Web Services and API Development CA3 (35%) Project - Online Bank Web Service API

This is a group project, if you are not in a group contact the lecturer.

**Problem Domain**

Online Banking is a mainstream service offered by most banks today. A typical consumer online banking application requires the following:

● **A Customer** Customers are individuals with a name, correspondence address, email and security credentials. A customer can hold one or more accounts.

● **An Account** An account has a sort code (identifying the branch), an account number and a current balance, The account has a list of transactions.

● **A Transaction** Each transaction is either a debit or credit on a particular date, with a description and post-transaction balance.

Customers will be able to do the following:

**Create** Customers should be able to create an account with the bank, and a customer who has an account should be able to add additional accounts. For example a typical customer may have a current account and a savings account.

**Lodgment** For the lodgment, a bank customer can specify the amount to lodge with the credit card that will be debited.

**Transfer** For the transfer, the bank customer can specify the amount to transfer and an account to transfer to.

**Withdrawal** For withdrawal, the bank customer can specify the amount to withdraw and the card that will be credited.

**Balance** The customer can request a balance on any account at any time.

**Project Requirements**

You are required to design, document, and develop an API for the problem domain above. The document will have the following sections:

**1. An Introduction:**

The introduction will expand the problem statement to describe how the API fits together. For instance:

a. Textual description of the problem and the proposed solution.

b. An Entity-Relationship diagram would be a useful addition.

c. Security will be a concern across the whole API. Describe how you tackle that issue.

**(10 Marks)**

Introduction is to be maximum 2 pages.

**2. The RESTful API:**

The API describes all the entry points for the above problem domain. Each entry point should be documented under the following headings:

a.  **API Name:** e.g. Users Resource

b. **Description:** e.g. This allows the retrieval of all user resources.

c. **URI:** e.g. /users

d. **HTTP verb:** e.g. GET

e.  **Parameters:** e.g. user\_id (Integer, Required), name (String, Optional), for a POST this would be a JSON object.

f.  **Resource contents:** e.g. an example of the returned resource.

g.  **Pre-Conditions:** e.g. no record for the user with the specified user\_id must exist.

h.  **Post-Conditions:** e.g. a new record for the user with the specified user\_id exists.

**(30 Marks)**

**3. Prototype:**

The API prototype, must be implemented according to the following requirements:

a. BACKEND: A server implementing at least THREE of the entry points listed above (Lodgment / Transfer / Withdrawal / Balance) returning resource representations in XML and JSON.

b. BACKEND: A server developed in Java which implements at least THREE of the entry points listed above (Lodgment / Transfer / Withdrawal / Balance) of the API using in-memory objects. Constraints should be implemented, balances should be updated and transactions should be remembered as the API is called. This should be achieved with a database.

c. FRONTEND: An HTML+JavaScript or Mobile or Desktop client calling ALL portions of the API. E.g., the client should check the account balance allowing a transfer or withdrawal.

You should provide screenshots to demonstrate the operation of the Web Service.

The prototype source code should also be included with the submission in a ZIP file.

**(60 Marks)**