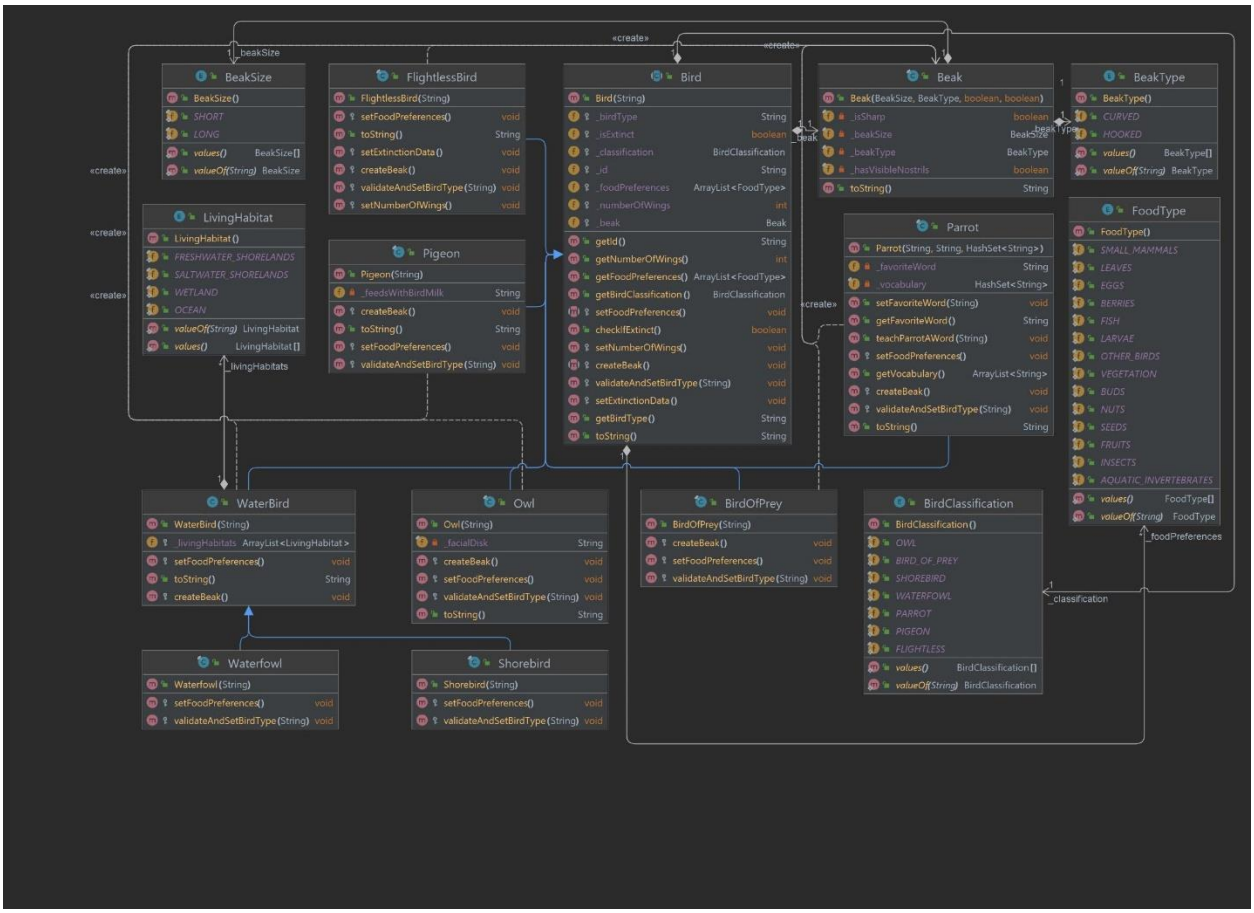
Alexander Khoperia

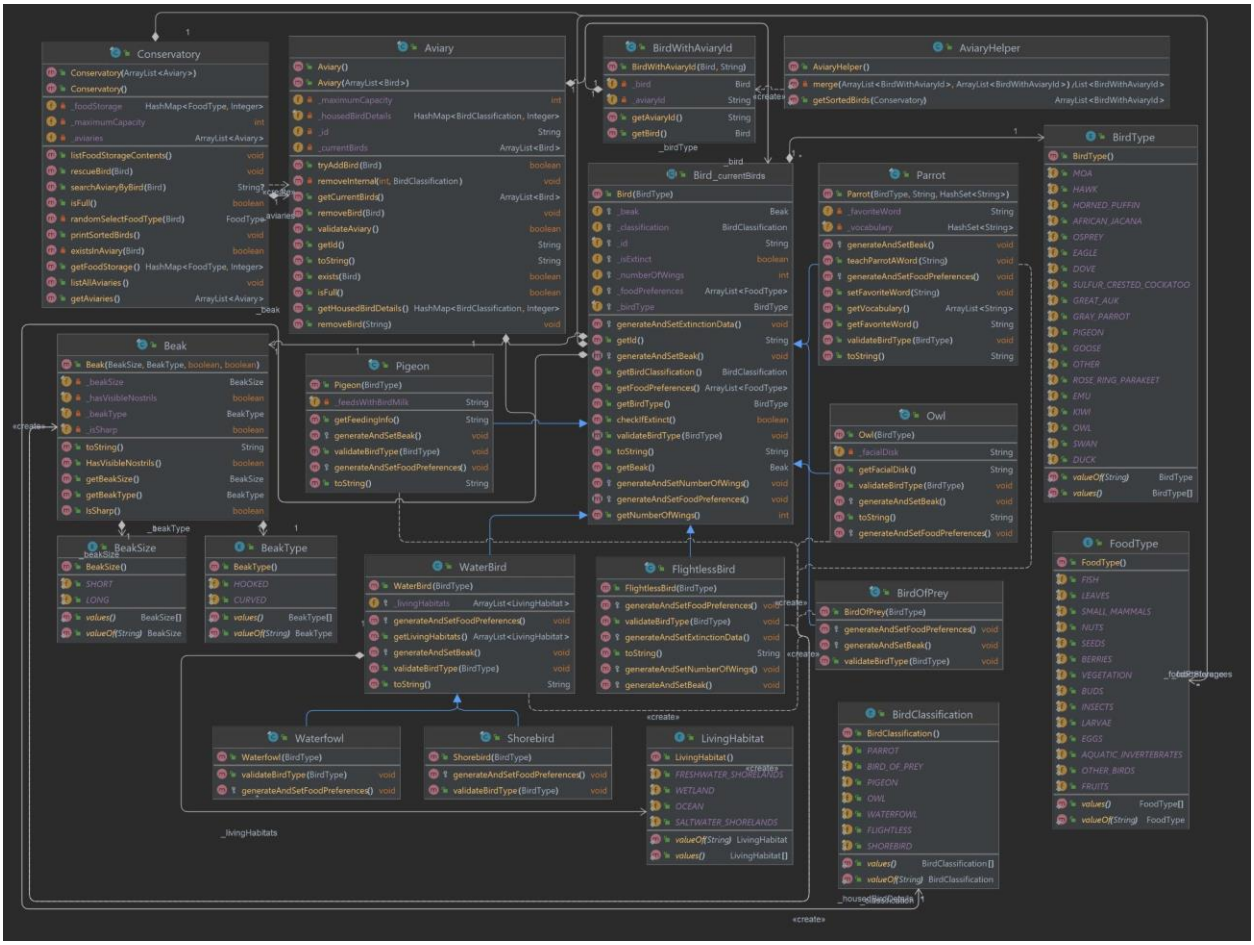
CS 5010

Lab 1: Birds

Original UML Diagram:



Final UML Diagram:



Testing Strategy:

My testing strategy can be categorized into three groups: Testing correct input, testing incorrect input, and testing runtime results of the functionality.

1) Testing correct input mainly will consist of creating the objects of each bird correctly. Namely, gray parrot which has a special favorite word or owl with special facial disk feature. Test cases will target these types of cases where the input is correct and therefore, output should also be correct. Also, these birds have many unique features that is only shared among their types, there is a lot of test cases that can target each of those features. Namely, eagle must have two wings and should have predatory food preferences, etc. This will ensure that all objects are created correctly.
2) Another big aspect of testing this design is to create test cases for incorrect inputs. What if I create a parrot object with the bird type of a goose? Clearly such cases should throw exception, and the job for test cases would be to cover specifically wrong input cases to ensure that no such case can exist within the running code. This will ensure that no incorrect object can be created to avoid further bugs in the functionality.

3) Another case is to test the functionality of the program. I intend to test the various methods that this design will implement. Namely, teaching parrot a word, what if parrot already knows the word? Then it should not be added to its vocabulary. Also, the huge aspect of this project will be to test the functionality of Aviaries and the Conservatoire. For instance, rescuing a bird, what if that all aviaries are full? Or what would happen if conservatoire cannot house a particular bird because of the bird type conflict? Also, there is a lookup and sorting of birds involved in this lab. Those functionalities need a proper testing as well. Therefore, several cases need to be implemented to cover the functionality aspect of the project as well.

All in all, I believe that these three types of testing will ensure that no correct input produces incorrect output, no incorrect input produces correct output, and the methods function as intended to create a bug-free program.