**Experiment No.: 10**

**Designing Java Applications for Mainframe Trading Systems**

**Learning Objective:** Design, develop, and deploy Java applications for mainframe trading systems that are secure, performant, and scalable.

**Theory:**

As with many large organizations, Bank of New Amsterdam (BNA) has several different operating systems running many kinds of software systems. The primary platform for the production business data is the IBM Mainframe S/390 (the mainframe). In addition, Microsoft NT boxes use Internet Information Services (IIS) and other Microsoft software to provide reporting capabilities, using a SQL Server shadow copy of the business data. BNA also has an investment in Oracle on the S/390. Several preexisting legacy applications provide trade-processing links to the major securities exchanges. These applications work well and have been developed and maintained for the past two decades, during which time a great deal of time and money has been spent. The applications do not need modification; instead, they need a new web front end to make them look good. The development environment is primarily Windows-based.

The advantages of using the mainframe include its reliability, scalability, flexibility, and security. The mainframe has been running continuously for years in BNA without major problems. The mainframe can easily be amended to add hardware resources, such as CPU, memory, disk, or networking hardware, to increase capacity without changing the operating system or application systems. In addition, the CPU, memory, and disk space can be redistributed as application requirements change.

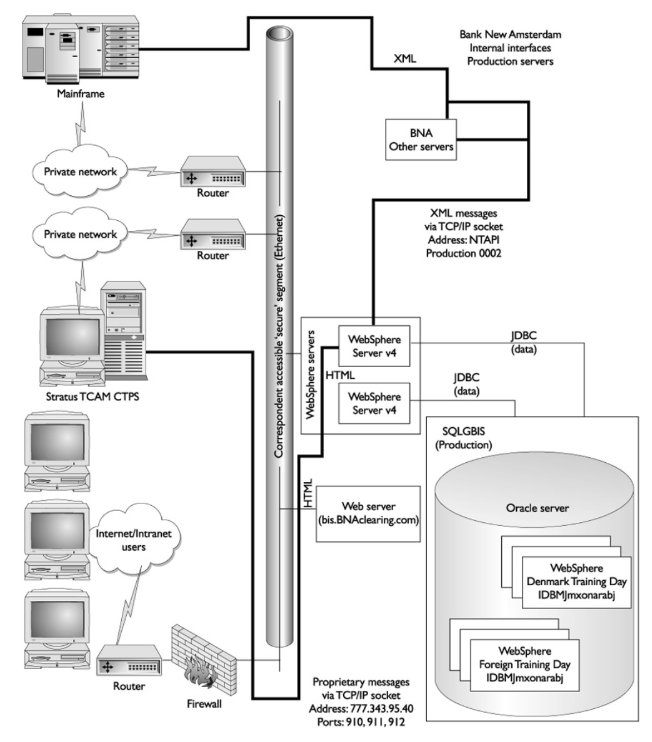
|  |  |
| --- | --- |
| Actor | Role |
| Customer | Trades with the application based on limits. |
| Trader | Trades with the application based without limits and routes orders to the mainframe trading system. |
| Continuous trade processing system (CTPS) | Routes orders to the mainframe trading system. |
| Mainframe trading system | Executes trades and manages account balances. |

Traditionally, the downside to the mainframe has been the user interface—the 3270 dumb terminal (the green screen), which is not user friendly. Prior to the commencement of BNA’s Java project, another group spent a few months trying to develop Windows Active Server Pages (ASPs) to talk to the mainframe. The only solution providing ASP reports, a portfolio management system, solved the interface problem, but it was difficult to connect to the mainframe using ASP for trading.

BNA has experienced difficulty getting the order messages to the mainframe. Along with performance problems, it seemed that every ASP order transaction required multiple dedicated connections to the SQL Server database. Fortunately, Java Database Connectivity (JDBC) connection pooling and Java’s platform independence provided the performance and scalability that was needed. Moreover, it allowed us to take advantage of the mainframe for deployment and Windows for development. (Because budget is a limiting factor in an economic downturn, where every developer is competing for business, it is critical that you deliver a solution quickly that will integrate with an organization’s existing infrastructure.)

The IBM HTTP server, WebSphere application server, and Oracle 9i Enterprise relational database management system (RDBMS) are all in existence on the company’s legacy IBM mainframe enterprise server

|  |  |  |  |
| --- | --- | --- | --- |
| Type of Component | Vendor/Component Name/Version | Description | Software/Hardware Required for Support |
| Application Server | IBM WebSphere Application Server 3/4 | Serves up JSPs and servlets, run EJBs, and provides JDBC connection pools to user data | IBM S/390, Windows NT 4.0 SP4 |
| DBMS | Oracle 9.0i | Database required for application data (orderdb) | Oracle 9i |
| JavaScript | Netscape/JavaScript | ECMA/JavaScript | IE 5 and above or Navigator 4 and above |
| Browser | Microsoft or Netscape | Web browser required to support JavaScript | IE 5 and above or Navigator 4 and above |
| XML Tool | IBM/XML Parser for Java/3.1.1 | A libary for parsing and generating XML documents | Windows NT 4.0 SP4 |



Production Environment

**WebSphere Application Server**

The WebSphere partition is also connected via TCP/IP sockets to the BNACS API. This API is used to communicate with and retrieve information from the mainframe system via an XML-based message format. Finally, the WebSphere partition is connected to the Stratus TCAM CTPS (the continuous trade processing system from TCAM) application. The WebSphere application sends and maintains orders, and it can look up order status by communicating with CTPS via a proprietary message format.

**Continuous Trade Processing**

The CTPS system is an order-routing system that is connected to several exchanges and market makers (firms that stand ready to buy and sell a particular stock on a regular and continuous basis at a publicly quoted price). The system receives orders either by direct entry into its terminals or via an in-house–built TCP/IP socket server (sometimes known as the Stratus Gateway Interface). These orders are routed to the appropriate exchange or market maker based on a set of correspondent-defined rules. Executions are then passed back from the exchange or market maker to CTPS, which updates the order file and forwards the result of the execution to the mainframe.

**SQLBIS Database Server**

The SQLBIS database server was created to service BNA’s trading website, Brokerage Information System (BIS). On the database server are several databases (or data marts) that are used by BIS and the BIS Trading Area to look up account and application access as well as cross references and other information. The ORDERDB database was created exclusively to support the WebSphere applications. In development, the tables and views were created by developers and migrated to production by the database administration (DBA) group.

**Result and Discussion:**

**Learning Outcomes:** Students should have be able to understand

LO1: Define software Architecture.

LO2: Identify different phases in software Architecture.

LO3: Explain Implementation of software Architecture .

.

**Course Outcomes:** Upon completion of the course students will be able to understand Implementation of software Architecture.

**Conclusion:**………………………………………………………………………………………

For Faculty Use

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Correction Parameters** | **Formative Assessment [40%]** | **Timely completion of Practical [ 40%]** | **Attendance / Learning Attitude [20%]** |  |
|  |  |  |