**Spring boot batch-2 – case study**

* Create a spring boot project named as **bank-application** with the following features:
  + Create an entity class named **Account** with data members ->
    - accountNumber,
    - customerName
    - balanceAmount
  + Another entity class named **Transactions** with data members ->
    - tranId
    - accountNumber
    - transactionDate
    - description
    - deposit
    - withdraw
    - balance
* Entity classes should have foreign key(one-one) relationship
* Account number should be generated in specified format
* tranId auto generated
* Create Repositories for Account and Transactions using Spring-Data
* Create service layers with the following features:
  + AccountService
    - To create an account
    - To deposit amount
    - To withdraw amount
    - To check balance
  + Transaction\_Service
    - To get transactions statement by acno
    - To get transactions statement by acno, datewise
  + \*\* define the methods with relevant return types and method signatures
* \*\* whenever the deposit or withdraw operations are performed the transaction details should be added to transactions database.
* Create controller layers with the following endpoints:
  + /createaccount
  + /deposit
  + /withdraw
  + /checkbalance
  + /viewTransactionsById
  + /viewTransactionsDatewise
  + Apart from the above, you can define suitable end points based on the operations performed.
  + Application should run on port: **9292**
* Throw the relevant custom exceptions for transactions like not found etc.
* Display meaningful messages for prompt and exceptions.