|  |
| --- |
| **Data Structures and Algorithms** |
| [INTERNET BANKING AND PAYMEN SYSTEM] |
| **Course Project Report** |

|  |
| --- |
| **School of Computer Science and Engineering**  **2020-21** |

**Contents**

|  |  |
| --- | --- |
| **Si. No.** | **Topics** |
| 1. | Course and Team Details |
| 2. | Introduction |
| 3. | Problem Definition |
| 4. | Functionalities |
| 5. | Tools and Techniques |
| 6. | Learning and Takeaway |
| 7. | References |

**1. Course and Team Details**

**1.1 Course details**

|  |  |
| --- | --- |
| **Course Name** | Data Structures and Algorithms  (Theory and Lab) |
| **Course Code** | 20ECSC205 and 19ECSP201 |
| **Semester** | III |
| **Division** | A |
| **Year** | 2020-21 |
| **Instructor** | Prakash Hegade |

**1.2 Team Details**

|  |  |  |
| --- | --- | --- |
| **Si.No.** | **Roll No.** | **Name** |
| 1. | 146 | ANKITA A MANE |
| 2. | 165 | MUSKAN R HAVALDAR |
| 3. | 145 | DISHA R DESHPANDE |
| 4. | 114 | YASHASVI A R |

**2. Introduction**

**Data Structure** is about organising and managing **data** effectively such that we can perform specific operation efficiently, while **Algorithm** is a step-by-step procedure to be followed to reach the desired output.

Using a cooking analogy definitely works to explain it to someone who is a beginner.

* Data structure → cooking pot
* Data → ingredients
* Algorithm → recipe

In short, **data structures and algorithm are two different things but they always go hand-in-hand**.

PROGRAM = DATA STRUCTURES + OPERATIONS/ALGORITHMS.

Coming onto why this is so important. This entire world functions on the concepts of data structures and algorithms. DSA is the heart of software development/software engineering. And one must good at it in order to develop efficient softwares. One cannot claim competency in computer science without being competent in data structure and algorithms! “Design of a programmer” is an e-book written by Prakash Hegade. It holds a very strong quote which proved to be true while performing this course project i.e., if we can think, we can code. If we can think better, we can code better!

**3. Problem Definition**

To device a code to access your bank account and carry out financial transactions through the internet, that enhances the normal banking experience as it is quick, free and allows us to carry out a number of tasks without actually having to visit your bank or ATM. It maintains a validated transaction tracking record and history. It reduces the chances of unwanted risk factor. Passwords are protected by using encryption.

A user can create an account by providing the name of the account holder, account number, select amount type whether it’s Saving account or Current account and providing an initial amount.

For certain purpose, he/she can also check for the balance inquiry which displays the account holder’s name with account number type and amount. He/she can also check for all the account holder’s list.

**4. Functionalities**

|  |  |  |  |
| --- | --- | --- | --- |
| **SI. No.** | **Function Name** | **Description** | **DS and Algorithm Used** |
|  | new\_acc () | It allows the customer and admin to create a new account. | FILES |
|  | edit () | It allows the customer and admin to edit your details. | FILES |
|  | erase () | It allows the customer to delete his/her account. | FILES |
|  | transact () | It allows the customer to deposit or withdraw amount. | FILES |
|  | balance\_enquiry () | It allows the customer to check the amount in his/her account. | FILES |
|  | near\_bank () | It allows the customer to view the nearby ATM with respect to your current location | BRUTE FORCE SEARCHING ALGORITHM |
|  | sort\_acc () | It allows the admin to sort the accounts with respect to their account number | INSERTION SORT  ALGORITHM |
|  | sort\_amt\_m ()  sort\_amt (first, last) | It allows the admin to sort the accounts with respect to the amount in their accounts | QUICK SORT  ALGORITHM |

**5. Tools and Techniques**

**5.1 Data Structures and Algorithms**

* File Data Structure is used to minimize the access time and the storage space. We used files to store details of account holders. File Handling helps in preserving the data or information generated after running the program. There are certain programs that require a lot of input from the user, considering ours which involves internet banking we can easily access any part of the code with the help of certain commands. You can easily transfer the contents of a file from one computer system to another without having to worry about the loss of data.
* Sorting the account holder details based on account number in their increasing order using **insertion sort** technique (called gambler’s invention). It is a stable algorithm. Run time of insertion sort in the best case is O(n) but in the worst case and avg. Case is O(n^2). So, **insertion sort** is simple and less time consuming and have less overhead. So, insertion is better in best and worst case.
* Sorted user’s information based on the amount present in their bank account by using quick sort. Even though **quick-sort** has a worst case run time of O(n2), **quicksort** is considered the **best** sorting because it is VERY efficient on the average: its expected running time is O(nlogn) where the constants are VERY SMALL compared to other sorting algorithms.
* Usage of brute force string search to search for the nearest banks available to the user’s location. For successful search the pattern found at first is the best case with O(m)no of comparisons and O(m\*n) no of comparisons for worst case when the pattern is found at last. For unsuccessful search the best is when the pattern is not found with O(n) complexity and the worst case is when the pattern is found but it is compared with O(nm).

**5.2 Project Statistics**

|  |  |  |
| --- | --- | --- |
| **Si. No.** | **Measure** | **Value** |
|  | Total Functions in Project | 29 |
|  | Total number of lines of code  (Including comments, newlines etc.) | 1613 |
|  | Number of Errors | NULL |
|  | Number of Warnings | NULL |
|  | Team Satisfaction about Project | 80 |

**6. Learning and Takeaway**

The key takeaways are we learnt how to choose the correct algorithm and data structure for a given problem. We were able to correlate between real world problems and their solutions. On how to compare the complexity and data structures of different algorithm for code performance and efficiency. Don’t fret about on how the code appears!

Always be concerned about the different data structures and algorithms being used which will make your code more efficient. That is what will make you a good programmer!

1. **References**

[1] <https://blog.codechef.com/2020/07/24/the-role-of-data-structure-and-algorithms-in-programming/>

Why DATA STRUCTURES AND ALGORITHMS is the HEART of COMPUTER SCIENCE!

[2] <https://ecomputernotes.com/c-program/write-a-program-for-bank-operation>

[3] <http://www.cprograms4future.com/p/atm-machine.html>

[4] <http://see-programming.blogspot.com/2013/07/c-program-for-bank-transactions.html>

**~\*~\*~\*~\*~\*~\*~\*~**