



Project Report on
CREDIT CARD MANAGEMENT SYSTEM

Submitted by:
ANEESH PANCHAL
2K20/MC/21

Submitted to:
PROF. GOONJAN JAIN

Department of Applied Mathematics
Delhi Technological University





Index

- Introduction
- Linked List
- Advantages of linked list
- Disadvantages of linked list
- Application of linked list
- Application of linked list in real world
- Luhn Algorithm
- Credit Card Management System
- Description of Functions
- Description of Commands
- Conclusion
- References

CARD MANAGEMENT SYSTEM

maintenance and management of the card

maintenance and management of the card



Introduction:

For the project I tried to built a **Credit Card Management System** which mainly used the data structure linked list.

Card Management involves controlling the life-cycle of a card. It could be a Credit card, ID card or Transport card, and it may include some electronic credential on the card (card serial number, e-wallet, digital ID).

The life-cycle is the process that starts from generating the need for a card, i.e. a request, moving to approval and then issuance, and follows with the monitoring of use, expiry, blocking and re-issuance of expired, lost or stolen cards. Each business will have its own rules around each stage of this process.

A Card Management System (CMS) is a software system that administers and facilitates the tracking and control of the card life cycle. Usually, the system will incorporate business rules to automate and streamline the card management process.

The most commonly referenced CMS when you search online are “Credit Card Management Systems” widely used in financial sectors to manage the issuance and management of Credit and Debit Cards. Credit Card Management Systems are typically specialised systems designed only for this purpose.

Some description of data structures I used in my project is provided below.





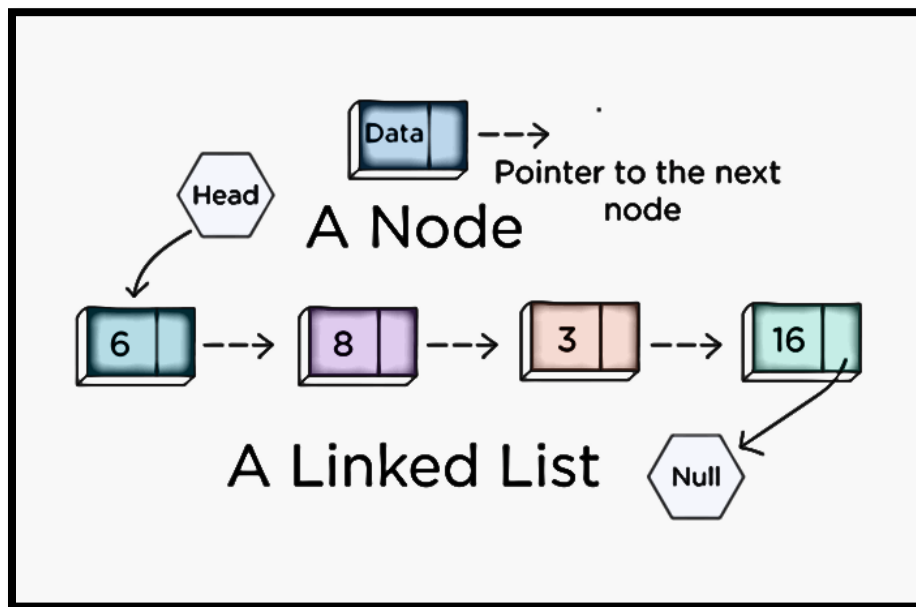
Linked list

Linked List is a very commonly used linear data structure which consists of group of nodes in a sequence.

A linked list is a linear data structure, in which the elements are not stored at contiguous memory locations

Each node holds its own data and the address of the next node hence forming a chain like structure.

Linked Lists are used to create trees and graphs.



Advantages of Linked Lists

- ❖ They are a dynamic in nature which allocates the memory when required.
- ❖ Insertion and deletion operations can be easily implemented.
- ❖ Stacks and queues can be easily executed.
- ❖ Linked List reduces the access time.

Disadvantages of Linked Lists

- ❖ The memory is wasted as pointers require extra memory for storage.
- ❖ No element can be accessed randomly; it has to access each node sequentially.
- ❖ Reverse Traversing is difficult in linked list.



Applications of Linked Lists

- ❖ Linked lists are used to implement stacks, queues, graphs, etc.
- ❖ Linked lists let you insert elements at the beginning and end of the list.
- ❖ In Linked Lists we don't need to know the size in advance.
- ❖ implementation of stack and queues
- ❖ implementation of graphs adjacency list representation of graphs is most popular which uses link list to store adjacent vertices
- ❖ dynamic memory allocation : we used list of free blocks
- ❖ maintain directory of names
- ❖ manipulation of polynomials by storing constants in the node of linked list
- ❖ representing sparse matrices

Application of linked list in real world

- ❖ image viewer - previous and next images are linked hence can be accessed by next and previous button
- ❖ web browser - previous and next url searched in web browser by pressing back and next button they are linked as linked list.
- ❖ music player - songs in music player are linked to previous and next song you can play songs either from starting or ending of the list





Luhn Algorithm:

The Luhn algorithm, also known as the modulus 10 or mod 10 algorithm, is a simple checksum formula used to validate a variety of identification numbers, such as credit card numbers, IMEI numbers, Canadian Social Insurance Numbers. The Luhn formula was created in the late 1960s by a group of mathematicians. Shortly thereafter, credit card companies adopted it. Because the algorithm is in the public domain, it can be used by anyone. Most credit cards and many government identification numbers use the algorithm as a simple method of distinguishing valid numbers from mistyped or otherwise incorrect numbers. It was designed to protect against accidental errors, not malicious attacks.

Seps Involved:

1. Starting from the rightmost digit, double the value of every second digit.
2. If doubling of a number results in a 2 digits number, then add the digits of the product, to get a single digit number.
3. Take the sum of all the digits.
4. If the total modulo 10 is equal to 0, then the number is valid according to the Luhn formula, else it is not valid.

4	0	1	2	8	8	8	8	8	8	8	8	1	8	8	1
x2		x2		x2		x2		x2		x2		x2		x2	
8		2		7		7		7		7		2		7	
	0		2		8		8		8		8		8		1

$$8+0+2+2+7+8+7+8+7+8+7+8+2+8+7+1$$
$$= 90$$



Credit Card Management System:

Functions inside the Class LinkedList:

- **read()**
It reads the data of the csv file and make a Linked list corresponding to the lines of the csv file saving the card number, name and age of the customer.
- **valid_name()**
It will return whether the entered name is valid or not.
- **insert()**
It inserts the given data into the linked list making a new node corresponding to the given card number if the given card number is valid.
- **delete_card()**
It will delete the specific card with the information in it.
- **delete_all()**
It will delete the whole linked list with all the data in it.
- **Update_name()**
It will update the name of the specific card owner.
- **Update_age()**
It will update the age of the specific card owner.
- **Save()**
It will save the file without checking whether the file contain valid cards or invalid cards (not recommended until you make the csv file from scratch).
- **Save_valid()**
It will save the information of only the valid cards.
- **Save_invalid()**
It will save the information of only the invalid cards.



- **Get_data()**
It will return the information corresponding to the specific card number.
- **totalCreditCards()**
It will return the total number of credit cards whether they are valid or not.
- **cValid()**
It will return the total number of valid cards in the file.
- **Show()**
It will print all of the cards and their details whether they are valid or not.
- **Show_valid()**
It will print all of the valid cards and their details.
- **Show_invalid()**
It will print all of the invalid cards and their details.

Functions outside the Class LinkedList:

- **Int_age()**
It will return the integer corresponding to the input string which contains the age of the credit card owner.
- **Splitdelim()**
It will return the array after splitting it according to the delimiter.
- **Split()**
It will return the array after splitting it according to the space “ ”.
- **Card()**
It will return the string corresponding to the input card number.
This function is just for precaution if input file have card number separated by spaces. For example – 1100 1100 1100



➤ **isValid()**

It will return the bool value corresponding to whether the card number is valid or not. It uses Luhn algorithm.

➤ **Lower()**

It will return the string of lower alphabet of the input string.

Commands:

➤ **open hello_test.csv**

it will open the csv file with the name as hello_test.csv

➤ **create hello_test.csv**

it will create the csv file with the name as hello_test.csv

➤ **help**

it will show all the commands that you can use.

➤ **show**

it will print all of the credit cards and their details present in the file.

➤ **show valid**

it will print only the valid credit cards and their details.

➤ **show invalid**

it will print only the invalid credit cards and their details.

➤ **insert 1100110011001100 Aneesh Panchal**

It will insert the card with number **1100110011001100** and credit card owner name as **Aneesh Panchal** if it is a valid number.

➤ **delete card 1100110011001100**

it will delete the card **1100110011001100** and all of its details if it exists.



- **delete all**
it will clear full file components that is it will delete the full linked list.
- **update (name || age) 1100110011001100**
it will update the name or age whichever you want to change corresponding to the given number if it exists.
- **valid**
it will print the number of valid card numbers present in the file.
- **invalid**
it will print the number of invalid card numbers present in the file.
- **total**
it will print the total number of card numbers present in the file.
- **getdata 1100110011001100**
it will print the information corresponding to the card number 1100110011001100.
- **save**
it will save the file with updates you have done in the file.
- **save valid**
it will save the file with information corresponding to only valid card numbers in a new file.
- **save invalid**
it will save the file with information corresponding to only invalid card numbers in a new file.
- **exit**
it will close the program. (make sure to save the changes before closing the file)

Conclusion:

I have implemented a **Credit Card Management System** which uses the Luhn algorithm to check the validity of the card number and also inserted some basic data corresponding to the owner of the card.

The information of the owner is as follows: Name and Age.

To get assigned a credit card one's age must be above 18 years and the card number must be a valid card as per the Luhn Algorithm.

References:

- Wikipedia.org
- Geeksforgeeks.org
- Tutorialspoint.com
- stackoverflow.com

