An On-Line Travel Reservation System in Java

Class Project, SWE 645-001, Spring 2006
Instructor: Nick Duan

Objective:

- To understand the component software engineering concepts and to utilize J2EE technologies via the development of a distributed application in J2EE.
- To obtain hands-on experience with both EJB components and web components with a J2EE compliant application server, and to gain skills with real-world software development in a team environment.
- To understand the concepts of component frameworks by designing and creating a customized component frameworks from the ground up using RMI.

Requirement:

1. The project is to implement an on-line travel reservation application with web-based functions for both the travelers and the reservation manager. The customer should be able to register him/herself with the reservation site, enter personal profile information (name, address, email, and credit card information), prepare for itineraries, and book flights. The reservation manager should be able to publish and update flight information, and generate inventory report.

2. The following functionalities should be provided for the reservation manager:

   - A web page for the manager to log in with username and password.

   - One or more web pages for the user to add/cancel/list flights with the following information: airline code/name, flight number, departure location, departure day of the week/time, arrival location, arrival day of the week/time, cost of business class and cost of economy class ticket. The flight number is a 3-digit number, prefixed with 0's if less the actual number is less than 100. For the simplicity reasons, it is predefined that there are 5 seats of business class and 10 seats of economy class seats in each flight.

   - The airline code must be a two-letter code defined in http://www.tvlon.com/resources/airlinecodes.htm. The airport location must be one of airport with a three-letter code defined in http://www.orbitz.com/App/global/airportCodes.jsp.

   - When a flight is canceled in the application, the all itineraries that are reserved and booked will become canceled as well.

   - The inventory report should contain a summary of all fights in the reservation system that still have any unsold seats, the total number of unsold seats of business class and economy class.

   - The user should be able to log out at any stage during his/her session.

3. The following functionalities should be provided for on-line travelers:

   - A web page for a traveler to register him/herself with name, address, username and password, email address, as well as credit card information (optional), including a 16-digit card number and a 4-digit expiration date. As a convenience for this project, a valid combination of credit card number and expiration date is defined in such a way that the 4-digit expiration date is defined with a valid *mmdd* format, and as an integer is a denominator of the 16-digit card number.

   - A web page for a traveler to log in with username and password. Once a traveler is logged in, he/she will be shown with a list of travel itineraries along with the status of each itinerary (reserved, booked, or canceled). A travel itinerary is a travel arrangement with one or more flights. It has the status reserved when it is created by a traveler, booked when it is paid, and canceled when it is deleted by the traveler from his/her itinerary list, or one of the flights in the itinerary becomes canceled by the reservation manager, or payment not received after 2 minutes (simulating the 24 hours holding period in the real world).

   - A traveler is able to create and book an travel itinerary by going through the following steps:

     - Search for flight information by providing departure/arrival date/time and location, number of passengers, one-way or round trip, and the max number of legs.

     - A list of available itinerary options will be shown to the on-line traveler, with departure/arrival and cost information. The departure time will be plus or minus hours within the specified departure time. It is possible that one itinerary contains one or more flights from one or more different airlines.

     - Once the traveler selected an itinerary from the list, he or she has the option to reserve it. Before the traveler can reserve the selected itinerary, he or she must be logged in.

     - Once the traveler reserved an itinerary, he or she has the option to book it by providing payment information via credit card. If the credit card information is not on-file for the traveler, he or she will be prompt to enter the credit card information (number and expiration date).

     - The credit card information has to be validated first before booking. The validation is done via a validation service, which is an EJB application provided by the instructor.

     - Once the credit card is validated, the traveler will be shown with the actual ticket information. The ticket number is automatically generated by the application in the format of XX-FFF-YYYYYY-ZZZ, where XX is the airline code, FF is the 3-digit flight number, YYYYYY is the traveler's

login name, and ZZZ is a 3-digit sequence number generated by the application. The sequence number is unique for each traveler.

- If the credit card number is invalid, a web page with an error message will be displayed to the user and ask the user to re-enter the credit card information.

- A traveler is able to cancel an itinerary from his/her itinerary list if it has not been paid. Cancellation after payment is not permitted.

- The traveler should be log out at any stage during his/her session. If the traveler is logged out during the preparation of an itinerary, the next time when the traveler is logged in again, he or she should be able to continue with the existing preparation process.

4. The application is to be developed in a multi-tier architecture with each tier to be implemented using the technologies and free software defined as follows.

   - The presentation tier is a web application based on Java Servlet/JSP technologies. The JSPs may contain JavaScripts functions for validating input fields with predefined string and/or number format (e.g. credit card number and expiration date). Sun's application server should be used as the servlet engine and the APIs should be compliant with the Java Servlet 2.4 and the JSP 2.0 standard.

   - The business tier of the application contains the business logic and processes with connections to the backend database via JDBC. There are two options for implementing the business tier of the application. The first one is to use the EJB, with Sun's application server as the EJB server. The second one is to use RMI, using the RMI server and utility tools provided by J2SE 5.0. The APIs for the two options shall be compliant with EJB 2.1, and the RMI APIs in J2SE 5.0, respectively.

   - The database tier of the application is a database server used to store all relevant data of the application. The PointBase database, which is bundled with Sun's application server, may be used as the database server in this project. It is a light-weighted database packaged with JDBC driver and equipped with a GUI client. If a student or a project group wishes to select another database for the project, please consult with the instructor first. Detailed information of usage of PointBase is available on the class web site.

   - In addition to the database, the backend of the application also contains a credit card validation server (provided by the instructor). The validation server application is implemented as an EJB application and provided by the instructor for free download. The detailed usage and access APIs will be given later during the class.

5. The purpose of the RMI option is to provide the participants with an opportunity to learn how to design and construct a component-based framework from the ground-up. In other words, a project group or individuals who selected this option is going to be

responsible for implementing both the container/server, and the components for the on-line travel management application. In addition to the business requirements for the application, the following requirements shall be met for the container and component implementation in RMI:

- All components shall be RMI objects and able to be registered with the standard RMI registry (bundled with JDK 1.5).
- The standard XML configuration file (provided by the instructor) shall be used to load the components into container (either statically or dynamically during run-time).
- The standard Jar file shall be used for application packaging and deployment.
- Each component shall be capable to provide session persistency via database or Java serialization. Container managed persistency is not necessary.
- Basic authentication using username/password is required.
- Admin interface for activating, deactivating components is not required.

Due to the challenging tasks and scope, the RMI option is only recommended to students with advanced Java experiences.

Project Team Formation:

The project can be completed individually or by a team of two or three students. Once a project team is formed (by the first deliverable due date), it should not be changed. Each team shall have a team leader whose name and ID are going to be used for identify the team and for submitting deliverables.

If both options of the business tier are implemented in a project, the presentation tier and the database tier are to be implemented only once, and expected to be re-used for both the RMI-based and the EJB-based business tier.

All members of a group will receive the same score for the project after the final deliverable is completed. However, the instructor reserves the right to lower the scores of certain group members if they did not contribute to the project and meet the expectations of the group.

Project Deliverables:

The project is to be delivered in three (3) phases in sequence at different dates with the final phase due on the date of project presentation, May 2, 2006. Each deliverable is an augmentation of the previous one, with the final deliverable containing the final product. Each deliverable contains a standard ear file (EJB option), and/or a standard jar file (RMI option), and two PointBase database files (with the extension of dbn and wal). They shall be copied/ftp-ed to designated directories defined by the instructor. Please read the

following instructions carefully since the project delivery (both content and format) will affect your project scores directly.

File Naming Convention and Delivery Locations:

All deliverable file names shall have the format of *projXXXX.yyy*, where *XXXX* is the last four digits of the lead person's GMU student ID number, and *yyy* is the standard file extensions, ear (EJB application), jar (RMI application), dbn (PointBase data) and wal (PointBase support info).

Two directories will be set up on instructor's home directory on hermes under */home/faculty/nduan*. One is the *earapps* directory, to which all ear files (or jar files in case of option II) shall be delivered. The other is the *projdata* directory, to which all PointBase data files (dbn and wal files) shall be delivered. Once you delivered your files to the designated directories, please change the file permission to 600 (using UNIX command: chmod 600 filename). Note that you may have a quota on hermes. You may have to delete some of your other files before delivering the project files.

Phase I Deliverable (Due on 2/21/2006 at 5 PM):

The deliverable of Phase I is a design document containing the following contents. All HTML pages shall be hyper-linked together for easy navigation. The deliverable shall be bundled in an ear file with the aforementioned file naming convention.

- A cover page containing your name and student ID. In case of a team, clearly indicates the team leader's name and ID number. In addition, it also contains the travel administrator's username and password. The cover page shall be the default index.html page as the starting page for the project, accessible via project URL: http://serverurl:8080/projXXXX, where XXXX is the last four digit of the lead person's SSN.
- Table of contents, with hyperlink to each topic.
- Overview of your application, e.g. what it does and why it is good (be a salesman/woman), and a brief description of the system architecture and interconnection among the tiers.
- Web page design, including a general navigation chart of all pages.
- Database design, with description of each table and an E-R diagram, or a domain diagram to describe the relationships.
- A task list and a brief project schedule, including design, implementation and testing.
- Description of each team member's responsibilities in accordance to the tasks.

Phase II Deliverable (Due on 3/28/2006 at 5 PM):

The deliverable of phase II consists of a design document, an ear file containing only the web components of the application, and database files (dbn and wal files) containing all database tables and necessary data.

The design document based on Phase I shall be augmented with the following contents:

- Design of web components with a brief description of how the web component technologies are used (Servlet, JSPs, JavaScripts).
- Design of EJB components with reasons of why a particular bean technology is used in the application (EJB option only). Describe the Bean interfaces (both Home and Bean) that you intend to create and use.
- Design of RMI-based component framework and individual components (RMI option only). Describe the RMI interfaces you intend to create and use. Describe the directory structure for deployment and the content of your XML-based configuration file. Provide instructions on how to launch the container/server and the registry server.
- Design of database access APIs via DAOs (data access object).

The ear file shall contain only the web components in the form of a war file (a war file contained within the ear file).

- All web pages (html or JSPs) along with Servlet and tag classes including Java source code and optional third party libraries shall be bundled within the war file.
- A hyper-link to the default starting page for customers or administrator shall be made on the cover page or the page of table of contents. Each page shall be navigable via the hyperlinks with possible dummy data displayed to show the look-and-feel of the page.

Phase III Deliverable (Due on 5/2/2005 at 5 PM):

The final deliverable contains the complete application and the final design document. The design document shall be augmented with the following contents:

- Summary of lessons learned and ideas for future enhancement (use your imagination and creativity).
- A complete API documentation of all your Java classes, generated by *javadoc*. It must be accessible from the page of table of contents.

For project teams or individuals doing the EJB option, an ear file with the following contents shall be submitted:

- All EJB components in the form of ejb-jar jar file (ejb-jar within the ear file). It contains all the bean implementations along with source code. The source code (java files) can be located in the same directory as the class files or in a different directory (e.g. src directory). Proper coding and comment standard should be followed.
- The web components delivered in Phase II shall be augmented with APIs to access EJBs.

For project teams or individuals doing the RMI option, a jar file shall be submitted along with the source code. The source code can be located within the same directory or in a separate directory within the jar file. Multi-level jar files are possible. In addition, the XML-based configuration file shall be bundled within the jar file. Additional script files (shell or batch files) for launching the RMI server may be included as well. Instructions for the script file shall be given for the script files.

Configuration Requirements for Phase III Deliverable

Although the configuration parameters are required only for phase III deliverable, it is recommended you use these configuration parameters during your project development.

1. The database connection pool in your application shall be configured with the url *jdbc:pointbase:server://localhost:9092/projxxxx*, where *xxxx* is the last four digit of your or your group leader's G#, and username defined as *projuser*, and password specified as *password*.
2. The JDBC resource name for the connection pool shall be configured as *jdbc/projxxxx*, where *xxxx* is the last 4 digit of your or your group leader's G number.
3. Please use the username *tomcat* and password *tomcat* as the username/password for credit card validation. No need to submit the validation server as part of the deliverable. The validation server will be provided by the demo server and running on local host. Use the usergroup *managergroup* to define the user *tomcat* in your app server.
4. If you are going to use the J2EE security feature to authenticate the travel manager in your demo application, please use the username *manager* with the password *managerpw* for logging in. Configure a usergroup named *managergroup* to include the user *manager* in your app server.
5. Make sure that the java source code of your EJBs and servlets are included in the ejb package, not in the war package of the ear file. You can certainly include the web component source code in the ejb package.
6. If you're doing the RMI option, please contact the instructor for database configuration options.

Project Presentation:

The project demo is going to be on hermes, a UNIX machine of the SWE department, or on a designated machine provided by the instructor. No other machine (personal) is allowed for the demo. The instructor and/or the GTA will be responsible for setting up the demo for each project team or individuals. If doing a group project, at least one person shall be doing the presentation. Each demo shall take about 10 to 15 minutes. A sign-up sheet of the demo will be provided one week prior to the demo in class. A hard copy of the cover page is to be submitted to the instructor before the demo.

Acceptance Policy:

Each deliverable is due at 5:00 PM on the due date. There will be NO exception.

Grading policy:

The entire project takes 40% of the class score. An additional 4% may be given to projects completed with exceptional quality and innovation. The instructor reserves the right to deduct up to 20% for late submissions.

Deliverable Phase	Deliverable Contents	Percentage of project
Phase I	Required Contents of Design Document	10%
	Proper Deliverable Format (ear file) and Process (context root, file name, due time)	2%
Phase II	Required Content of Design Document	3%
	Required Content and Functionalities of Web Components	5%
	Database Files	5%
	Proper Deliverable Format (ear, dbn, war) and Process (context root, file name, due time)	3%
Phase III	Additional Content of Design Document	2%
	Reservation Manager Functions	20%
	Traveler Functions	25%
	Credit Card Validation	5%
	Database Access	5%
	Session Persistency	5%
	Proper Deliverable Format (ear, dbn, wal, jar) and Process (username/password, due time)	5%
Presentation	In class Presentation	5%

Suggested Development Steps:

1. It is recommended that you follow the standard software development process, i.e. from analysis to design, then to implementation and testing. A good start would be trying to understand the application requirement and to layout the web pages and their relationship to servlets/JSPs, EJBs/RMI, and data entities. Don't rush to implementation. A good design would make your implementation much easier.
2. You need to divide the application into modules according to their tiers. Once you have a thorough design completed and the interface among these modules defined, you may proceed to implement and test each module one at a time, without having them interact with each other. Once you have each module fully implemented and tested, you can proceed with the integration. This is especially true if you have three people in a group and you want to have one person responsible for one or two modules. This would be also a good opportunity to develop your teamwork skills.
3. It is recommended to complete your design, and then start implementing your system based on your design (not vise versa). If anything you found out during the implementation stage that something is wrong with your design, go back to your

documentation and correct it before continuing with the implementation. Update your design document as you redesign and code.

4. Before the delivering the final package, please test the ear files and database files on another machine of your own, or simply delete your existing EJB application and database files, then drop the ear file into the *autodeploy* directory, and the database files into the database directory of the application server, and see if everything would deploy and work the same as before.
5. **Please read the entire project description and follow the delivery procedure carefully**. Failure to follow the procedures and use non-standard naming will result in deduction of points.

Software Required:

J2SE 5.0: http://java.sun.com/j2se
J2EE SDK 1.4 (including Sun App Server): http://java.sun.com/j2ee/1.4/download.html