Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2019, 6(7):38-46

Research Article ISSN: 2394 - 658X

Design and Implementation of Allocation Algorithm for Networks with Scarce Resources

J.V. Joshua[1], O.D. Alao[2] and P.O. Lamina[2]

[1]Department of Software Engineering, Babcock University, Ilishan-Remo, Nigeria
[2]Department of Computer Science and Information Technology, Babcock University, Ilishan-Remo, Nigeria
joshuaj@babcock.edu.ng

ABSTRACT

Network is simply two or more computers that are connected to share resources. Network resources include software, hardware, and files.

In a network computing environment, we have substantial access to resources by many users; these users are competing for limited resources especially when there are several concurrent users.

These resources are normally administered by network administrator, and this lead to poor management or allocation of network resources.

In this paper, we design and implement a real time automatic network resource allocation algorithm for scarce resources in a network setting with emphasize on memory. This ultimately will help the network administrator to administer resources effectively and efficiently.

We adopted waterfall methodology and the algorithm was implemented using Java language

Key words: Algorithm, Memory, Resources, Network, User, allocation

INTRODUCTION

In recent years, many tools have been used to model and solve a variety of practical resource allocation issues, including optimization theory [1]. The allocation of resources in the computing system is therefore a pressing research topic with huge applications. Computing and information technology has become the center of productivity, communication and link to every other activity engaged by man. The power of computing comes from the core engine called Algorithm. Algorithm is a sequence of simple steps that can be followed to solve a problem [2]. These steps must be organized in a logical, and clear manner. These algorithms are built into systems and then powers such systems. Whatever systems runs on information technology, be it accounting, medical, communication, governance and others runs by the power of the algorithm built into it. Therefore, even though algorithm may be intangible and are usually coded, they are the power of computing.

Also from economics we studied how to manage and make appropriate choice from scare resources because wants are always more than resources available [3]. The process of choosing or allocating scarce resources to meet the needs requires a great planning [4]. When such process will constantly be repeated, there is need to formulate the procedure to be followed at every time. In modern day activities where computer will be at the center of the operation, such system will require that the procedure be computed as algorithm in the system.

Network memory or disk space, typically measured in kilobytes, is quantified as the amount of disk space mapped for network use. In the past, external physical memory is attached to the server system or in some situations, except for domain networks, no means of sharing memory over the network [5]. In today's computing paradigms, collaborative computing will always be needed, and sharing infrastructure including workspace or directory will be one of the best ways to achieve this. Because of undue access and the catastrophe of overrunning the limited space available, maintaining a network sharing place may not be an easy task. The current operating systems may have management directory functionalities, they may not be able to allocate and limit the volume or limit to which the size of allocated directories may grow in share memories. This paper is to design and implement an application that will make it possible to create share drive and directories inside the share drive for network users but with limited sizes such that any directory that is growing beyond its specified size will no longer be permitted to add volume to such directory.

PROBLEM STATEMENT

The challenge of poor management or allocation of network resources manually by the network administrators every time the need arises necessitate the need to have an automated network allocation algorithm which can help the network administrator to administer network resources effectively.

AIM AND OBJECTIVES

The aim of this paper is to design and implement an algorithm for managing network resources with emphasize on memory. The specific objectives are to

a. create a shared drive for storage of files.
b. create users that will use the shared drives.
c. allocate memory to shared drives according to users need

SYSTEM REQUIREMENTS

In the development of computer systems, the requirement for the effective operation of any system must always be defined. The basic requirements or minimum system requirements are always stated. We will specify the requirements in the following sub-sections.

Functional Requirements

These are service statements that the system should provide what reactions the system expects when certain inputs are delivered and the system's expected behaviour in specific situations. The functional requirements are listed below:

- The system should allow admin login using their username and password.
- The system should allow the administrator create new share drive.
- The system should allow the administrator to list users that can access the drive and their limit.
- The system should allow the administrator to add new users to the shared drive and allocate space to them.
- The system should allow the administrator to delete users from the shared drive.
- The system should allow added users to access and use the shared drive.
- The system should not allow users that have reached the allocated limit of space to add files or data to the shared drive.
- The system should allow users to delete files from the drive.

Non-Functional Requirements

They define system properties and constraints on functions offered e.g. reliability, response time. This are listed below:

- Portability: The system is platform independent.
- The applications failure occurrence has to be minimal.
- The applications response to requests should be fast.

User Requirements

This describes the needs, desires and expectations of end users from a system. These requirements are listed below:

- Administrator shall be able to login using their username and password.
- Administrator shall be able to manage share drives.
- Administrator shall be able to remove users from shared drive accessibility.
- User should be able to add files or any form of data into the shared drive.
- User should be able to delete data/files from the shared drive.

Hardware Requirements

The following are the minimum hardware requirements

- Processor: 1GHz speed and above
- RAM: 512 MB or higher
- Hard disk: 20GB or higher

Software Requirements

The following are the require software specification

- Operating systems: Microsoft Windows, Linux/Unix, OS X
- Java Virtual machine

ALGORITHM

An algorithm, named after the ninth-century Persian mathematician Abu Jafar Mohammed bin Musa al-Khwarizmi, is simply a set of rules for carrying out some task, either by hand or, more usually, on a machine [6].

Good algorithm design is, therefore, essential for the performance of all software systems. An algorithm should have the following characteristics [7-8]

- **Unambiguous** − Algorithm should be clear and unambiguous. Each of its steps *or* phases, and their inputs/outputs should be clear and must lead to only one meaning.
- **Input** − an algorithm should have 0 or more well-defined inputs.
- **Output** − an algorithm should have 1 or more well-defined outputs, and should match the desired output.
- **Finiteness** − Algorithms must terminate after a finite number of steps.
- **Feasibility** − should be feasible with the available resources.
- **Independent** − An algorithm should have step-by-step directions, which should be independent of any programming code

We discuss two types of resource allocation algorithm namely

- Bidding algorithm and
- Round Robin algorithm

Bidding Algorithms: This uses the bidding concept to determine how processes can be allocated to the different core of the system. Bidding algorithms have been widely used as part of auction-based algorithms to solve several optimization issues. There are typically n possible "bidders" for a specific number of items, with each "bidder" placing a bid in an attempt to "buy" a number of items [9].

Round Robin Algorithm: It is one of the oldest, simplest, fairest and most widely used algorithms for scheduling, especially designed for time-sharing systems. A small unit of time is defined, referred to as time slice or quantum [10]. In a circular queue, all runnable processes are kept. The CPU scheduler handles this queue, allocating the CPU for a time interval of one quantity to each process.

DESIGN METHODOLOGY

The Waterfall development methodology, fig 1 is the software development model used in this system. In the waterfall model, requirements are fully gathered before the design and development of the system, so that after deployment the built system cannot be modified except for another version.

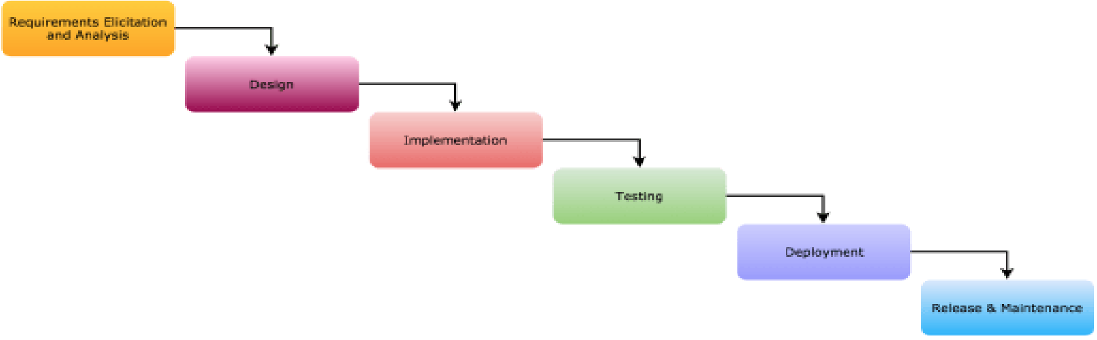


Fig. 1 Phases of waterfall model

Although this may not seem the most appropriate for today's dynamic system, the methodology choice depends on the factors listed below

1. The nature of the project
2. The choice of programming language
3. The scope of the project

We implemented the use of file management across the network which is a hardware task which involved programming the system. To achieve this, the system was built as native application that accommodates hardware attached to a local system to work. The nature and scope of the system as a desktop application and not a web application makes the deployment to be constraint to being an executable system which when deployed cannot be modified since it is going to be distributed and meet the security requirement. Java is the programming language used for the implementation, because java is vast in its application and can support cross platform applications. The system is defined to be used locally with maximum security which made the choice of the development methodology to be confined.

ANALYSIS OF THE SYSTEM

The scope of this project is to manage network resources and in this case memory specifically. The system will work such that the administrator will deploy the application on any system that will allow for the creation of a network share drive. The system will allow the network users to access the drive from network. The main difference of this system from windows sharing system is limit to which a network user can use the drive. A limit of size of content that a user can accumulate on the share drive can be set such that when the limit is exceeded, the drive will no longer expand and the user will no longer be permitted to be able to add more content.

SOFTWARE MODEL AND ARCHITECTURE

Software architecture is the physical and functional architecture that depicts the structural and operational process and representation of an integrated software product [11].

Unified Modeling Language (UML) is "a graphical language for visualizing, specifying, constructing and documenting the artifacts of a software-intensive system [12]. The UML diagram was used in the modeling of the system

Use cases are the primary modeling tool of UML employed to define the behaviour of the system, it describes how the user interacts with the system to perform some activity. It is use to identify and communicate the requirements.

Administrator

The administrator is responsible for creating shared drives and adding users to the system who have access to the drive. The admin creates users and allocates memory to them and when the memory allocated is reached the user can't add more data or files to the shared drive. Fig. 2 shows the administrator use case diagram.

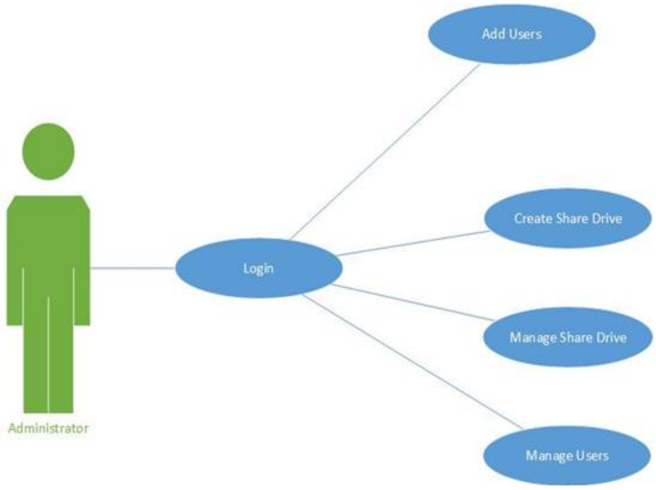


Fig. 2 Administrator use case diagram

User

The user is added by the administrator and they can add files to the shared drive for storage or to share to other users. When the allocated memory is reached they can delete files in the shared drive and add more files as long as the limit is not reached. Figure show the user use case diagram.

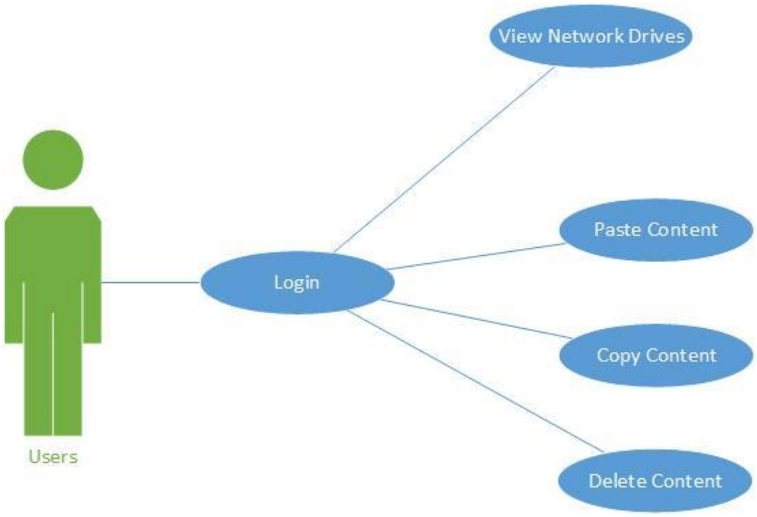


Fig. 3 User use case diagram

The allocation algorithm is depicted by the system flowchart in Fig. 4.

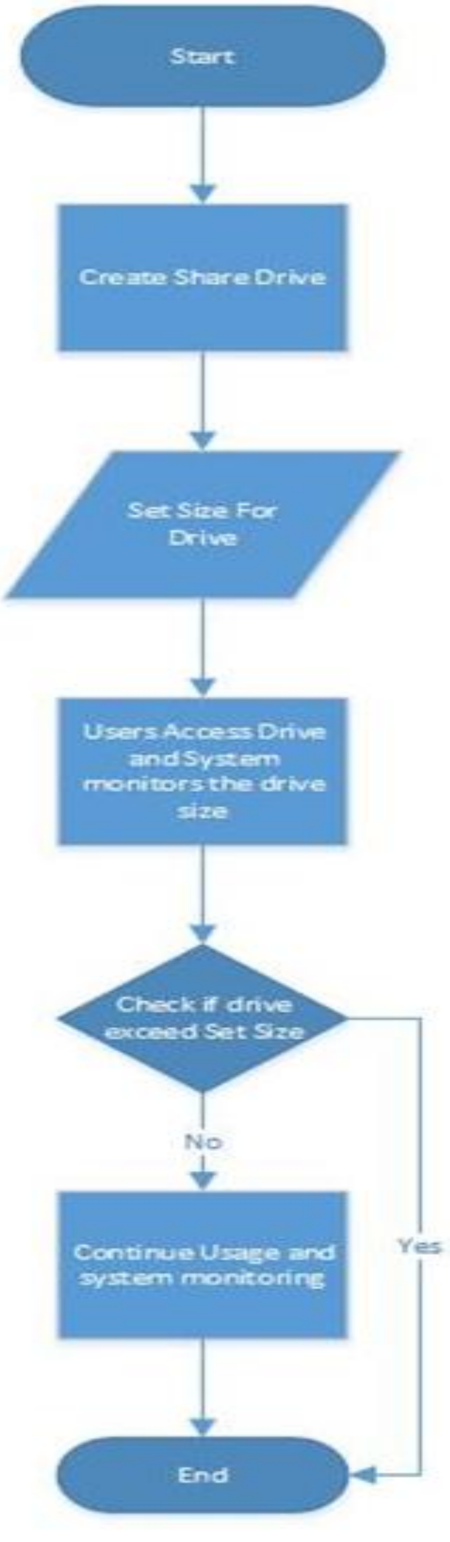

Start
Create Share Drive
Set Size For Drive
Users Access Drive and System monitors the drive size
Check if drive exceed Set Size
No
Yes
Continue Usage and system monitoring
End

Fig. 4 System Flowchart

RESULTS AND DISCUSSION

When the system is started, a login page will be displayed where a user (administrator) is required to login to the system with a valid username and password. Then the application will be open to the menu page depending on the user. The menu page will now be displayed with the various menus. The operations that the administrator can perform include:

1. Adding users.
2. Creating Shared Drives.
3. Removing Users from share drive.
4. Managing share Drives.

We described the user interface pages designed for the system as follows

Login page

This is the first page or the welcome page. It will require admin to login with username and password, the diagram in Fig. 5 displays the login page which prompts the administrator user to enter his username and password

Menu page

After a successful login of the administrator, the menu page in Fig. 6 will be displayed from where an option can be selected based on the operation to be performed.

Add/Create new Share drive

This is the page from where the administrator will create share folder on the network and map how users will use such folder. In this menu, you create the drive by selecting a path where the folder will be stored and when selected you supply the drive name to create the drive, as shown in Fig. 7

When the directory has been selected the shared drive is stored there and a success message is displayed as shown in Fig. 8

Add Users

This menu allows the administrator to create users by giving the user username and password for the use of the drive on the network as shown in Fig. 9

Manage Users

This menu enables the administrator to manage all the network users by changing the limit that applies to each user. Fig. 5 diagram below displays the login page which prompts the administrator user to enter his/her username and password.

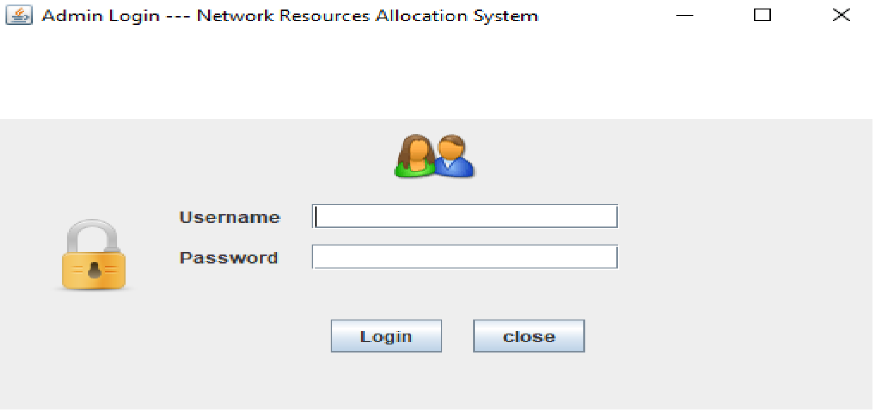


Fig. 5 Login page

On successful login, the menu page indicated in Fig. 6 will be displayed from which we can choose what task to perform on the list

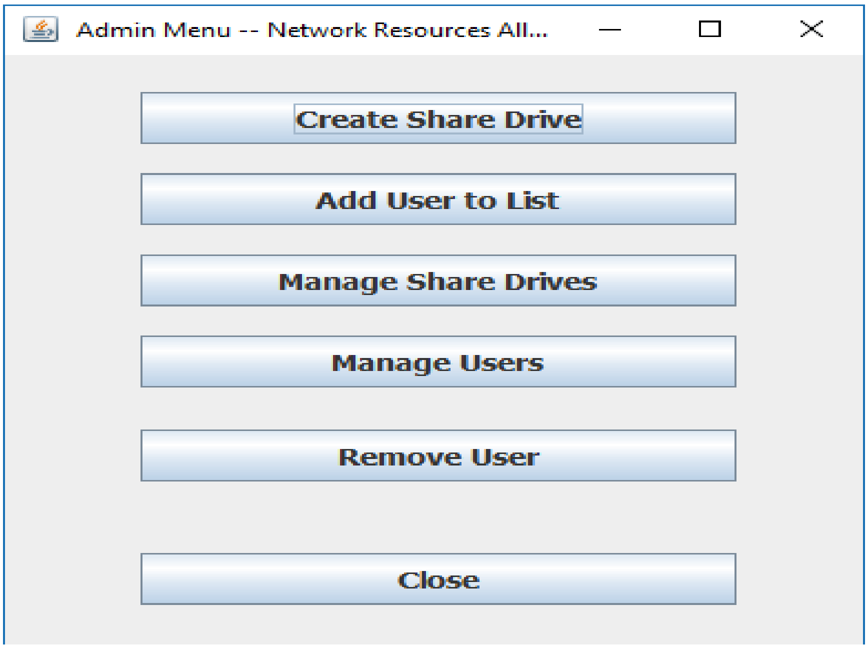


Fig. 6 Menu page

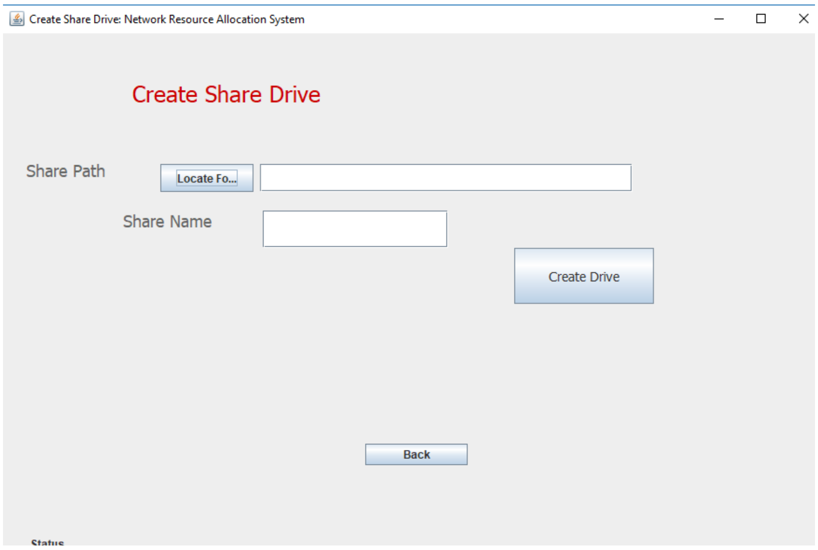
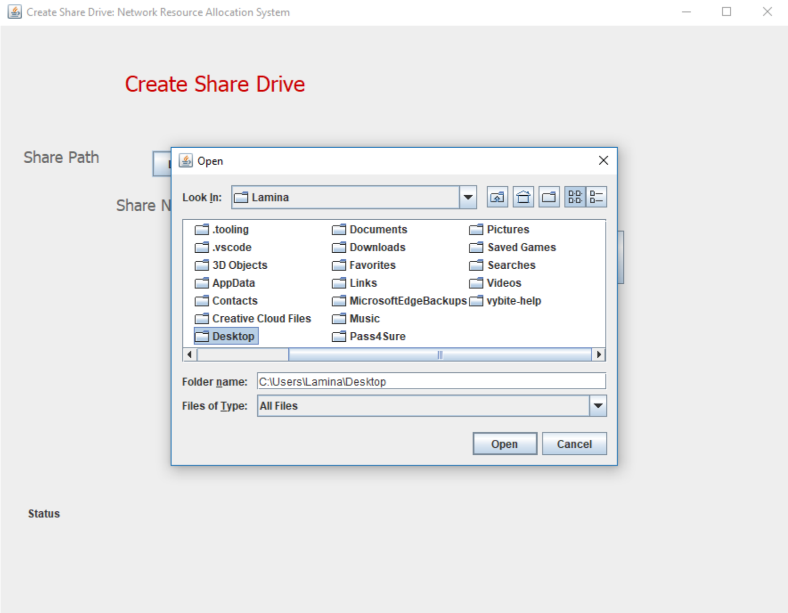


Fig. 7 Create share drive window



Look In: Lamina

Folder name: C:\Users\Lamina\Desktop

Files of Type: All Files

Fig. 8 Select shared drive dialog window

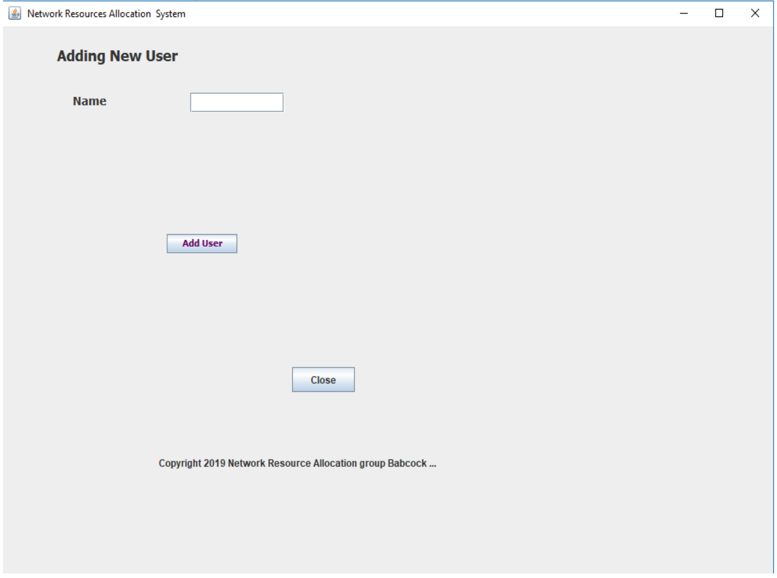
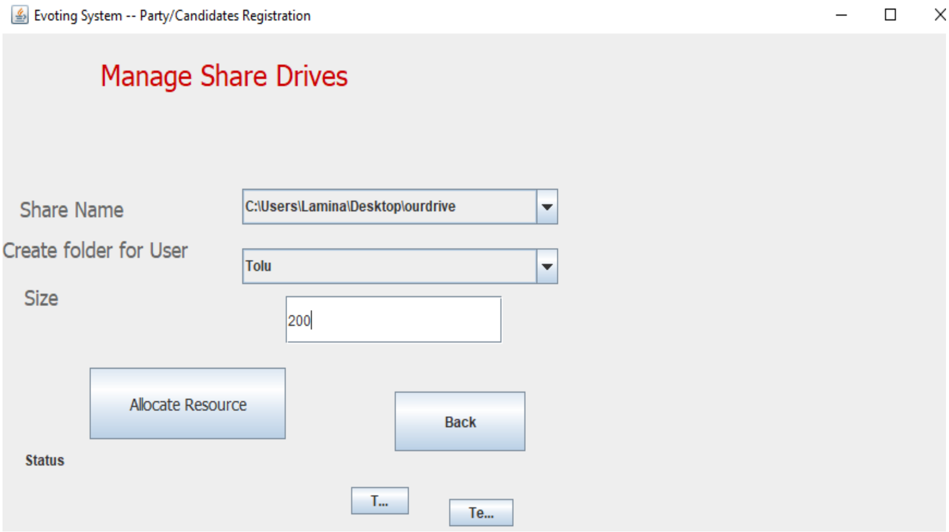

Network Resources Allocation System
Adding New User
Name
Add User
Close
Copyright 2019 Network Resource Allocation group Babcock ...

Fig. 9 Add new user window



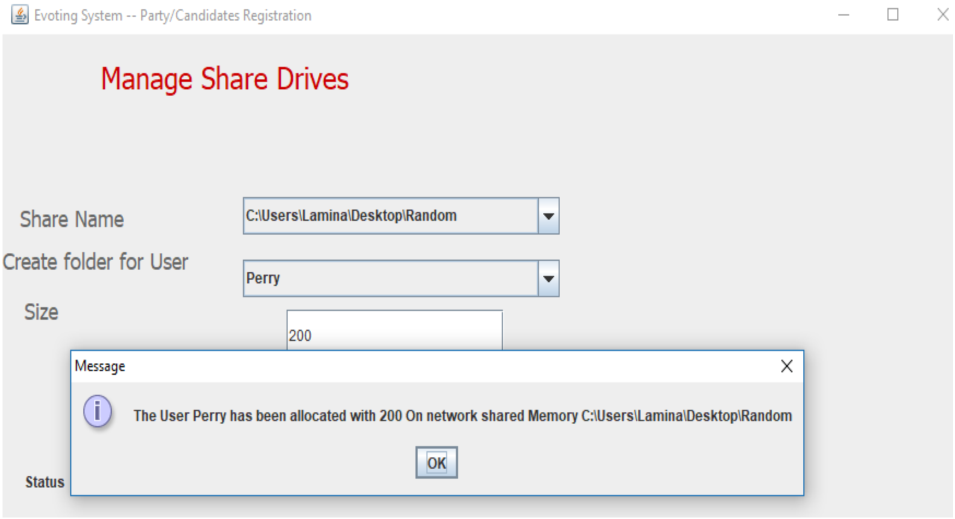
C:\Users\Lamina\Desktop\ourdrive

Tolu

200

Fig. 10 Manage share drive window

When resource is successfully allocated to the user a success resource allocation dialog will be displayed as shown in Fig. 11



C:\Users\Lamina\Desktop\Random

Perry

200

The User Perry has been allocated with 200 On network shared Memory C:\Users\Lamina\Desktop\Random

Fig. 11 Success resource allocation dialog window

CONCLUSION

The Concept, design, construction and implementation of the system clearly demonstrates how resources can be allocated on networks with scarce resources. In this paper, the scarce resource is “Memory” and in the implementation of the system we showed how memory can be allocated with shared drives to multiple users with a limit set and we built this form of allocation using a file system on the computer. This is not a new system but what makes this system different is the input of the limit to how much memory can be allocated to the users on the shared drive.

REFERENCES

[1]. Yi Su, Fangwen Fu, and Shuo Guo (2013). Resource Allocation in Communications and Computing. Journal of Electrical and Computer Engineering Volume 2013, Article ID 328395 2 pages. https://doi.org/10.1155/2013/328395

[2]. Programming Building Blocks Accessed on 6/4/19 from http://www.dcs.bbk.ac.uk/~gr/itapps/PBB.pdf

[3]. Economics and finance, Khan Academy Accessed on 19/4/19 https://www.khanacademy.org/economics-finance-domain.

[4]. Resource Allocation in Management: Methods, Process & Strategy. Accessed on 27/4/19 https://study.com/academy/lesson/resource-allocation-in-management-methods-process-strategy.html.

[5]. Wordpress. (2019). Accessed on 18/5/19https://matteocappelli.files.wordpress.com/2011/02/storage_basics.pdf.

[6]. Varsha H. Patil (2012). Data Structures using C++, 1st edition, Oxford University press.

[7]. https://www.tutorialspoint.com/data_structures_algorithms/dsa_quick_guide.htm. Accessed on 21/5/19.

[8]. Knuth D. (2014). The Art of Computer Programming Volume 1: Fundamental Algorithms, 3rd Edition. Addison Wesley

[9]. Greenwald, Amy & Boyan, Justin. (2004). Bidding Algorithms for Simultaneous Auctions: A Case Study. Autonomous Agents and Multi-Agent Systems. Proceedings of the 3rd ACM conference on Electronic Commerce.

[10]. Abass Noon, A. K. (2011). New Round Robin Based Scheduling Algorithm for Operating Systems: Dynamic Quantum Using the Mean Average. IJCSI International Journal of Computer Science Issues, 8(3), 5.

[11]. Schmidt, Richard F. (2013). Software Engineering: Architecture-Driven Software Development. Elsevier Science and Technology Books Inc.

[12]. Weisfeld, Matt A. (2008): The object-oriented thought process, 3rd ed, Addison-Wesley.