

## Refactoring Document, Iteration 3:

Old Design/Problem	<ul style="list-style-type: none"><li>- Study area format was not ideal</li><li>- Caused click offset on map as well</li></ul>
Refactoring Done	<ul style="list-style-type: none"><li>- Updated map and study area visuals</li><li>- Removed old design elements</li></ul>
New Design	<ul style="list-style-type: none"><li>- Much better user experience</li><li>- Fixed offset issue</li></ul>

Old Design/Problem	<ul style="list-style-type: none"><li>- Getting directions was unintuitive</li><li>- Didn't work with real database</li></ul>
Refactoring Done	<ul style="list-style-type: none"><li>- Implemented visual response</li><li>- Synced stub and database info</li></ul>
New Design	<ul style="list-style-type: none"><li>- Intuitive directions</li></ul>

Old Design/Problem	<ul style="list-style-type: none"><li>- If the API request fails, there is no error handling, which could cause the entire function to break.</li></ul>
Refactoring Done	<ul style="list-style-type: none"><li>- Wrap the fetch call inside a try-catch block.</li></ul>
New Design	<ul style="list-style-type: none"><li>- Avoid function to break</li></ul>

Old Design/Problem	<ul style="list-style-type: none"><li>- Lack of type safety for property "map". The map property is typed as any, which reduces type safety.</li></ul>
Refactoring Done	<ul style="list-style-type: none"><li>- Use mapboxgl.Map as the property's type instead: map!: mapboxgl.Map;</li></ul>
New Design	<ul style="list-style-type: none"><li>- Increase type safety</li></ul>

Old Design/Problem	<ul style="list-style-type: none"><li>- Incorrect RouterLink usage applied to the button. The &lt;button&gt; element</li></ul>
--------------------	--

	doesn't support routerLink directly.
Refactoring Done	<ul style="list-style-type: none"> <li>- Since we do not need routerLink here, we can remove it altogether.</li> </ul>
New Design	<ul style="list-style-type: none"> <li>- Will support routerLink directly</li> </ul>

Old Design/Problem	<ul style="list-style-type: none"> <li>- Opening hours filter does not provide any meaningful details for the user and may be unnecessary.</li> </ul>
Refactoring Done	<ul style="list-style-type: none"> <li>- Include more switch cases in filterByCategory() to include Morning, Afternoon and evening hours.</li> </ul>
New Design	<ul style="list-style-type: none"> <li>- Intuitive for user</li> <li>- Remove unnecessary code</li> </ul>

Old Design/Problem	<ul style="list-style-type: none"> <li>- The filtering logic for searching a study area filterStudyAreas() and filtering a study area based on their categories filterByCategory(category: filterCaterogy)have similar logic (duplicate code).</li> </ul>
Refactoring Done	<ul style="list-style-type: none"> <li>- Combine the filtering function (filterByCategory() method) and the search logic (filterStudyAreas() method) into one function</li> </ul>
New Design	<ul style="list-style-type: none"> <li>- Eliminate redundancy</li> </ul>

Old Design/Problem	<ul style="list-style-type: none"> <li>- The rating system for loudness and cleanliness is not well-defined or explained within the code or the GUI.</li> </ul>
Refactoring Done	<ul style="list-style-type: none"> <li>- Add a label with description in the GUI to explain how the loudness and cleanliness rating are evaluated.</li> </ul>

New Design	- More user friendly and clear
------------	--------------------------------

Old Design/Problem	- Lack of error handling in StudyArea(String name, boolean chargingOutlets, int cleanlinessRating, boolean accessible, int loudness, Location location, int openingTime, int closingTime, int business)
Refactoring Done	- Add an IllegalArgumentException handling for busyness that checks for inputs that are greater than 5 and less than 1.
New Design	- Handle exceptions correctly

Old Design/Problem	- Unclear method naming for addBusynessRating. Without context, the method name can be confusing. For example, it could mean it is adding a busyness rating to something or adding the rating number to the busyness rating.
Refactoring Done	- A method name like "incrementTotalBusynessRating" clarifies what the method does.
New Design	- Adds clarity

Old Design/Problem	- Using both "Busyness" and "Business" is confusing and can cause a developer to think they represent and refer to different things in the code. Also, "Business" doesn't mean busy.
Refactoring Done	- Change every use of "Business" to "Busyness".
New Design	- Eliminates redundancy and confusion

Old Design/Problem	<ul style="list-style-type: none"><li>- Login Page shows Old Map design in background</li></ul>
Refactoring Done	<ul style="list-style-type: none"><li>- Changes map image of both register and login page</li></ul>
New Design	<ul style="list-style-type: none"><li>- Consistency</li></ul>