Warung Bude

Data Structure and Object Oriented Programming

Outcomes

1. Demonstrate how to create any learned data structure
2. Analyze the usage of data structure in an application
3. Demonstrate basic problem solving skills: analyzing problems, creating algorithms, and implementing logic and algorithms in both C and Java
4. Design, develop and test Java programs using classes, methods, conditionals, loops, etc.

Topics

1. Linked List in C
2. Hash Table in C
3. Collections in Java
4. Object Oriented Programming in Java

Sub-Topics

1. Linked List basic operations (Push, Pop, View)
2. Hash Table basic operations (Put, Remove, Get)
3. Hash Table Collision Handling (separated chaining)
4. Hash Table hashing techniques (DJB2)
5. Basic data types and control flow statements in Java
6. Encapsulation, Inheritance, and Polymorphism
7. Collection Framework (Vector / Array List)

Scoring in C

Subject	Implementation	Percentage
Doubly Linked List (40%)	Push Head	5%
	Push Mid	7.5%
	Push Tail	5%
	Pop Head	5%
	Pop Mid	7.5%
	Pop Tail	5%
	View Nodes	5%
Hashtable (25%)	Insert Item (chaining)	7.5%
	DJB2 Hashing Algorithm	2.5%
	View Item	5%

	Search Item	5%
	Delete Item	5%
Others (35%)	Logic and Validation	25%
	Local Time	3%
	Operating System	2%
	File Reading	5%

Scoring in Java

Subject	Implementation	Percentage
Collections (25%)	Create item	6.25%
	Update Item	6.25%
	Delete item	6.25%
	View Item	6.25%
Hashtable (25%)	Insert Item (linear probing)	10%
	View Item	5%
	Search Item	5%
	Delete Item	5%
Others (50%)	Logic and Validation	25%
	File Reading	7.5%
	OOP	17.5%

Case

Many food stalls, also called *Warteg* are growing around us, spread like hectic because of basic needs that must be met despite material difficulties or time limitations. Warteg Bude is one of them. At every meal break, it becomes crowded by starving people who have to queue for a long time and chuckled out during food taking even when they want to eat it too.

Bude is puzzled to serve every customer that results in the open risk of bad customer intentions, yet at the same time, she wants to maximize profits while preparing for the next dish stock. Hence, Bude was conceived to implement a *Warteg* management application.

The objective of this application is the seller simply requires to fill in the customer's identification and the order based on dish options that have been prepared beforehand. All the data is stored automatically for later use by the seller. Please help Bude make this application so everyone can eat peacefully!

Defines & Scope

1. **Main Menu**

```
System: Mac OSX
Mon Mar 30 15:21:14 2020
1. Add Dish
2. Remove Dish
3. Add Customer
4. Search Customer
5. View Warteg
6. Order
7. Payment
8. Exit Warteg
>>
```

Detect and print user's operating system, current time, and the available options for every feature below then ask a user to input a transition choice, the given parameters should be met are; choice in numeric form and in an inclusive range between 1 to total features. Validate each of the criteria then transition to the chosen menu.

2. **Add Dish**

```
Insert the name of the dish [Lowercase letters]: Tempe Goreng
Insert the name of the dish [Lowercase letters]: telur asin
Insert the price of the dish [1000..50000]: 2500
Insert the quantity of the dish [1..999]: 40
The dish has been added!
Press enter to continue...
```

Ask the user to input a dish to be prepared. The given models should be met; dish's name in lowercase letters and must be unique; price only in inclusive range between 1000 to 50000; quantity between 1 to 999. Validate each of the criteria then store it to the Linked-list data structure.

3. **Remove Dish**

```
            Bude's Menu
==============================
No.      Name        Quantity     Price
1.     telur asin        040       Rp2500
2.   tempe orek        030       Rp13000
3.       buncis          001       Rp6000
==============================
```

```
Insert dish's name to be deleted: telur asin

The dish has been removed!

Press enter to continue...
```

View all the available dishes. Validate each of the criteria then show it from the Linked-list data structure. Ask the user to input a dish name to be removed from the collection. The given controller parameters should be met are; dish name only in alphabet form and in lowercase. Validate each of the criteria then update it to the Linked-list data structure.

4. Add Customer

```
Insert the customer's name [Without space]: Romario Putra
Insert the customer's name [Without space]: Elvan

Customer has been added!

Press enter to continue...
```

Ask the user to input the customer's name to be added to the collection. The given models should be met are; customer name only in alphabet form and without space. Validate each of the criteria then put it to the Hash table data structure with DJB2 hashing.

5. Search Customer

```
Insert the customer's name to be searched: Mabel
Mabel is not present.
Press enter to continue…
```

Ask the user to input the customer name to be view it from the collection. The given controller parameters should be met are; customer name only in alpha form and without space in it. Validate each of the criteria then get it from the Hash table data structure.

6. View Warteg

```
Customer's List

12. Elvan

Press enter to continue...
```

View all the available customers. The given controller parameters should be met are; none. Validate each of the criteria then get it from the Hash table data structure.

7. Order

```
Insert the customer's name: Elvan
Insert the amount of dish: 2
[1] Insert the dish's name and quantity: telur asin x1
[2] Insert the dish's name and quantity: tempe orek x3
```

```
Order success!
Press enter to continue…
```

Ask the user to input the customer's name and his orders. Validate the customer is present. Validate dish is available and the quantity must only be in numeric form and in an inclusive range between 1 to total stock.

8. Payment

```
Insert the customer's index: 12

Elvan
[1] telur asin x1
[2] tempe orek x3
Total: Rp28500
Press enter to continue…
```

Ask the user to input the customer's name and validate the customer has ordered. Print his orders and total payment. After that, remove the customer from the hash table.

9. Exit Warteg

```
Please expand your terminal to full screen!
Press enter to continue...
```

Prompt the user the above message and read the file "*splash-screen.txt*". Then print the file in the terminal.

Please do this project seriously, good luck!