Gebze Technical University

Computer Engineering

CSE222-2020-SPRING

Homework-5_Part3 Report

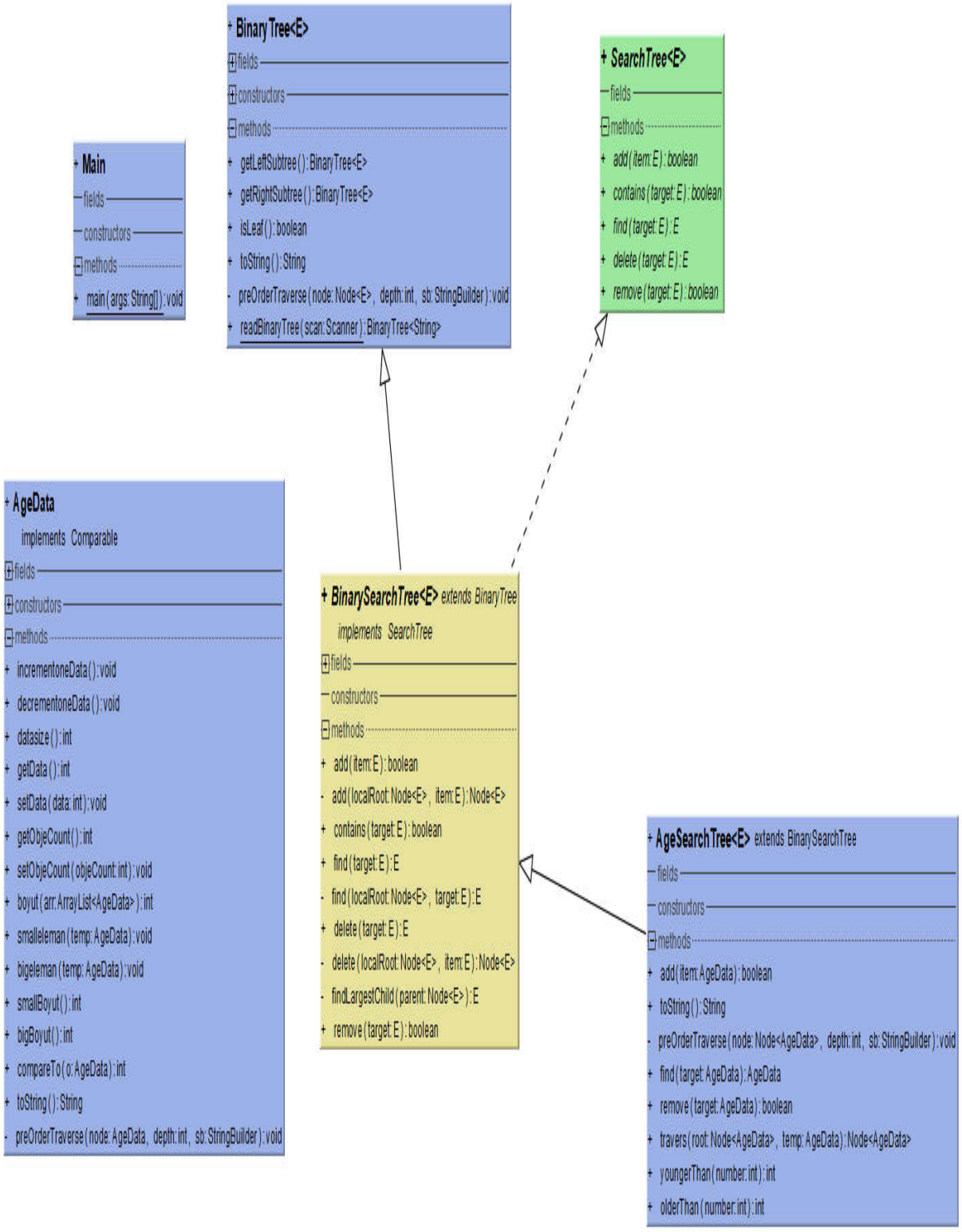
Ferdi Sönmez

161044046

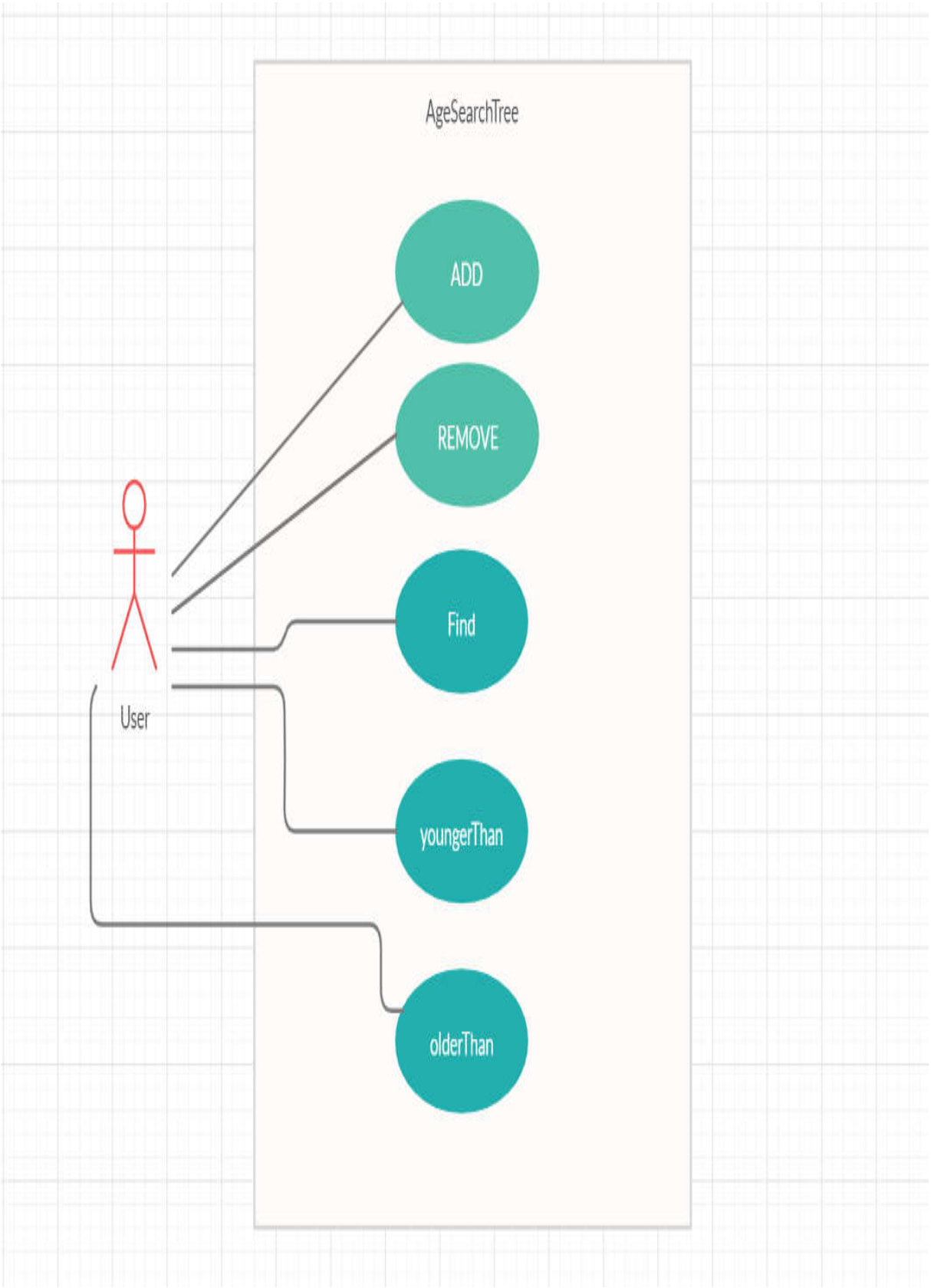
1)Problem Solutions Approach

It extends the BinarySearchTree and uses the add,find and remove functions in BinarySearchTree. Using these functions, changes in the methods override makes by itself perform the wanted operations.
The add function also finds the age to be used by adding the find function in the super class, and the number of people is increased. The find function is also found with the superclass find function.
The remove function finds the age with the find method in the super class, if the number of people is more than 1, it decreases the number of people or it completely delete the age. In the youngerThan function, it finds people younger than the given age. In the OlderThan function, it does the opposite.

2) Class Diagram



3)Use Case Diagram



4)Test Case

a)Add

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T1	with this method we will add the age data to the tree structure	Method call with valid input	`ageTree.add(new AgeData(10));`	Adding data to the tree structure	New item added to tree structure	Pass
T2	Sending age data that not include element	Method call with invalid input	`ageTree.add(new AgeData());`	Adding a zero-year-old item	As expected	Pass

```
AgeSearchTree<AgeData> ageTree = new AgeSearchTree<~>();
ageTree.add(new AgeData(10));
ageTree.add(new AgeData(20));
ageTree.add(new AgeData(5));
ageTree.add(new AgeData(15));
ageTree.add(new AgeData(10));
ageTree.add(new AgeData());
```

```
****Tree--Print****
10-2
5-1
0-1
null
null
null
20-1
15-1
null
null
null
```

b)Remove

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T1	Remove of the requested data from the tree structure	Method call with valid input	`ageTree.remove(new AgeData(0));`	Reduces or completely deletes the number of people of the given age item	Item deleted	Pass
T2	Remove of an item that is not in the tree structure	Method call with invalid input	`flag=ageTree.remove(new AgeData(1500));`	The item cannot be deleted because it is not in the tree structure	The item could not be deleted	Pass

```
ageTree.remove(new AgeData(0));
flag=ageTree.remove(new AgeData(1500));
    if (flag){
        System.out.println("Element Delete");
    }
    else
        System.out.println("There is no such element in the tree structure");
```

```
There is no such element in the tree structure
****Tree--Print****
10-2
5-1
null
null
20-1
15-1
null
null
null
```

c)Find

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T1	Searching for the item in the tree structure	Method call with valid input	`System.out.println(ageTree.find(new AgeData(10)).toString());`	The number of people and item value of the found item are shown	As Expected	Pass
T2	Searching for an item that is not in the tree structure	Method call with invalid input	`System.out.println(ageTree.find(new AgeData(1700)).toString());`	Throws exception if the item is not in the tree	As expected	Pass

```
System.out.println("****Find*****");
System.out.println(ageTree.find(new AgeData(10)).toString());
System.out.println(ageTree.find(new AgeData(1700)).toString());
```

```
****Find*****
10-2

There is no this element
```

d)olderThan

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T1	Shows the number of people older than the given age	Method call with valid input	`System.out.println("olderThan: " + ageTree.olderThan(5));`	The total number of people older than the given age	As Expected	Pass
T2	if the given age is not in the tree structure	Method call with invalid input	`System.out.println("olderThan: " + ageTree.olderThan(20));`	Returns 0 because the data is not in the tree structure	As expected	Pass

```
System.out.println("olderThan: " + ageTree.olderThan( number: 20));
System.out.println("olderThan: " + ageTree.olderThan( number: 5));
```

```
olderThan: 0
olderThan: 4
```

e)youngerThan

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Result	Actual Result	Pass/Fail
T1	Shows the number of people younger than the given age	Method call with valid input	`System.out.println("younger Than: " + ageTree.youngerThan(15));`	The total number of people younger than the given age	As Expected	Pass
T2	if the given age is not in the tree structure	Method call with invalid input	`System.out.println("younger Than: " + ageTree.youngerThan(1523));`	Returns 0 because the data is not in the tree structure	As Expected	Pass

```
System.out.println("youngerThan: " + ageTree.youngerThan( number: 15));
System.out.println("youngerThan: " + ageTree.youngerThan( number: 1523));
```

```
youngerThan: 3
youngerThan: 0
```