EPB

Easy Peasy Bank

EPB project idea from the lack of trusted online banks which come usually with few features.

EPB bank project will be the first project to offer the most important features for almost zero-fee, the project still in developing and I will just go through some features will be available for the users.

FEATURES FOR ZERO-FEE:

1- Free transactions worldwide.
2- Joint accounts*.
3- Instant account top-up & fast & cheap.

Why EPB?

- 24/7 availability- anytime, anywhere.
- Get approved fast.
- ALMOST ZERO-fees.
- Secure.

A deep overview on the system *(it might be expanded due to the developing process needs):*

The system will come with three types of users:

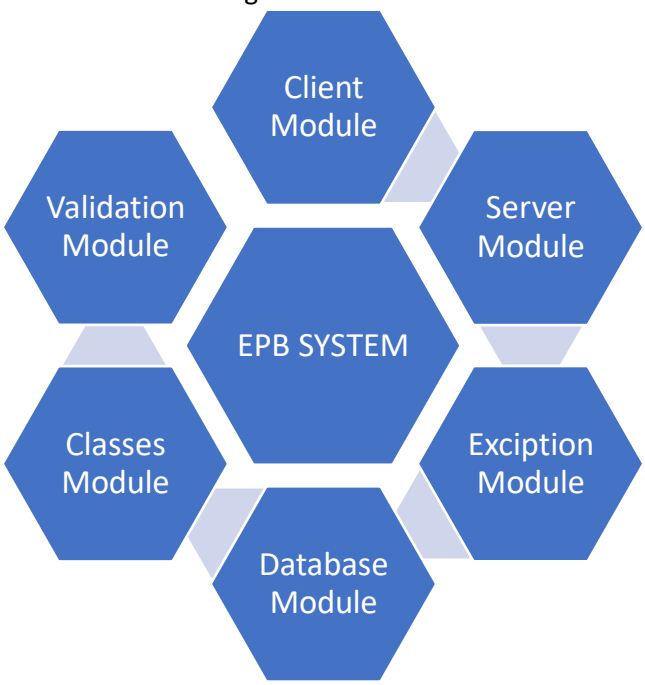
1- Customer.
2- Employee.
3- Admin.

The system will come with three types of accounts:

1- Normal account.
2- Joint accounts, in our case it's just for two people.

EPB

The system will be consisted of the fowling modules:



Providing a graphical user interface

Using the API in the database alongside of some methods in the graphical interfaces

Client module :

Which contains all the graphical interfaces for the users like:

1- AccountGui.java
2- CustomerBalanceInquery.java
3- CustomerChnagePinGui.java
4- CustomerGui.java
5- CustomerLogInGui.java
6- CustomerService.java
7- CustomerTransferGui.java
8- DepositCustomerGui.java
9- GuiTools.java
10- TransactionsHistoryCustomerGui.java

EPB

Server module:

Contains the majority of managing the system, creating customers and employees reviewing transactions

1- AccountsGui.java
2- ChatRoom.java(explicit use of multithreading)
    a. Using multithreading creates a chatroom for helping customers
3- CustomerEditGui.java
4- CustomerNewAccountGui.java
5- CustomerNewGui.java
6- CustomersJointAccount.java
7- EmployeeEditGui.java
8- EmployeeNewGui.java
9- EmployeesGui.java
10- EpbManageGui.java
11- EpbSystemLogInGui.java
12- TransactionHistoryGui.java

Validation module:

This module is designed to apply a validation on the input of some classes, I implemented it using the VISITOR pattern.

1- IValidator.java
2- IValidatable.java
3- ContactDetailsValidator.java
4- AccountBalanceValidator.java

Using a validation class together with the Visitor pattern to validate arbitrary business/infrastructure rules solved the problem of various types of future requested validations.

The Visitor will be the Validator, and the Visitee will be the entity class

"IValidatable" interface so I can keep track of what can be validated and what can't.

Using generics in own classes
a generic validation interface was created to solve the various classes which need the validation to be applied on

EPB

Classes module:

It contains the basic structure of the application, classes which is used to encapsulate the important information's along side with its features (methods).

1- Account.java
2- Admin.java extends person (inheritance)
3- ContactDetails.java
4- Customer.java
5- Employee.java extends person (inheritance)
6- JavaMailUtil.java
    a. It used to send the credentials for the customers
7- JointAccount.java extends account (inheritance)
8- Person.java
9- Transaction.java

A customer HAS-A list of various types of accounts(aggregation), normal or joint stored in one place.

Using nested classes
The contact details store some important information about the persons as long as to their address using the ADDRES inner class.

Database module:

It stores the data as long as it contains an API with various implemented methods.

Exceptions module:

handling exceptional states using own exceptions
It has two own exceptions to handle two specific

list of the major program versions submitted:

Building validation module using visitor pattern

Abdo-Saleh finished applying validation module

new hierarchy

Abdo-Saleh change the hierarchy of the application by adding some modules and change the address class to be a nested class of the contact details class

client version 1

EPB

Abdo-Saleh first working tested of the EPB application

Dictionary:

EPB

A **joint account** is a bank account that has been opened by two or more individuals or entities. Joint accounts are commonly opened by close relatives (such as by a husband and wife) or by business partners, but it can be used in other circumstances, such as by a club committee