Deliverable Description and Explanations:

All methods written have their http handlers in UBCSubletAPI.js

Deliverable 1:

In our first deliverable we were required to identify the types of users that our application would use. We identified a subletter and a sublettee, since this application was created to facilitate subletting rooms at UBC. Our code for these is located in the UsersManager.js file in src. Here we allow users to sign up and log in. All subletters and sublettees are users, these categories aren’t actually in the code, but rather the way one approaches the application.

Deliverable 2:

Our second deliverable was to create the Create Post method. Our code for this is located in the PostsManager.js file in src. This method simply inserts a new post offering into the database in the Posts table.

Deliverable 3:

Here, we created the delete post method. It simply deletes an existing post from the Posts table. The code is in PostsManager.js file.

Deliverable 4:

Deliverable 4 handled editing a post in the Posts table. The code for this is in the PostsManager.js file.

Deliverable 5:

Since our application allows you to filter through posts to find the residence, room type, etc you’re looking for. To achieve this we join the tables: Rooms, Residences, and UnitTypes to display the complete virtual post table, which would be used to run other queries on it in order to filter out the appropriate posts. Located in PostManager.js

Deliverable 6:

This deliverable handles how the client pulls the data from the database for post creation. UnitTypes and Residences have limited options, so we join the residences and UnitTypes tables to give that data to the client when the user is creating his or her post.

Deliverable 7:

For this deliverable, we use a GROUP BY query in order to determine how many posts filter out according to the subletee’s preferred criteria by using COUNT. The count will be applied to each field the user decides to filter. Then, the UI will display the posts that fall under the criteria, as well as the total number of posts that are filtered out.

Deliverable 8:

Deliverable 8 simply handles user log in. User log in is used in every http post request since we have to authorize before accessing user info. The code for this is in UsersManager.js.

Deliverable 9:

Deliverable 9 was about returning post information to the client. The code for getting a posts from the Posts table is in PostsManager.js.

Deliverable 10:

Deliverable 10 changed from just creating sublettee info to also editing it. Users can create sublettee information to tell subletters (post creators) their names, and some information about them such as where and how they can be contacted. The code for this is in UsersManager.js.

Deliverable 11:

This deliverable was concerned with returning filtered data to the user. When the user searches for posts with filters, they are shown only the posts that meet their criteria, so we join tables to display that to the user. The code for this is in PostsManager.js.