Design document

analysis of design choices

the MySQL database created for this CA contains a unique id, as well as appropriately named table elements. It would be possible to normalise the example data given, which would mean creating more tables to reduce the amount of redundant information stored within it.
I have decided not normalise this database as the sample data and average use case is sufficiently small enough that no huge advantage in performance or efficiency would be brought about from normalisation in this case. For scaling this project, I would modify the data to third normal form so that the company names are not repeated.

The SQL statements used to access the database are mostly stored in their own separate PHP files which allows for easier updating and modification. If this project is to be developed further and store potentially sensitive information, it would be necessary to protect against SQL injections. One way the security of this database could be further improved would be through the use of prepared statements. The SQL instructions are passed by use of HTML forms and passwords are stored in a hashed format which could be extended to include usernames for extra security.

Access to the catalogue was originally controlled by an initial login page which denied entry without registering a username and password. By requiring the user to register it is hoped that any malicious data entry could be traced to the source and handled by the proper authorities as required. Another considered design was to allow limited access to unregistered users which would include browsing but not database manipulation privileges.

The HTML was mostly used for forms and including CSS options. JavaScript forms were researched after most of the project had been developed, however these features are increasingly found to not be active on users machines. PHP was used throughout the development as the main focus as its use complemented previous experience gained in computer programming.

As with the SQL statements, many functions or pages were placed in their own individual files which is useful in a modular approach to implementing future features. The organisation of the files could be improved by creating many subdirectories, however the reasoning used for placing the majority of files in a single directory was based on the sometimes slow speed of the recommend hardware for this project.

Files used

add_ship
html and PHP is used to display a form to take information from a user , as well as to send a SQL statement to add the information to the database. Logical checks are performed to make sure all possible fields are filled in to stop hanging entries for normalisation as well as guarding against errors caused by missing data.

Builder_list
see company list

Check
ensures the session is being conducted by a registered user. Will redirected a non-signed in session to the login and register page. Used on accessible pages as a security feature.

Company_list
a PHP file searches the database for companies and adds the returned results to a HTML drop down menu. Used to search by company name as an alternative to expecting the user to already know which company they are searching for and then typing it in manually.

Connection
an important file which is used to connect and verify connection to the database. Stored as a separate file as is frequently used by many final files as well as test cases.

Delete_check
this file is a simple html page which is used as a buffer to protect against the user accidentally clicking on and deleting a ship, especially if accessed on a mobile device. Called only from an attempt to delete an entry

delete_ship
Contains the SQL command for deleting an entry from the table

edit_ship
this file creates a web form and also handles the processing of said form to adjust pre-existing entries. The form will update as long as id and at least one of entry is modified.

Error_check
used right at the beginning of development for its verbose output until more familiarity with PHP was developed as well as the discovery of online tools

index
this is considered the home page as it is the first page a user will see when logging in, as well as containing most of the logic which was focused on during this CA. The CSS style was based on this page and enhanced for the others.

Landing
this page was originally chosen to be the home page however a summary of the ships was required by the brief to be accessed on a separate page. An important aspect of this PHP file is that it can create links to ships pages without having the information hard coded. This is close to replicating the use of technologies such as AJAX and would be a prominent feature in a more rapidly changing environment.

Login
handles initiating the session of a pre-existing user and also the necessary redirects, and destruction of sessions.

Logout
a simple yet important PHP file which simply ends the users session. Chosen to be a separate file in the case of future updates requiring more features to sessions.

Manageusers
contains the functions required to add and remove users to the database

register
sends username and password to be validated

return_builder/capacity/company/names
a PHP file which accesses the database using SQL and then parses html to display the respective results.

Ship_detail
presents to the user all the information held on an individual ship. This page can be accessed by clicking on the name or picture of a ship anywhere on the site.

style_sheet
contains CSS to style the webpage.