Project Part 3

Part 1 - Code Refactoring

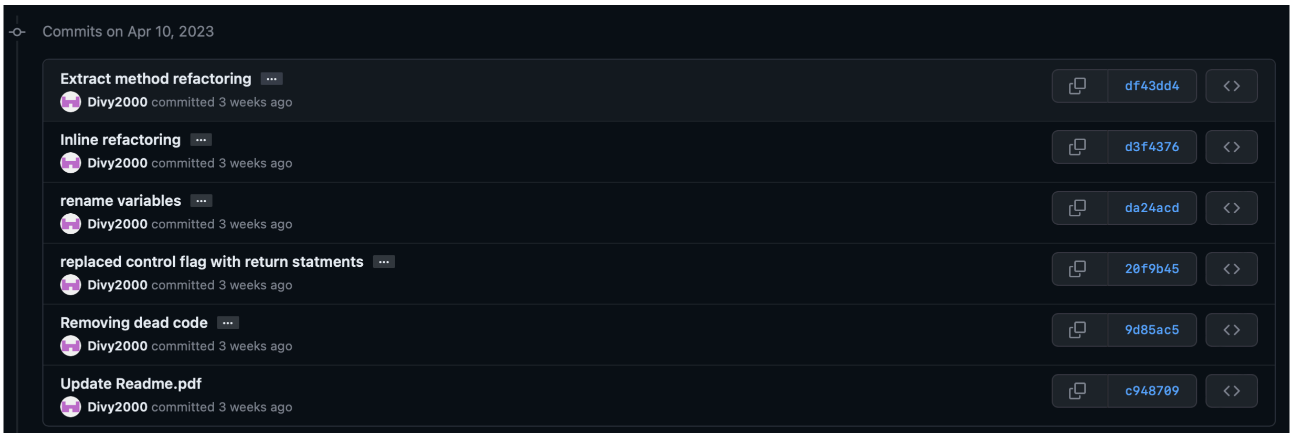
1st Commit
Extract Method Refactoring: https://github.com/Divy2000/CSE464_2023_dpate164/commit/df43dd44ff8433b66ca8657331e6dc46126e20eb

2nd Commit
Inline Refactoring: https://github.com/Divy2000/CSE464_2023_dpate164/commit/d3f43769b68d95cb1adb44e9cc0315068fc2512b

3rd Commit
Renaming Variables: https://github.com/Divy2000/CSE464_2023_dpate164/commit/da24acdaf5b617fd184807aed35d445859a9f898

4th Commit
Replaced control flag with return statement: https://github.com/Divy2000/CSE464_2023_dpate164/commit/20f9b45535ba22629360444e9d89ee9c0f0cb905

5th Commit
Removing the dead code: https://github.com/Divy2000/CSE464_2023_dpate164/commit/9d85ac57609bcbfa33e74323049b1a3b75d63fd8


Commits on Apr 10, 2023

Extract method refactoring ···
Divy2000 committed 3 weeks ago — df43dd4

Inline refactoring ···
Divy2000 committed 3 weeks ago — d3f4376

rename variables ···
Divy2000 committed 3 weeks ago — da24acd

replaced control flag with return statments ···
Divy2000 committed 3 weeks ago — 20f9b45

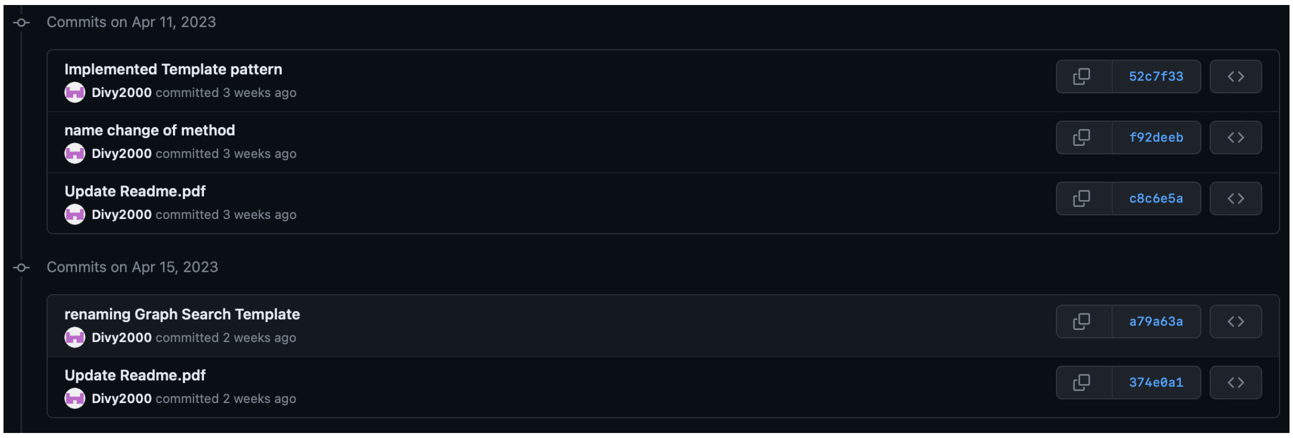
Removing dead code ···
Divy2000 committed 3 weeks ago — 9d85ac5

Update Readme.pdf
Divy2000 committed 3 weeks ago — c948709

Part 2 - Template Design Pattern

- In implementing template design pattern, I have extracted the common steps in "**src/main/application/GraphSearch.java**".
- Both BFS and DFS have only one difference, BFS uses Queue while DFS uses stack.
- Therefor I have implemented everything in GraphSearch.java except the part where we add and remove from the queue/stack and the part where we check if queue/stack is empty.
- I have made 3 different abstract methods for that part :-
  - addToDataStructure
  - removeFromDataStructure
  - ifEmpty **(Note: I changed the method name to isEmpty in next commit)**
- In BFS class I declare a Queue and implement abstract methods according to that.
- In DFS class I declare a Stack and implement abstract methods according to that.

Commit Link: https://github.com/Divy2000/CSE464_2023_dpate164/commit/a79a63ad9249bf99742825b2c7acda88e04ec72f


Commits on Apr 11, 2023

Implemented Template pattern
Divy2000 committed 3 weeks ago — 52c7f33

name change of method
Divy2000 committed 3 weeks ago — f92deeb

Update Readme.pdf
Divy2000 committed 3 weeks ago — c8c6e5a

Commits on Apr 15, 2023

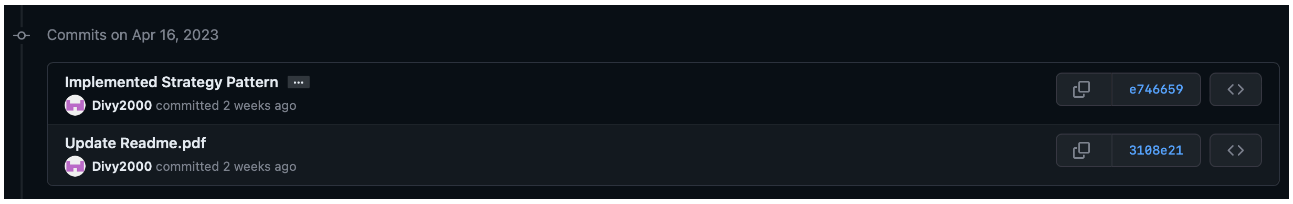
renaming Graph Search Template
Divy2000 committed 2 weeks ago — a79a63a

Update Readme.pdf
Divy2000 committed 2 weeks ago — 374e0a1

Part 3 - Strategy Design Pattern

- For this implementation I have created an abstract class GraphSearch_Stragey, which takes MutableGraph and GraphSearch_interface as its input to constructor.
- GraphSearch_interface have 3 methods, addToDataStructure, removeFromDataStructure and isEmpty.
- Classes BFS and DFS implements these methods.
- The GraphSearch class extends the abstract class of GraphSearch_Stragey and calls the super constructor.
- In GraphSearch method in muGraphClass, I am creating objects of the type of search and passing them to constructor of GraphSearch class.

Commit link: https://github.com/Divy2000/CSE464_2023_dpate164/commit/e746659a58d4015f3e24ef76d6fa79e427f70aab


Commits on Apr 16, 2023

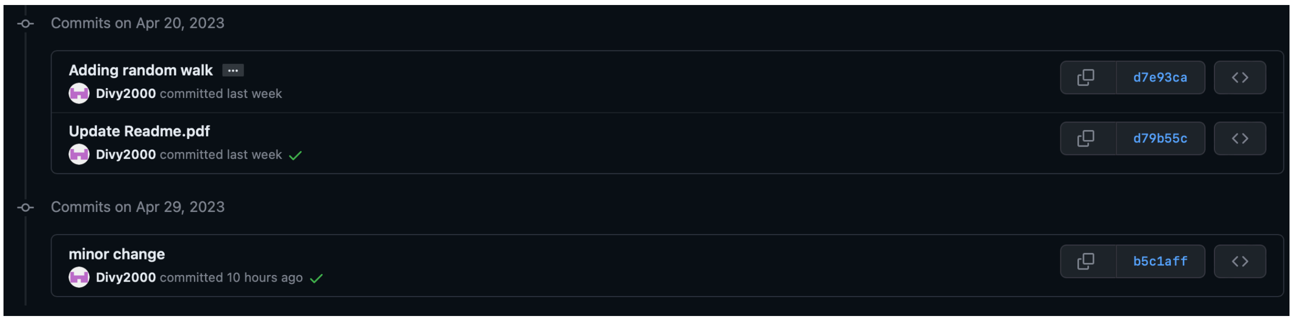
Implemented Strategy Pattern ···
Divy2000 committed 2 weeks ago — e746659

Update Readme.pdf
Divy2000 committed 2 weeks ago — 3108e21

Part 4 - Random walk

- For this I have taken the common steps of the previous strategy pattern and that of the random walk and applied it accordingly.
- The BFS and DFS had the same iteration with only difference of the data structure, but random walk had different type of iterations, so I tried to see whats different in the iteration and put them separate keeping other steps common.
- To print the path for each of the visited node in random walk and implement the behaviour of looping till not finding the path, I used if else statement in the GraphSearch function.
- I again added the BFS and DFS Template implementation, which I removed while implementing the strategy pattern.

Commit Link: https://github.com/Divy2000/CSE464_2023_dpate164/commit/d7e93ca544637b9c1528202ade4cb161e1ccd803


Commits on Apr 20, 2023

Adding random walk ···
Divy2000 committed last week — d7e93ca

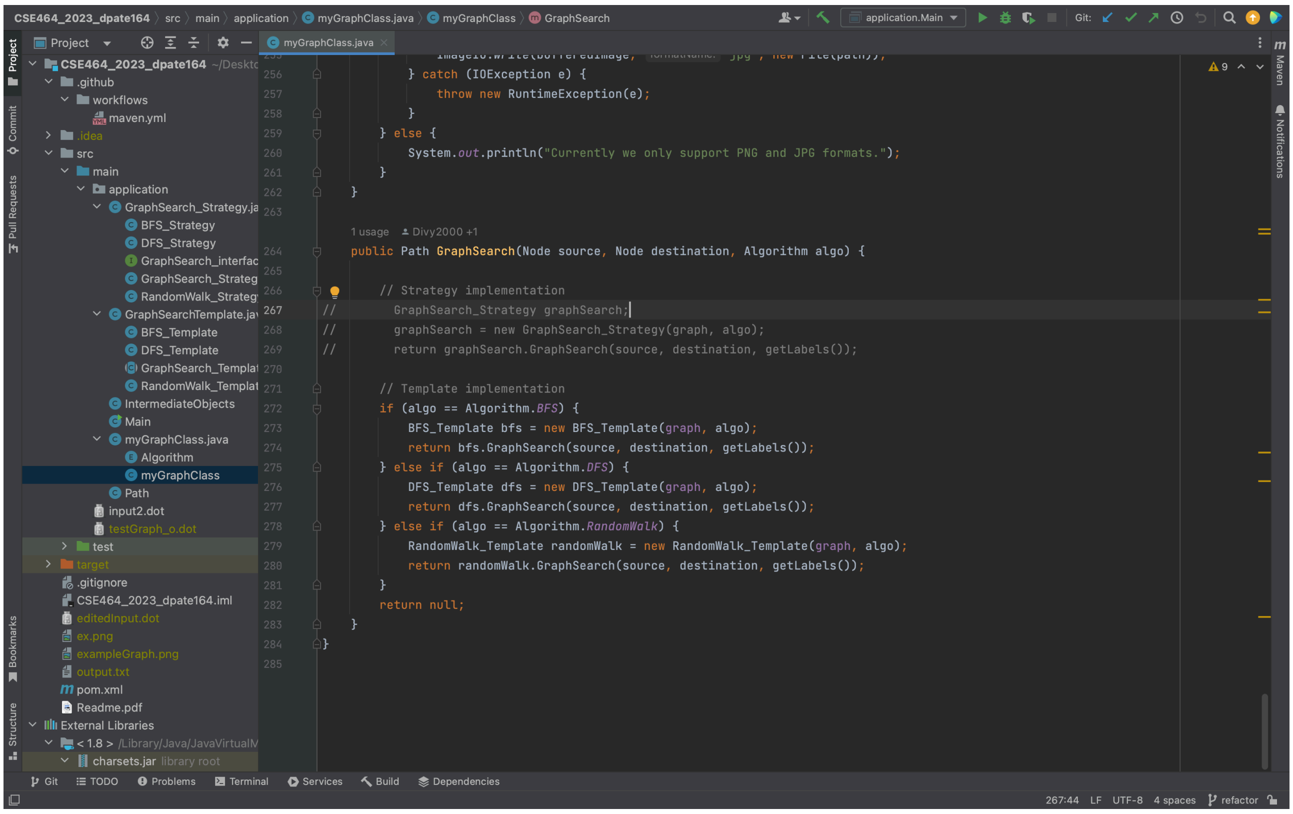
Update Readme.pdf
Divy2000 committed last week ✓ — d79b55c

Commits on Apr 29, 2023

minor change
Divy2000 committed 10 hours ago ✓ — b5c1aff

The Implementation is same as BFS and DFS previously, you have to run it with the command menu given. The option to run Strategy Pattern or Template Pattern is controlled by commenting/uncommenting the code.

Uncomment the code under template implementation to run it.



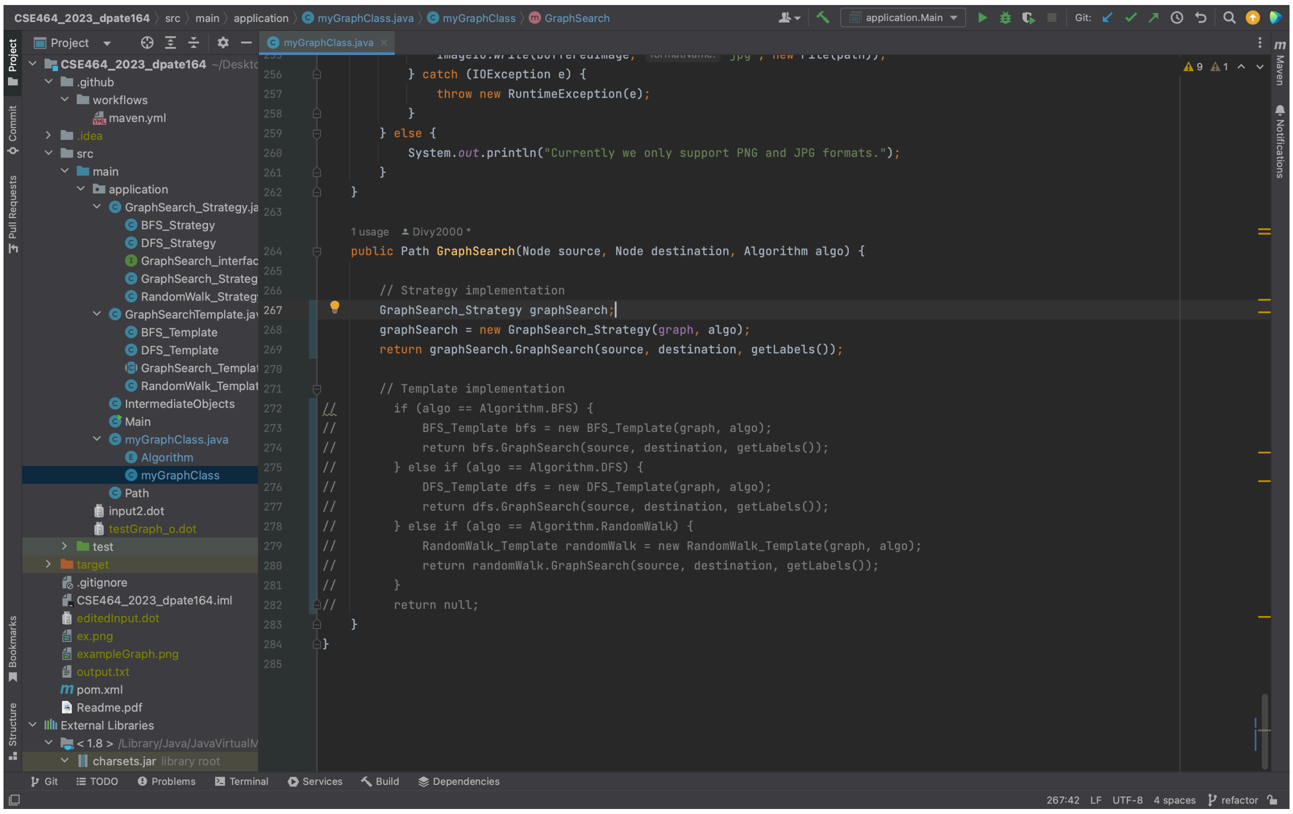
```java
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        } else {
            System.out.println("Currently we only support PNG and JPG formats.");
        }
    }

    public Path GraphSearch(Node source, Node destination, Algorithm algo) {

        // Strategy implementation
//      GraphSearch_Strategy graphSearch;
//      graphSearch = new GraphSearch_Strategy(graph, algo);
//      return graphSearch.GraphSearch(source, destination, getLabels());

        // Template implementation
        if (algo == Algorithm.BFS) {
            BFS_Template bfs = new BFS_Template(graph, algo);
            return bfs.GraphSearch(source, destination, getLabels());
        } else if (algo == Algorithm.DFS) {
            DFS_Template dfs = new DFS_Template(graph, algo);
            return dfs.GraphSearch(source, destination, getLabels());
        } else if (algo == Algorithm.RandomWalk) {
            RandomWalk_Template randomWalk = new RandomWalk_Template(graph, algo);
            return randomWalk.GraphSearch(source, destination, getLabels());
        }
        return null;
    }
}
```

Uncomment the code below strategy comment and comment the template one to run template pattern.



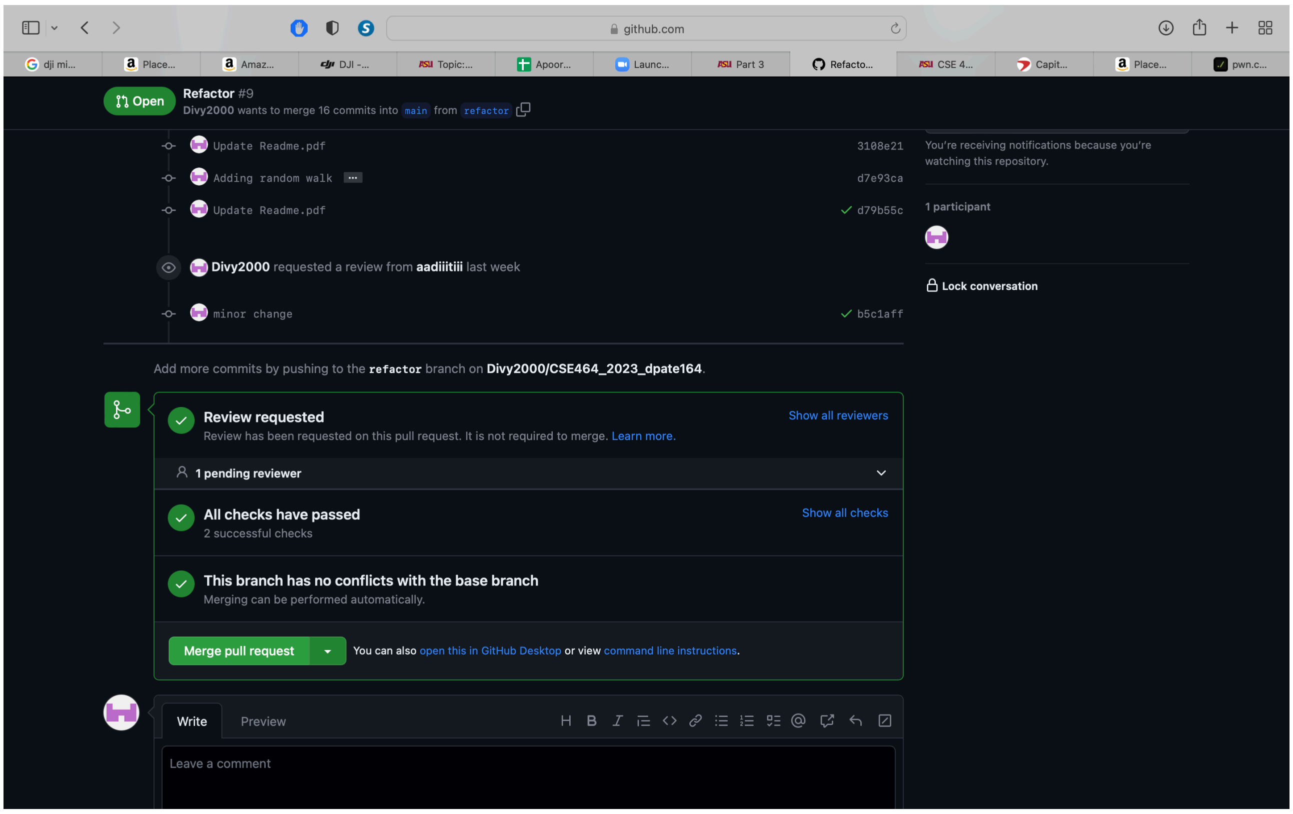
```java
            } catch (IOException e) {
                throw new RuntimeException(e);
            }
        } else {
            System.out.println("Currently we only support PNG and JPG formats.");
        }
    }

    public Path GraphSearch(Node source, Node destination, Algorithm algo) {

        // Strategy implementation
        GraphSearch_Strategy graphSearch;
        graphSearch = new GraphSearch_Strategy(graph, algo);
        return graphSearch.GraphSearch(source, destination, getLabels());

        // Template implementation
//      if (algo == Algorithm.BFS) {
//          BFS_Template bfs = new BFS_Template(graph, algo);
//          return bfs.GraphSearch(source, destination, getLabels());
//      } else if (algo == Algorithm.DFS) {
//          DFS_Template dfs = new DFS_Template(graph, algo);
//          return dfs.GraphSearch(source, destination, getLabels());
//      } else if (algo == Algorithm.RandomWalk) {
//          RandomWalk_Template randomWalk = new RandomWalk_Template(graph, algo);
//          return randomWalk.GraphSearch(source, destination, getLabels());
//      }
//      return null;
    }
}
```

Pull request link: https://github.com/Divy2000/CSE464_2023_dpate164/pull/9


Refactor #9
Divy2000 wants to merge 16 commits into main from refactor

Update Readme.pdf 3108e21
Adding random walk d7e93ca
Update Readme.pdf ✓ d79b55c

Divy2000 requested a review from aadiiitiii last week

minor change ✓ b5c1aff

Add more commits by pushing to the **refactor** branch on **Divy2000/CSE464_2023_dpate164**.

Review requested
Review has been requested on this pull request. It is not required to merge. Learn more.

1 pending reviewer

All checks have passed
2 successful checks

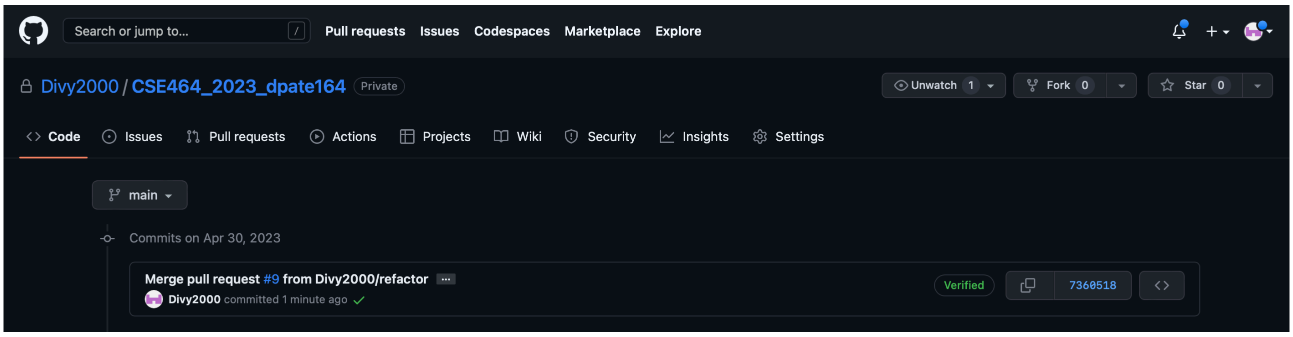
This branch has no conflicts with the base branch
Merging can be performed automatically.

You can also open this in GitHub Desktop or view command line instructions.

You're receiving notifications because you're watching this repository.

1 participant

Merge commit link: https://github.com/Divy2000/CSE464_2023_dpate164/commit/7360518dec1145f87e982e6ff47789019a6761e9


Divy2000 / CSE464_2023_dpate164 Private

Commits on Apr 30, 2023

Merge pull request #9 from Divy2000/refactor
Divy2000 committed 1 minute ago ✓ Verified 7360518