National Institute of Business Management

School of Computing and Engineering

Project Report

Temporary Network Access Control System

Data Structure & Algorithms

HNDNE 25.2F

Group Members

Index No	Name
COHNDNE252F-001	FERNANDO M V S
COHNDNE252F-003	METHNULI J K P D D

STATEMENT OF AUTHENTICITY

We hereby come into affirmation of declaring that the thesis is the original work as a result from our study research and endeavours. We have not submitted this work in prior to this for any academic award or any other institution. As far as we know, the thesis includes no material neither authored nor published by anyone else without due knowledge.

Students' Names and Signatures:

1. M.V.S Feranando

   …………………………………… ………………………………..
   Signature Date

2. METHNULI J K P D D

   …………………………………… ………………………………..
   Signature Date

This is to certify that this project is the work commenced by the aforementioned students under my supervision. The report is as per the required format and at a level of expected and acceptable submission standard.

Certified by:

………………………………………………………
Supervisor Name: Dr. Thisara Weerasinghe

Signature Date:

WORD OF ACKNOWLEDGEMENT

We would like express our gratitude on everyone who gave their best at supporting in assisting and advising on the path of forming a successful project. Their excessive endeavour in delivering us an affirmative assistance lead the project into generating best possible outcome of it.

The heartiest gratitude goes to **Dr. Thisara Weerasinghe** for bestowing us with the knowledge by improving our logical & critical thinking while encouraging us to learn with willingness via a practical-based environment. Moreover, the knowledge gained through and the learning outcome from the module of data structure and algorithm did a huge contribution in generating a perfect vision for a problem, planning and developing the concept more broadly than we intended to. Therefore, we are obliged & in indebt for our lecturer for keeping us enlightened and educated while bringing us on the right path under a meticulous guidance at each step towards an effective and rational output with the project implementation.

Thus, the sincerest admiration must go to the team-members as they sacrifice their dedication and the effort at the expense of sacrificing the time. Under the co-operated work distribution, were able to derive with creative solutions and improved the ability to surpass the challenges as a team engraving another skill to work under a collaborative environment.

Finally, we extend our appreciation towards everyone who assisted us on each step directly or indirectly letting our determination end up in generating a solution for a tricky and challenging problem exist within the real-world.

TABLE OF CONTENT

PROJECT SYNOPSIS

In accessing the network of an organization, the person get access via intranet or extranet based on the role he possessed. But when it comes to temporary access for external users like visitors, guest users and auditors will have access temporarily via a manually operated system. This thesis proposes how a manual system is being transformed into an automatic based system software with the usage of Min Heap data structure. As it let the manual way of handling access grant, session appointing, revocation of user access and user behaviour monitoring and etc. flaws to be covered within the automated version. The supposed solution data structure Min heap, each node is a representative of user compromised with identification details and time-slot key for priority cases. In addition, system is supportive of assuring tasks like insertion, deletion or removal, and update of the heap tree efficiently, engendering a methodical operation without scanning all nodes for every situation. The software-based automatic extranet access control system operate for several ultimate scenarios:

- Access Allocation of the users
- Enforcement of expiry time-slots
- Automatic access revocation in urgencies
- Extension & modification of expiry time

CONTEXT OF THE PROJECT

Within a modern organization the network infrastructure basically is concerning due to several reasons like communication, resource-sharing, productivity-effectiveness, connectivity and efficiency. In order to get connected to the organization network system employees and non-employees are fundamentally gaining access via intranet and extranet. Intranet is highly secured and restricted private network ultimately for the employees the internal staff who work under the organization fulltime.

As for the clients, vendors and suppliers who require access to the network yet not the internal network falls upon the extranet. But when it comes for the temporary external users like guests, visitors, consultants, and auditors, they must be controlled via a secure environment since it can generate some risks for the organization network infrastructure and internal private network.

When temporary external users get linked through extranet the process can be described like this; from zero procedure of getting access, session assignment and allocation of the users for a limited time for each users, continuous monitoring of each specific session for malicious activities are the prime formulae in handling the temporary external users.

However, the issue is the if the current network access controlling system of the company seems to be working under the influence of human labour we will have to come up with a better automated version of that. Since we already know that human based manual management is inclined to be error based. Subsequently, the automated system need to meet those specific requirements of the whole procedure while covering up the loopholes and flaws of where the human intervention based system basically did. This is when to apply the knowledge of data structure and algorithm to solve a real-world problem mentioned above.

The **Min Heap** will be used as the strategic clarification for the above scenario as it can answer all our problems that are present within the manual based system. It has the ability to define access to the users, time allocation based on the role policy of the company, maintain the order in accordance with who to be expired soon, continuous monitoring each user actions for abnormal activities due to security concerns, the capability of searching for specific session and immediate annulment of the user access in case of unusual behaviours.

NETWORK-SCENARIO

As mentioned the circumstances for the external users who get access to the company network infrastructure must linked only via extranet, an isolated network segment for non-internal staff members which is basically not prohibited for the external users like vendors, clients and suppliers. But the situation is there is another category of network access users who interconnect with the network for a limited time.

For such situation there is a whole procedure ongoing and it indicates the lack of a system that can operate as an automated-system with following requirements in order to maintain the process efficient, fast and non-error prone and a network infrastructure secured, controlled, compromised and undisturbed while it's restricted only for external users.

The exact process within the of the automated version ought to be like this; recognition of temporary external users, time allocation must be established on the order of the role-based regulated with the company policies, identification of session that is to be expired momentarily, continuous-monitoring of the sessions and detection of mal-activities & immediate revoke access of such users as the results of abnormal behaviours.

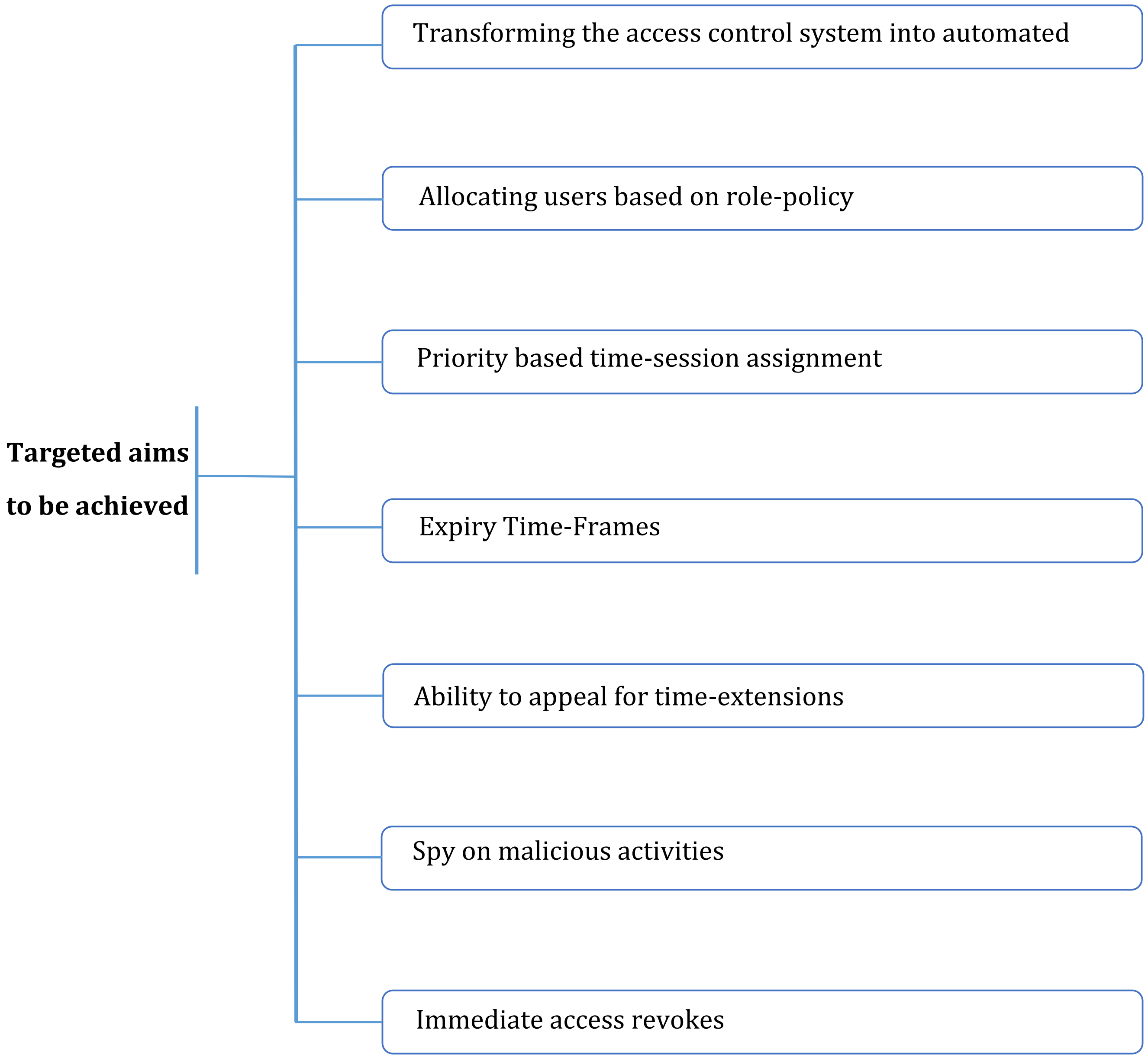
STATEMENT OF THE PROBLEM

Contemporarily, the issue of the system is that the whole system operates at the hands of human, which is more likely to be an error biased system. The possibility of getting exposed to a fault at any time due to the human intervention is highly risk and can place the whole network schema in vulnerability leading to threats from outside users who visits temporarily and connect to the organisation temporarily.

The manual based system comes with some certain points of flaws and loopholes;

- Inefficiency in managing the network access control system
- Dealing out with the access for the users can be slow and error prone
- Complexity in following the role-based policy in assigning users
- Assignment of the time frames for each user can get messier
- Identification of time slots and the user to be expired is challenging
- Observation of each access enrolment for unusual acts
- Potentiality of internal network being exposed to threats
- Finding & identifying the precise user is complex
- Immediate cancellation under different circumstances can be problematic

INTENDED OUTCOMES



Targeted aims to be achieved

- Transforming the access control system into automated
- Allocating users based on role-policy
- Priority based time-session assignment
- Expiry Time-Frames
- Ability to appeal for time-extensions
- Spy on malicious activities
- Immediate access revokes

PROPOSED-SOLUTION

DATA STRUCTURE SELECTION

As the main goal for the current situation of the system can be resolved by transforming the network access control panel into a system of:

- Software-based automatic extranet access control system
- Time-based

In addition, in accordance with the requirements as mentioned in the intended outcomes of the scheme the most appropriate data-structure to apply for the development of this particular **software-based automatic extranet access control system** and **time-based** will be:

Selected Data Structure → *Min Heap Data Structure*

HEAP DATA STRUCTURE OVERVIEW

Definition:

> "A heap is a complete binary tree which satisfies the heap property: the key at each node is greater than or equal to the key at its parent node in a Max Heap, or less than or equal to the key at its parent node in a Min Heap" (GeeksforGeeks, 2024).

Min Heap is a complete binary tree data structure which means the value of each parent-node is always less than or equal to the values of its child-node. Hence, making the root node as the node with minimum value of the whole tree. Consequently, Min heap will make it efficient with the fact that access to the smallest element at any time.

CHARACTERISTICS OF THE MIN-HEAP

- Binary Tree Structure

Node-Structure:-User-ID,-Role, Expiry-time-&-Access-Scope

Each node holds the info of each user ID, Role Expiry time frames allocated and the access-scope. Binary tree is fundamentally beneficial for such scenarios as storing them in an orderly manner where access and traverse must be done efficiently and quickly.

- Priority-Based Order

Prioritize-the-Earliest-Expiry-Time

Guarantees that the user with the earliest expiry time, means minimum time-slot bearer is always the parent node or the root, ultimately permitting the process give priority to revoke of access session spared efficiently

- Fast Access to Minimum

Fast-Insertion-and-Removal-in-Logarithmic-time

Instead of searching and scanning the whole tree Min heap is able of identifying who to be removed or expired first, since root node holds the minimum value out of all. It's crucial for automated access cancellations and notification process.

- Dynamic

Optimal-Ordering: -Always-keeps-the-earliest-expiry-at-Root

As elements or users are added or deleted, the Min Heap is capable of dynamically maintaining the correct order, keeping the earliest expiry or the key with least time slot that is determined to be removed from the tree at root

JUSTIFICATION

With this system's time-based temporary access control requirements, Min Heap is the most suitable data structure due to its efficiency and alignment with priority driven access control systems. This can be justified with different categories.

Time-Based Access: This system manages temporary access of the users according to a pre-defined validity period. Each user is assigned with an expiry time based on their role which will be efficiently manages by the Min heap with time-based priorities.

Prioritize Earliest Expiry Time: Since users with the earliest expiry time should be removed according to the system, by organizing users in a min heap, the system will automatically prioritize the earliest expiry of the user access. This ensures that the notifications and access removal are performed in the correct order.

Fast Scanning: The system does not need to scan all the node to find the next earliest expiry since min heap always keeps the earliest expiry user at the root node. Therefore, by scanning the root node, the system can find the next earliest expiry user which improves performance in a system with multiple temporary users.

Efficient Operations: The min heap allows the critical operation within the system such as insertion and deletion to be performed efficiently with a low time complexity.

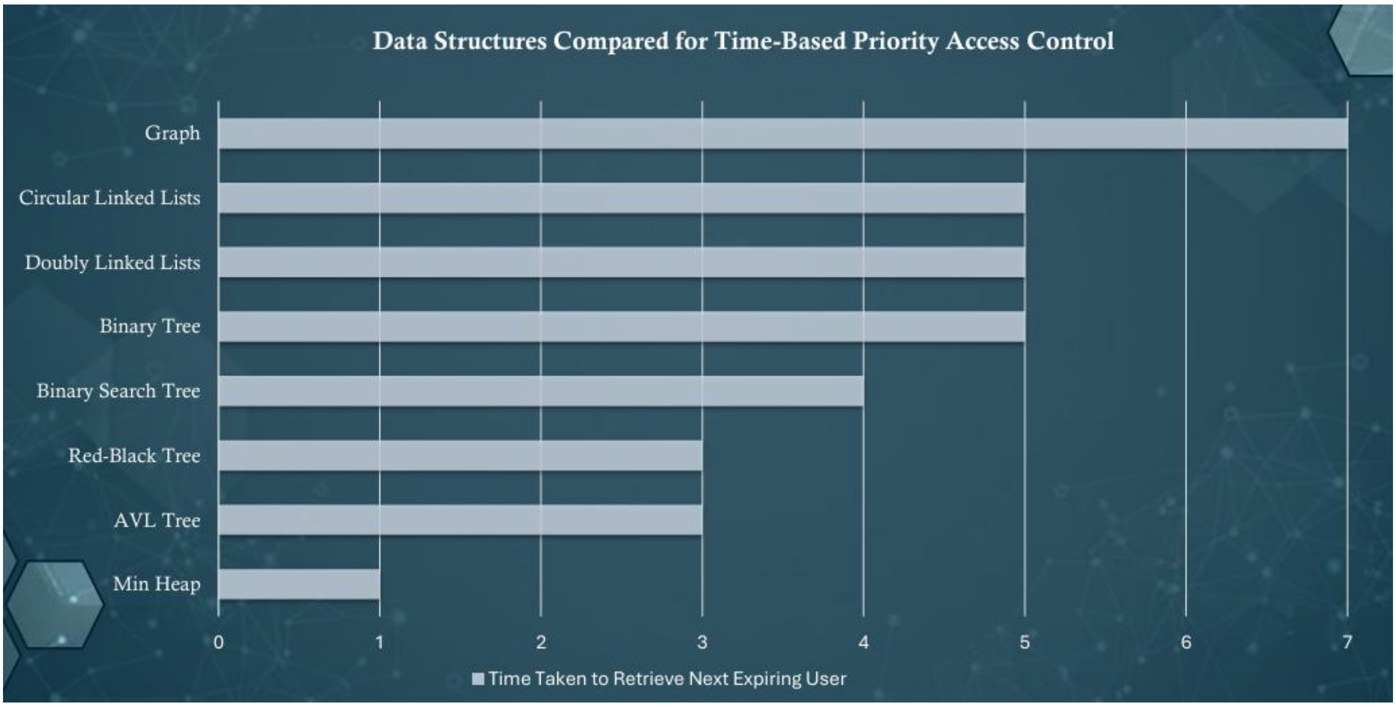
- Insertion – O (log n)
- Deletion – O (log n)
- Search – O (1)

Both the insertion and deletion operations have logarithmic time complexity meaning it scales efficiently even the numbers of the users grows making it ideal for the system scalability.

After considering all the factors above it is clear that by using a Min heap the system achieves efficient, scalable and automated access control management making it ideal for temporary, priority driven user access**.**

COMPARATIVE ANALYSIS

The below graph is a depiction of how relative time bound with the given data structures to reclaim the user to be expired soon. Scanning multiple elements will require time with the structures like graph and link list. Balanced tree preferably for performance yet require for traversal process. In comparison Min heap maintains the earliest removal at root, endowing direct access in constant updates in logarithmic time. Therefore, in conclusion min heap is the most appropriate, efficient and scalable solution for our network scenario; temporary access control system.



WORKFLOW OF THE SYSTEM

The system operates through an organized and automated process to effectively manage temporary extranet access.

➢ Access Allocation

Initially the workflow of the system starts by gathering user information upon user requests through a digital form. The gathered user information includes the role, access scope based on the role and some identification information. Once the system gathers the information it automatically generates a User ID and Expiry Time based on pre- defined policies within the organization that associated with the user role.

Afterwards, with the gathered user information and generated UID & expiry time, the system creates a heap node. The created heap node is then inserted into the Min Heap ensuring an expiry time prioritized user organization. Once the node is successfully inserted into the heap, the network access is granted for the user.

ALGORITHM IN STEPS:

Step 01: Gathering user information through a digital form.

↓

Step 02: Assign User ID and Expiry Time Automatically

↓

Step 03: Create the Heap node with user information. (UID, Role, Access Scope, Expiry time)

↓

Step 04: Insert the user into the Heap

↓

Step 05: Enable User Access

➢ Expiry Enforcement & Extension

This system is designed to notify the user 5 minutes before their access expiry. Hence, the root of the Min Heap is continuously monitored by the system to check the next earliest expiry. Therefore, if the difference between current system time and expiry time is less than or equal to 5 minutes, the system sends an extension notification to the user.

If the user approves the extension notification the system updates expiry time stored in the min heap and then rearrange the heap accordingly without interrupting the active user session. And if the user declined the extension notification, the system automatically revokes the network access of the user and removes the corresponding node from the heap.

Additionally, if the system detects an inactive session (Idle user, Network disconnection…) it removes the node with the detected UID from the min heap to ensure accurate access management even before the expiry time exceeds.

ALGORITHM IN STEPS:

```
Step 01: Checks the Root of the heat for next user to be expired.
                                 ↓
Step 02: Check if;
                   current time >= Expiry Time
             If yes;
                                 ↓
Step 03: Sends an Extension request notification to the user.
                                 ↓
Step 04: If user needs to extend the Time;
            Update Expiry time in the node.
            Rebalance the heap if needed
                                 ↓
Step 05: If user don't need extension;
            Remove the user (Root) from heap
            Revoke Access.
            Update the Heap
```

➢ Unusual Behaviour Detection

While the system operates continuously, if any unusual or suspicious behaviour is detected by the monitoring systems, it sends the UID of the user to this system. Once the system receives the UID, it searches for the node that contains the respective UID within the min heap and removes that node from the min heap and network access is revoked immediately without waiting for the expiry time to be exceed. After removing the node, the heap is rebalanced accordingly ensuring secure time-based access control.

ALGORITHM IN STEPS:

Step 01: Receives detected user's UID.

↓

Step 02: Find the node of that UID within the heap.

↓

Step 03: Remove the User from the heap.

↓

Step 04: Update the Heap accordingly.

↓

Step 05: Revoke the user access immediately.

USE CASES IN REAL WORLD

The ability of the min heap in creating solution for the addressed issue within thesis of our project demonstrates how the temporary network access control system will be operated via an automatic and time-based priority system. Hence, it can be determined that the proposed solution, the concept invented will not be limited to the project scenario but also a valid developed model system software for several cases as briefly mentioned below;

NAC (Network Access Control) System

The authenticated guest visitors who logs on to the system temporarily are allowed with permission to access the extranet for a limited period. Towards the end of their session, with expiration time frame gets disconnect from the network. It is complex and inefficient under the manual management for the process once the number of users increases. Will be administered through the automatic software based system which operates via the data structure of Min heap fundamentally, keeping order of expiry slots, allowing the control panel handle session ends automatically, identifying users for mal-activities and immediate annulment execution.

VPN (Virtual Private Network) Session Management

The system is usually for remote users within an organization who would want to connect to the company network via VPN. As the number of users increase, they will be managed via a same Min heap conceptual system as this project suggested. The employees, Consultants partners will have active sessions via VPN environment and VPN basically will need to deal with maintenance of multiple sessions simultaneously, session timeout determination and forced disconnections for urgent cases. Using a Min heap powers up a system like VPN session management by tracking the sessions efficiently, termination of connections at early leading to minimum overhead, improving both performance and security.

Firewall Timeout System

This also falls under the same concept of using Min heap to automate the whole procedure in firewall timeouts. Principally, the firewall is responsible of maintaining thousands of temporary connection cases and states for active-traffic-flow filtration process. And Each connection requests for inactivity and timeout periods which is basically removal of memory to make it free and to prevent stale sessions. With the usage of Min Heap, it upkeeps for organizing the timeout periods and efficient clean-up of stale connections even without scanning the entire table of session expired culminating a better version of the system faster, scalable and secure.

Visitor Wi-Fi Automatic Management System

In private general places like hotels, offices, schools or campuses with guest networks, it more common to have a visitor Wi-Fi for the internet access. Users access the Wi-Fi with given credentials and after each log-on, the session must be under limited data and internet access. In addition, it must be exhausted and expired automatically with the exceeding of the limits. The same mechanism discussed above, 'Min Heap' will maintain a large number of temporary loggings while ensuring the efficiency and access grant and revocation with zero intervention from human beings.

FUTURE-ENHANCEMENTS

1. Intelligent Behaviour Analysis

Within the current system the users with unusual behaviours/activities are removed based on pre-defined rules within the organization. The proposed enhancement using intelligent behaviour analysis we can improve the system's detection accuracy. For instance, by monitoring normal usage patterns the system can detect hidden inconsistencies such as irregular source usage and unusual login times.

Working Strategy:

- ➢ System continuously absorbs normal user behaviour profiles.
- ➢ When a user's activity changes significantly, the system identifies it as suspicious.
- ➢ The user's access is revoked as same as the existing mechanism.

Benefits:

- ➢ Early detection of potential security threats.
- ➢ Minimize false positivists.

2. Integration with Real Network Access Controllers

The current access system is a virtual based system. With this enhancement we can integrate the Min Heap based system with the real-world network access controllers such as authentication servers, firewall gateways etc.

Working Strategy:

- ➢ Heap logic runs on a central controller.
- ➢ When a user is removed from the heap, the network access is immediately disabled on the actual network device by the controller.
- ➢ Min Heap logic remains unchanged.

Benefits:

- ➢ Ensure real-time enforcement of access policies.
- ➢ Improves reliability, scalability and practical deployment.

FINAL-ANALYSIS

In brief, schema of the project itself is a reflection of how a manual based is transformed into an automated system generated for controlling a system used for temporary access admission for the extranet network for external users. The weaknesses that was lied within the traditional way of conducting and performing seems to be impractical, inefficient and insecure for the organization.

The real challenge was not to identify the problem but to follow the strategic movement by providing it with the most suitable and appropriate data structure and generating an algorithm accordingly. The several issues beneath the manual system prevailed; allocating users in comply with role-policy regulations of the organization, automatic revocation of the users after the limited period assigned, and annulment of the user session at will and emergency cases like detecting malicious activities.

However, with the suggested solution, applying the data structure Min Heap carried us to great lengths while solving all issues exist with the manual handling while delivering a better improved version of the intended the automated software based. Rather constructing the system with general abilities of access grant and automatic revocation the newly designed software model will offer abilities in appeal for time extensions, time-slot decision making that complies with regulations of the company and monitor for abnormal activities and immediate user session access elimination.

In conclusion, the thesis is a significant breakthrough for a problematic situation under a network circumstance as discussed above and a tactically calculated practical solution aligned with few more improved features by subtly approaching the addressed matter with the data structure Min heap.

REFERENCES

- GeeksforGeeks. (2024). *Heap Data Structure*. Retrieved February 9, 2026, Available at: https://www.geeksforgeeks.org/heap-data-structure/

- Programiz – Heap Data Structure, Available at: https://www.programiz.com/dsa/heap-data-structure

- Cisco – Network Access Control (NAC) Basics, Available at: https://www.cisco.com/c/en/us/products/security/identity-services-engine/index.html

- Cloudflare Learning – VPN Explanation, Available at: https://www.cloudflare.com/learning/access-management/what-is-a-vpn/

APPENDIX A

Pseudo Code for Access Allocation

```
FUNCTION AllocateAccess(UserForm):
  UID= GenerateUserID()
  ExpiryTime = AssignExpiry(UserForm.Role)
  Node = CreateHeapNode(UID, UserForm.Role, UserForm.AccessScope, ExpiryTime)

  HeapInsert(Node)
  EnableAccess(UID)
END FUNCTION
```

Pseudo Code for Expiry Enforcement & Extension

```
FUNCTION CheckExpiryAndExtension():
  RootNode = HeapRoot()

  IF CurrentTime >= RootNode.ExpiryTime THEN
    NotifyUserForExtension(RootNode.UID)

    IF UserWantsExtension(RootNode.UID) THEN
      RootNode.ExpiryTime = GetNewExpiryTime()
      Heapify(RootNode)
    ELSE
      HeapRemoveRoot()
      RevokeAccess(RootNode.UID)
    END IF
  END IF
END FUNCTION
```

Pseudo Code for Unusual Behavior Detection

```
FUNCTION HandleUnusualBehavior(UID):
        Node = FindNodeInHeap(UID)

        IF Node EXISTS THEN
            HeapRemove(Node)
        RevokeAccess(UID)
        END IF
END FUNCTION
```

APPENDIX B

GitHub Repository Link

Link: https://github.com/mvsfernando190-create/DSA_Project-HNDNE-25.2F

Follow the above provided link to navigate further for the complete project files including README, algorithm documentation and presentation materials.

The repository contains:

- README.md with project details with a brief explanation
- algorithm/ folder with pseudocodes
- pptx-report/ folder with presentation(.pptx) and report (.pdf)