**Question**

Personal Financial Manager:

Explain how you would design a personal financial manager (like Mint.com). This system would connect to your bank accounts, analyze your spending habits, and make recommendations.

**Solution**

**Modules Involved:**

1. User Login Module
2. Security
3. Gathering the Initial Details [Savings Bank Details, Credit Card Details, Business Account, Loan Details, Assets and Liabilities]
4. Web service to pull information from the banks and other places
5. Storing the details which are not obtainable from other sources in the Application’s Database

**Application’s Database**

**User ID**

**User Name**

**User Password**

**User Details**

**User ID**

**Account Type**

**Account Number**

**Account Details (Including Assets and Liabilities)**

**User ID**

**Account Number**

**Transaction Type**

**Amount**

**Date**

**Transaction Details (Only store the details which are not accessible otherwise)**

1. Recommendations
   1. Based on the type of spending account we could assimilate the data on how much has been earned and spent in particular types.
   2. Month wise Expense Report
   3. Income vs Expense Report
   4. Next month’s forecast based on current month credit spending and other possible income
2. If we want we could maintain a separate Database gathering data from banking but we could leverage the web services. However if these data needs to be modified into some other format at a later time we can maintain a dedicated database for the implementation.

**Working**

1. Whenever a person accesses the application next time (after providing the initial details during first login) we could gather the initial details from the banks for displaying graphical representation.
2. If a person requests a particular format of recommendation then information can be queried and provided in real time.

**Application’s Database**

**Application’s Server**

**Web service**

**Other Bank’s Database**

**Client Browser**

**Modification (Optimal Solution)**

Some of the process has to take place in a order and incase if it fails it should be retried after a certain interval and we could leverage queue concept and priority for each tasks (pulling bank data, re-analyzing budget, assigning categories to the transaction, etc.)

Email-Alerts need to be given when the budget exceeds and there should be a mechanism to fix the budget by the user.

Data can be stored in normal textual format or other less storage and periodically updating to the application’s database.