
CS401 Project

Kyungjin Kwak

Intro to Advanced Studies 1 (CS401)

Dr. Michael Y. Choi

24th of April

1. Project managing schedule (Timeline)

Tasks Plan	Daily progress (dates of April)														
	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Problem analysis	1	0.5													
Requirements identification			0.5												
Design & Code writing															
▪ Master LinkedList & BST (Code outline)				2	1	1									
▪ Searching method for BST and LinkedList						1	3	2	1						
▪ Main class (file reader, menu)										1	1				
▪ Sorting method										3	2	1	1		
▪ Main class (user's manual input method)											1	2	4		
Testing(w/ data input)												1	1	2	1

(* number in the green box indicates hours/day to work on the task)

2. Problem specification – What problems are solving?

This Coupon Inventory System solves problem that happens in real-world as follows:

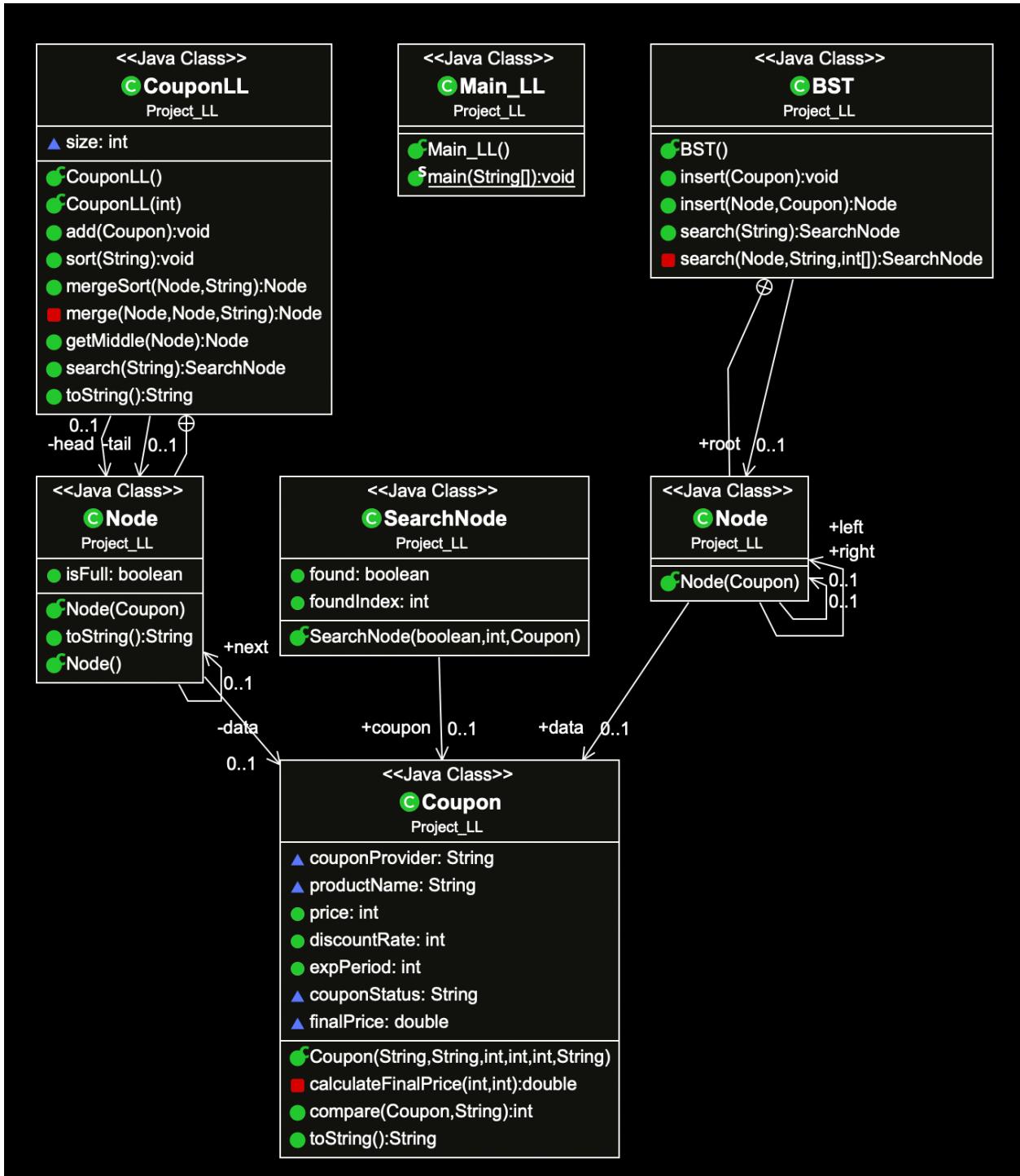
- i) How to storage coupon data both in sorted list and unsorted list?
- ii) How to make coupon list both from reading data file and user's manual input?
- iii) How to make and show a user sorted list of coupons by key field in ascending order?
- iv) How to search coupon of the product? (by Binary Search Tree and Linear Search)
- v) How to count number of searching? (by Binary Search Tree and Linear Search)
- vi) How to show a searched coupon to a user?

3. Software specification - What functions are there?

The Coupon Inventory System has functions for:

- i) Coupon purchase
- ii) Adding more coupon data both by user's manual input and data file
- iii) Showing unsorted list to a user
- iv) Make and show to user the sorted list of coupon by a certain key field that user input on the menu
- v) Searching a coupon by name of the product that user input on the menu
- vi) Showing where the coupon was found by BST and Linear search

4. UML diagram of the Program



5. Project Operation Manual for Users

- a. Run CIS program.
- b. First check the available product name (as of now) and type of criteria of coupons.
- c. Input data of a coupon
(In order: Name of Coupon provider, Product name, Original price, Discount rate, Expiration period, Coupon status (unused or redeemed))
- d. Then the program will show a user Unsored List of Coupon including the coupon what user just input
- e. Enter key field to make Sorted List of Coupons based on the key field
- f. Enter 1 Product name (from the list stated above initial screen) to search a coupon detail
- g. Then the program will show a user where the coupon is found from the list with coupon details
- h. Enter 1 key field (from the list stated above initial screen) to view the Sorted List of Coupons
- i. Then the program will show a user the Sorted List of Coupons based on the key field that user just input on the menu

6. Testing log with input data file

- a. 04/21/2022 ~ 04/22/2022
 - i) Sorting method not working
→ Heap sort and quick sort didn't work at first.
→ Found out that merge sort works well on linked list -> created merge sort method and it worked out well
- b. 04/23/2022
 - i) User's manual input not working
 - ii) It kept showing error of nullPointerException because rest of empty nodes that are waiting for user to input more data is 'null'
→ Created 20 nodes first, and made dummy data with max value for the rest of the empty nodes
- c. 04/24/2022
 - i) The program worked well with all functions stated in the instruction

7. Explanation of how the program works with screen shots

- a. Coupon list (text data file for coupons)

```
coupon list.txt X | J Coupon.java | J SearchNode.java
1 Innisfree SkinToner 20 30 30 redeemed
2 Etude FaceCleanser 15 35 14 unused
3 Etude FaceOil 11 35 14 redeemed
4 Skinfood Lipstick 10 20 30 unused
5 Youtube YoutubePremier 12 45 90 unused
6 CostCo Hotdogs 2 50 14 redeemed
7 Loreal HairBleach 50 60 30 unused
8 Loreal HairDye 40 55 30 redeemed|
```

b. Add function from linked list for user to add more coupon data

```
!!!Coupon inventory System!!!
***Available products: SkinToner, FaceCleanser, FaceOil, Lipstick, YoutubePremier, Hotdogs, HairBleach, HairDye
***Type of criteria: couponProvider, productName, price, discountRate, finalPrice, expPeriod, couponStatus

Add more coupon data.
(IN ORDER: Name of Coupon provider, Product name, Original price, Discount rate, Expiration period, Coupon status)

→ Above picture shows the initial screen that user sees first when the program runs. And it asks a user to add more coupon data. This add function and CouponLL constructor in below picture is created in the CouponLL class which is a master Linked List for this program, and it makes 20 max nodes exist in the list at first, and then user can have total 20 data in the list adding more coupon data with the add function.

//constructor for user to input more data manually
public CouponLL() {

}

public CouponLL (int size) { //user size input method
    this.size = size;
    Node prev = null; // create empty node
    // to create number of nodes that user input(size)
    for (int i = 0; i < size; i++) {
        Coupon s = new Coupon("ZZZZZZZZ", "ZZZZZZZZ", Integer.MAX_VALUE, Integer.MAX_VALUE, Integer.MAX_VALUE, "ZZZZ");
        s.finalPrice = Double.MAX_VALUE; //dummy data to fill the nodes that was not in the data file
        Node node = new Node(s); //make nodes for 'size' times
        if(head == null) { //if head is empty, the first node becomes head and tail
            head = node;
            tail = head;
        }
        if(prev != null) { //if previous node is not empty, the new node is linked after the prev node, and now prev becomes previous node
            prev.next = node;
        }
        prev = node;
    }
}

// add method for Linked List
public void add(Coupon input) {
    if(tail != null) {
        tail.isFull = true; //meaning that list is filled with data from user input or data file
        tail.data = input;
        tail = tail.next;
    }
    //if tail is the last node of the list, link the new node after
    else {
        Node node = new Node(input);
        size++;
        node.isFull = true;
        tail.next = node;
        tail = node;
    }
}
```

c. Show Unsorted List of Coupons

→ When a user add coupon data on the screen, it automatically shows Unsorted List of existing Coupons like below

```
Add more coupon data.
(IN ORDER: Name of Coupon provider, Product name, Original price, Discount rate, Expiration period, Coupon status)
target milk 3 50 10 unused

[Unsorted List of Coupons]
Coupon provider: Innisfree, Product name: SkinToner, Original price: $20, Discount rate: 30%, Final price: $14.0, Expiration period: 30days, Coupon status: redeemed,
Coupon provider: Etude, Product name: FaceCleanser, Original price: $15, Discount rate: 35%, Final price: $9.75, Expiration period: 14days, Coupon status: unused,
Coupon provider: Etude, Product name: FaceOil, Original price: $11, Discount rate: 35%, Final price: $7.15, Expiration period: 14days, Coupon status: redeemed,
Coupon provider: Skinfood, Product name: Lipstick, Original price: $10, Discount rate: 20%, Final price: $8.0, Expiration period: 30days, Coupon status: unused,
Coupon provider: Youtube, Product name: YoutubePremier, Original price: $12, Discount rate: 45%, Final price: $6.6, Expiration period: 90days, Coupon status: unused,
Coupon provider: Costco, Product name: Hotdogs, Original price: $2, Discount rate: 50%, Final price: $1.0, Expiration period: 14days, Coupon status: redeemed,
Coupon provider: Loreal, Product name: HairBleach, Original price: $50, Discount rate: 60%, Final price: $20.0, Expiration period: 30days, Coupon status: unused,
Coupon provider: Loreal, Product name: HairDye, Original price: $40, Discount rate: 55%, Final price: $18.0, Expiration period: 300days, Coupon status: redeemed,
Coupon provider: target, Product name: milk, Original price: $3, Discount rate: 50%, Final price: $1.5, Expiration period: 10days, Coupon status: unused,

Enter the key field to make sorted list of the coupons
```

d. Sorting function(merge sort)

```
Enter the key field to make sorted list of the coupons  
price  
-----Search 'Coupon of product' -----  
Enter 1 available products listed above to search the coupon of the product
```

→ When a user enter key field (e.g. price), it will process merge sort which is method in CouponLL class as below

```
// merge sort method by the keyField that user input on the menu  
public void sort(String key) { //constructor  
    head = mergeSort(head, key);  
}  
  
public Node mergeSort(Node head, String key) {  
    //base case  
    if (head == null || head.next == null)  
        return head;  
    Node mid = getMiddle(head);  
    Node midNext = mid.next;  
    mid.next = null;  
  
    Node leftHead = mergeSort(head, key);  
    Node rightHead = mergeSort(midNext, key);  
  
    return merge(leftHead, rightHead, key);  
}  
private Node merge(Node left, Node right, String key) {  
    if (left == null)  
        return right;  
    if (right == null)  
        return left;  
  
    if (left.data.compareTo(right.data, key) < 0) {  
        left.next = merge(left.next, right, key);  
        return left;  
    } else {  
        right.next = merge(left, right.next, key);  
        return right;  
    }  
}  
  
//separate node  
public Node getMiddle(Node head) {  
    if (head == null)  
        return head;  
    Node slow = head;  
    Node fast = head;  
    while (fast.next != null && fast.next.next != null) {  
        slow = slow.next;  
        fast = fast.next.next;  
    }  
    return slow;  
}
```

e. Searching function both by BST and Linear Search

```
-----Search 'Coupon of product' -----  
Enter 1 available products listed above to search the coupon of the product  
Lipstick  
The coupon is found in 4th by BST  
The coupon is found in 3th by Linear Search  
Check for the Coupon details below:  
Coupon provider: Skinfood, Product name: Lipstick, Original price: $10, Discount rate: 20%, Final price: $8.0, Expiration period: 30days, Coupon status: unused  
-----View the 'List of Coupons'-----  
By what criteria do you want to view the list of coupons? Enter 1 criteria from the list above:
```

→ After sorting the list, the program will ask user to enter 1 product name(e.g. Lipstick) to search for the coupon detail and where it was found from the list both by BST and Linear Search.

```

Scanner input = new Scanner(System.in);
String enterProduct = input.nextLine();

//search coupon by BST
SearchNode searchedNode = bst.search(enterProduct);
if (searchedNode.found) {
    System.out.println(String.format("The coupon is found in %dth by BST", searchedNode.findIndex));
} else {
    System.out.println(String.format("No coupon is found - %dth by BST", searchedNode.findIndex));
}
//search coupon by Linear Search
SearchNode searchedNodeLL = CouponLinkedList.search(enterProduct);
if (searchedNodeLL.found) {
    System.out.println(String.format("The coupon is found in %dth by Linear Search", searchedNodeLL.findIndex));
} else {
    System.out.println(String.format("No coupon is found - %dth by Linear Search", searchedNodeLL.findIndex));
}

System.out.println("Check for the Coupon details below:");
System.out.println(searchedNodeLL.coupon);

```

→ Main class to return the output of where the coupon was found by BST and linear search, and detail of the coupon.

```

// create linear search method by name of the product
public SearchNode search(String productName) {
    Node currPointer = head;
    int index = 1;
    //search the node until the head is equal to the productName that user input on the menu
    while (currPointer != null) {
        if (currPointer.data.productName.equals(productName)) {
            return new SearchNode(true, index, currPointer.data);
        }
        index++;
        currPointer = currPointer.next;
    }
    return new SearchNode(false, index, null); // the coupon is not found on the list
}

```

→ Linear searching (Linked List) method to search coupon and count the number of searching.

```

// search method to search by the productName that user input on the menu
public SearchNode search(String productName) {
    int[] counter = new int[] { 1 }; //first counter starts with 1
    return search(this.root, productName, counter);
}

private SearchNode search(Node root, String productName, int[] counter) {
    //if root is empty, nothing is found from the tree
    if (root == null) {
        return new SearchNode(false, counter[0], null);
    }
    //if root data is equal to the productName that user input on the menu, the data is found from the list and return
    if (root.data.productName.equals(productName)) {
        return new SearchNode(true, counter[0], root.data);
    }
    //if root data is greater than user input, search the data from left sub tree and increment the count until it's found
    else if (root.data.productName.compareTo(productName) > 0) {
        counter[0] = counter[0] + 1;
        return search(root.left, productName, counter);
    }
    //if root data is less than user input, search the data from right sub tree and increment the count until it's found
    counter[0] = counter[0] + 1;
    return search(root.right, productName, counter);
}

```

→ Binary Search Tree method to search coupon and count the number of searching.

f. Show Sorted List of Coupons to user

-----View the 'List of Coupons'-----
By what key field do you want to view the list of coupons? Enter 1 key field from the list above:
finalPrice
Coupon provider: CostCo, Product name: Hotdogs, Original price: \$2, Discount rate: 50%, Final price: \$1.0, Expiration period: 14days, Coupon status: redeemed,
Coupon provider: target, Product name: milk, Original price: \$3, Discount rate: 50%, Final price: \$1.5, Expiration period: 10days, Coupon status: unused,
Coupon provider: Youtube, Product name:YoutubePremier, Original price: \$12, Discount rate: 45%, Final price: \$6.6, Expiration period: 90days, Coupon status: unused,
Coupon provider: Etude, Product name: FaceOil, Original price: \$11, Discount rate: 35%, Final price: \$7.15, Expiration period: 14days, Coupon status: redeemed,
Coupon provider: Skinfood, Product name: Lipstick, Original price: \$10, Discount rate: 20%, Final price: \$8.0, Expiration period: 30days, Coupon status: unused,
Coupon provider: Etude, Product name: FaceCleanser, Original price: \$15, Discount rate: 35%, Final price: \$9.75, Expiration period: 14days, Coupon status: unused,
Coupon provider: Innisfree, Product name: SkinToner, Original price: \$20, Discount rate: 30%, Final price: \$14.0, Expiration period: 30days, Coupon status: redeemed,
Coupon provider: Loreal, Product name: HairDye, Original price: \$40, Discount rate: 55%, Final price: \$18.0, Expiration period: 30days, Coupon status: redeemed,
Coupon provider: Loreal, Product name: HairBleach, Original price: \$50, Discount rate: 60%, Final price: \$20.0, Expiration period: 30days, Coupon status: unused,

→ After searching the coupon, the program will ask user to enter 1 key field to sort the list by that criteria and view the Sorted List of Coupons automatically