2

Introduce Rich Client Applications

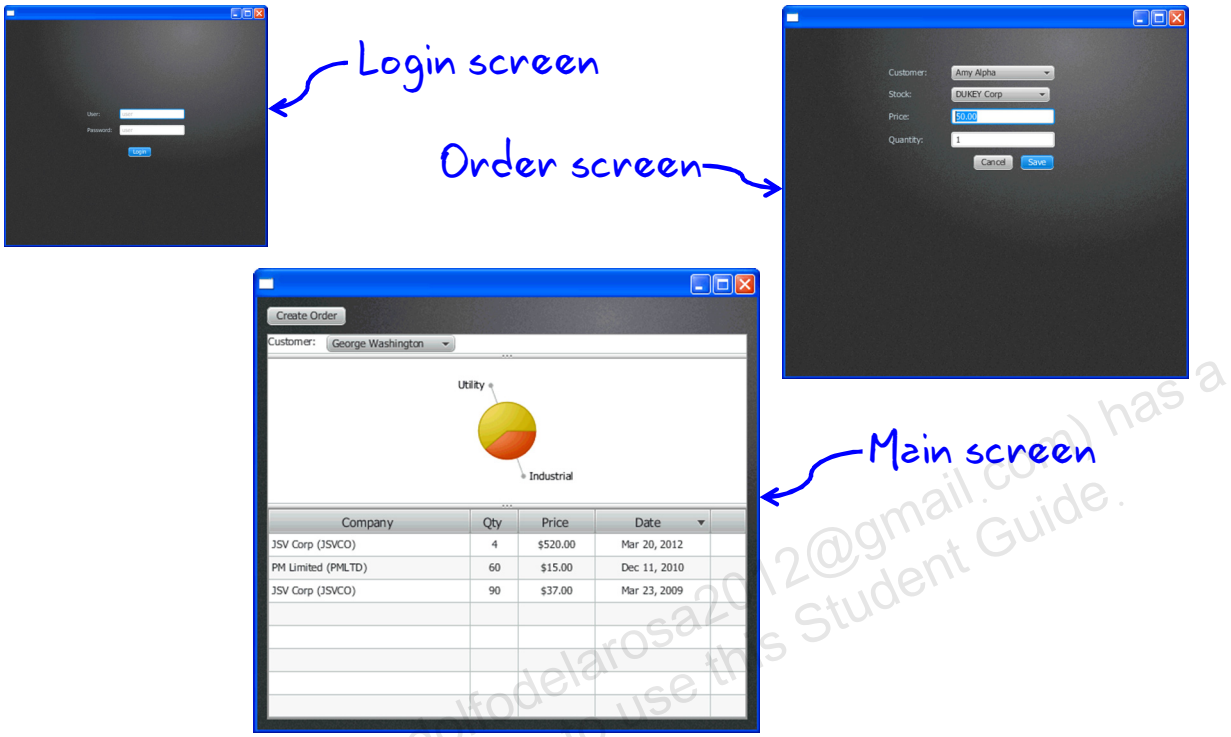
ORACLE

Objectives

After completing this lesson, you should be able to:

- Describe an overview of the BrokerTool application
- Explain the problem statement of the BrokerTool project

BrokerTool Application Overview



The BrokerTool application is a rich client, three-tier application that you will develop in this course. BrokerTool is Java technology-based, interactive, client-server system for creating, updating, and viewing customer and stock information that is contained in the database.

The application has been developed using Java, JavaFX, JDBC, and JavaDB.

The Login screen includes layout containers, text fields, labels, and a button.

The Order screen includes layout containers, combination boxes, labels, text fields, and buttons.

The Main screen includes buttons, labels, a combination box, an animated chart, a smart table , and layout containers.

BrokerTool Application Customer Problem Statement: Requirements

Functional requirements:

- Buy, sell, and update stocks for customers in the database by using the graphical user interface (GUI).
- Add customers to and remove customers from the database.
- Modify customer's name and address.
- View the current price of any stock in the database.

You are a working with a customer who wants you to develop an application based on the functional requirements. If you had enough time during this course, you would implement all of the features listed above. However, we have scaled back on the features list in the application that you develop during this course.

BrokerTool Application Customer Problem Statement: Resources & Constraints

Project resources:

- A server machine, on which the Java DB database is installed
- Multiple client graphic workstations on which the JDK is installed

Project constraints:

- Database queries should use the customer's name or a unique stock symbol.
- Some initial work was completed by the consultant.

ORACLE

The customer's requirement is that the application will post data in a JavaDB that is hosted on a server machine. In the case of this course, the server is hosted locally on your student machine.

BrokerTool Application: Customer Table

The `Customer` Table

Field Name	Type	Comment
customer_ID	int	Primary Key
account	char(15)	
fullname	char (255)	
address	char (255)	
broker_ID	int	

BrokerTool Application: Shares Table

The `Shares` Table

Field Name	Type	Comment
shares_id	int	Primary Key
symbol	char (8)	
quantity	int	
customer_id	int	
purchaseprice	double	
purchasedate	timestamp	

BrokerTool Application: Stock Table

The `Stock` Table

Field Name	Type	Comment
symbol	char (8)	Primary Key
stockname	char (255)	
sector	char (255)	

BrokerTool Application: Broker Table

The `Broker` Table

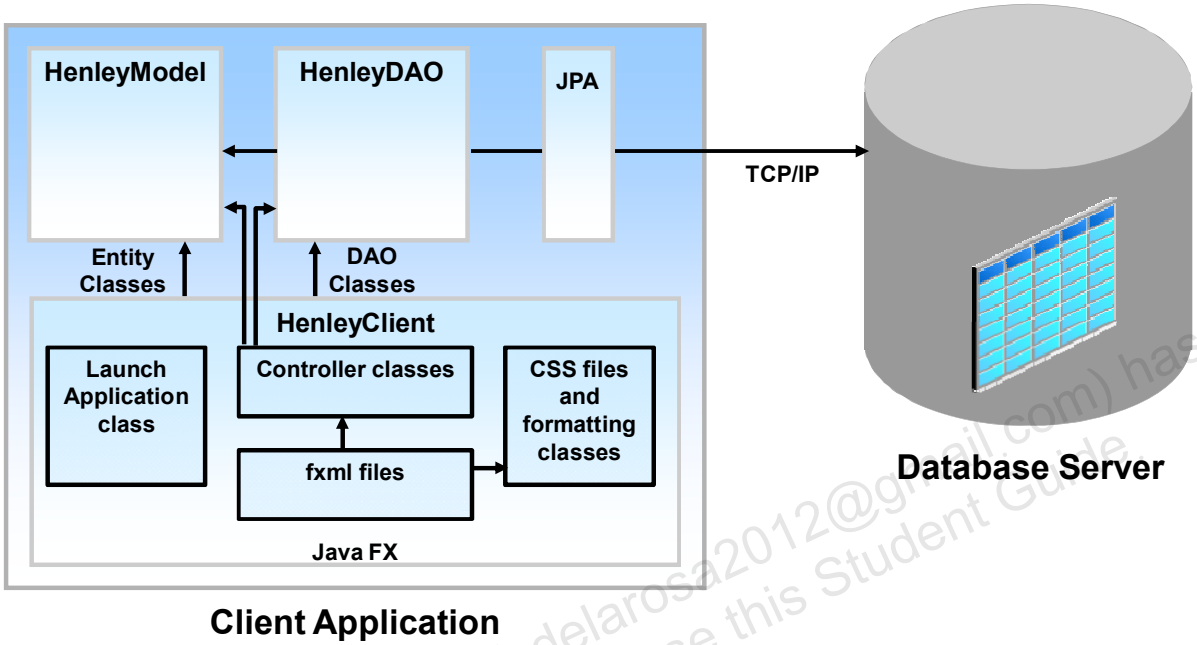
Field Name	Type	Comment
broker_id	int	Primary Key
brokername	char (255)	
address	char (255)	

Your Assignment

You need to enhance the BrokerTool application to a full-fledged Portfolio Manager application using the following technologies:

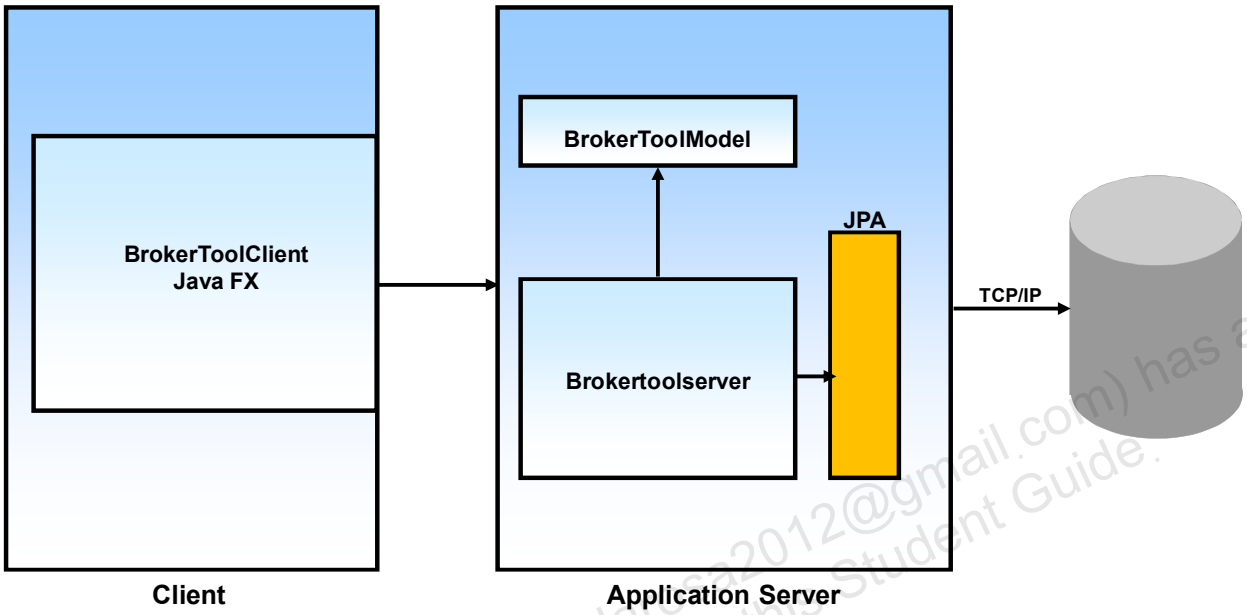
- NetBeans IDE with Java EE
  - JavaSE 7
  - JavaFX 2
  - JavaDB
  - Glassfish Server
- JPA
- JUnit for testing
- Logging API (Log4j or Java logging APIs)

Two-Tier Application Design



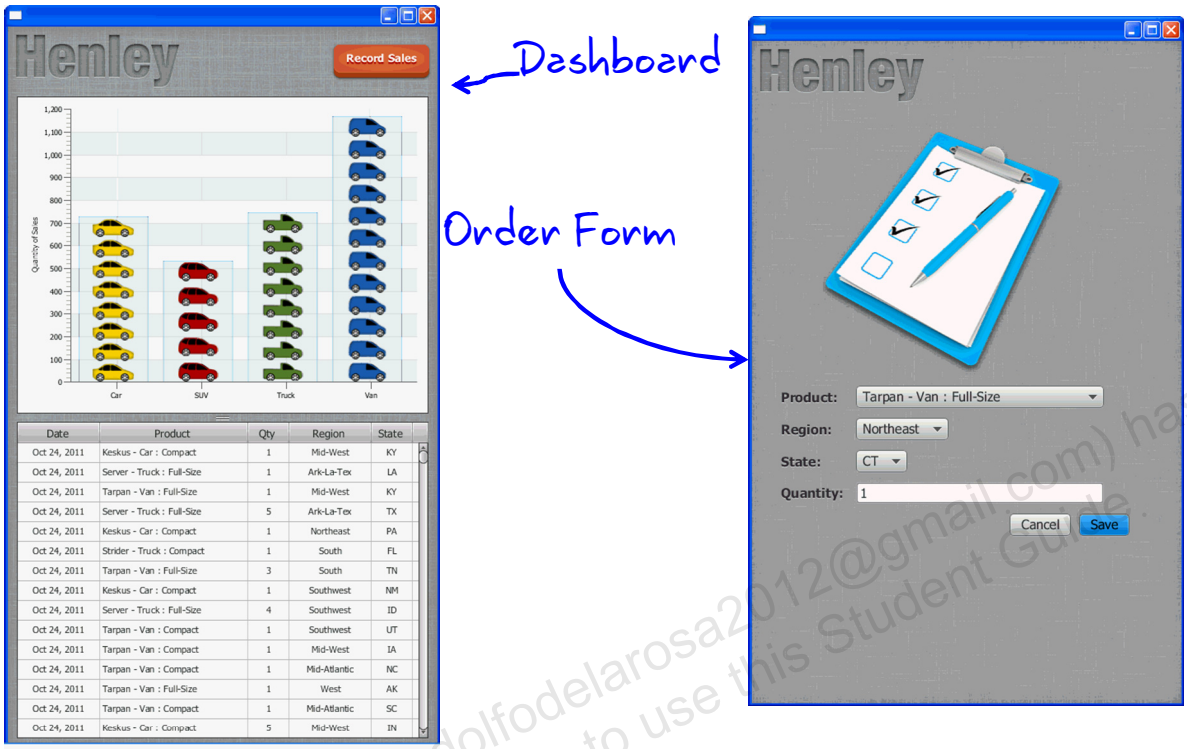
The figure in the slide shows the relationship between the model, server, and client projects. Later in the course, you will learn about multi-tier applications. You get a chance to create a two-tier application similar to the drawing in the slide.

BrokerTool Application Design: Three-Tier



Later in the course, you will learn about multi-tier applications. You get a chance to create a three-tier application similar to the drawing in the slide.

HenleyApp Car Sales Application



The HenleyApp car sales application is an example of an enterprise-level application that will be used throughout this course by the instructor for example purposes. The application includes a main screen that is used as a dashboard to display current car sales and an order form that is used to register car sales. The Henley Car Sales Application includes:

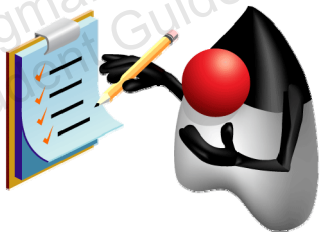
- FXML
- Tables
- Threading
- Layout
- Charts
- CSS
- UI Controls
- Animation
- Custom cells

Sales person can log sales for the product type, region, state, and quantity. The data is logged in the database, and the dashboard reflects the latest sale.

Summary

In this lesson, you should have learned how to:

- Describe an overview of the BrokerTool application
- Explain the problem statement of the BrokerTool project



Lesson 2 Practice Overview: Set up environment and use BrokerTool

This practice covers the following topics:

- 2-1: Set up the development environment
- 2-2: Run the BrokerTool application
- 2-3: Run the JavaFX Ensemble sample application