Electricity Management System Report

Team Members :

Full Name	Group	Email	School ID
Slama Lina Nour	G10	lina.slama@ensia.edu.dz	212132041287
Demmouche Lina	G8	lina.demmouche@ensia.edu.dz	212133022373
Lekhdari Wissam	G10	wissam.lakhdari@ensia.edu.dz	222231588609
Mami Nourhane	G10	nourhane.mami@ensia.edu.dz	222235345110

Problem treated :

The Electricity Network Management System is a C++ based program that stimulates the electricity national company software . The purpose of this system is managing efficiently power consumption and injection among customers with solar energy generation capabilities, located in regions ,cities and districts to eventually calculate their bills .
The problem treated in this project consists of conceptualising an Electricity management system which provides company administrators with the ability to track customer data , generate electricity bills, supervise profit in different areas ,in addition to analysing performance of marketing department , while ensuring fast data acess via using convenient ADTs such as Hashtables , Binary searchand AVL trees to minimise search duration and optimize the overall efficiency of the Electricity Company.

Explanation of Data chosen :

In the context of our project, we generated datasets to simulate the electricity management system database , representing the geographic structure of Algeria (regions, cities, districts), along with the details of household (customer) , each of them identified by a unique ID that we generated , in addition to amounts of power consumption and production by each customer , and the weather conditions for each day .Below are tables representing samples of CSV files used in Our program .

1-Region,Cities ,Districts data :

region_id	region_name	city_id	city_name	district_id	district_name,
01	Adrar	01	Adrar	006	Bouda
01	Adrar	01	Adrar	013	Ouled Ahmed Timmi,
01	Adrar	01	Adrar	001	Adrar
01	Adrar	02	Aoulef	022	Timekten
01	Adrar	02	Aoulef	026	Tit
01	Adrar	02	Aoulef	002	Akabli

2-Customer data :

FirstName	LastName	RegionName	CityName	DistrictName	BankAccount	FamilyMem	Ages	Ages	Ages	Ages	Ages	Ages	Ages	ID
Lina	Zemirli	Adrar	Adrar	Bouda	7252441194	7	85	37	83	14	65	17	65	935640796
Malek	Lassouioui	Souk Ahras	Haddada	Haddada	6887966960	2	28	55	0	0	0	0	0	153431794
Zakaria	Larbi	Adrar	Adrar	Bouda	1961432782	2	32	75	0	0	0	0	0	936009736
Abdelmale	Sasi	Adrar	Adrar	Adrar	5785965514	6	29	17	64	53	80	52	0	674724762
Moussa	Cherchar	Adrar	Adrar	Bouda	7717211114	6	73	23	73	51	25	8	0	654833760
Amin	Zemirli	Souk Ahras	Haddada	Ouled Moum	2829332584	4	54	60	60	69	0	0	0	444219373
Mebarka	Nessah	Adrar	Adrar	Adrar	2863606800	7	40	79	37	86	40	57	60	388529680

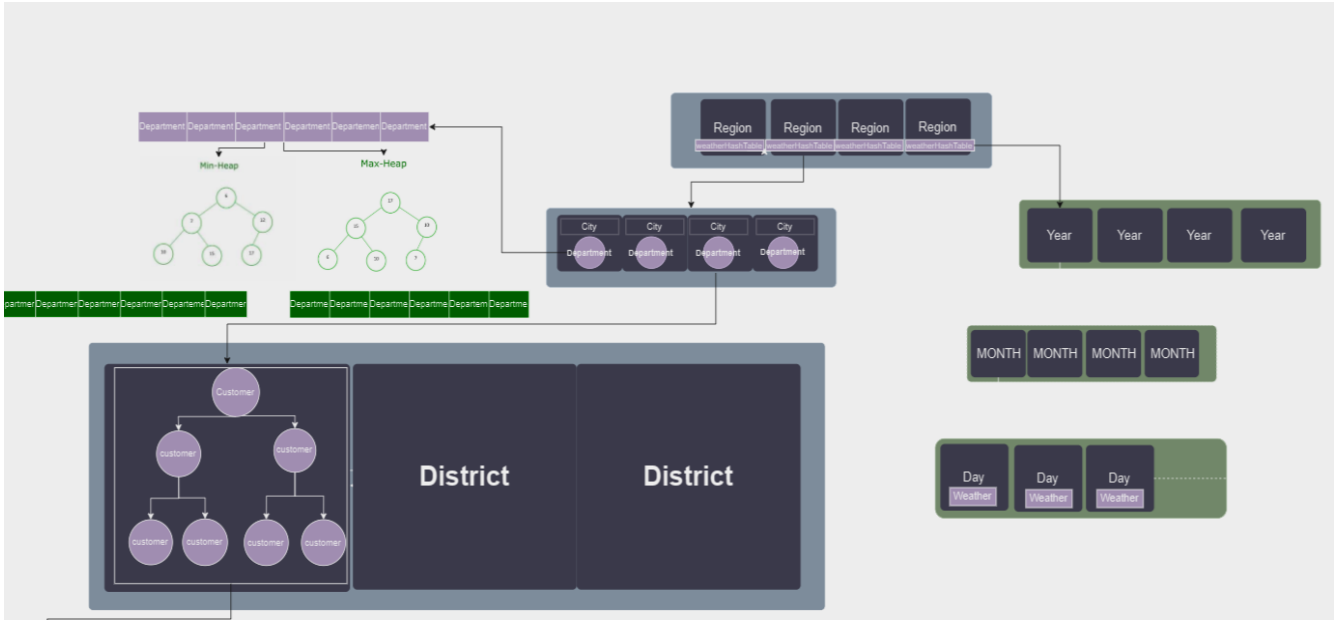
3-Weather Dataset :

Date	MaxTemperature	MinTemperature	SunnyHours	Condition
01-01-202	5	0	2	snowy
02-01-202	4	-1	2	snowy
03-01-202	6	1	3	cloudy
04-01-202	5	0	2	snowy
05-01-202	7	1	3	cloudy
06-01-202	5	-1	2	snowy
07-01-202	4	-2	1	rainy
08-01-202	3	-1	2	snowy
09-01-202	6	2	3	cloudy
10-01-202	5	1	2	snowy

Customer ID : To ensure uniqueness and decrease search time complexity , we chose a specific format for generation of ID in each insertion of Customers .When the Customer is inserted , according to his address(region , city , district) the suitable RegionID , CityID and DistrictID are read from the file and concatinated along with the National ID card number entered when setting his information . by That each customer will uniquely exist in the BST/AVL .

Explanation of Design Choices (ADTs) :

1-Part A :



Hash Tables for Regions, Cities, Districts

A main phase in this project is the distribution of regions, cities, districts to retrieve the bills within any region or city or district in a fast way (O(1) time complexity for acccess)

Hash table of Regions: is a hash table containing many regions, each one is hashed into it using its ID such that the index of the hash table represents the ID of the region. Each Region is linked to a hash table of cities using a pointer (Cities)

Hash Table of Cities: is a hash table containing many cities, each one is hashed into it using its ID such that the index of the hash table represents the ID of the city. Each city is linked to a hash table of districts using a pointer (Districts)

Hash Table of Districts: is a hash table containing many districts, each one is hashed into it using its ID such that the index of the hash table represents the ID of the district. Each district contains a BST (Or AVL in part B) of Customers.

- This hashing method serves to access to any BST easily and Fastly just by knowing the Ids of regions, cities, districts that customers belong to.

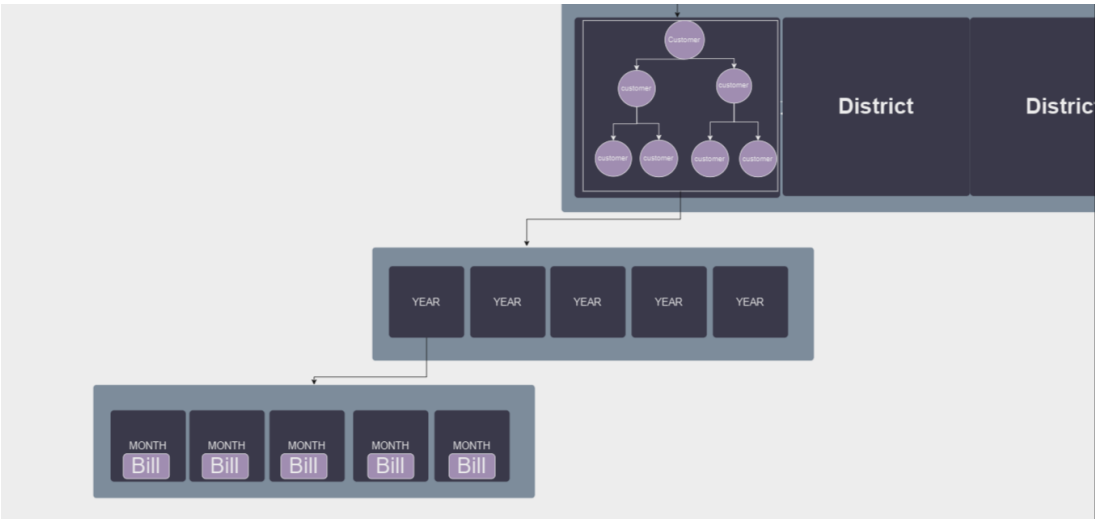
Customers BST : the customers that live in the same region, city and district are in the same

BST stored within each district, (inserted into it after generating their IDs). The choice of a BST for

customers is due to their unfixed size thus the BST is better than hash Tables which need

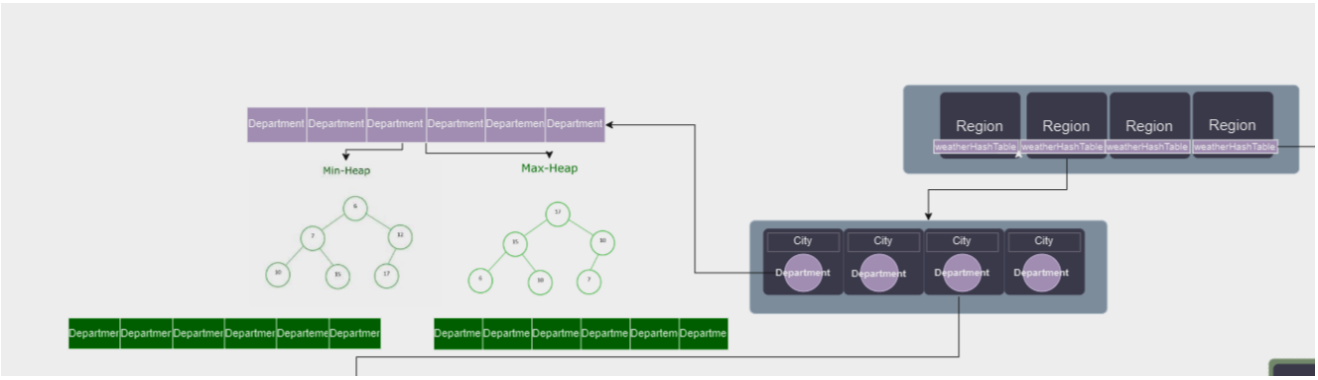
collisions handling and rehashing each time along with loosing memory when allocating size with arrays . Also the BST is a good choice for an efficient and quick search of any customer within it (as the insertion and search operations take on the Average O(log N) and can be on the worst case O(n) , but this can be fixed in Part B using AVL).

Hash Tables for years and months :



Each BST of Customers has a pointer to hashtable of years (the hashing key is the year value) which also points to a hashtable of months storing an object of bill , to enable the administrator to retrieve any bill of any customer in a given month and year .

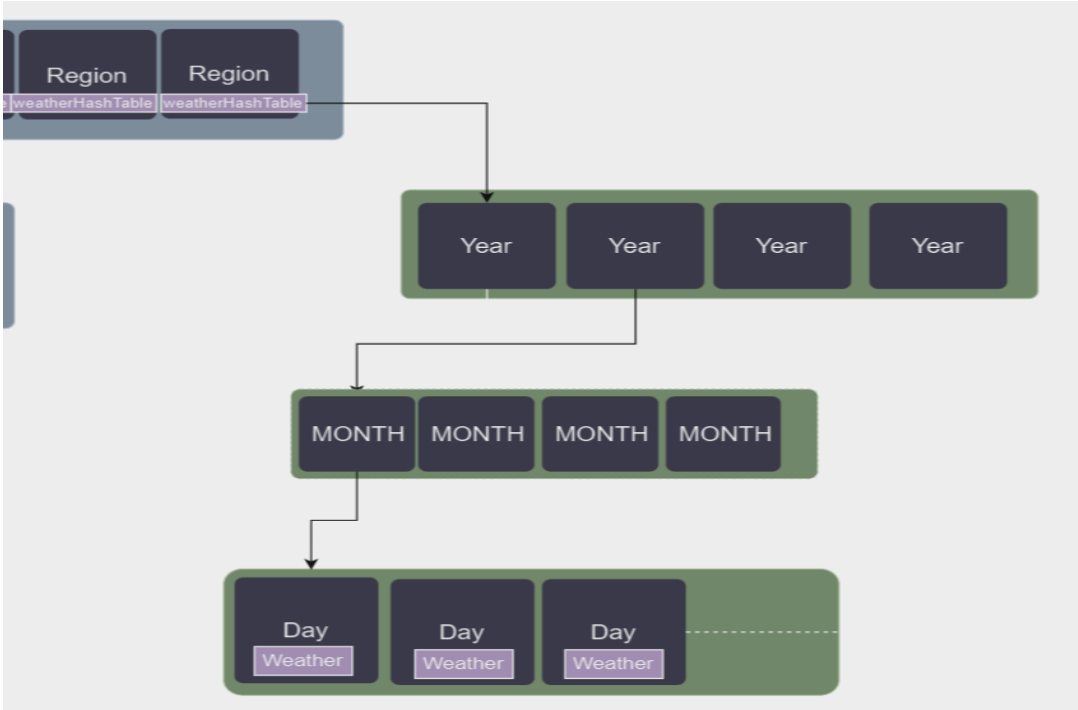
Department heap :



In the company , each department belongs to a city so we stored each department within a city class to have access to the total paid by all customers in that city within a year .

As each year the marketing departments need to be sorted according to their profit , we implemented them in two different heaps max and min that have O((NlogN)) for sorting which is better than sorting them in a regular array .

Weather Vectors :



The weather information are set in every region , thus every region class has a weather vector which has a vector of years , each year is a class that points to a vector of months and within every month there is a weather object that stores information about temperature , date , condition ..

Weather data is set once from the appropriate file once Regions are stored into their hashtable , and can be retrieved by the administrator via “Display Weather” functionality .

2 -Part B :

All ADTs are kept the same for this part , except for the Data structure that stores the Customers , which is transformed from a BST to an AVL .

AVL for customers : The AVL also stores the customers that live in the same Region , City , and District , but performs better when it comes to time complexity in search (O(logN)) in the worst case due to the rotations that balance the nodes at each insertion .

Test Cases :

Test Condition	Test Steps	Test Input	Expected Result	Actual Result	Status	Comment
Ability to Login as administrator	Choose option 1 as an option to login in Write appropriate password only shared for admins	yrhrhe	incorrect password!	incorrect password!	pass	Admins should have a shared password and enter it correctly
Ability to login in as customer	Choose option 2 Enter your ID	Correct Digit of 15 or 16 characters	-Menu appears -Option to display customer bills appears	-Menu appears -Option to display customer bills appears	pass	Customer should enter correct ID to have access for bills
		Uncorrect 1234 (less than 15 digits or more than 17) Or non existent in data base	-Menu appears -Option to display customer bills appears	Customer Not found !	unpass	
checking existence of Region , City,District	-Log in as administrator -Add customer or display bills in District , City , Region	Paris (not found in file of Regions , Cities , Districts)	Inserting customer or displaying bills	Region , City , District doesn't exist	Unpass	Administrator should enter a valid city , region , district that is present in the dataset
Program Efficiency	Inserting data set into hash tables	Regions 58 City 1547 Districts	Program runs efficiently	Takes 2 seconds Accessing	Pass	/

	containing 986 lines	**498**		**data in hash tables take 0.07s**		
Checking validity of Day , month , year , ages ..	**-choose option display Bill in District , City , Region or Choose option display weather in given date or inserting ages of family members**	**Age : -3 Month : 13 Year: 165**	**-Displaying bill of that month/year -displaying weather of tha date -setting ages**	**Invalid input -Menu shows up again**	**Unpass**	**Administrator should set valid date , ages**

Work of each Member :

Lina Nour Slama : Classes of Region, City , District ,, all generating Id functions , debugging , report (test cases , graphical representation of ADTs) , linking classes

Lina Demmouche : Classes of Weather , w_years , w_day , w_hashtable , interface , generating files , report (explanation of ADTs)

Wissam Lekhdari : Classes of Customer , Customers , Bills , Calendar , debudding , linking classes , report (explanation of ADTs) .

Nouha Mami: Department , rotation of AVL , generating Region,City District files , report (Explanation of the problem)

1- to run the program :

For Admin :

password is : helloAdmin.

References :
Stackoverflow.com
Implementation aids from lectures
Chatgpt for debugging .