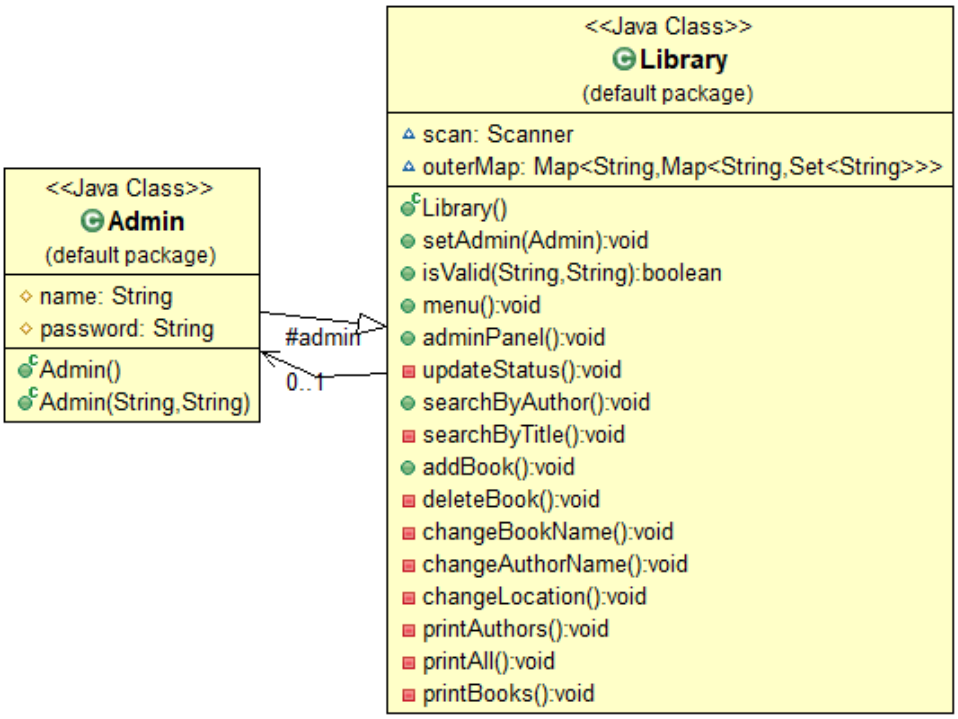
GIT Department of Computer Engineering
CSE 222/505 - Spring 2020

Homework #6 Report

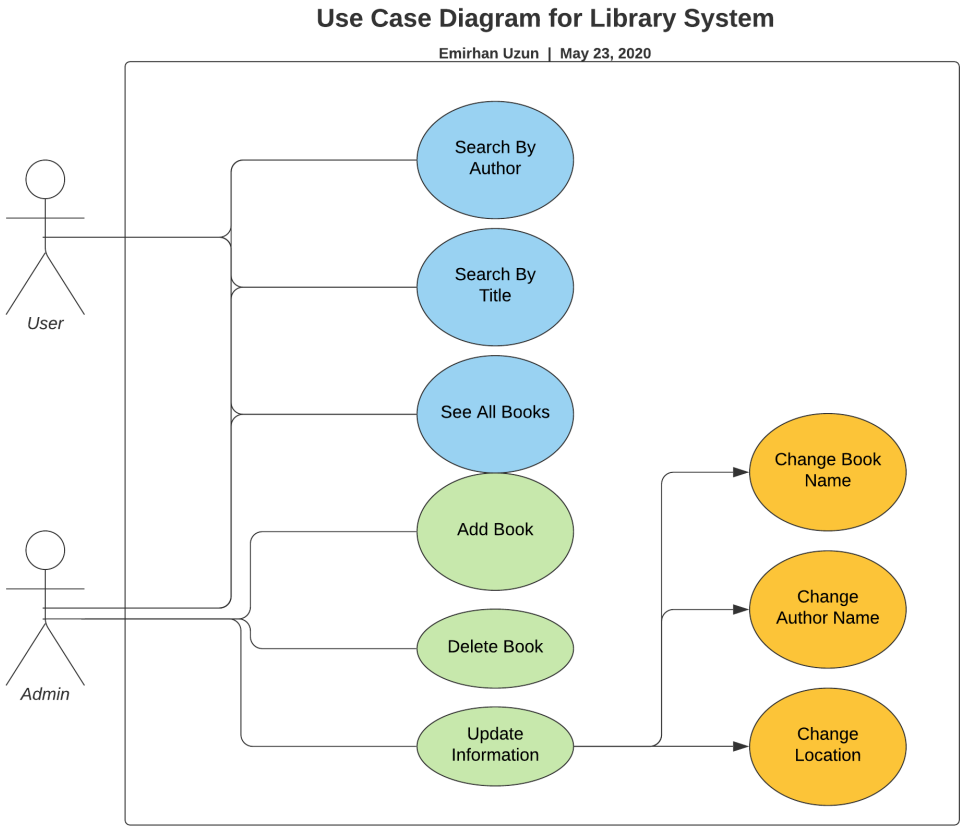
Emirhan Uzun
171044019

QUESTION 3

1) Class Diagram



2) Use Case Diagram



3) Problem Solution Approach

Firstly, I created a outer HashMap. In this map, author names are keys and their values are the other inner map. In inner map, book names are keys and their values are in set.

Except those, I create a admin object. The admin has a name and password for login the system.

When the program runs, the menu expects us. The first choice is search by author name. I take the input for author name and I search in outer map keys. If the map contains it, I print the books. After that if user choose a book, the program shows its location(s).

The second choice is search by title name. Again I take input for title name. And if the inner map keys contain the name, I Show the outer map key which is book's author and location(s).

The third choice is admin panel. When the user choose that, he has to enter the admin system with name and password. If the informations are true, then he can add,delete book or update the informations which are change author name,change book name and change locations. When I add book, first I check that the author exist in map. If it exist I check that the book name exist. If it exists, then I add the location to the set. If the book doesn't exist, then I add the map without extra operation. In delete book method, I search the book name first, then I search the author name because different authors can have same book name. So if the book exist, I delete it.

The fourth choice is print all books. If user chooses it, I print the all books name, author name and location(s).

4) Test Cases

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
T01	Admin Informations	1.Write the admin informations. 2.Create a admin object	- Admin object	Object should be created and informations must be initialized	As Expected	Pass
T02	Search by author method	1- Write the author name 2- Prints the books	-Outer Map -Inner Map	The author's book must be printed	As Expected	Pass
T03	Search by author method if the map is empty	1- Write the author name 2- Give an error	-Outer Map -Inner Map	The program must give an error mesage and turn back the menu	As Expected	Pass
T04	Choose the book informations in search by author method	1- Give an input for book name 2- Prints the book informations	-Outer Map -Inner Map	The method prints the book informations.	As Expected	Pass
T05	Search by title method	1- Write the book name 2- Prints the book informations	-Outer Map -Inner Map	The book informations should be printed	As Expected	Pass
T06	Search by title method if the map is empty	1- Write the book name 2- Give an error	-Outer Map -Inner Map	The program must give an error mesage and turn back the menu	As Expected	Pass
T07	Admin Login Panel method	1.Write the login informations 2- Check and login or logout	- Admin object - Given inputs	If the informations right, then enter the panel, otherwise not	As Expected	Pass
T08	Add Book method	1- Write the book informations 2- Add book	-Outer Map -Inner Map -Book informations	Add book to the map	As Expected	Pass
T09	Delete Book method	1-Write author and book names	-Outer Map -Inner Map -Book informations	If this book exist, then remove it from map	As Expected	Pass
T10	Change Book Name	1-Write the book name	-Outer Map -Inner Map -Book Name	If this book name exist in all the authors books, then change all of	As Expected	Pass

				them name in order		
T11	Change Author Name	1- Write the author name	-Outer Map -Inner Map -Author Name	If this author name exist in map, change it	As Expected	Pass
T12	Change Locations	1- Write the book and author name	-Outer Map -Inner Map -Book Informations	If the informations are exist, then change the location of that book	As Expected	Pass
T13	Print Books	1- Choose the print menu	-Outer Map -Inner Map -Book	Prints all book informations	As Expected	Pass

5) Running Command and Results

```
Please enter a name for admin :
emirhan uzun
Plase enter a password for admin :
123456
The new admin profile was created!
Name : emirhan uzun
Password : 123456
```

This is the create admin. It comes out at program works.

```
----------------------------------------------------
Please enter choice  :
3
Please enter your name :
emirhan uzun
Please enter the your password :
123456
1. Add Book
2. Delete Book
3. Update Information
4. Exit
----------------------------------------------------
```

```
----------------------------------------------------
Please enter choice  :
3
Please enter your name :
emirhanuzun
Please enter the your password :
123456
The name or password are false !
```

This is the admin panel. If the informations are not true then goes to the panel, otherwise not

```
------------------------------------------------------

Please select menu :
1
Please input the Book Name :
gtu pc
Please input Author :
cse222
Please input Location(c*s*.number -> c corridor,s shel
c2s3.1907
The book was added !
Name : gtu pc
Author : cse222
Location : c2s3.1907
```

The add book method. It takes input and adds the map

```
1. Add Book
2. Delete Book
3. Update Information
4. Exit
------------------------------------------------------

Please select menu :
2
Please input the book name to remove :

Authors   -    Books
cse222 - [gtu pc]
---------------------
gtu pc
Please input the author name
cse919
This author was not found !
```

If the author is not true, the program will give an error mesage and won't delete the book.

```
------------------------------------------------------

Please select menu :
1
Please input the Book Name :
gtu pc
Please input Author :
cse222
Please input Location(c*s*.number -> c corridor,s shel
c2s3.1907
The book was added !
Name : gtu pc
Author : cse222
Location : c2s3.1907
```

If the book and author names are true, then the book will removed.

```
Please select menu :
1
Please input the Book Name :
hw6
Please input Author :
data
Please input Location(c*s*.number -> c corridor,s shelf) :
c1s2.1453
The book was added !
Name : hw6
Author : data
Location : c1s2.1453
1. Add Book
2. Delete Book
3. Update Information
4. Exit
```

```
Authors   -    Books
data - [hw6]
---------------------
Please enter the input for old book namehw6
Please enter the input for new book namehw7(unfortunately)
```

```
ALL BOOKS :
Authors - Books - Locations
data-[hw7(unfortunately)]-[[c1s2.1453]]
---------------------
```

I add the book and change book name method changes the book name. The example is above.

```
ALL BOOKS :
Authors - Books - Locations
data-[hw7(unfortunately)]-[[c1s2.1453]]
```

First the book name.

```
Please select menu :
3
1. Change Book Name
2. Change Author Name
3. Change Location
4. Exit
Please select menu :
2

Authors :
 - data
----------------------
Please enter the input for old author namedata
Please enter the input for new author namedata structures
This author name was changed !
```

Now the change author name method works.

```
----------------------------------------------------------------------
Please enter choice  :
4
ALL BOOKS :
Authors - Books - Locations
data structures-[hw7(unfortunately)]-[[c1s2.1453]]
```

The last author name. The change author name method works.

```
Please select menu :
3
1. Change Book Name
2. Change Author Name
3. Change Location
4. Exit
Please select menu :
3
Please input the book name to change location :
ALL BOOKS :
Authors - Books - Locations
data structures-[hw7(unfortunately)]-[[c1s2.1453]]
---------------------
hw7(unfortunately)
Please input the author name
data structures
Please enter the input old location c1s2.1453
Please enter the input new location c6s9.1071
The location was changed !
ALL BOOKS :
Authors - Books - Locations
data structures-[hw7(unfortunately)]-[[c6s9.1071]]
```

This is the change location method. If the book and author exist in the maps, then change the location with new location. This method works.

```
Please select menu :
1
Please input the Book Name :
data structures
Please input Author :
cse 222
Please input Location(c*s*.number -> c corridor,s shelf) :
c4s7.1000
The book was added !
Name : data structures
Author : cse 222
Location : c4s7.1000
```

I added this book for now.

```
----------------------LIBRARY AUTOMATION SYSTEM----------------------
1. Search Book by Author
2. Search Book by Book Title
3. Update (Admin Login)
4. Show all books
5. Exit
-----------------------------------------------------------------
Please enter choice  :
1
Please enter author name:

cse 222
Book 1 : data structures
Which book would you like to see location ? Press 0 for exit : Please enter choice :
1
Book : data structures
Location(s) : [c4s7.1000]
Which book would you like to see location ? Press 0 for exit : Please enter choice :
0
Backing to main menu...
```

If the user choose the search by author, he enters the author name and if it exists, the books are printed. The user choose the book and see the location(s) of that book. This method works.

```
--------------------LIBRARY AUTOMATION SYSTEM----------------------
1. Search Book by Author
2. Search Book by Book Title
3. Update (Admin Login)
4. Show all books
5. Exit
-----------------------------------------------------------------
Please enter choice  :
2
Please enter book name:
data structures
title : data structures
Author : cse 222

Book : data structures
Location(s) : [c4s7.1000]
--------------------LIBRARY AUTOMATION SYSTEM----------------------
1. Search Book by Author
2. Search Book by Book Title
3. Update (Admin Login)
4. Show all books
5. Exit
-----------------------------------------------------------------
```

If user choose the search by title, then he enters the book name. If the book exists, then program prints the author and location(s) informations. This method works.

```
Please select menu :
1
Please input the Book Name :
data structures
Please input Author :
cse 222
Please input Location(c*s*.number -> c corridor,s shelf) :
c6s9.4231
The book was added !
Name : data structures
Author : cse 222
Location : c6s9.4231
```

```
ALL BOOKS :
Authors - Books - Locations
cse 222-[data structures]-[[c4s7.1000, c6s9.4231]]
----------------------
```

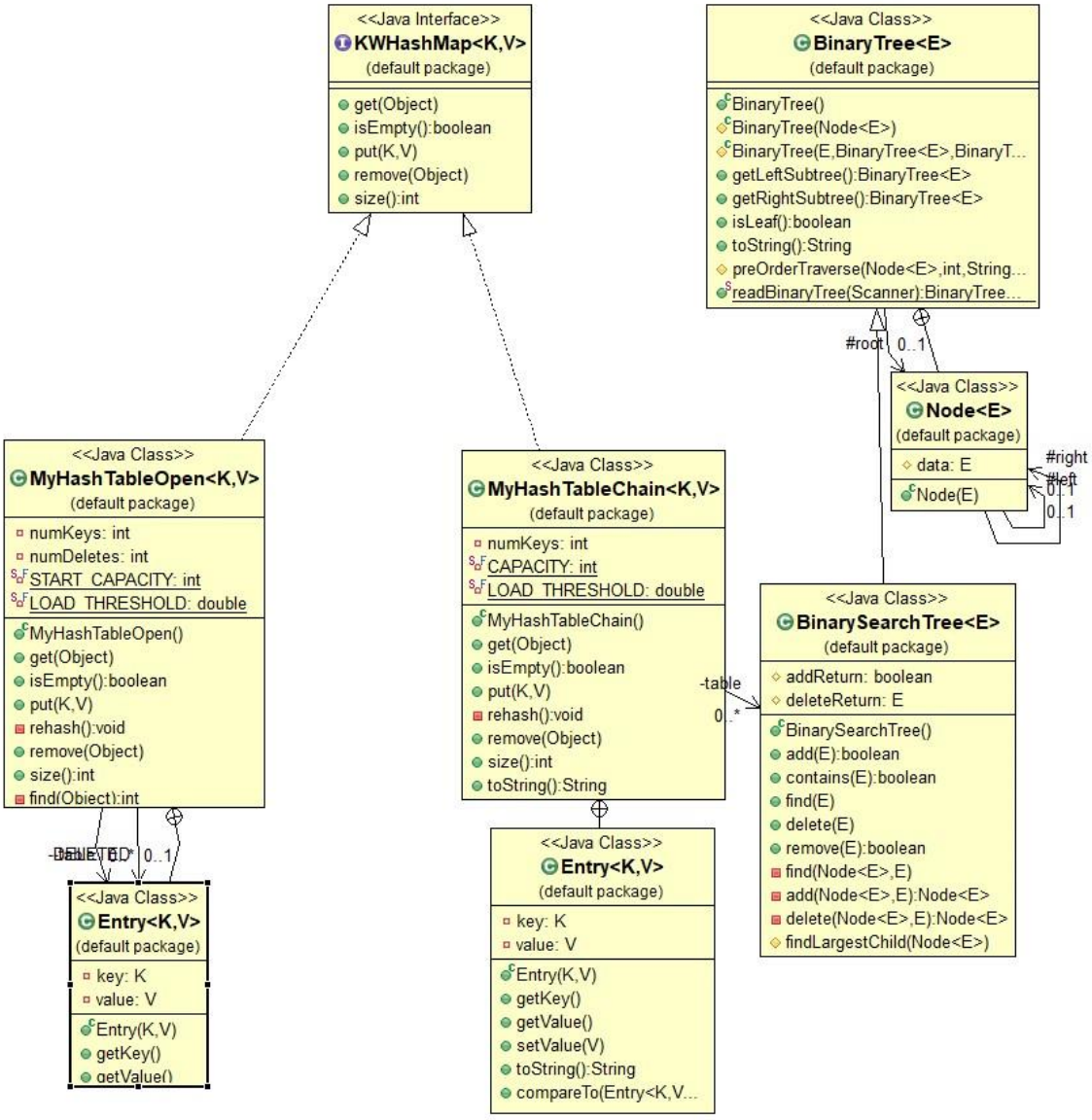
If the same author and same book name exists, then I added the same map with extra location which is given by user. This also works.

Commands:

In this part, I learnt use the map in map with set. Also when I writing this program, also I learnt the methods of set and map like contains, add and remove etc. For each key, we have multiple values since the values are in map. So somewhere I use tempMap and tempSet to put values in specified keys.

QUESTION 3

1) Class Diagram



2) Problem Solution Approach

- **HashTable Chaining**

In chaining method, we should use binary tree instead of linked list implementation. Firstly I have to specify that I use binary search tree. Because I mailed my assistant teachter and she said that it can be used.

My K and V types have to be comparable type. So I implement like "K extends Comparable<K>,". In Entry inner class also should be comparable. To compare, I use compareTo in Entry inner class.

Firstly I create put method. The index is key's hashCode modes table capacity. If the index is not null, I send the key to BST find method. If the key is found then I change the value of that key.

In rehash, I extend the doubled array size.

In remove, I search the key and if it is found in one of these tree, remove that data.

- **HashTable Open**

In open addressing, I just change the hashCode method. In normally, the index is not null, then it increments by one. But out homework, we have to use second method for this. So I researched on Internet and the second hash is generally PRIME – hash % PRIME used. So I multiplied this with table.length which is i. I found the new hashing method.

3) Test Cases

Test Case ID	Test Scenario	Test Steps	Test Data	Expected Results	Actual Results	Pass/Fail
T01	Put method (Chaining)	1.Find the index 2. Add item to tree	- Table array	Find the right index and add or sets the new valeu of that key	As Expected	Pass
T02	Get method (Chaining)	1- Find the key and return the value of key	- Table array	Return the value of that key or if it is not found return null	As Expected	Pass
T03	Remove method (Chaining)	1- Write the key 2- Remove this entry	-Table array	Remove that entry or return null if it is not in tree array	As Expected	Pass
T04	Put method (Open Addressing)	1.Find the index 2. Add item to tree	- Table array	Find the right index and add or sets the new valeu of that key	As Expected	Pass
T05	Get method (Open Addressing)	1- Find the key and return the value of key	- Table array	Return the value of that key or if it is not found return null	As Expected	Pass
T06	Remove method (Open Addressing)	1- Write the key 2- Remove this entry	-Table array	Remove that entry or return null if it is not in tree array	As Expected	Pass

4) Compare Results

- Put Method

```
Table My Chaining : 316 ms
Table My Open Addressing : 160 ms
Table Open Addressing : 124 ms
Table Chaining : 300 ms
```

```
long start = System.currentTimeMillis();
for(int i =0;i<1000000;++i) {
    tableMyChain.put(i+1, i+15);
}
long end = System.currentTimeMillis();
long difTime = end -start;
System.out.println("Table My Chaining : " + difTime + " ms ");
```

In 1 million data, the put method running time is above. The fastest method is open addressing. After that my open addressing implementation is second. The bad times are chaining methods. Because in chaining, each element has a tree. So find and add can be slower than open addressing.

- Remove Method

```
 start = System.currentTimeMillis();
for(int i =0;i<100000;++i) {
    tableMyOpen.put(rand.nextInt(), i+15);
    tableMyOpen.remove(rand.nextInt());
}
 end = System.currentTimeMillis();
 difTime = end -start;
System.out.println("Table My Open Addressing : " + difTime + " ms ")
```

```
Table My Chaining : 19  ms
Table My Open Addressing : 809 ms
Table Open Addressing : 140 ms
Table Chaining : 3 ms
```

In remove and put method (mixed) , the chaining is very very takes time shorter than open addressing. Because it remove the node and it takes less time.

Firstly, to compare all of them I can say these :
My two implementations take much time than the other book implementations.
In add and find, the open addressing takes less time according to chaining.
In remove method, the open addressing takes less time according to chaining.(Generally).

I didn’t try so big data numbers. My computer got very hot and started to slow down. Also, when I run the program, I didn’t get any results.

I create a graph for random operations. It is in the other pfd. Thank you..