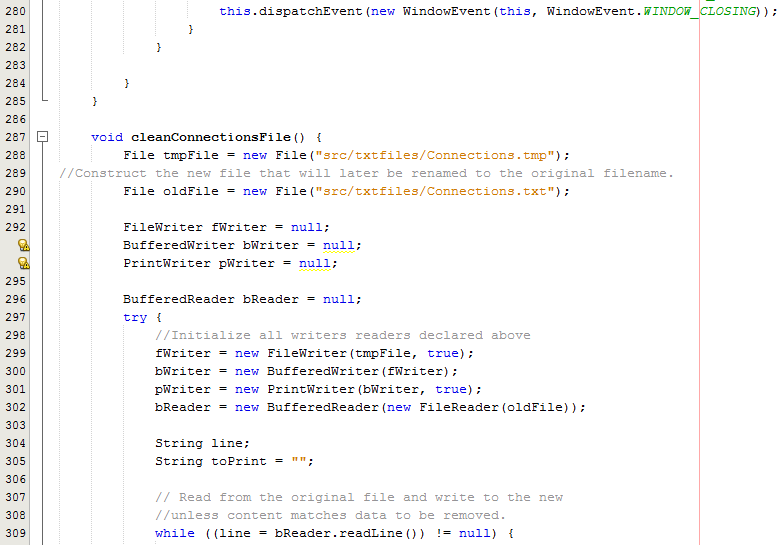
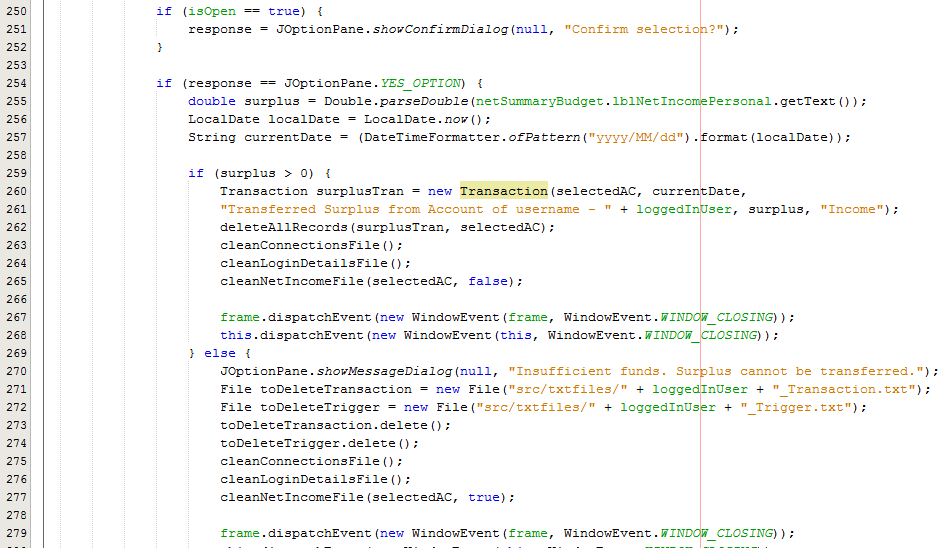
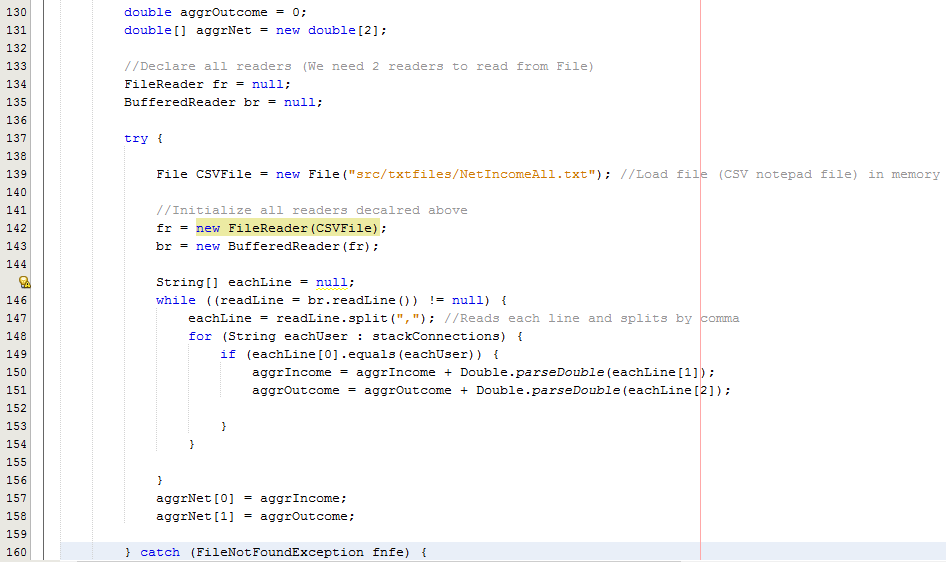
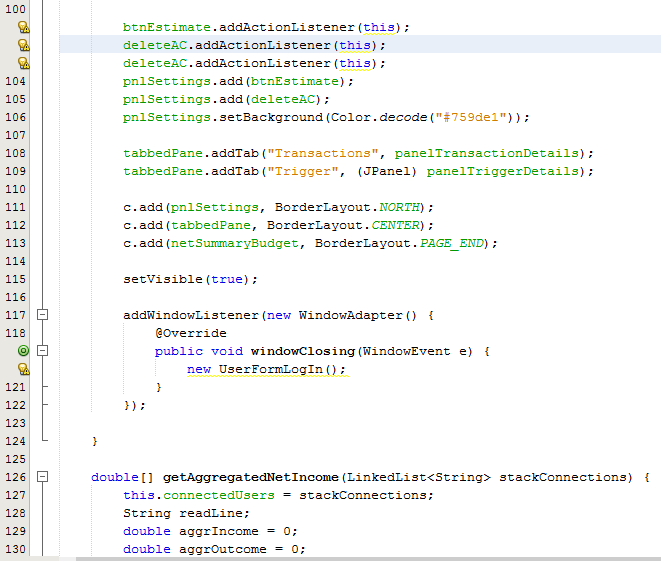
**Introduction**

Budget Manager Application is an application which is capable of recording transactions of users across various account types. Transaction Data relates to the income and expenditure of family or individual. Account types that can be created are – joint (shared between pairs), discretionary (single user) & home (multiple users with aggregated view). It has a two systems of recording data that are transaction and trigger. In transaction section users keep the input of expenditures and incomes where as in trigger section monthly ending balance is estimated. For the estimation of balance monthly income and expenditure is recorded from which monthly average balance is calculated and then helps to estimate remaining balance or losses across a period of 6 months. Both transaction and trigger contain a summery where total income and outcome is recorded and gives net income of home and personal users. In this system specific users with accounts discretionary, home and shared are allowed to create, edit and delete the data. If the users with several amounts wished to delete their user account it allows to allocate surplus money to an account according to the user’s choice. Which has simplified the act of keeping the record for home and individual users.

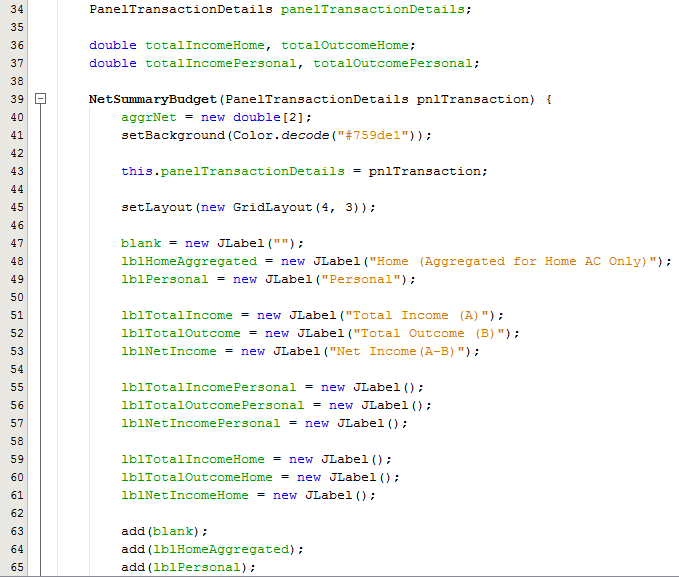
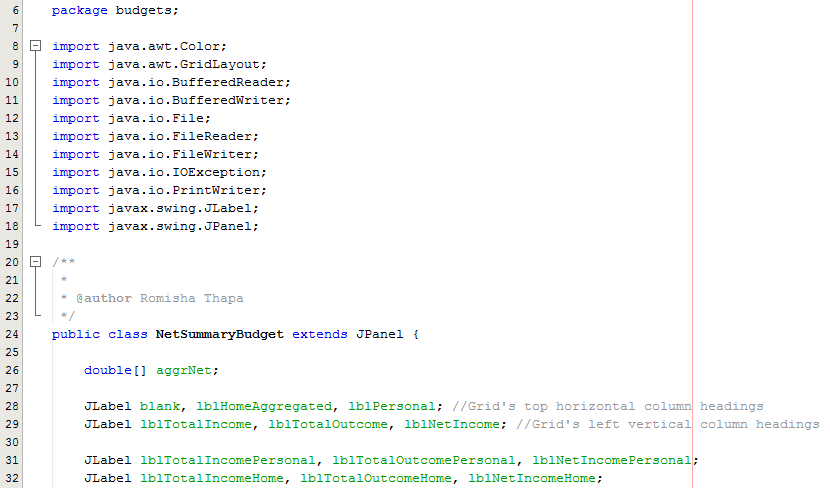
**1st Task: Source code**

**Main\_Dashboard**

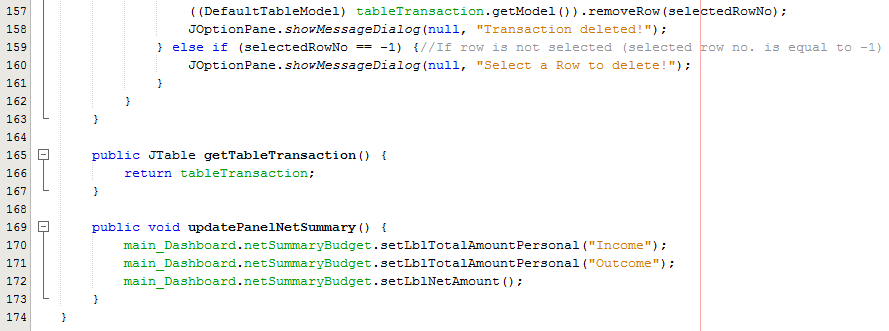
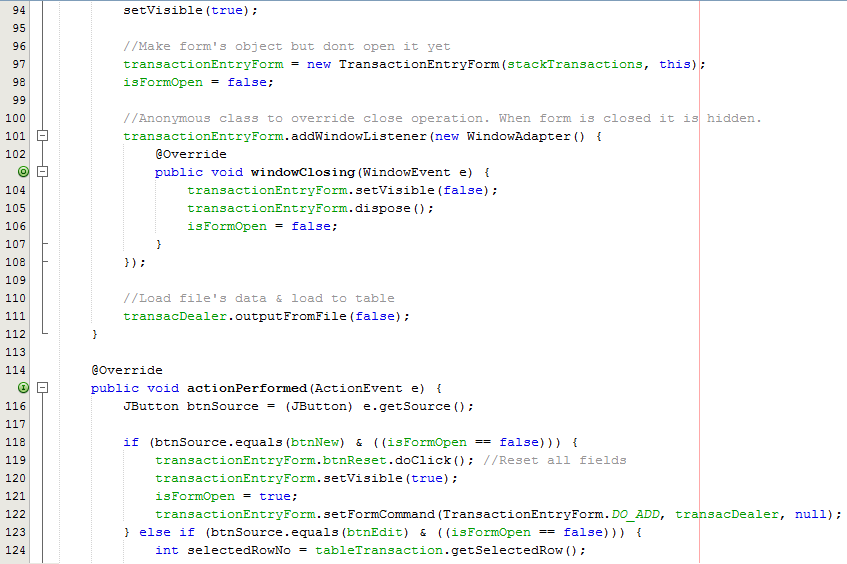
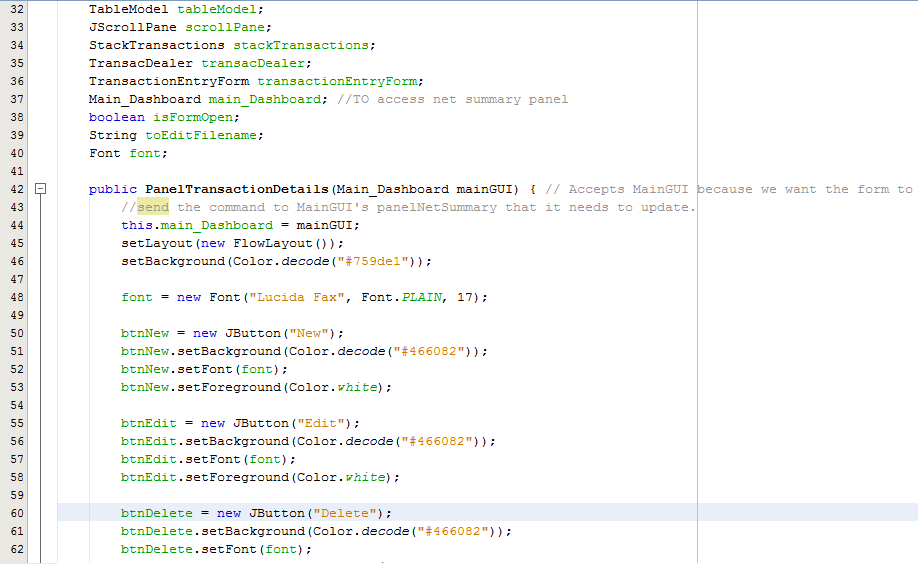
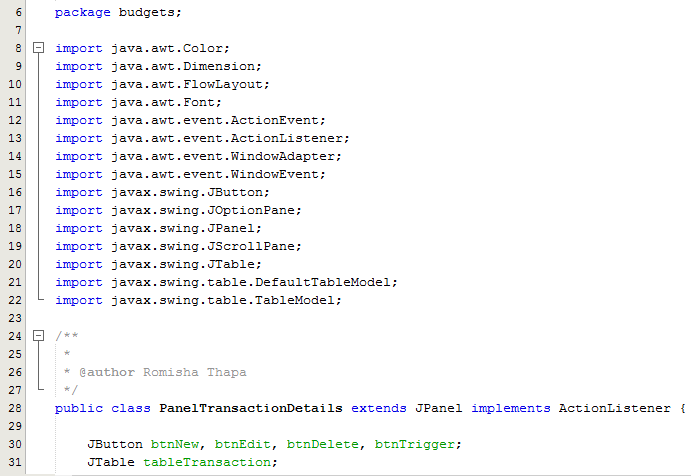
****



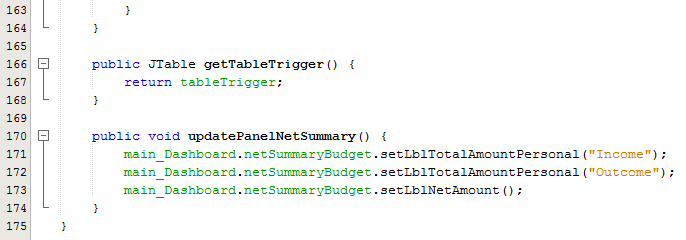
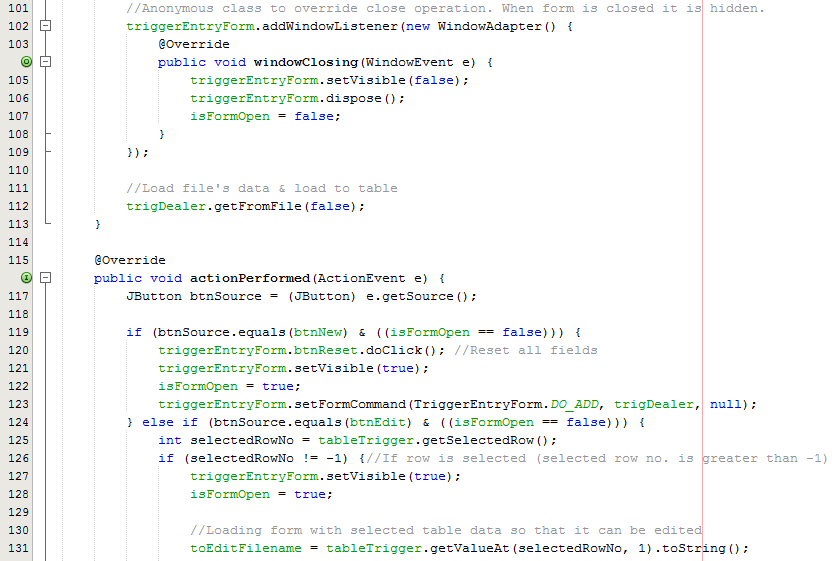
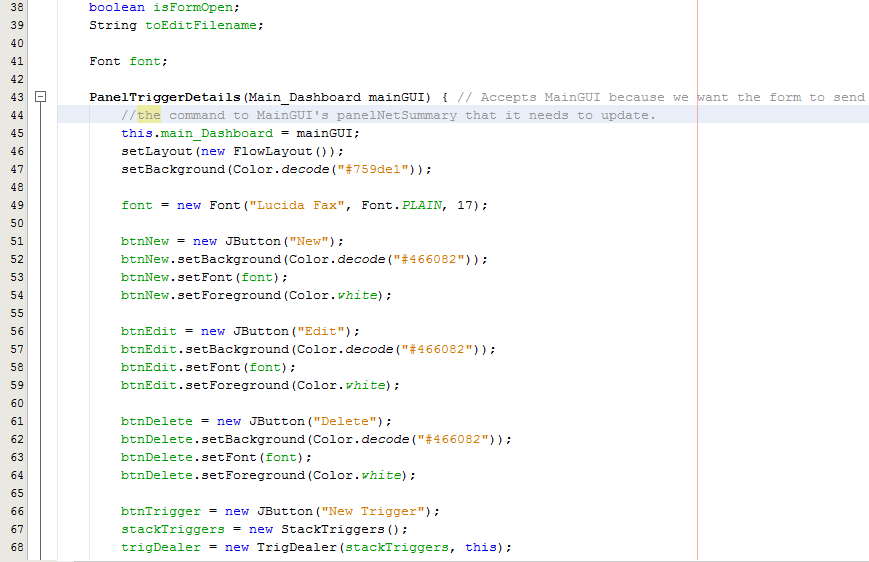
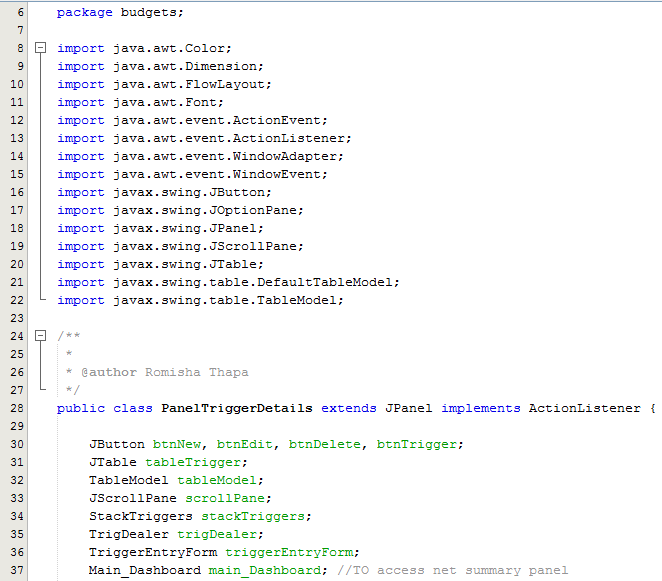
**NetSummeryBudget**



**PanelTransactionDetails**



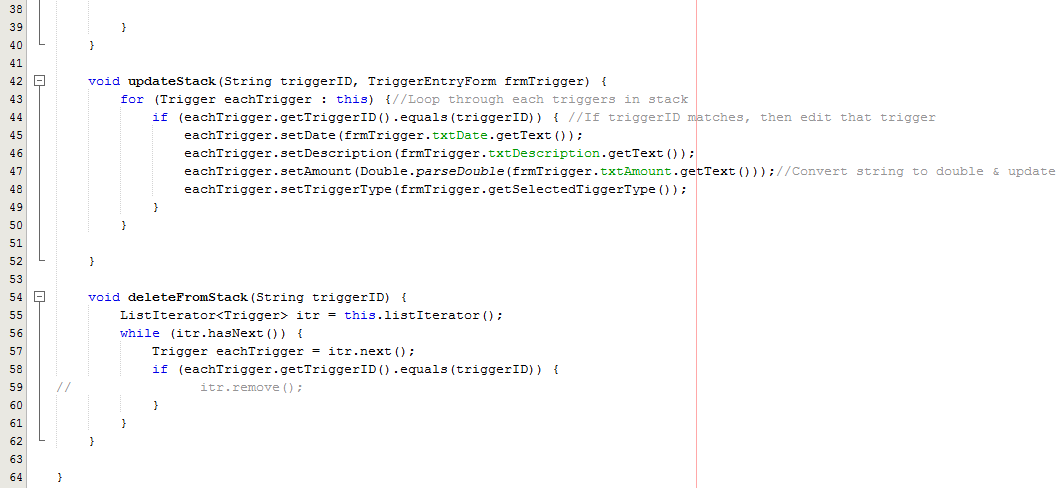
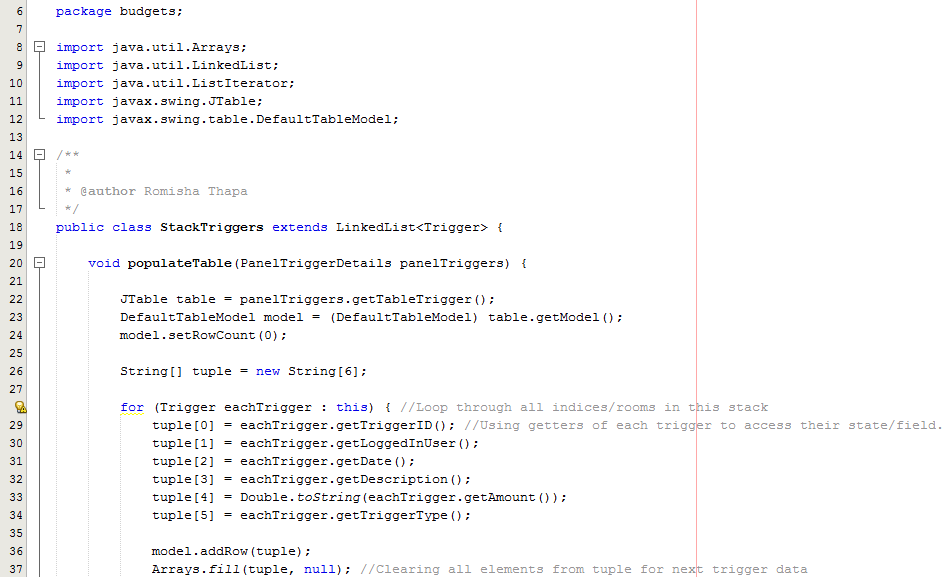
**PanelTriggerDetails:**



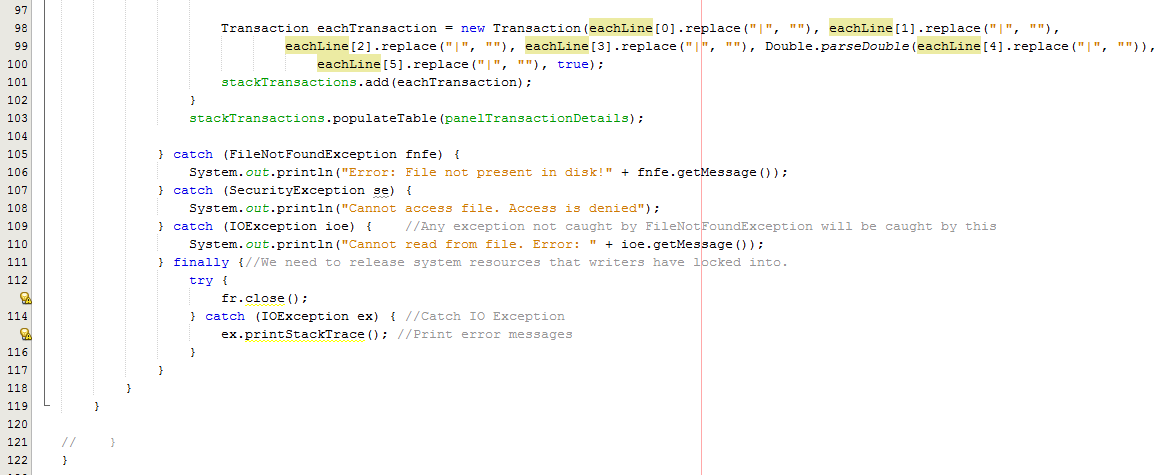
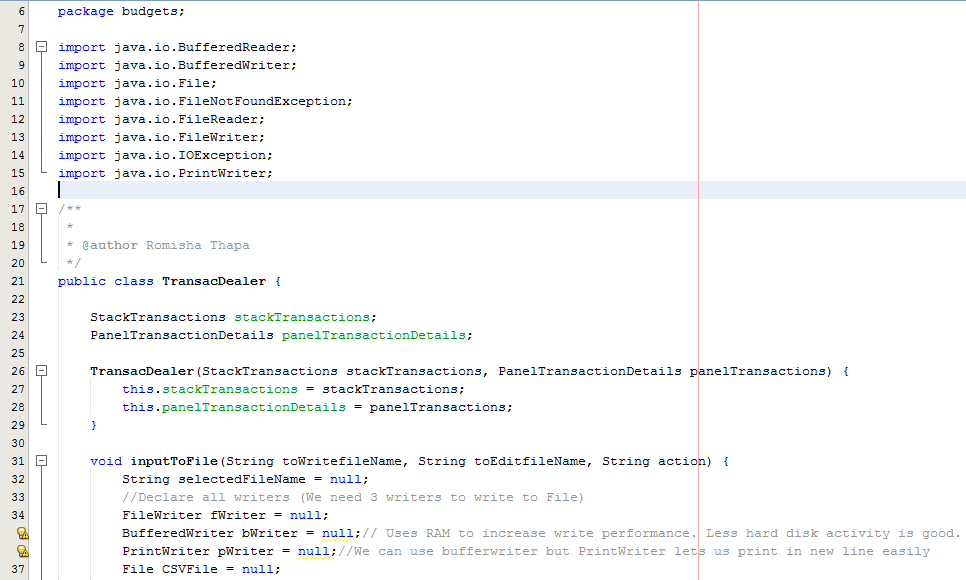
**StackTransaction**



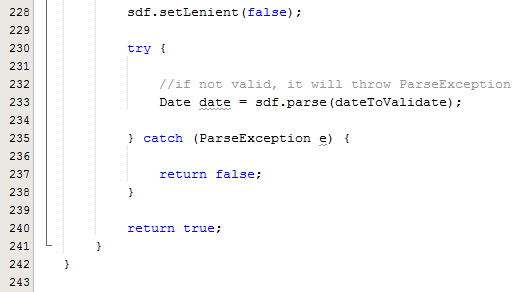
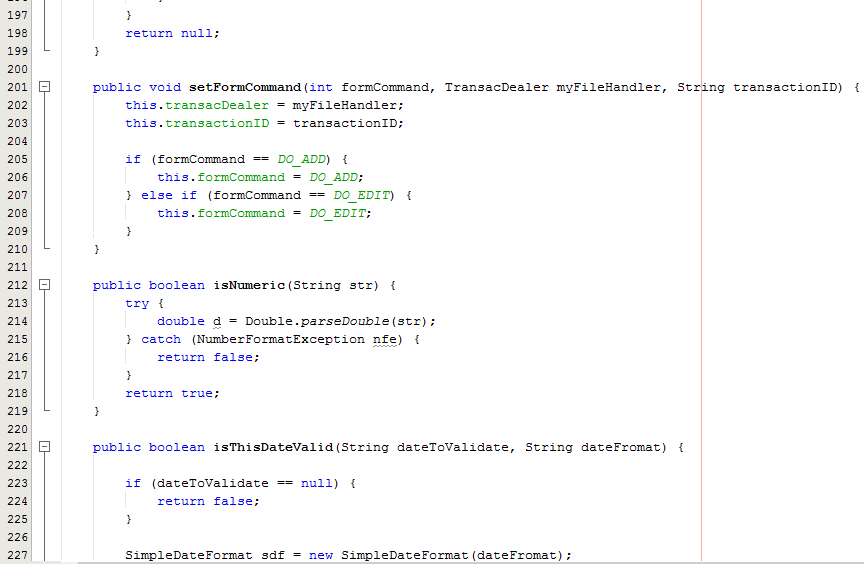
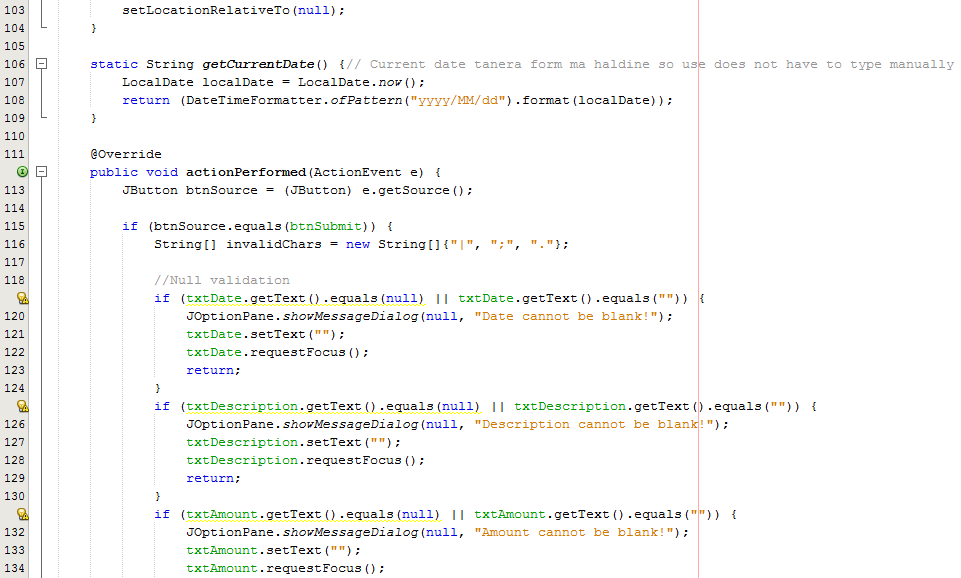
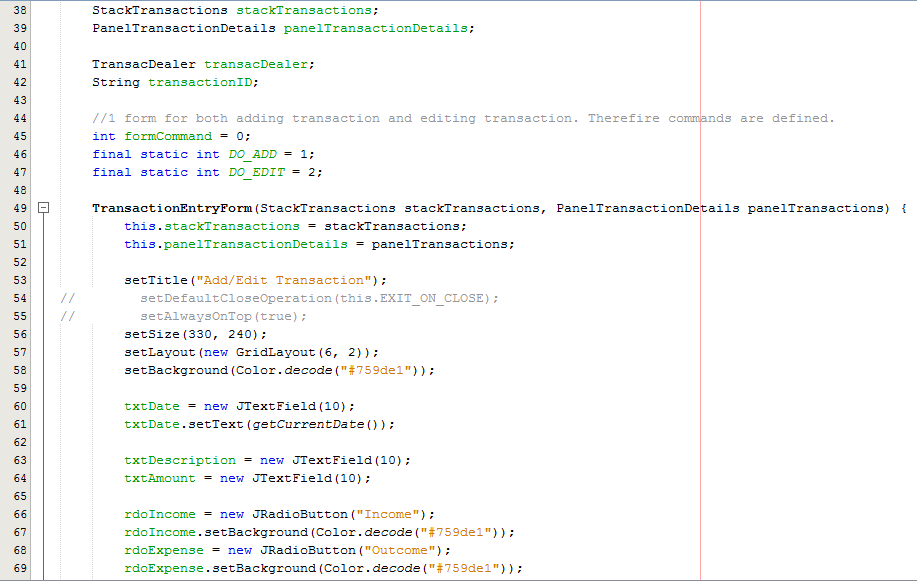
**StackTriggers**



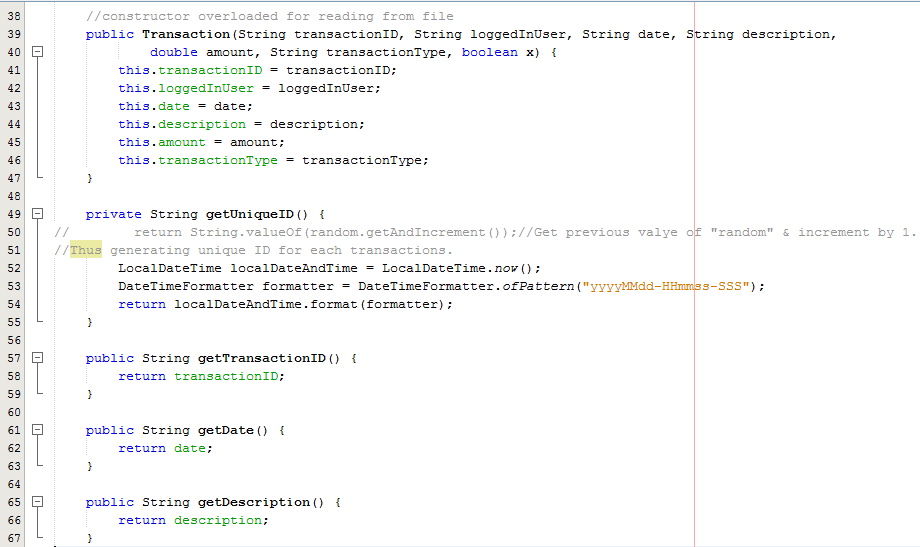
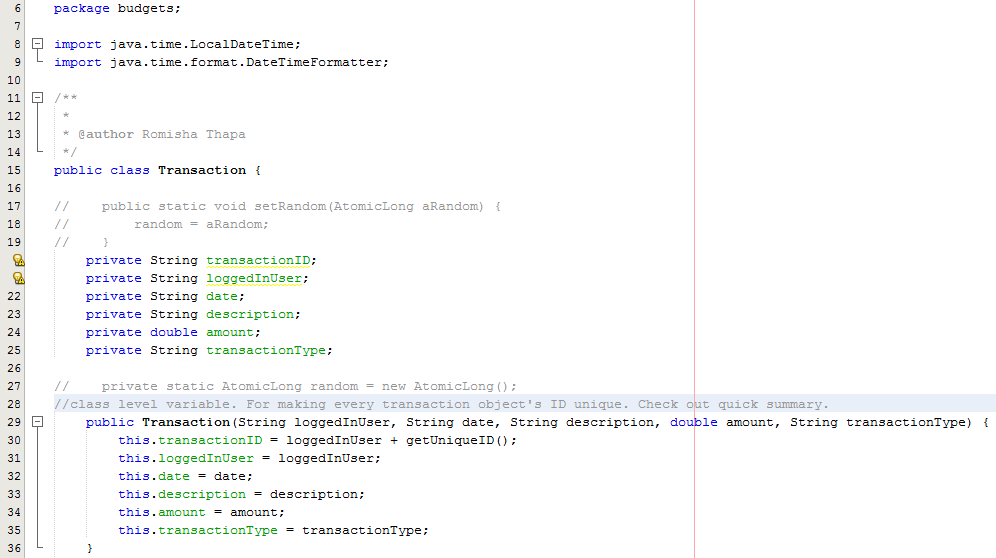
**TransacDealer**



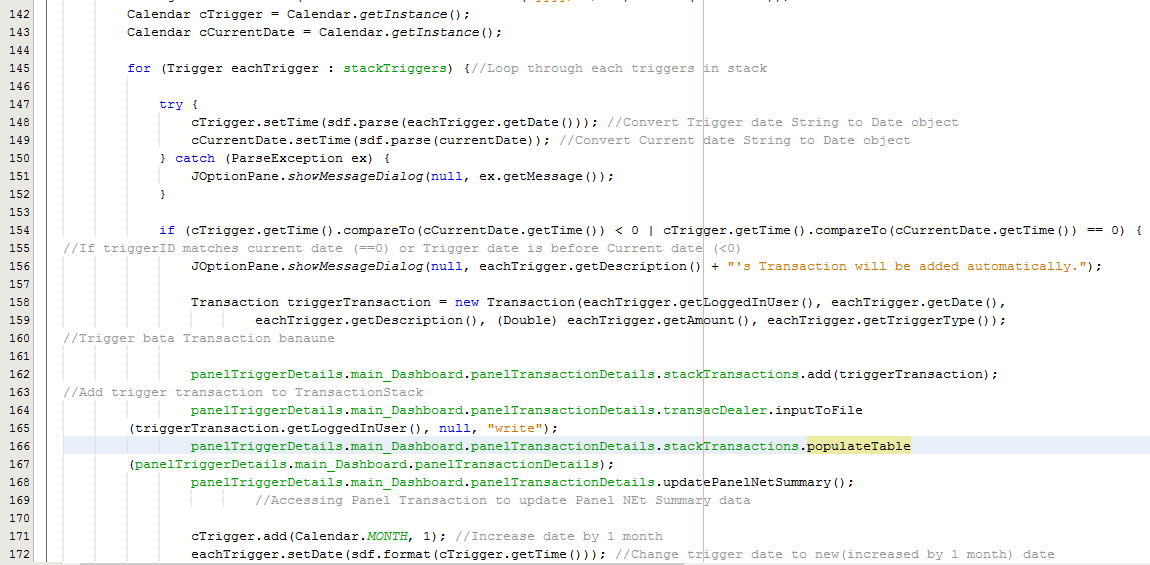
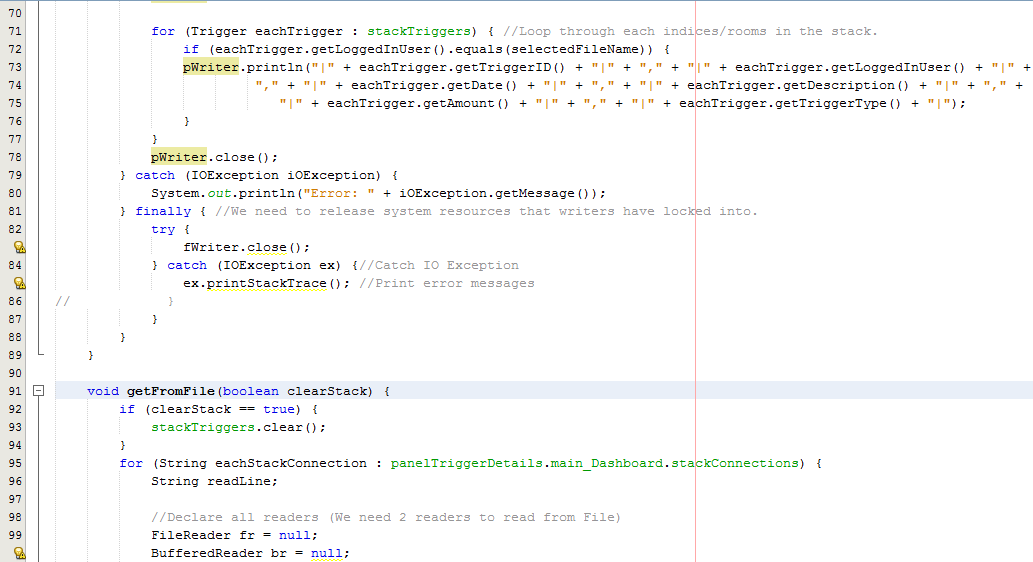
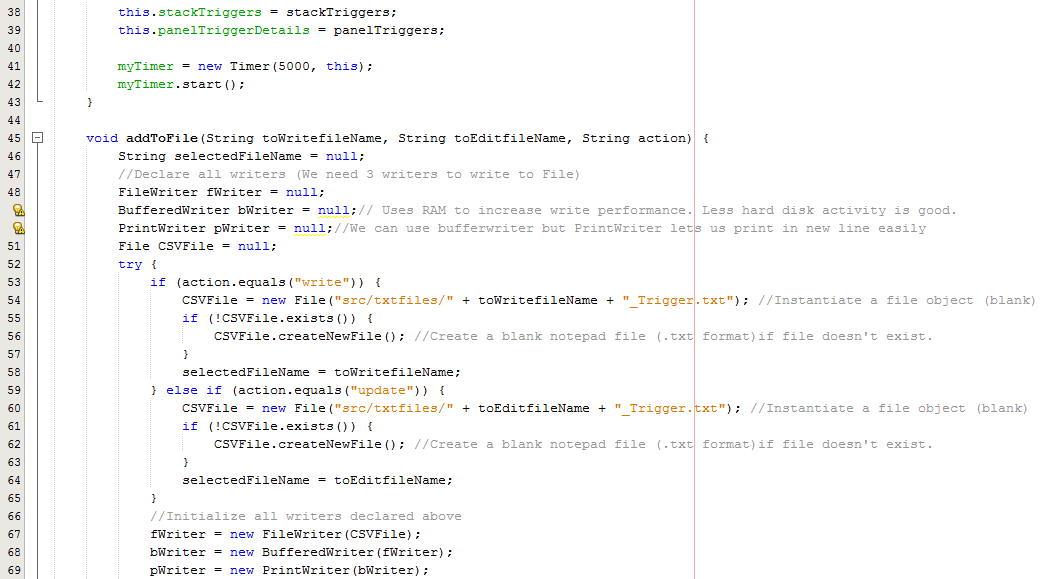
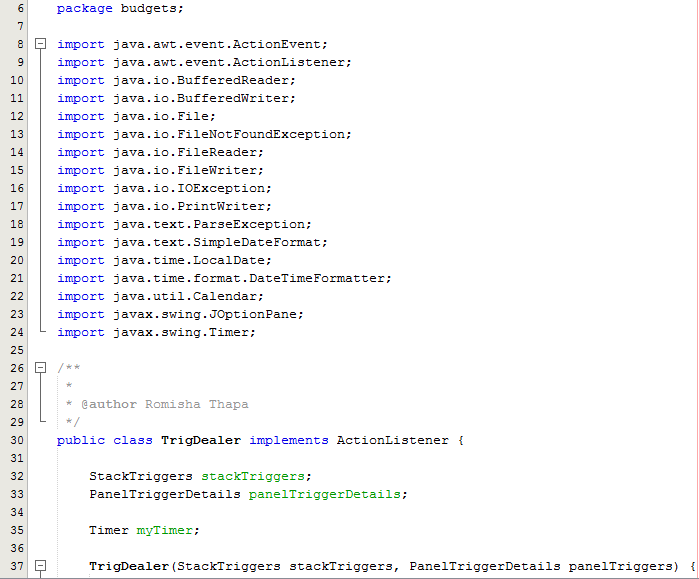
**TransactionEntryForm:**



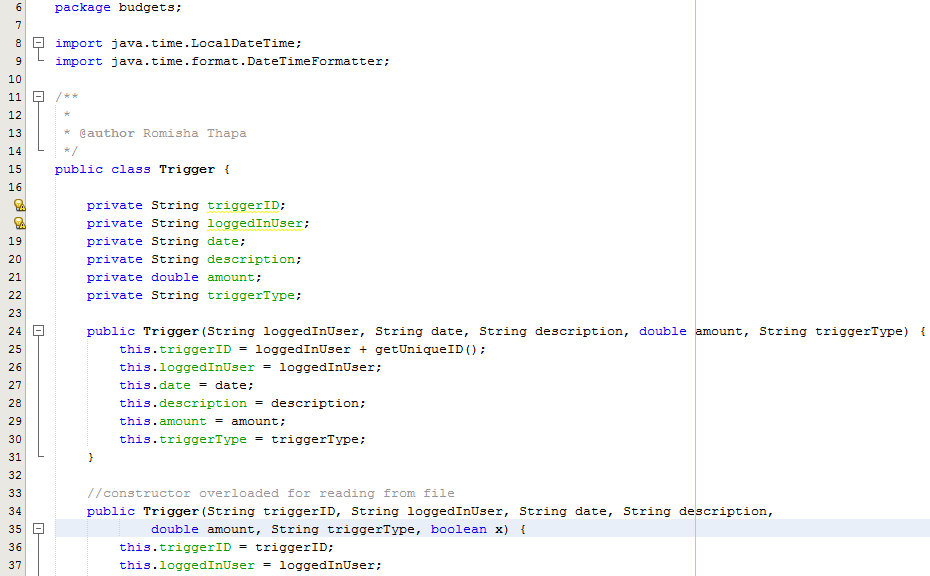
**Transaction**



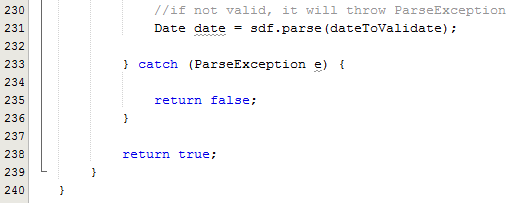
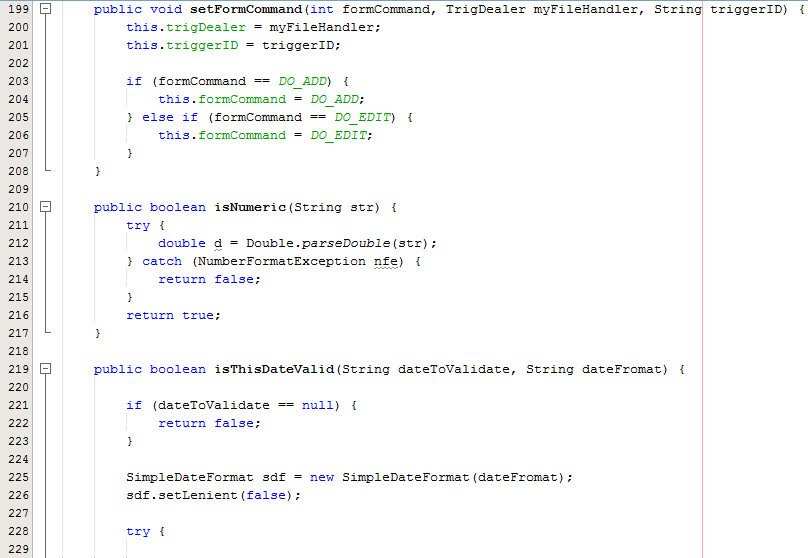
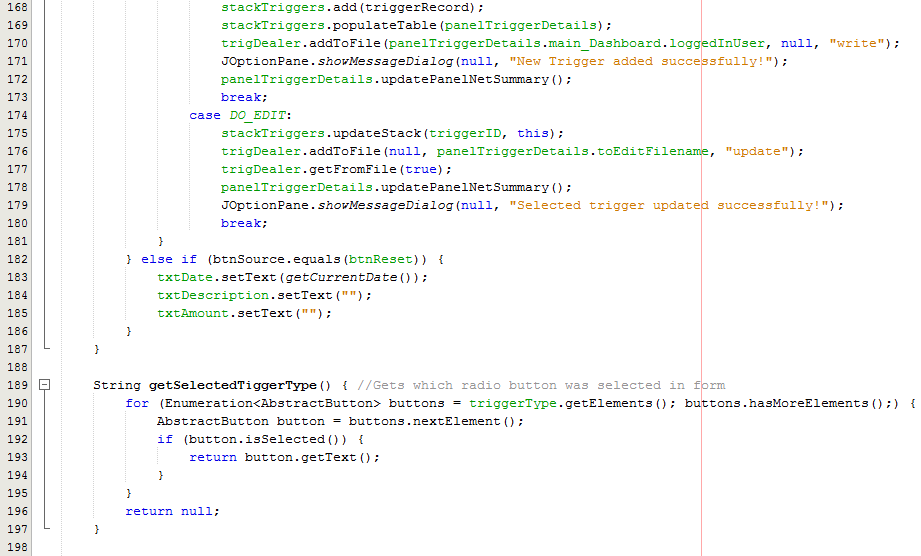
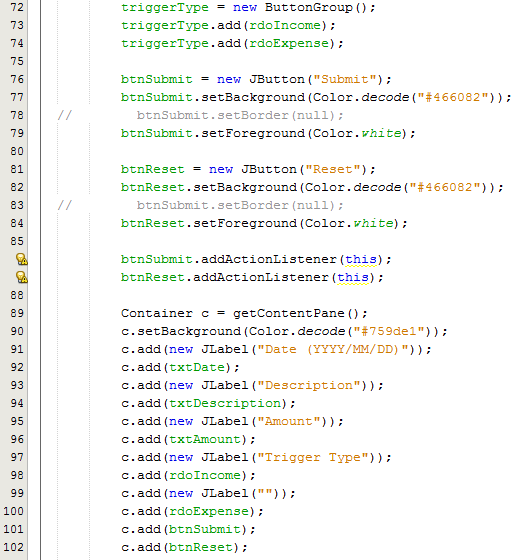
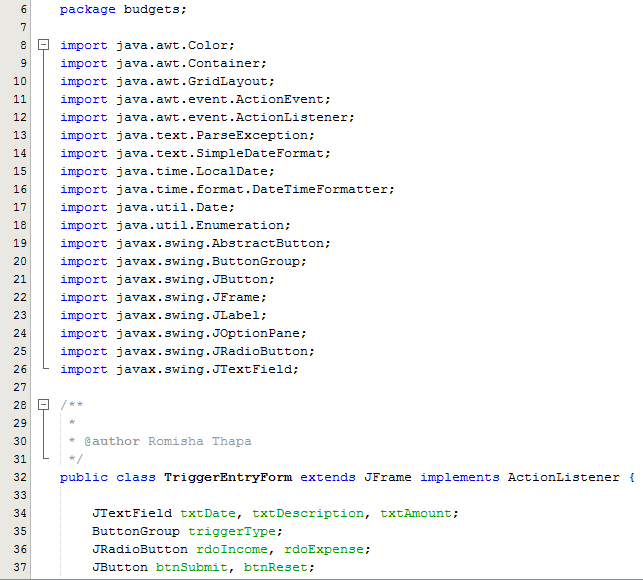
**TrigDealer:**



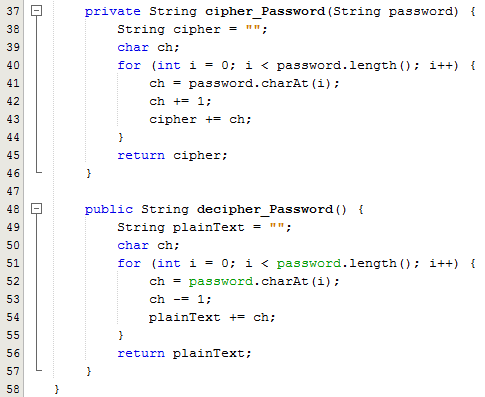
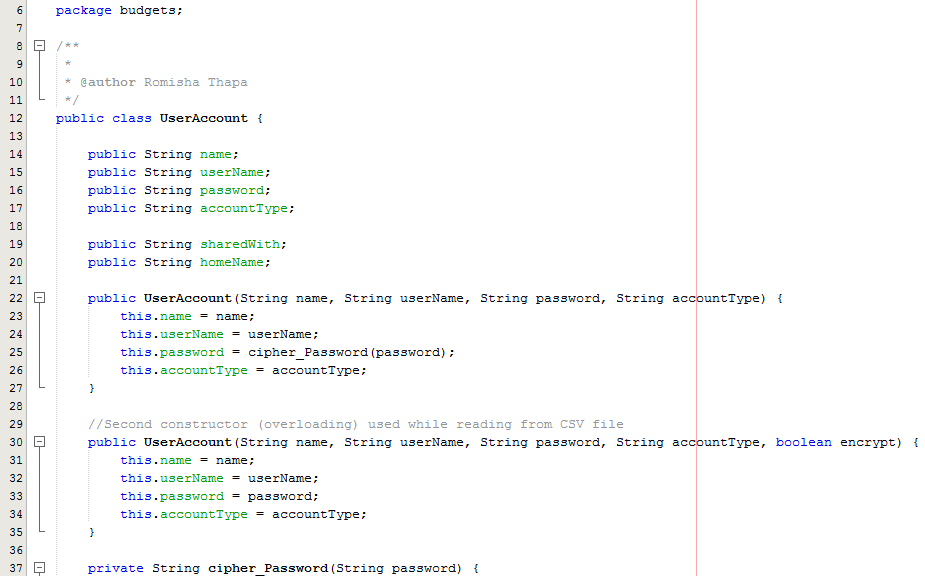
**Trigger**



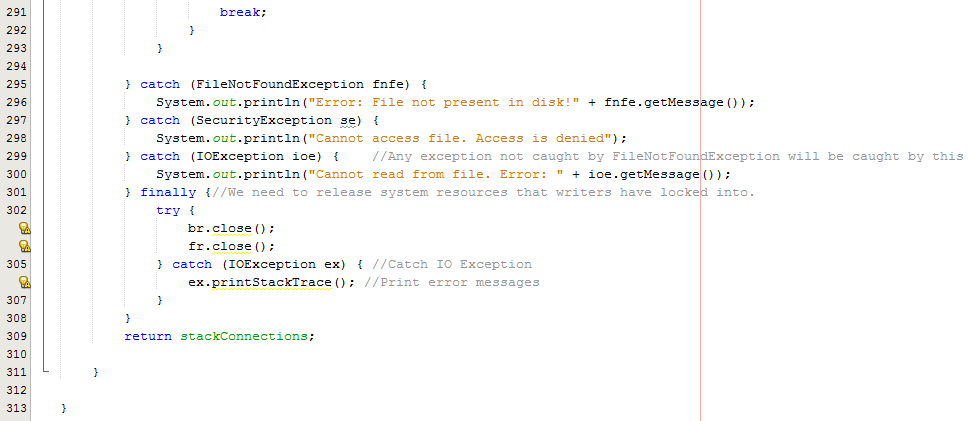
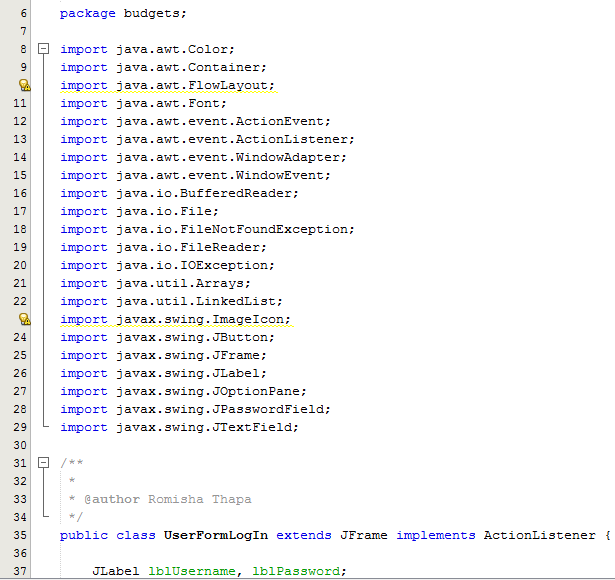
**TriggerEntryForm**



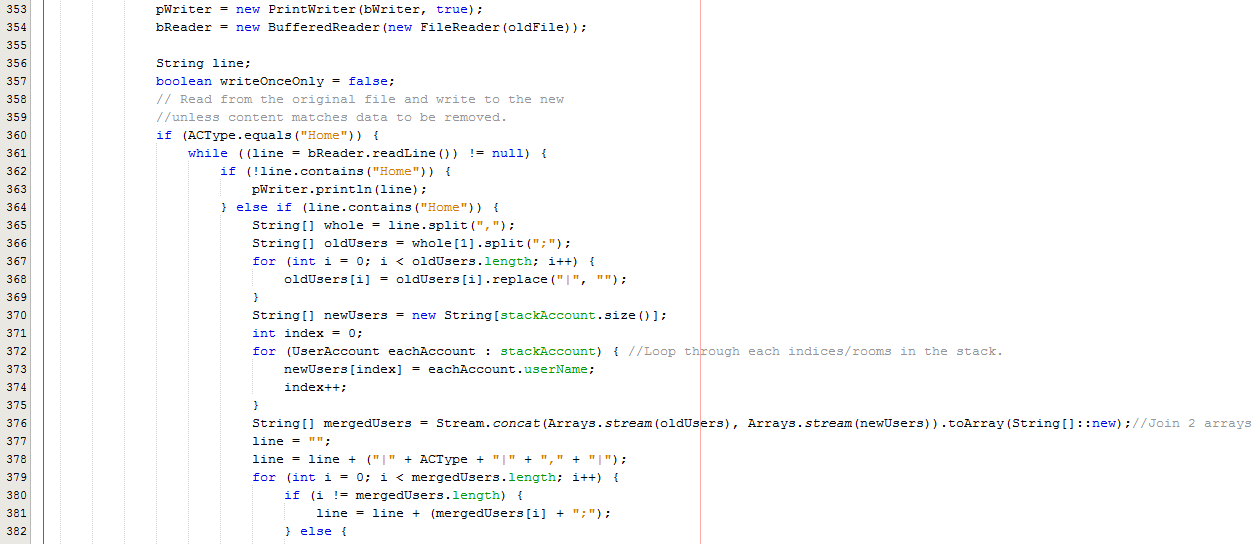
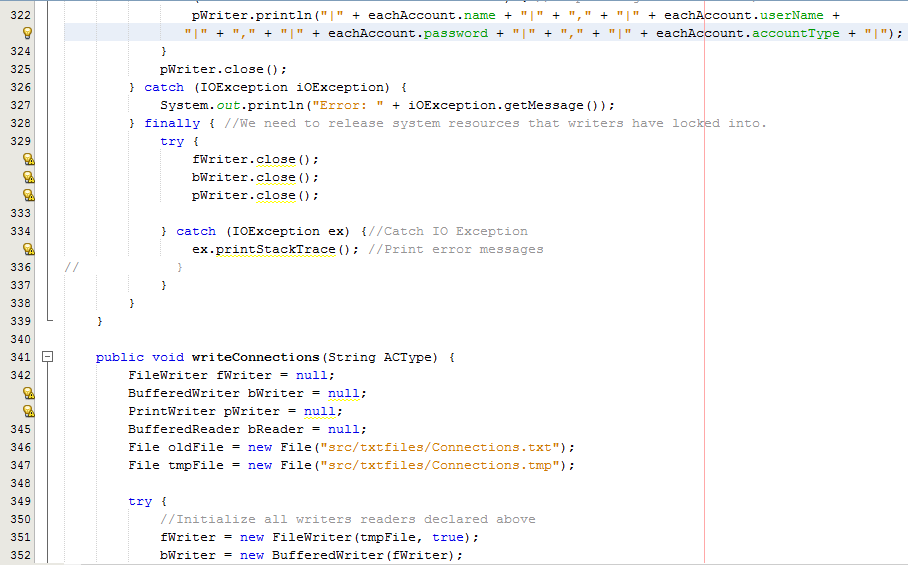
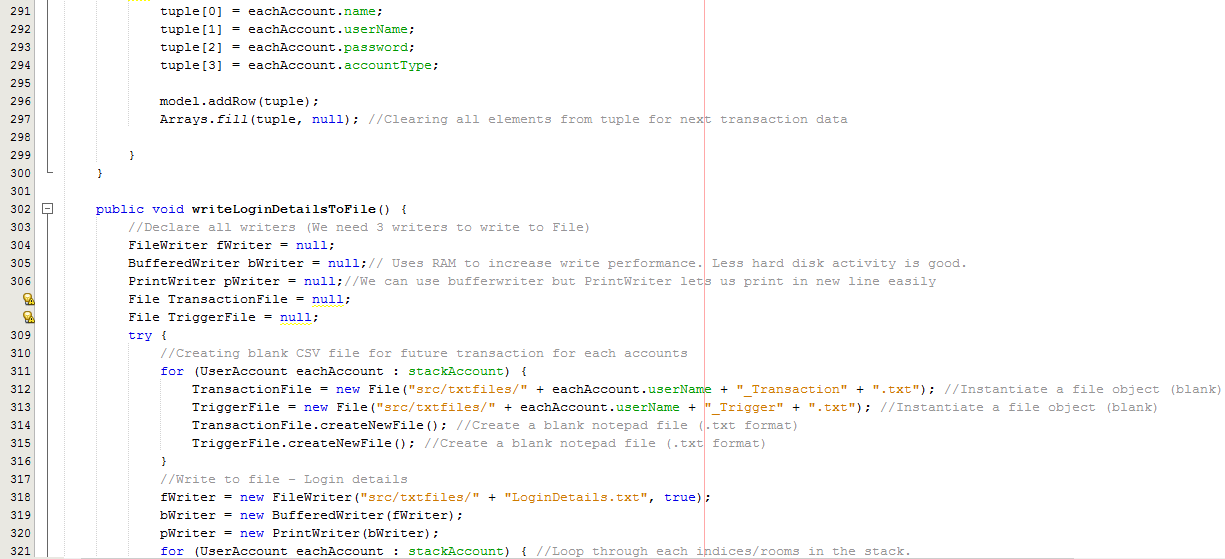
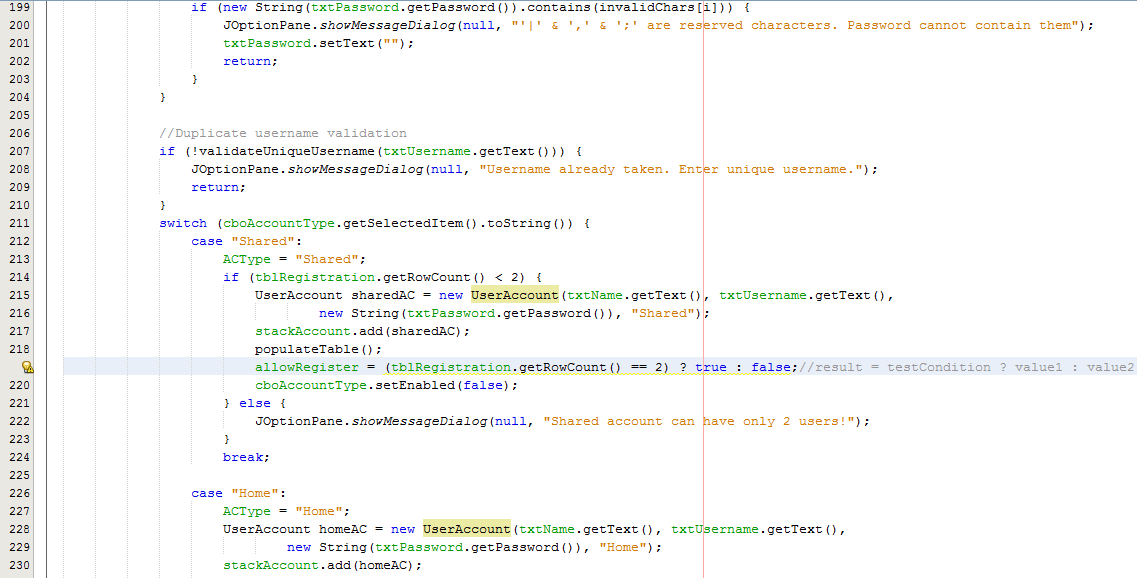
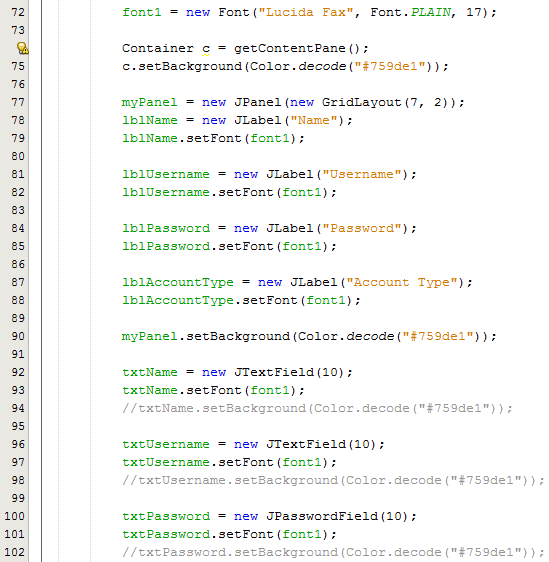
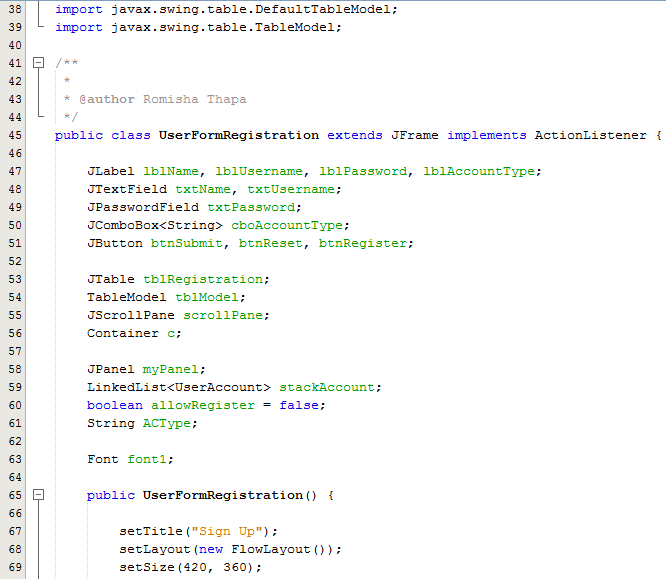
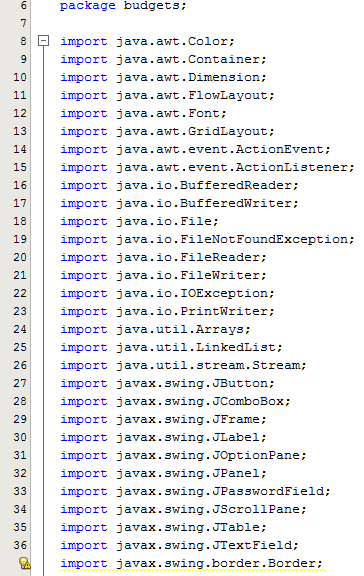
**UserAccount**



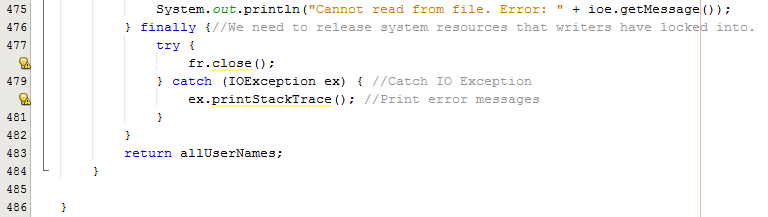
**UserFormLogin**



**UserFormRegistration**

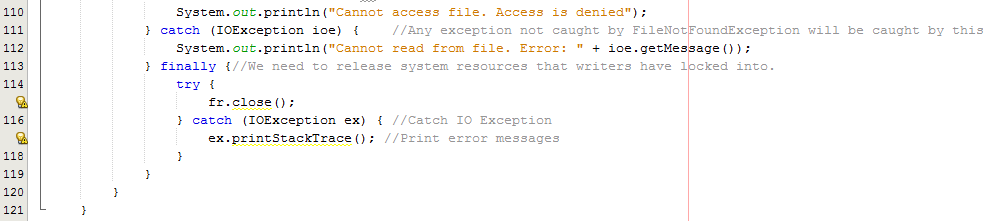
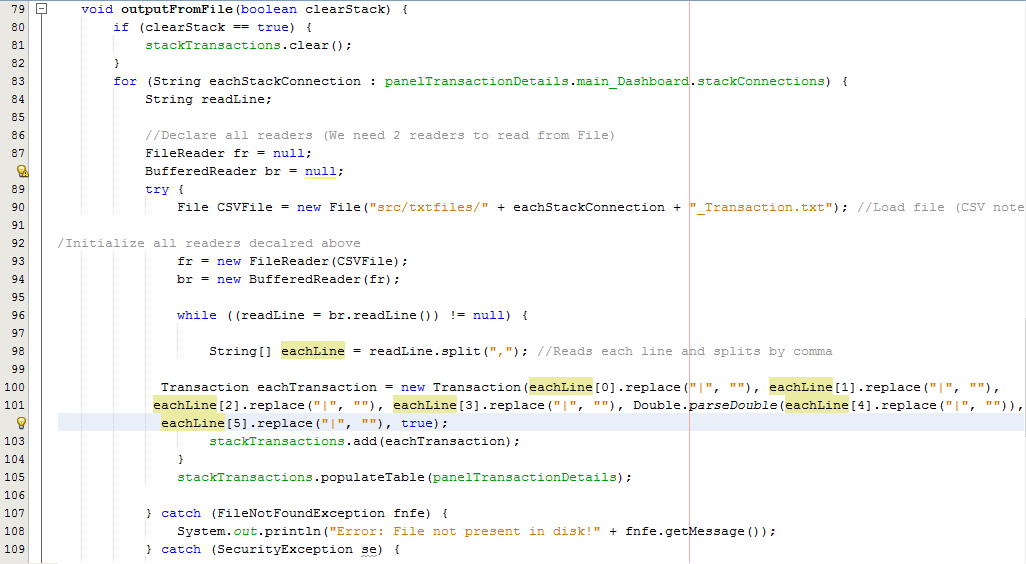




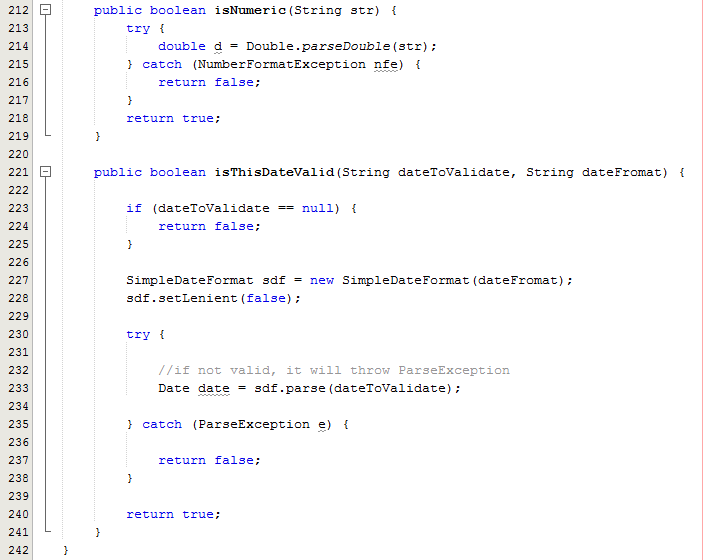


**2nd Task: Exception Handling**

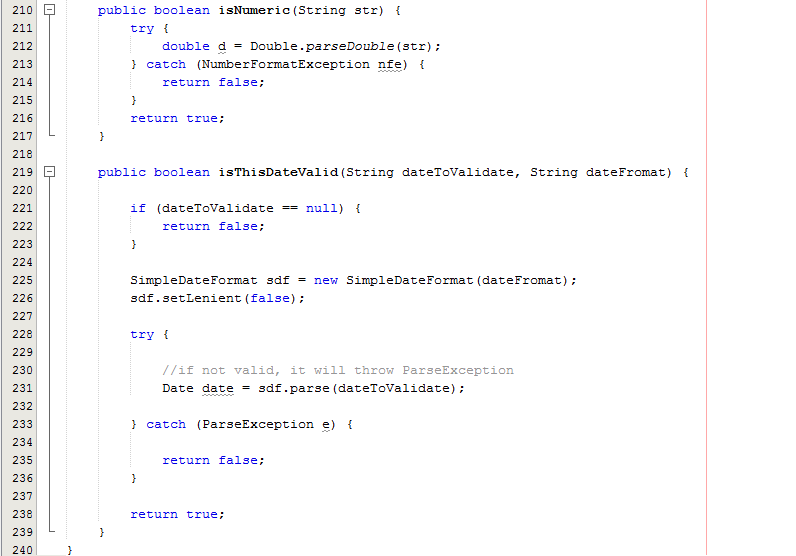
**TransactionDealer**



**TransactionEntryForm**

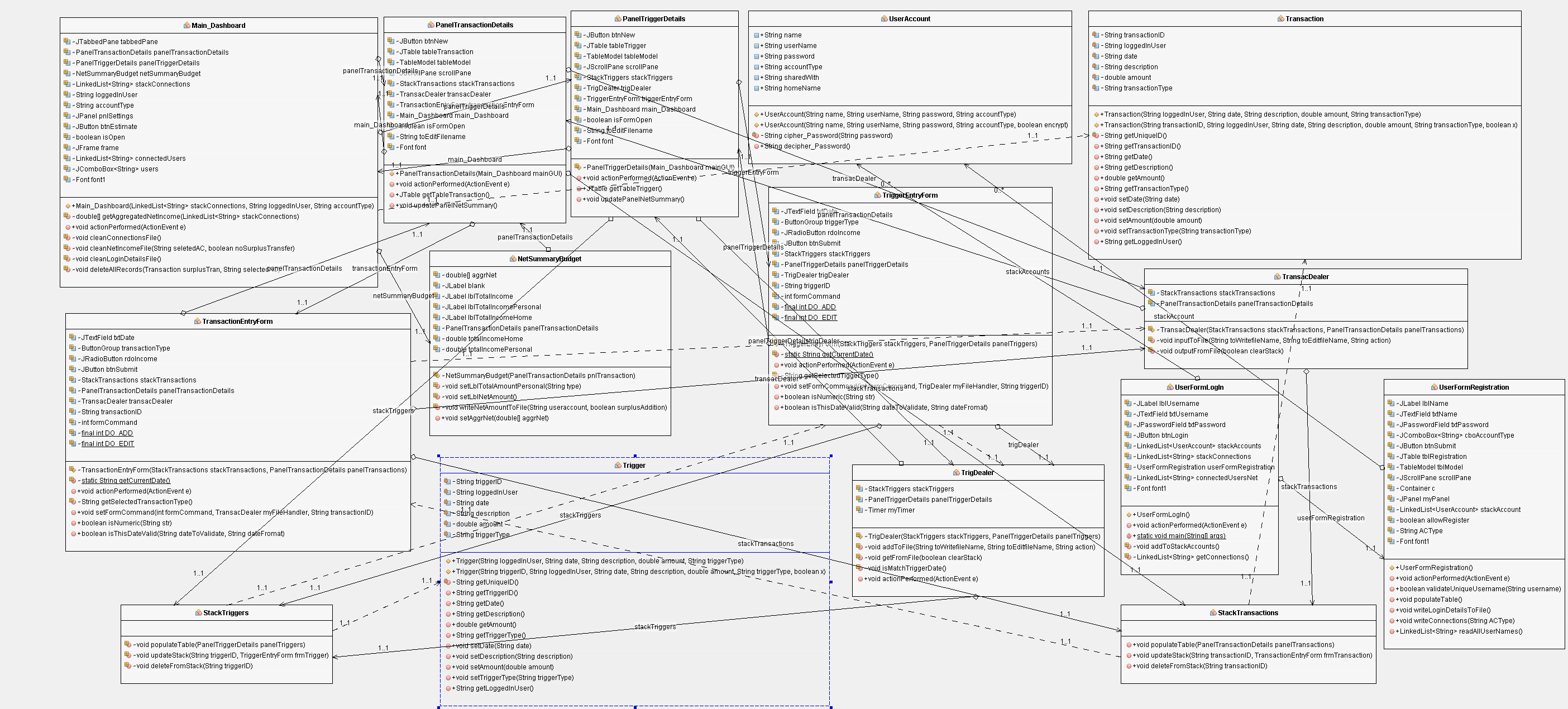


**TriggerEntryForm**



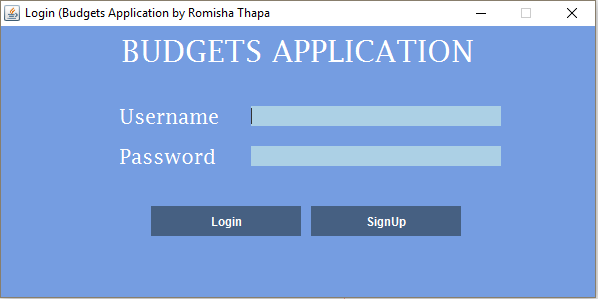
**Exceptions handling was done similarly for other classes too. This is available in the Task1 section in the source code of this assignment file.**

**3rd Task: Class Diagram**

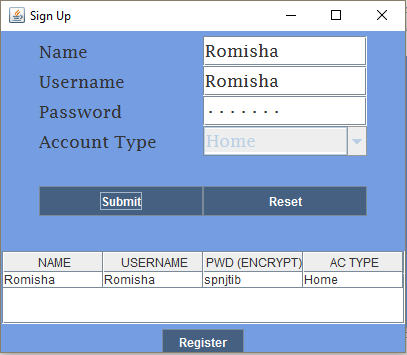
****

**4th Task: Integration (Testing)**

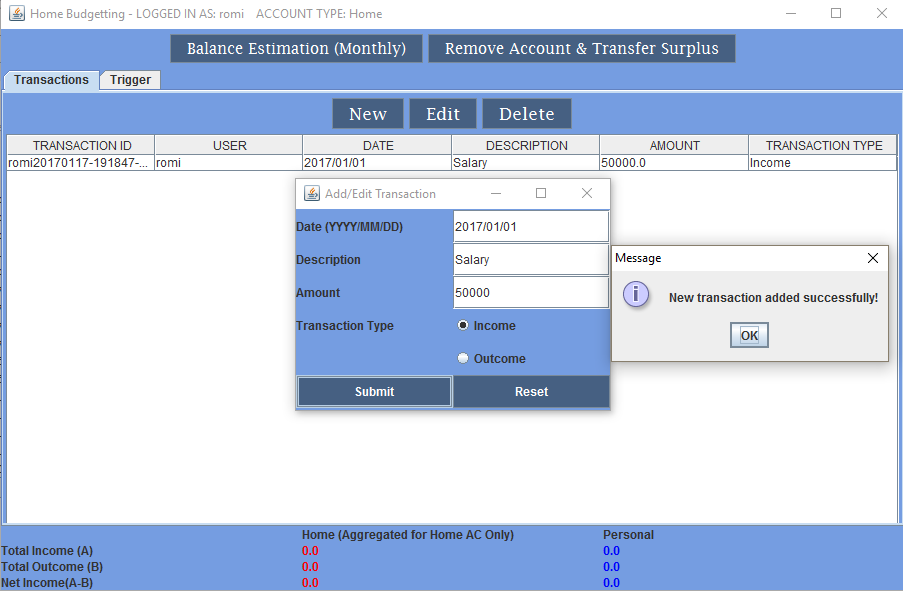
**Form(LogIn)**



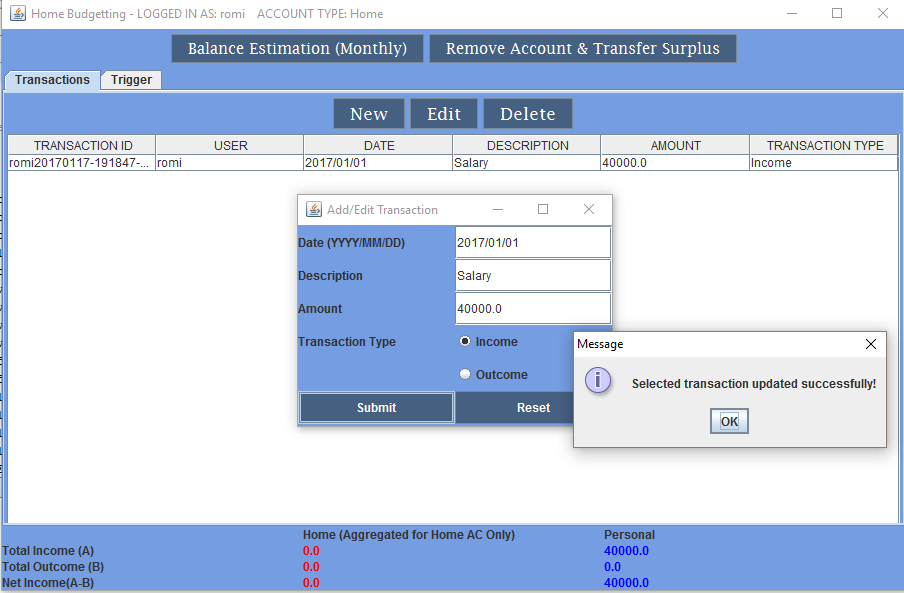
**Form(Registration)**



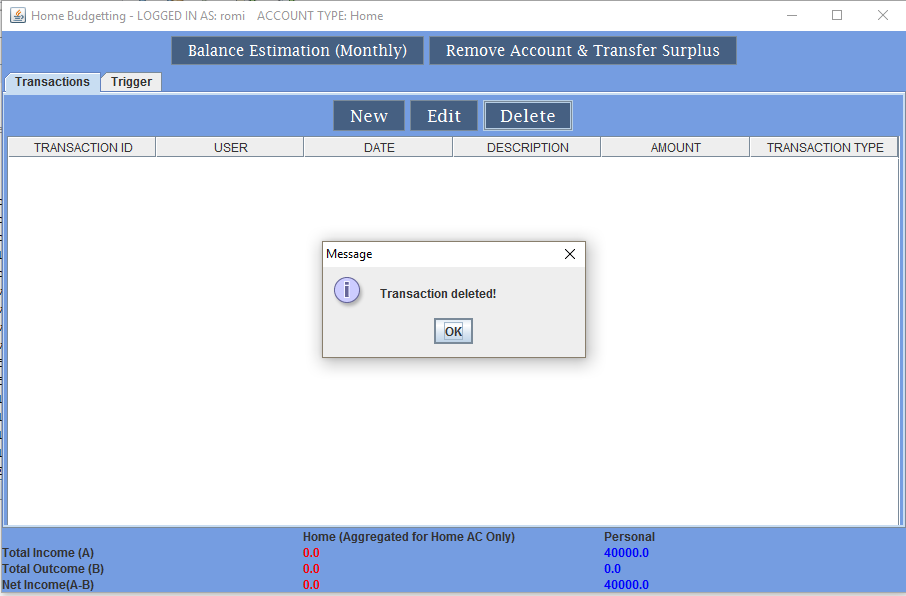
**Data input in transaction:**



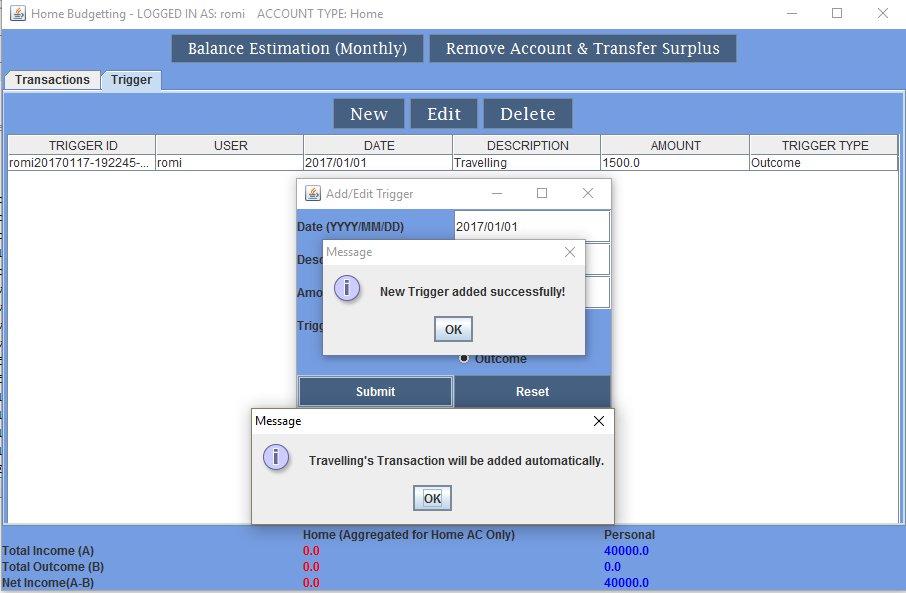
**Update of transaction:**



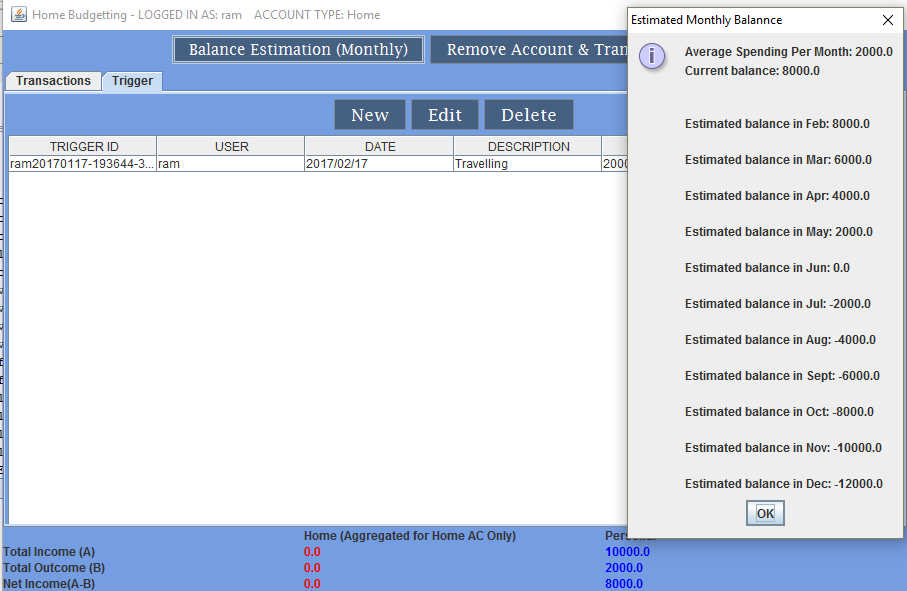
**Data Deletion:**



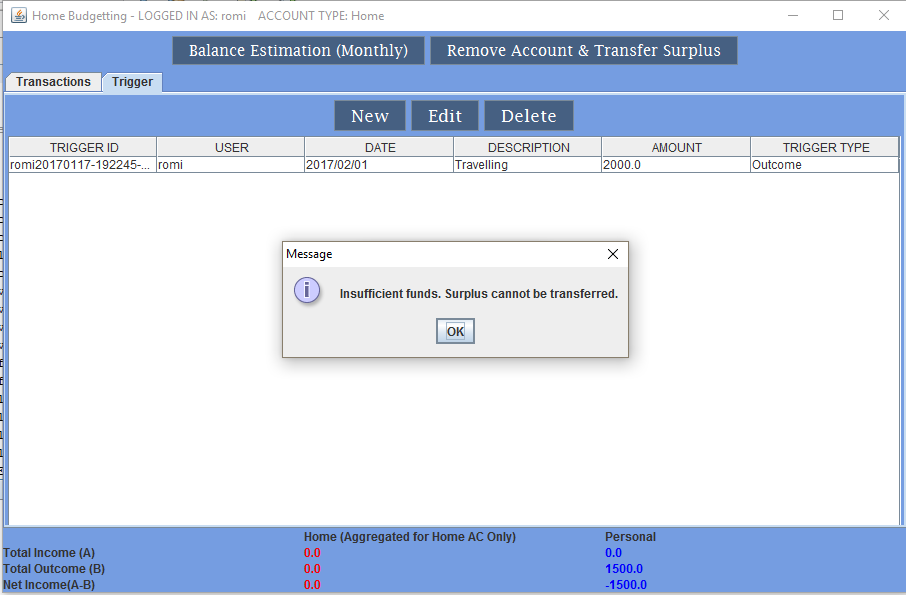
**Addition of trigger:**



**Balance Estimation:**



**Account deletion:**



**Conclusion**

**Budget Application was developed successfully using object oriented programming approach. Several OO features and programming elements were used such as polymorphism, encapsulation, file handing, String manipulation & so on.**

**UML class diagram was developed to give an overview of the relationships between the classes. Unit testing & integration testing was carried out and documented.**

**The application will benefit the user by providing transaction details recording feature with account management & estimation. Also, triggers could be set for automated entry.**