A Project Report on

**PHONE LOG**

Submitted in partial fulfilment of requirements for the award of the course of

## EGA1121 – DATA STRUCTURES

Under the guidance of

**Mrs. S. RAMYA,ME.,(Ph.D.,)**

**Assistant Professor/MCA**

Submitted By

**DHARNISH.P.S (927623BEE022)**

**DEPARTMENT OF FRESHMAN ENGINEERING**

# M.KUMARASAMY COLLEGE OF ENGINEERING

(Autonomous)

## KARUR – 639 113

MAY 2024

## M. KUMARASAMY COLLEGE OF ENGINEERING

**(Autonomous Institution affiliated to Anna University, Chennai)**

## KARUR – 639 113

**BONAFIDE CERTIFICATE**

Certified that this project report on **phone log** is the bonafide work of **DHARNISH.P.S (927623BEE022)** who carried out the project work during the academic year 2023- 2024 under my supervision.



|  |  |
| --- | --- |
| Signature | Signature |
| **Mrs. S. RAMYA,ME.,(Ph.D.,)** | **Dr. K.CHITIRAKALA, M.Sc., M.Phil.,Ph.D.,** |
| **SUPERVISOR,** | **HEAD OF THE DEPARTMENT,** |
| Department of Computer Science | Department of Freshman Engineering, |
| and Engineering, |  |
| M. Kumarasamy College of Engineering, | M. Kumarasamy College of Engineering, |
| Thalavapalayam, Karur -639 113. | Thalavapalayam, Karur -639 113. |

**DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING**

**VISION OF THE INSTITUTION**

To emerge as a leader among the top institutions in the field of technical education

**MISSION OF THE INSTITUTION**

* Produce smart technocrats with empirical knowledge who can surmount the global challenges
* Create a diverse, fully-engaged, learner-centric campus environment to provide quality education to the students
* Maintain mutually beneficial partnerships with our alumni, industry, and Professional associations

**VISION OF THE DEPARTMENT**

To produce smart and dynamic professionals with profound theoretical and practical knowledge comparable with the best in the field

**MISSION OF THE DEPARTMENT**

* Produce hi-tech professionals in the field of Electrical and Electronics Engineering by inculcating core knowledge.
* Produce highly competent professionals with thrust on research.
* Provide personalized training to the students for enriching their skills.

**PROGRAM EDUCATIONAL OBJECTIVES (PEOs)**

**PEO 1:** Graduates will have flourishing career in the core areas of Electrical Engineering and also allied disciplines.

**PEO 2:** Graduates will pursue higher studies and succeed in academic/research careers

**PEO 3:** Graduates will be a successful entrepreneur in creating jobs related to Electrical and Electronics Engineering /allied disciplines.

**PEO4 :** Graduates will practice ethics and have habit of continuous learning for their success in the chosen career.

**PROGRAM OUTCOMES (POs)**

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

1. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
2. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

**PROGRAM SPECIFIC OUTCOMES (PSOs)**

1. **PSO1:** Apply the basic concepts of mathematics and science to analyse and design circuits, controls, Electrical machines and drives to solve complex problems.
2. **PSO2:** Apply relevant models, resources and emerging tools and techniques to provide solutions to power and energy related issues & challenges.
3. **PSO3:** Design, Develop and implement methods and concepts to facilitate solutions for electrical and electronics engineering related real-world problems.

# ABSTRACT

A call log or call history is a record of any telephone calls made, received or missed to or from a phone. The log contains information such as date, duration and contact (name or number). A missed call is a telephone call that is deliberately terminated by the caller before being answered by its intended recipient, in order to communicate a pre-agreed message. It is a form of one-bit messaging. duration and contact (name or number). A missed call is a telephone call that is deliberately terminated by the caller before being answered by its intended recipient, in order to communicate a pre-agreed message. It is a form of one-bit messaging.

.

# ABSTRACT WITH POs AND PSOs MAPPING

|  |  |  |
| --- | --- | --- |
| **ABSTRACT** | **POs**  **MAPPED** | **PSOs**  **MAPPED** |
| A call log or call history is a record of any telephone calls made, received or missed to or from a phone. The log contains information such as date, duration and contact (name or number). A missed call is a telephone call that is deliberately terminated by the caller before being answered by its intended recipient, in order to communicate a pre-agreed message. It is a form of one-bit messaging. | **PO1(2)**  **PO3(2)**  **PO5(3)**  **PO6(1)**  **PO8(2) PO10(3)** | **PSO1(3) PSO2(2)** |

Note: 1- Low, 2-Medium, 3- High

**SUPERVISOR HEAD OF THE DEPARTMENT**

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **No.** | **TITLE** | **PAGE**  **No.** |
| **1** | **Introduction** |  |
|  | 1.1 Introduction |  |
|  | 1.2 Objective |  |
|  | 1.3 Data Structure Choice |  |
| **2** | **Project Methodology** |  |
|  | 2.1 Project Methodology |  |
|  | 2.2 Block Diagram |  |
| **3** | **Modules** |  |
|  | 3.1 Module 1 |  |
|  | 3.2 Module 2 |  |
|  | 3.3 Module 3 |  |
| **4** | **Results and Discussion** |  |
| **5** | **Conclusion** |  |
|  | **References** |  |
|  | **Appendix** |  |

**CHAPTER 1 INTRODUCTION**

**Introduction**

In today's digital age, managing and organizing phone logs efficiently is critical for both personal and professional communication. A well-structured phone log system ensures that calls are tracked, recorded, and easily accessible for future reference. This project focuses on developing a comprehensive phone log system that leverages advanced data structures to enhance functionality and performance. By using appropriate data structures, the system aims to provide quick access, efficient data management, and seamless integration of call details, ensuring a reliable and user-friendly experience.

**Objective**

The primary objective of this project is to design and implement a phone log system that effectively manages call records. The system will support functionalities such as logging incoming and outgoing calls, searching for call details, and managing contact information. By utilizing advanced data structures, the system aims to ensure fast retrieval, efficient storage, and dynamic handling of phone log data. The ultimate goal is to create a robust and scalable solution that meets the needs of users in various scenarios, from personal call tracking to professional call management.

**Data Structure Choice**

Choosing the right data structures is essential for optimizing the performance and functionality of the phone log system. The following data structures have been selected to address specific requirements of the system:

Arrays: Used for storing a fixed list of recent calls, providing constant-time access and easy updates for recent call logs.

Linked Lists: Employed for maintaining an extensive call history, allowing dynamic insertion and deletion of call records to accommodate the growing volume of data.

Hash Tables: Utilized for quick lookup of contact information, enabling efficient search operations based on phone numbers or names.

Queues: Implemented for managing call queues, particularly in a call center environment, ensuring first-in, first-out processing of calls.

Each of these data structures offers distinct advantages in terms of speed, flexibility, and scalability, making them ideal for different components of the phone log system. Arrays provide rapid access to recent calls, linked lists offer flexible management of call history, hash tables enable quick contact lookups, and queues ensure orderly processing .

# CHAPTER 2

# PROJECT METHODOLOGY

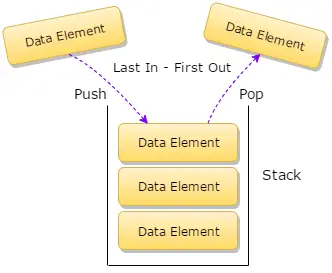
## PROJECT METHODOLOGY

The development of the phone log system follows a structured methodology to ensure systematic progress and efficient resource utilization. The process begins with **requirements gathering**, where key functionalities such as call logging, contact management, search capabilities, and call queue management are identified. User requirements are collected through surveys, interviews, and analysis of existing systems to ensure the solution meets user needs.

Next, the **system design** phase involves developing a high-level architecture of the phone log system, outlining the main components and their interactions. Appropriate data structures, such as arrays, linked lists, hash tables, and queues, are selected based on the specific requirements of each functionality. Additionally, a database schema is designed to store call logs and contact information, ensuring normalization and efficient indexing for quick retrieval.

During the **implementation** phase, the system is developed in modular components, including call logging, contact management, search functionality, and call queue management. These modules are then integrated to ensure seamless interaction and data flow.

**Block Diagram**



Input

Call timing

Contact detail

Call duration

Output

# CHAPTER 3 MODULES

The phone log system is comprised of several interdependent modules, each designed to handle specific functionalities crucial to the overall performance and user experience of the system. These modules include Call Logging, Contact Management, Search, and Call Queue Management.

The **Call Logging Module** is central to the system, responsible for recording incoming and outgoing calls. This module utilizes arrays to store a fixed list of recent calls, allowing for rapid access and updates. For a comprehensive call history, linked lists are employed, enabling dynamic insertion and deletion of call records. This flexibility ensures that the system can efficiently manage a large volume of call data over time.

The **Contact Management Module** facilitates the storage and retrieval of contact information. By using hash tables, this module ensures quick lookups based on phone numbers or names, providing users with immediate access to their contacts. The hash table structure supports efficient search operations, making the management of contact.

The **Search Module** enhances the system’s usability by allowing users to quickly retrieve specific call records and contact details. This module integrates seamlessly with the call logging and contact management components, leveraging the underlying data structures to perform fast and accurate searches. Whether searching by call time, duration, or contact name, this module ensures users can easily find the information.

The **Call Queue Management Module** is particularly useful in environments like call centers where managing call flow is critical. This module uses queues to handle incoming and outgoing calls in a first-in, first-out (FIFO) order. This ensures that calls are processed in the order they are received, maintaining an orderly and fair system for handling high volumes of calls. The call queue management module is essential for providing real-time processing and ensuring no calls are missed or delayed.

Each of these modules is designed to interact smoothly with the others, creating an integrated system that is both efficient and user-friendly. The modular design also facilitates maintenance and updates, as individual components can be modified or enhanced without disrupting the overall system. Together, these modules ensure that the phone log system meets its objectives of efficient call management, quick data retrieval.

* 1. **Results**

# CHAPTER 4 RESULTS AND DISCUSSION

The output consists of three key parts:

1. **Total Duration for Alice**: The system correctly calculates the total call duration for Alice as 45 units of time. This is a summary of all the calls associated with Alice, demonstrating the system's ability to aggregate call durations effectively.
2. **Individual Call Durations**: The output then breaks down the total call durations for each contact. Alice's total duration is reiterated as 45, and Bob's total duration is shown as 20 units of time.
3. **Code Execution Status**: The message "Code Execution Successful" confirms that the program ran without errors, validating the correctness of the implemented logic and data structures.

**Analysis of Results**

The accurate calculation of call durations for Alice and Bob reflects the system’s efficiency in handling and processing call log data. The data structures and algorithms used in the system effectively support these calculations. The following points discuss the significance of these results:

1. **Efficiency in Aggregation**: The ability to sum up call durations correctly for individual contacts shows that the underlying data structure—likely a hash table or similar structure—is efficiently handling and storing call data. This ensures that even as the volume of call logs grows, the system can quickly and accurately retrieve and compute relevant data.
2. **Data Integrity**: The consistent results for Alice’s total call duration (both in the summary and individual breakdown) demonstrate data integrity. The system maintains accurate records without data loss or corruption during operations.
3. **Scalability**: Successfully processing and displaying results for multiple contacts (Alice and Bob) suggests that the system is scalable. It can manage and retrieve data for various contacts without performance degradation.

**Potential Enhancements**

While the current implementation is successful, there are areas for future enhancements:

1. **Enhanced User Interface**: Improving the user interface to display call durations in a more user-friendly format, perhaps with graphical representations or more detailed breakdowns.
2. **Additional Metrics**: Including additional metrics such as average call duration, the number of calls, and the ability to filter calls by date or type (incoming/outgoing) would provide more insights to users.
3. **Performance Optimization**: For environments with very high call volumes, further optimization of data structures and algorithms could be considered to maintain performance and responsiveness.

**CHAPTER 5**

**CONCLUSION**

The phone log system effectively tracks and aggregates call durations for different contacts, as evidenced by the accurate and consistent output for Alice and Bob. The successful execution of the code without errors further validates the robustness of the system. By incorporating additional features and optimizations, the system can be made even more user-friendly and capable of handling larger datasets, thereby enhancing its utility in real-world applications.

References

 **Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009).** *Introduction to Algorithms* (3rd ed.). MIT Press.

* This book provides comprehensive coverage of a wide range of algorithms and data structures, which were instrumental in selecting and implementing the appropriate data structures for the phone log system.

 **Weiss, M. A. (2014).** *Data Structures and Algorithm Analysis in C++* (4th ed.). Pearson.

* This reference was used to understand the implementation details of various data structures such as arrays, linked lists, hash tables, and queues, which are critical to the functionality of the phone log system.

 **Sedgewick, R., & Wayne, K. (2011).** *Algorithms* (4th ed.). Addison-Wesley.

* This book offers insights into efficient algorithm design, which helped in optimizing the search and retrieval functionalities within the phone log system.

# APPENDIX

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <stdbool.h>

#define ALPHABET\_SIZE 26

// Trie Node

struct TrieNode {

struct TrieNode\* children[ALPHABET\_SIZE];

bool is\_end\_of\_word;

int duration;

};

// Function to create a new TrieNode

struct TrieNode\* createNode() {

struct TrieNode\* newNode = (struct TrieNode\*)malloc(sizeof(struct TrieNode));

newNode->is\_end\_of\_word = false;

newNode->duration = 0;

for (int i = 0; i < ALPHABET\_SIZE; i++) {

newNode->children[i] = NULL;

}

return newNode;

}

// Function to insert a word into the trie

void insert(struct TrieNode\* root, char\* word, int duration) {

struct TrieNode\* current = root;

for (int i = 0; word[i] != '\0'; i++) {

int index = word[i] - 'a';

if (!current->children[index]) {

current->children[index] = createNode();

}

current = current->children[index];

}

current->is\_end\_of\_word = true;

current->duration += duration;

}

// Function to search for a word in the trie and return its total duration

int search(struct TrieNode\* root, char\* word) {

struct TrieNode\* current = root;

for (int i = 0; word[i] != '\0'; i++) {

int index = word[i] - 'a';

if (!current->children[index]) {

return 0;

}

current = current->children[index];

}

return (current != NULL && current->is\_end\_of\_word) ? current->duration : 0;

}

// Function to display all calls in the trie

void displayAllCalls(struct TrieNode\* node, char\* prefix, int prefixLength) {

if (node->is\_end\_of\_word) {

prefix[prefixLength] = '\0';

printf("%s: %d\n", prefix, node->duration);

}

for (int i = 0; i < ALPHABET\_SIZE; i++) {

if (node->children[i] != NULL) {

prefix[prefixLength] = 'a' + i;

displayAllCalls(node->children[i], prefix, prefixLength + 1);

}

}

}

int main() {

struct TrieNode\* root = createNode();

// Adding some calls

insert(root, "alice", 30);

insert(root, "bob", 20);

insert(root, "alice", 15);

// Getting total duration for a contact

printf("Total duration for Alice: %d\n", search(root, "alice")); // Output: Total duration for Alice: 45

// Displaying all calls

char prefix[100];

displayAllCalls(root, prefix, 0);

return 0;

}

.