

Geek Automated Bidding e-System (GABeS)

Phase IV: Completion

1. You are asked to implement a complete Web-based system that supports at least 12 functionalities. **Each member is responsible for a total of 4 functionalities which will form the basis of your grade on this phase.**
2. Design a complete Web-based user interface for your system BUT ONLY for the following system components
 - **LOGIN** for both Administrators and customers each of which must display the proper menu afterwards (Figures A1, A2, C1 and C2) --- **1 FUNCTIONALITY: login**
 - **Administrator REPORTS** (Figures A4 & A5) --- **2 FUNCTIONALITIES: view report 1, view report 2**
 - **UPDATE PROFILE** for customers (Figures C3) --- **1 FUNCTIONALITY: update profile**
 - **SELLING MANAGEMENT** (Figures C4, C5, C6, C7, in addition to Add Item) --- **4 FUNCTIONALITIES: list my items, show item info, show list of bidders, add new item**
 - **BIDDING MANAGEMENT** (Figures C8, C9, C10, C11, C6, & C12) --- **3 NEW FUNCTIONALITIES (4 in total): search for items, show item info (SAME ONE FROM SELLING MANAGEMENT), bid on an item, show items bid on**
 - **VIEW MY FEEDBACK: 1 FUNCTIONALITY: View my feedback**

PS: Some of the missing components (such as adding users, leaving feedback, etc.) results in lack of important database data (such as customers as well as feedback ratings in the previous example); as a result, please make sure you manually enter such missing data in your database.
3. Create all needed (fully documented & tested) Javabeans classes (one per table and one per functionality etc.) for components listed in above. NB: You are expected to use **PreparedStatement**s and **CallableStatement**s as often as possible
4. Connect your HTML Web-based user interface to the Javabeans using JSP and test your system
5. What to include in your last report:
 - **Section 1 (PHASE SUMMARY):** DETAILED Task decomposition (i.e., who did what especially the functionalities completed by each member and meeting minutes. Also include the URL where I can run and test your system on tomcat ... more on this later on.
 - **Section 2 (REFLECTION):** Reflect back on the set of requirements that you started with and figure out
 - What you've completed as required
 - What you've modified and completed (and briefly state why you modified them)
 - What you've omitted (and briefly state why you dropped them)
 - What you've added and completed (and briefly state why you added them)
 - **Section 3 (ARGUMENT):** Tell me why you deserve a B or more (see note below).
6. Finally, copy your final project folder and paste into a folder `/usr/people/handins/CS331`. Make sure your submitted folder includes all classes (i.e. **.java** not just **.class** files). Include report in submitted folder.

IMPORTANT POINTS

A working project that satisfies ALL requirements above and includes a complete report (as described above) will earn your team a grade in the C range.

To earn a B, you're expected to complete two of the following in addition to the above (you're welcome to suggest others as well but please run them by me first); completing three will increase your grade to an A.

- **NEW: Normalization analysis: for each table, derive a minimal set of FDs, depict FDs in a dependency diagram and derive the normal form. Be complete in your analysis and explanations.**
- **3 GABeS features not required herein (but discussed in the description document)**
- **2 additional (useful) GABeS features of your own choosing (not discussed in the description document)**
- **A highly sophisticated Web interface with CSS, JavaScript, etc...**

- Heavy reliance (significantly more than required in phase III; you need to make your case) on SQL stored procedures, functions, triggers, views, etc. instead of dynamic SQL code
- Using (at least 4) extra Oracle or JSP features not covered in class. E.g.: Oracle events, useful JSP predefined tags, JSP user-defined tags, etc.
- Adequate error handling & security checks (e.g., a user shouldn't be able to visit some pages without being logged in)

Submit to the N: drive a folder containing your report and all of your code --- DUE BY CLASS TIME ON TUESDAY 12/13

When project is complete, please follow steps below to deploy it (STILL NOT FINALIZED)

1. Have any of the group members launch the Eclipse IDE (only one of you, please). Open the project and right-click on your Java classes package under **src** (if you have multiple packages, please repeat the following for each package)
 - a. Select **Refactor > Rename**
 - b. Make sure all check boxes are selected
 - c. Press **Preview >** and then **OK**
 - d. For the **New name** please prefix the current package name with **your_actual_linux_username**.: e.g., if the package name is **productdeals** it now becomes **irahal.productsdeals** for me
 - e. Save your work (if need be) and close Eclipse
 - f. Launch Eclipse again to test that your project still works
 - g. Close Eclipse when done
2. Deploying the Web pages:
 - a. Open folder **java-docs** on your home; create a subfolder for your project called **gabes**. Keep this folder open.
 - b. In a separate window, browse to the project folder in your workspace (where Eclipse stores projects) and open the **WebContent** folder
 - c. Copy all the files and folders there (except for **META-INF** & **WEB-INF** folders) into your **~/java-docs/gabes** folder created in **2.a**
3. Deploying source code java classes (JavaBeans):
 - a. Open folder **java-docs/beans/your_actual_linux_username** on your home; please note that folder **java-docs/beans/your_actual_linux_username** NEVER BE DELETED OR REPLACED
 - b. In a separate window, browse to the project folder in your workspace (where Eclipse stores projects) and open the **src/your_actual_linux_username** folder
 - c. Copy all the subfolders and files from **src/your_actual_linux_username** into your **~/java-docs/beans/your_actual_linux_username** folder
 - d. Using a simple Java IDE like DrJava, open and compile your java classes in **~/java-docs/beans**
4. Changing permissions:
 - a. Open a Linux terminal and browse to folder **~/java-docs** (i.e. **cd ~/java-docs**)
 - b. Change permissions using command: **chmod -R 755 ***
5. Running project from a browser:
 - a. Open a browser (on Windows or Linux) --- FROM A CAMPUS MACHINE
 - b. Go to following URL:
http://tomcat.csbsju.edu/csci331/your_actual_linux_username/gabes/ (that would be <http://tomcat.csbsju.edu/csci331/irahal/gabes/> for me)
 - c. Your project should work fine now