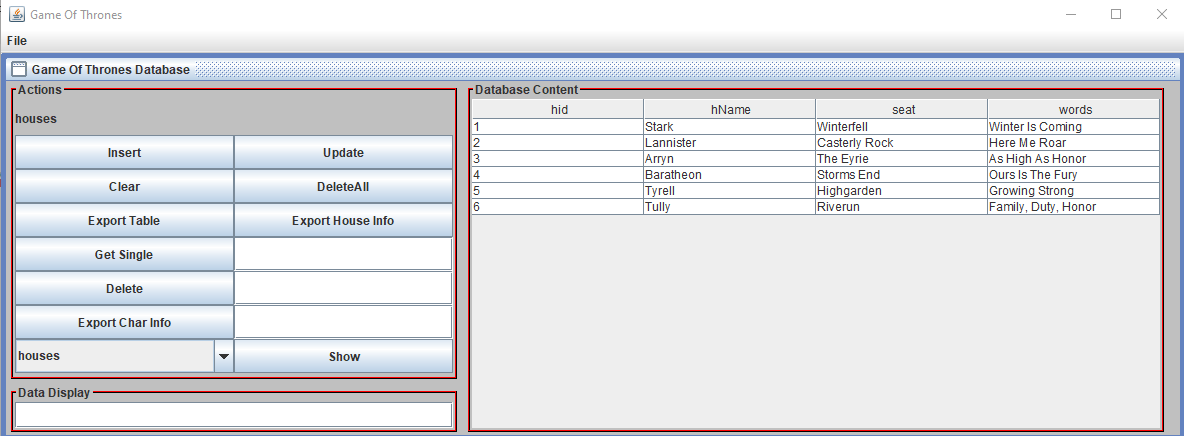
**Distributed Systems Assignment 2022**

**Oisin Anderson – A00260168**

**Project Overview**

This project displays a java window to manage the storage of database information. The database stores information in three tables which are houses, characters and forts. These tables are all HSQLDB tables, and each contains four columns. The House table connects to both the Characters and Forts table by having a foreign key (hid) in both of those tables. The data is retrieved and sent to that database using http requests and the data that is collected in the response is parsed using the XMLPullParser. The server the project is ran on is Apache Tomcat 7.0 and uses JAX:RS api to access the database.

**Main Gui**

****

This is the main window for my application. On the left it contains all the action buttons and a display text field to show messages on, then on the right the table data for the table selected is displayed. Each of the buttons and functions are outlined below.

**Main Functions**

Insert – When pressed a pop up appears where the user can insert data into the database table shown currently.

Update – When pressed a pop up appears where the user can update a row in the database table shown currently.

Clear – When pressed this clears all Texfields on the main window.

Delete All – When pressed this deletes all rows from the database table shown currently.

Export Table – When pressed this exports the database table shown currently into a csv file.

Export House Info – When pressed this exports data from all three database tables into a csv file.

Get Single – Returns a single row of from a database table by entering an id and is displayed in data display.

Delete – When pressed this deletes a row from the database table shown currently.

Export Char Info – When pressed this exports data about the characters from the house specified into a csv file.

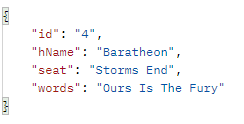
Combo Box – When the show button is pressed depending on the option in the combo box selected a table is displayed on the main window.

Get All – Used to load the table and does this by getting all rows from a database table. This is done automatically and refreshes after and insert, delete or update.

**Requirements**

1. Build a client application that sends all of the HTTP requests GET/PUT/POST/DELETE.
2. Build a server application using tomcat server, that responds to all of the HTTP requests GET/PUT/POST/DELETE"
3. The client application will parses the response using XMLPullParser and outputs to the GUI" + "A tomcat server that responds to all of the HTTP requests GET/PUT/POST/DELETE"
4. The data in the response will be taken from an HSQLDB database

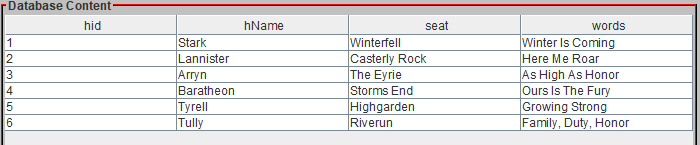
This application follows all of the requirements outlined above. The main gui seen above allows the user to have access to the HSQLDB database. The gui acts as the client and when buttons are pressed it activates a function that sends a http request to the server to send or retrieve data.

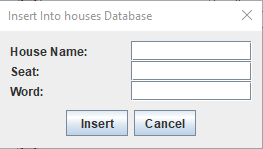


When the request is sent to the server responses are sent back using the JAX RS api to access the database. Using the single get as an example the image above is what the server returns, I enter the digit 4 in the text field and click the Get Single button. It returns a response in the form of xml which later has to be parsed using the XMLPullParser. The XMLPullParser is used to loop through the xml tags to find the data that is stored between the opening and closing tags and display that data on the main gui. Below an image of the Get Single display is shown.

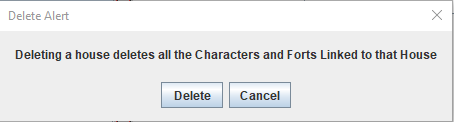


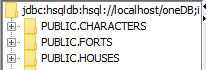
The Get All is used to load the tables and does this by requesting the data from the server which returns a response. This response is parsed using the XMLPullParser and display in a table on the main gui as seen below. This get All function is called every time the database has been edited as to keep showing the user the latest data available.



The put and post command works similar in the sense that when there buttons are pressed update and insert, a pop up appears allowing the user to enter the data into the database or cancel as can be seen below. Both of the functions send the data to the server as form data. This data is received as a form param and either inserted into the database table or updates it. 

Delete when pressed send a delete request to the server which in turn deletes the specified item or all the items from a table if Delete All is pressed. If deleting from the house database a pop up appears telling the user that removing a house from that table will cause any forts or characters in there tables that are linked to the house deleted to be deletd also. If the user selects delete it will continue but cancel will stop the delete as seen below.





When data is requested from the server it accesses the HSQLDB database. Above is an image of the three databases in the HSQLDB manager.

1. To access this data a select command is used.



1. To Insert new data an insert command is used.



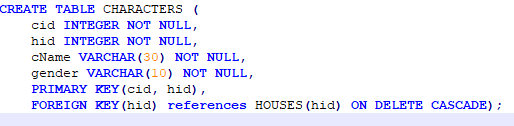
1. To update data an update command is used.



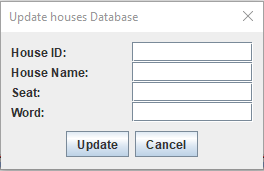
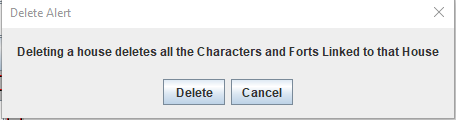
1. To delete data a delete command is used.



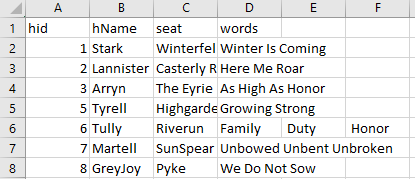
**Extra Features**



Cascade delete) When deleting from the house database any rows in the forts or characters database that were linked to that deleted row would need to be deleted. I achieved this by using a cascade delete that automatically deletes the row of a foreign key in a table if the foreign key no longer exists.

Alerts) Is used alerts in my project to simplify the look of the main gui as I already have a lot of buttons on the window. An insert and update alert pop up when those buttons are pressed and then the inputs into those Texfields are taken and sent to the server to be processed. The delete alerts are used to warn the user that when they delete a house, they will delete all forts and characters linked to that house.



Export Functions) The export table function exports all the table from the table currently on display into a csv file which can be viewed by the user. Another export function is the Char Info function. This function takes in the id entered by the user and uses and joins together data from both the houses and characters tables that the house id entered. The house info function is similar except it exports data to a csv file from all three of the tables.



Message Box) This box displays messages when actions are performed in the main gui. When data is retrieved and sent from and to the server this box is updated and shows the user what just happened.



Three tables) I decided to have three database tables in my project. This added to the complexity of the project as I had to created separate DAO and Resource files for each table. The user can choose which table they wish to view by selected an option from the combo box and pressing the show button. When this happens, the table is refreshed showing the data from the table the user selected.