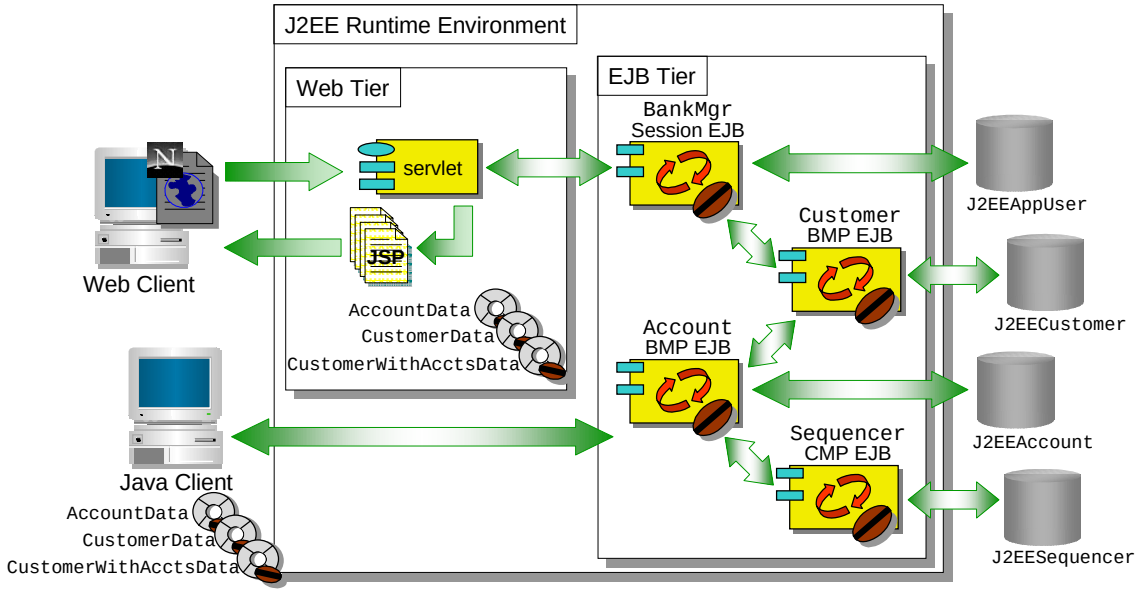
J2EE™ Lab Application Overview

Case Study: J2EE Lab Application Overview

The application used for the J2EE™ course lab exercises is based on a simple banking model that implements a few basic bank account maintenance routines used by customers and bank managers. In this model, customers login to the system with a user name and password to manage their accounts. Managers access a list of bank customers and can add or remove customers and accounts, as well as modify customer information (address, phone number, name, and so on). The lab project supports a one-to-many relationship between customers and their accounts. The maintenance of joint accounts is not supported.

During the course you create session and entity Enterprise JavaBeans™ that connect to a persistent store, such as an Oracle or SQL Server database. You code methods to access the EJBs™ and manage the bank customers and customer accounts. You also develop Java client and HTML browser-based interfaces to the banking system. A sample diagram of the system components is shown in Figure 1.



The data store that supports this application contains four tables: J2EECUSTOMER, J2EEACCOUNT, J2EEAPPUSER, and J2EESEQUENCER. The J2EECUSTOMER table stores information on bank customers, the J2EEACCOUNT table contains account information, the J2EEAPPUSER table controls security access to the system, and the J2EESEQUENCER table maintains primary key sequences used when adding new rows to the J2EEAPPUSER and J2EEACCOUNT tables. The structure of each of the tables is shown in Figure 2. The lab appendix contains a listing of the J2EEAPPUSER table and the associated customer and account data for one of the bank customers that can be used as a test instance for the lab exercises.

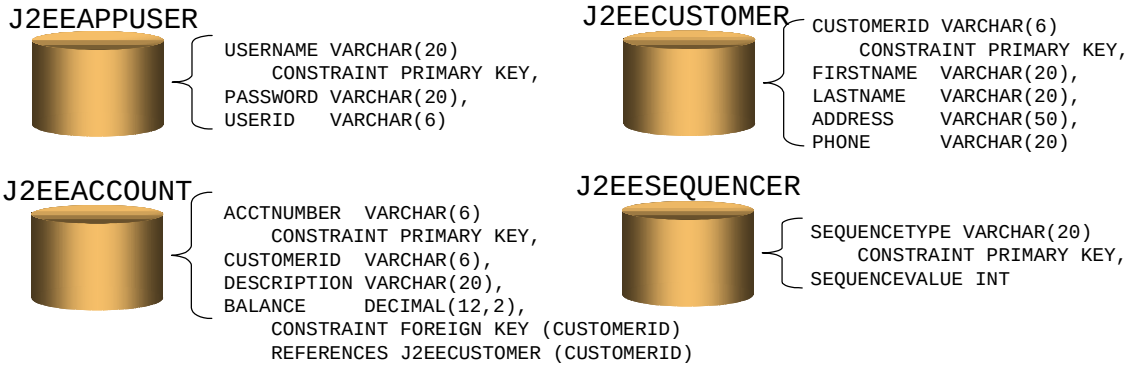


Figure 2 - Banking Application Data Tables

Figure 3 illustrates the application's EJB components and their public methods. Users login to the system using the `BankMgr` stateless session EJB's `loginUser` method. The `loginUser` method uses JDBC to connect to the `J2EEAPPUSER` table and validate system users. The functionality of the `Customer` BMP entity EJB is accessed through business methods on the `BankMgr`. The `BankMgr`'s `getCustomerData` and `getCustomerWithAcctsData` methods retrieve data model beans that reflect the entity state for the specified customer. `GetAllCustomers` returns a collection representing all of the customers in the `J2EECUSTOMER` table. The `Customer` table interfaces with the `Account` EJB when creating a `CustomerWithAcctsData` data model bean.

To reduce the complexity of the lab exercise, application clients access the `Account` BMP entity EJB directly. Using the public methods on the `Account` EJB a client application can add and delete rows from the data store, retrieve an `AccountData` data model bean, locate a specific account, and locate the set of accounts associated with a specified customer. The `Account` EJB retrieves primary keys from a `Sequencer` EJB when adding new rows to the `J2EEACCOUNT` table.

The `Sequencer` CMP entity EJB contains a single method that returns a unique primary key value for a specified table.

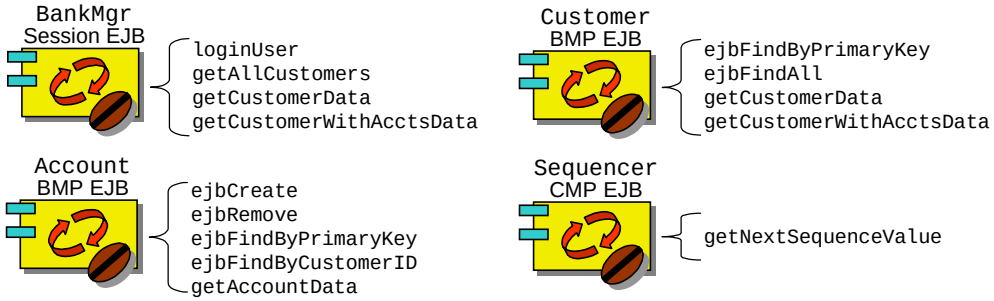


Figure 3 - EJB Method Details

Filename:	laboverview.doc
Directory:	H:\FJ310_revC_0301\BOOK\EXERCISE\Exercise_Originals
Template:	C:\Program Files\Microsoft Office\Templates\Normal.dot
Title:	Case Study: The New User Interface
Subject:	
Author:	jdibble
Keywords:	
Comments:	
Creation Date:	04/24/01 10:19 AM
Change Number:	2
Last Saved On:	04/24/01 10:19 AM
Last Saved By:	Nancy Clarke
Total Editing Time:	0 Minutes
Last Printed On:	04/24/01 12:40 PM

As of Last Complete Printing

Number of Pages:	3
Number of Words:	441 (approx.)
Number of Characters:	2,519 (approx.)