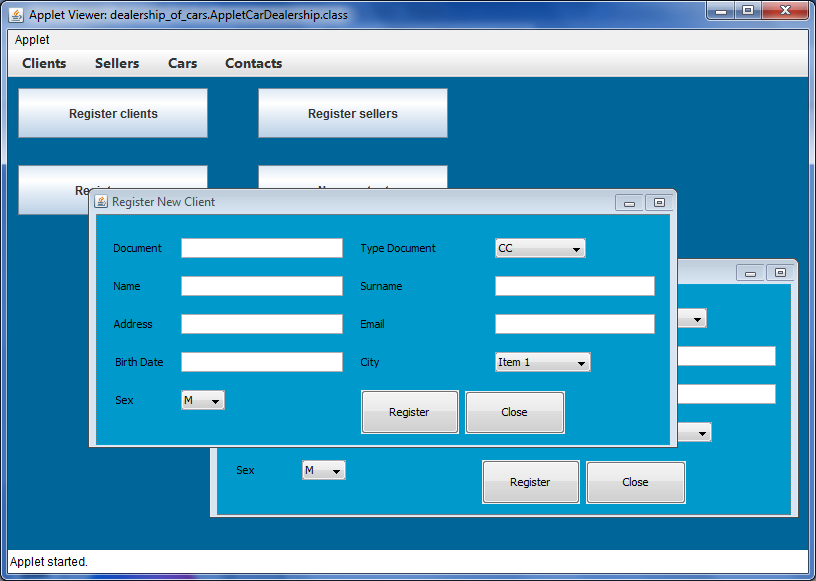
**Car Dealership Management**

Our concept was to create an application that could be used by a car dealership to manage its clients, sellers, and car inventory. The initial idea we had was to make it a Java Applet. We had plans to make it an MDI applet so that it would feel like using a regular desktop application. We did the initial work for that and then started running into problems once we actually would run the applet in a web browser. In the Applet Viewer it would communicate with the MySQL database just fine but in the browser there are apparently built-in protection in the Java Browser Plugin that prevent it from talking to a local database unless the applet is signed, the host machine has a keypair, there is an exception in Windows Firewall, etc etc etc. Since there were just going to be too many headaches we decided to scrap that idea and go with a web application. Here is what the applet looked like though:



The JSF application we ended up making is pretty straight forward. We created multiple pages and use a redirect model to navigate through the pages rather than having them come in through AJAX. We decided to do it that way so the user could make specific bookmarks to the pages they wanted as well as them being able to have multiple tabs open at the same time if they wanted (ie one with the Client List, one with the Car List, etc.).

We used GlassFish and JavaDB for this application and used Subversion for our source control. The “Core” functionality of what we wanted was to be able to have lists of Clients, Cars, Sellers, and then be able to “sell” the cars we had in the database. Additional features we decided to implement was having some error handling, having the tables be limited by pages, and having certain things be protected by a login so that only administrators could access them. As more future features we had ideas about having additional reports with date ranges and sales figures as well as an actual login page in order to even have access to use the application at all with some sort of “rights” system.

To get the application set up you’ll need to open the project itself and then configure GlasshFish with a database and some security settings. Below are the instructions to do that.

Overall we had a lot of fun with this project and both learned a lot. We are both very happy with the final product and the way it works.

**Setup Database**:

1. Follow the same basic steps that are outlined on pg 1279 of Java How To Program to set up a JavaDB database for a web application
2. Use “jdbc/carDealershipJavaDB” as the resource name and “carDealership” as the database name
3. NOTE - After following these steps make sure to go to the database connection pool you created in GlassFish and click the Ping button. Doing this appears to actually create the database (since you entered “;create=true”). If you don’t do this then you will end up getting username/password errors when you try and interact with the application.
4. Run the carDealershipDatabaseCreateSQL.sql file in the root of the project folder to create all the tables and populate them with data

**Setup Security In GlassFish:**

1. Open the GlassFish Admin console
2. Navigate to Configurations ---> server-config ---> Security--> Realms--->Click New...
3. Set values to (without quotes):
   1. Name: “myfilerealm”
   2. Class Name: “com.sun.enterprise.security.auth.realm.file.FileRealm”
   3. JAAS Context: “fileRealm”
   4. Key File: “${com.sun.aas.instanceRoot}/config/keyfile”
   5. Assign Groups: “admin”
4. Click OK button at top right corner
5. Click on myfilerealm---> Click the Manage Users button--->Click New...
6. Set values to (without quotes):
   1. User ID: “testadmin”
   2. Group List: “admin”
   3. New password: “testadmin”
   4. Confirm new password : “testadmin”
7. Click the OK button
8. You are now done with creating the security realm

**Add PrimeFaces JAR into Project:**

1. Right-click on the project and click Properties
2. Under Categories select Libraries
3. Click the Add JAR/Folder button
4. Browse to the location of the “primefaces-3.5.jar” file (should be in the root folder of the project)
5. Click on the JAR file and click the Open button
6. Click the OK button
7. Verify inside the project in the Libraries folder that it lists primefaces-3.5
8. Right-click on the project and click Run
9. The page should now open in your browser

NOTE - If you get errors when you try and interact with the various elements, follow step #3 of setting up the database (Ping the database from the Admin Console)