

## Objective:

This casestudy is on Internet Banking System where users register themselves on the bank's website and avail various services like seeing their balance, transactions, request for the chequebook and net transfers.

## Technologies used:

Java (JDBC/Hibernate, SOAP/REST Services) for Business and Data Layer.

UI -- HTML/CSS/JS (for expert level)

## Level 1 ( for Beginners)

1. Create following classes.

### Customer

- a. custId (autogenerated)
- b. email address
- c. panCardNo.
- d. address ( plotId,street,city,state,country)
- e. mobileNo.

2. Account.

- a. accountId.
- b. balance
- c. dateOfAccountOpening
- d. customer
- e. type of account (Savings or Current)
- f. username
- g. loginPassword
- h. transactionPassword.

- 3 Transaction

- a transDate
- b account
- c amount
- d type ( debit/credit)
- e party

A Create a Test client to test the above classes by creating objects and calling the getters and setters.

B create a class Bank that maintains a Map of Accounts and has the following business methods.

- a. Adding a new Account to the Map
- b. Retrieve the account details based on the account No.
- c. Update balance on the account based on the transaction type.

## Level 2 (Intermediate level)

A Create database tables as required based on the above classes or generate dynamically using hibernate framework.

B Create a DAO layer for the CRUD operations on the database using JDBC/Hibernate.(instead of Map as above)

C Customer should be able to get a statement between two dates.

D Customer is validated when he logs in.

Create a Test Client to test the above.

## Level 3 ( Expert level)

A Create a business layer where some REST/SOAP services are exposed for the basic CRUD operations.

B Test these with required UI REST clients.

Note:

- Validations to be done wherever applicable.
- Good practices of Exceptions and logging to be applied.
- Some relevant assumptions can be made.
- Date objects to be handled instead of strings.
- Participants can submit their projects in parts level wise.
- Any parameter or functionality expected missing can be added or assumed.

