Stock Management System

Final Project Report

By

MOHAMED NIYAZ

JUDE NISHANTH

DIVYA AVIRINENI

HARITHA MANDANAPU

Under the guidance of

Wensheng Wu, **Ph.D.**

Professor and Associate Chair

Department of Computer Science, College of Computing and Informatics

University of North Carolina at Charlotte

2012

Introduction:

There are already several applications out there that help you find to manage stocks. There are not however, any System that not only helps you find stocks across the globe, but this finds you stock from different location and helps to manage them.

This will be the purpose of the "Managing Stocks" belonging to different locations across the globe. It will be a system that helps you buy by location, sell by location, find by location, Indicate stock by location, monitor favorites by any location, represents stock data's graphical, Stock alerter , instant updates of stock.

"Stock Management System" will leverage existing technologies in order to add the new dimension of Stocking. The creation of "Stock Management System" will involve several developmental tiers: UI, User Management System, Database, and Web Screens.

Project Objectives:

To design and model the domain of an application by appropriately defining the application functionalities. Design and integrate a database system, the schema focusing on the basic functionalities like insertion, deletion and updating the respective records.

Project Implementation:

The project has been implemented using Oracle 11g, Java and Spring frameworks. The project focuses on a stock market paradigm where in, the users can buy or sell stocks.

An assortment of features like listing the stocks on the basis of continent selection, sell/buy the stocks by location, account history, purchase history, monitor the prices of favorite stocks and their graphical representation enable the project to adhere to the project objectives

A batch file has been designed to fetch data from an online website (yahoofinance.com) every 1 minute. The current market price of the selective stocks is obtained from the website and stored in a windows directory. Concurrently, the Oracle database uses this data fetched from the directory to update its records every 30 sec. This dynamic updating of the database enables users to look at the most up-to-date data whenever he wishes to make a transaction.

Project Functionalities:

Dashboard:

The advanced charts are displayed to user using the DMR and fusion charts. Four types of charts are displayed to User namely Pie, bar, doughnut and hierarchy. This has used user efficient interaction on the system.

Login Page:

This page allows users to login to the website before making any transactions. Any user without login credentials is allowed to register to the website.

Registration Page:

This page allows users to register by specifying some basic details like first name, last name, password and thereby redirects to the login page once done.

Home Page:

The home page displays all the favorite stocks of a user. The most updated price of the favorite stocks is fetched from a website and represented graphically (in the form of pie chart, do-nut chart, bar graph and histogram)

Favorites Page:

This page displays a list of all the favorite stocks of a user. By default, only 4 of the favorites can be listed out by any user.

Accounts Page:

The accounts page provides information about the current balance in a user's account. The user is also allowed to add balance into his account whenever desired. Any amount credited by selling any stocks will be added into the user's account.

A user can add multiple accounts into his accounts page. This page also includes an account history page embedded in it which provides information about all the earlier transactions made. This would include the account balance details along with time stamps whenever a stock was sold or brought or when balance is added to the respective user's account (paypal transactions).

Accounts Page:

The page maintains all the transaction details executed by the user , it has details such as transaction type , transaction type and time stamp. This page gives user the most feasible access to interpret the transaction in time. This also helps user to keep record of his every activity.

Purchases Page:

This page would allow any user to buy or sell the stocks. A user can buy up to a maximum of 3 stocks at a time and sell stocks on one by one basis.

Purchase History:

This page displays a list of all the purchases made at any time. All the stocks sold are displayed in this page.

Division of Labor:

Mohamed Zakriea Niyaz

Our project involves building a web base database which acts as a platform to help people to manage their Stock. Niyaz tasks will involve in setting the Oracle 11g Database on Windows server and mount a developer portability tool (either Toad or Pl-sql developer). Will also involve in creation of database schema with abstract privileges along with packages, procedures, triggers ,jobs , DBMS_refresh if needed. He will also take the responsibility of reviewing the codes and evaluating the keys to make sure the Database remains in BCNF.

Role: *Database Admin, Pl-sql developer, Application developer/designer*.

Jude Nishanth Wilfred

The Project is likely to build with the strong analytics functions as a part of web pages which provides all available determination of a stock. This elite functionality on stock are brought to life through spring frame works which serves one among the few prominent frame works in today's Java. Jude tasks involve in setting up the web server which also include creation of web pages with all the functionalities expressed vividly through JSP, HTML, and CSS. Jude Leads the team in assisting the implementation of Java script and J-query for the UI pages and will also involve setting up an API which will be responsible for culminating the data's into the system.

Role: *Web Admin, Web Designer*

Divya Arveni

Stocks refresh periodically where price, volume and respective related characteristics changes instantly . The instant changes are analyzed , validated and executed rigorously through built in Core Java packages. Divya task involves in implementation of instant changes of stock characteristics. This involves building of java files which builds stock information instantly every time when the system get refreshed and also responsible for all the stock characteristics math.

Role: *Java Developer, Java Admin.*

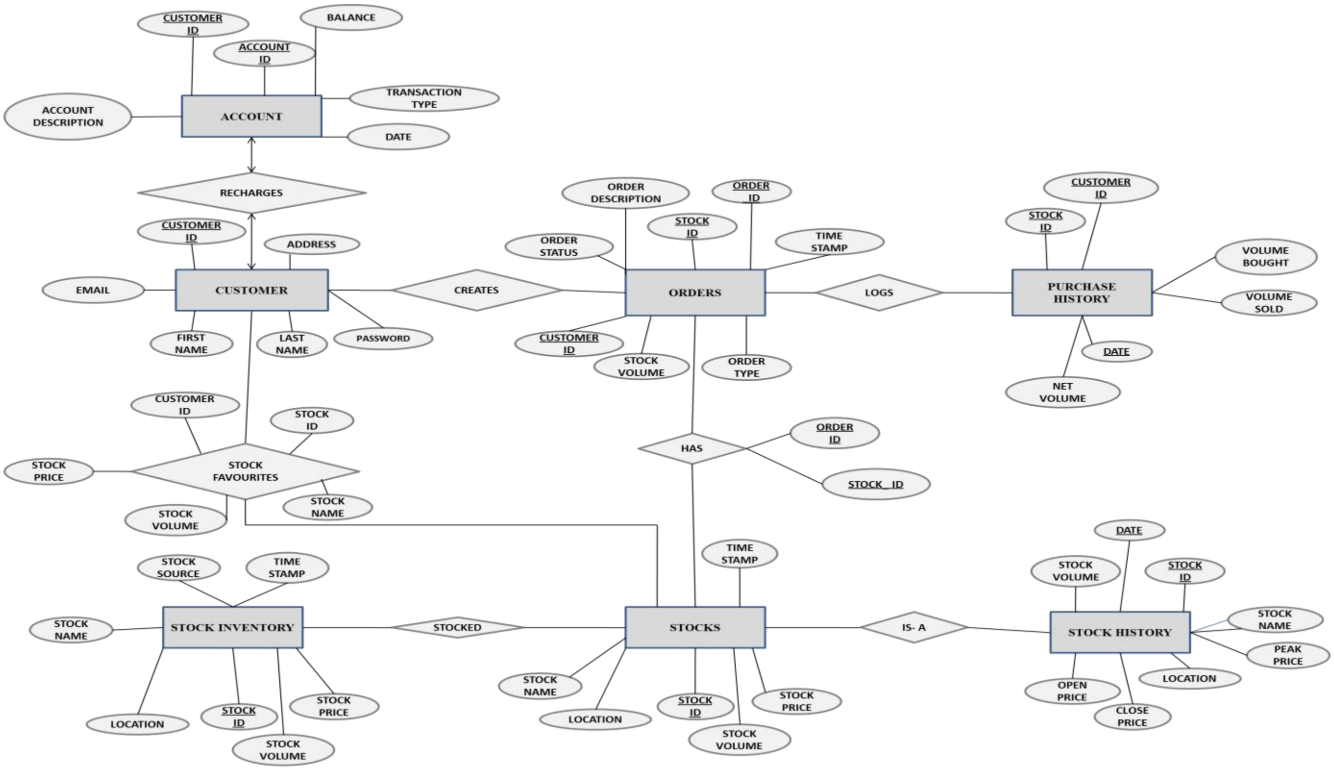
Haritha Mandanapu

A Stock web should have the best GUI only through which the analytics can be understood by the end user. In order to bring best GUI we adapt HTML pages merged with jquery , Haritha task involves is providing the right GUI which makes customer to ease the application without any difficulties. The one of the best GUI practices in JAVA and HTML are brought in the web pages by involving graphs and statics and these are extensively carried out through html and java plug-ins which promote high level simplicity in the application

Role*: Jsp, HTML Admin, HTML designer*.

Design Specifications:

Entity Relation-ship Diagram



Relational Schema

1. ***CUSTOMER***(**CUSTOMER_ID**, PASSWORD ,FIRST NAME,LAST NAME,ADDRESS,EMAIL)
2. ***ACCOUNT***(**ACCOUNT ID**,CUSTOMER_ID,BALANCE,ACCOUNT DESCRIPTION,TRANSACTION TYPE,DATE)
3. ***ORDER***(**ORDER_ID**,CUSTOMER_ID,STOCK_ID,STOCK VOLUME,ORDER DESCRIPTION,ORDER TYPE,ORDER STATUS,DATE)
4. ***STOCK***(**STOCK_ID**,STOCK NAME,STOCK_PRICE,STOCK VOLUME,LOCATION,TIMESTAMP)

5. ***STOCK_HISTORY***(**STOCK_ID**, **DATE,** STOCK NAME,OPEN PRICE,CLOSE PRICE,PEAK PRICE, STOCK VOLUME,LOCATION)
6. ***PURCHASE_HISTORY***(**STOCK_ID** ,**CUSTOMER_ID**, **DATE,** VOLUME BROUGHT,VOLUME SOLD,NET VOLUME,)
7. ***STOCK FAVORITES***(**CUSTOMER_ID** ,**STOCK_ID**,STOCK_NAME,STOCK_PRICE,STOCK VOLUME)
8. ***STOCK INVENTORY***(**STOCK_ID**,STOCK NAME,STOCK_PRICE,LOCATION, WEB_SOURCE,STOCK VOLUME)

Functional Dependencies

1. **CUSTOMER**

   CUSTOMER_ID → PASSWORD, FIRST NAME, LAST NAME, ADDRESS, EMAIL
2. **ACCOUNT**

   ACCOUNT_ID→CUSTOMER_ID, BALANCE, ACCOUNT DESCRIPTION, TRANSACTION TYPE, DATE
3. **ORDER**

   ORDER_ID → CUSTOMER_ID, STOCK_ID, STOCK VOLUME, ORDER DESCRIPTION, ORDER TYPE, ORDER STATUS, DATE
4. **STOCK**

   STOCK_ID→STOCK NAME, STOCK_PRICE, STOCK VOLUME, LOCATION, TIMESTAMP
5. **STOCK_HISTORY**

   STOCK_ID, **DATE**→STOCK NAME, OPEN PRICE, CLOSE PRICE, PEAK PRICE, STOCK VOLUME, LOCATION
6. **PURCHASE _HISTORY**

   STOCK_ID, CUSTOMER_ID, DATE → VOLUME BROUGHT, VOLUME SOLD, NET VOLUME
7. **STOCK FAVOURITES**

   CUSTOMER_ID, STOCK_ID →STOCK_NAME, STOCK_PRICE, STOCK VOLUME

8. **STOCK INVENTORY**

   STOCK_ID → STOCK NAME, STOCK_PRICE, LOCATION, WEB_SOURCE, STOCK VOLUME

The project has been implemented according to the plan and design specifications. No discrepancies were observed during the progress of the project. No changes have been observed or made in the ER diagram and it remained the same along with the functional dependencies.

Software Used:

1. **Java** - Java is an object-oriented programming language developed by Sun Microsystems. Java is a platform-independent, multi-threaded programming environment designed for creating programs and applications for the Internet and Intranets.

2. **JavaScript** - JavaScript is a scripting language developed by Netscape Communications designed for developing client and server Internet applications. Netscape Navigator is designed to interpret JavaScript embedded into Web pages. JavaScript is independent of Sun Microsystem's Java language.

3. **Spring** - Spring is the most popular application development framework for enterprise Java. It is used to create high performing, easily testable, reusable code without any lock-in

4. **Oracle** - The Oracle Database (commonly referred to as Oracle RDBMS or simply as Oracle) is an object-relational database management system (ORDBMS). It stores data logically in the form of table spaces and physically in the form of data files ("data files").

5. **Pl-sql –** This is a Oracle commands which will be used to provide a flexible database activity in order to promote best achievements in obtaining a elegant working system.

Future Scope:

- The project uses one data source to import the data’s but in future we are planning to integrate data’s from multiple systems.
- Updating the pages in constant time.
- Used vivid functionalities such as 3D graphs and instant updates.

Conclusion:

Our application is developed to support stocks from different continents and have a unique system to buy, sell and monitor stock across the globe.