

The Chittychitty Bank Project

The Chittychitty Bank is about to commission us with an application to deal with some currency movements for their customers. **An online solution is not envisaged**, so the customers ought to show up at a bank office to perform the operation at the counter.



- The bank has a number of different offices. Each office can have different accounts that belong to this office. Each account has a unique customer. Each customer can have one or more accounts, in one or several offices. All accounts hold money in euros.
- The system will have a client/server architecture with three layers: presentation, business logic and data access. The client side with the presentation layer will be installed at all the computers of all the offices of the bank. They will be operated by the bank cashiers. The server will be installed at the central office of the Chittychitty Bank in Panama City.
- Any customer can open an account at any office. The opened account will belong to that office. The customer must identify her/himself with the DNI card and the cashier will open the account in the system. The customer must provide some initial amount, no smaller than 1.000€, to open an account.
- The customer can then deposit or withdraw money from his/her accounts. S/he can perform these operations at any office, not necessarily at the one to which the account belongs. The customer cannot withdraw from an account any amount higher than its balance. S/he must identify her/himself with the DNI card and the cashier will collect or refund the money of the operation.
- The bank also markets currencies other than euros. A customer can buy to the bank some amount of a currency, paying it in euros from one of his/her accounts. He/she can also sell some amount of a currency, receiving the payment in euros in one of his/her accounts.
- Each office as a limited stock of each currency. We can suppose that any office has an unlimited supply of euros.
- A customer goes to an office and asks for a particular amount of a determinate foreign currency to buy, indicating the number of her/his account. If the bank has enough stock of that currency and the customer's account has enough balance to afford the purchase, then the transaction is completed and payed against the customer's account, and the cashier gives the currency to the customer.
- A customer goes to an office and shows a particular amount of a determinate currency to sell, indicating to into which account should be deposited the corresponding amount in euros. The transaction is completed, the cashier collects the currency from the customer and the amount is payed into the customer's account.
- To calculate how to convert the foreign currencies from or to euros, the bank knows the rate of each currency in euros. Apart form this there is a forex rate that is specific of each office. For instance, at airports this is much higher (typically 5-6%) than the standard values (around 1,5-2%). When the customer wants to buy a currency, the forex rate is added, and when s/she wants to sell, it is subtracted. For instance, if the rate for Canadian dollars is 0.679479, we are at the office of Rue de Rivoli in Paris (where the differential rate is 2,5%) and a customer wants to buy 1.000 CAD, we have that 1.000 CAD correspond to 679,48€. Since $2,5\% * 679,48€ = 16,99€$, we will deduct 696,47€ from the customer's account. But if the customer wanted to sell us 1.000 CAD we would pay him only 662,49€ in the account.

You have to develop a software project for this program, using a three-layered architecture. You have three phases for the three disciplines that we have worked in the course: Requirements, Design and Implementation. Please check the Moodlesite for details and/or consult with the teacher. Your program should run correctly in client-server mode.