WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

An Improved Database Fragment Allocation Technique

Ezeugbor I. C[1], Okolo C. C[2], Ogbunude F. O[3], Asogwa D. C[1]

[1]Department of Computer Science, Nnamdi Azikiwe University, Awka
[2]Electronics development institute, Federal min. of science and technology, Awka capital territory, Anambra state
[3]Department of Computer Science,Akanu Ibiam Federai Polythecnic, Unwana, Afikpo, Ebonyi state

ABSTRACT

In recent time, different industries and organizations use distributed system for their day to day business transactions in different geographical zones. Issues relating to the complexity, maintenance, performance and communication cost of data in distributed data warehouse for processing of query, according to the request of end users from different zones have critically been observed. Distributed system enables the allocation of data as fragmented, duplicated and distributed over the Intranet or Internet within and crosswise an organization. In this work, we developed an improved database fragment allocation technique. Algorithm-genetic was deployed to allocate database fragments to individual sites haven collated these distributed systems to one central store. Object oriented analysis and design methodology (OOADM) was utilized in the methodical study and design of the system. To achieve this system, bootstrap, cascading Style Sheet (CSS) and JavaScript were employed. MongoDB database engine was introduced to obtain the database of the system. The obtained result is a system with external fragmented database that has capabilities of storing and saving to a central database. This system can be put into practice by every individual, organizations and establishments that have need to allocate fragments in a distributed system for a maximum reduction of process time, for easy access of data and complexity reduction.

Keywords: Improved Model, Fragment Allocation, Distributed System, Genetic Technique, Communication

INTRODUCTION

The high rate complexity of information and communication technology (ICT) development in all dimensions of life, including the field of computer science has triggered the crucial need for designing an improved database allocation technique [1]. The worry for this has shown a distributed system as being pertinent in achieving better efficacy and easy retrieval of information in establishments; this is particularly observable in the banking system in Nigeria. It is a well-known fact that every organization has its own peculiar database but there appears to be a requisite to develop an improved database fragment allocation technique in order to be in line with the international best practices [2]. In today's financial organizations, majority of banks, if not all, have their own database system but still have need to be collated and run a fragmented central system for all banks irrespective of banking type or name. There is need for fragmentation because in a distributed system, the cumbersomeness of use of data, the difficulty in accessing and manipulating data and reduction of performance speeds in accessing data pose an undiluted problem in data retrieval. An improved database was developed for allocating fragments that are in distributed systems to one database using Genetic technique. This technique comes into play as to identify the activities which are already fragmented and the ones yet to be fragmented. In order words, there are lots of activities performed by banks in Nigeria, hence, this work is narrowed down to only transactions (credit and debit of account), and also payment of utility bills. Difficulty erupts in accessing and manipulation of data in a distributed system, reduction of performance speed in accessing a distributed system, as well as posing a problem in data retrieval, as such, introduction of fragmentation is very pertinent [3]. Allocation is the act of placing a fragmented site in distributed environment such that the system functions effectively and efficiently. The determination of the non-overlapping fragments that can be the logical unit of allocation necessitates the fragmentation design [4]. Similarly, allocation is the process of assigning a function or an activity to a database. This study focuses on the introduction of fragmentation concept in the distributed system and to enable the placement of data in close proximity to its place of use, which helps to reduce transmission cost and also the size of the relations that are involved, adding the effectiveness and data integrity of the system.

REVIEW OF RELATED WORKS

The presence of decomposition heuristics which brought about the problem of complexity, an author proposed an optimization model for non-replicated data allocation with the objective function to minimize total transaction processing cost [5]. The fragment allocation model needs transaction execution strategies and horizontal fragmentation schema as input. There is an assumption that the cost of the transmission unit is same among any of the two nodes. To contain the repeat of allocation of fragments within the whole relation, a greedy algorithm will be proposed though its only the predefined schedules which is comprised of basic operations are considered. Considering the number of possible vertical fragments of a relation, such approach can't be used with the vertical fragmentation. The study of fragment allocation is based on query processing strategies which treat non-redundant

allocation of fragments in a uniform way [6]. From observation, the problem of achieving a non-redundant optimized allocation for the fragments is NP-hard. The static/dynamic processing schedules, heuristic and optimal algorithms used for the minimization of total cost of transmission are investigated. It is assumed that the heuristic algorithm is a copy for all fragments for each query and this was why it was proposed though it is still very complicated while used during practical because both the number of fragments and number of copies need to be decided. Looking at the complexities of dynamic processing schedules, they are more reasonable but infeasible practically in applications. Another author, proposed an allocation strategy which integrates the problem of relation allocation with query strategy to optimize performance of a distributed database system [6]. To simplify the linear programming solution complications for practical application, it is assumed that the networks are fully connected with equal bandwidth for each link. In a similar way, a certain author went ahead to propose a comprehensive cost model that considers the following; operation allocation, consideration replication, and concurrency control.

MATERIALS AND METHODS

HTML and Cascading Style Sheet (CSS) based design templates for typography, forms, buttons, tables, navigations; modals which are under the umbrella of Bootstrap were used as a structure to help design sites faster and easier. This equally gave support for JavaScript plugins. CSS came into play for the purpose of describing the presentation and design of web pages that includes colors, fonts and layouts. It was practically put in order to enable the distinction between presentation and content, as well as colors, layouts and fonts. JavaScript is widely used as a client side scripting language, hence its dynamic nature, special effects on pages like rollover, and many types of graphics. It can fill up content into a document whenever the user requires it without having to reload their entire page. We equally adopted Nodejs as it is built on chrome's JavaScript runtime for the purpose of building fast and scalable network applications. Jquery was introduced to make much easier for JavaScript to be used at the sites because of its light weight, "write less, do more", JavaScript library. MongoDB database engine was used to implement a data store which provides high performance, high availability and automatic scaling. Npm, a node package manager was used to install packages locally into this work, particularly into the Node_modules folder. Genetic algorithm (Algorithm_Genetic**)** in allocating fragments to individual sites was deployed. Genetic algorithm is considered more attractive when efficiency and solution quality is more important. The aim of this technique is to overcome the problems associated with previous techniques, such as restrictions on the number of network sites, complexity, inefficient solutions etc. The system used genetic algorithms to achieve the result in this work, which contributed in searching the solution space for possible allocations, evaluating them using a developed allocation cost function and eventually finds the best allocation in which in most cases is the optimal allocation. The allocation cost function identifies the allocation status which is computed as a logical value for the comparison between the cost of remote access of the fragment to the cluster and the cost of allocating the fragment to the cluster. The system is out to improve data accessibility, security and speed of operation and data integrity of a distributed database system.

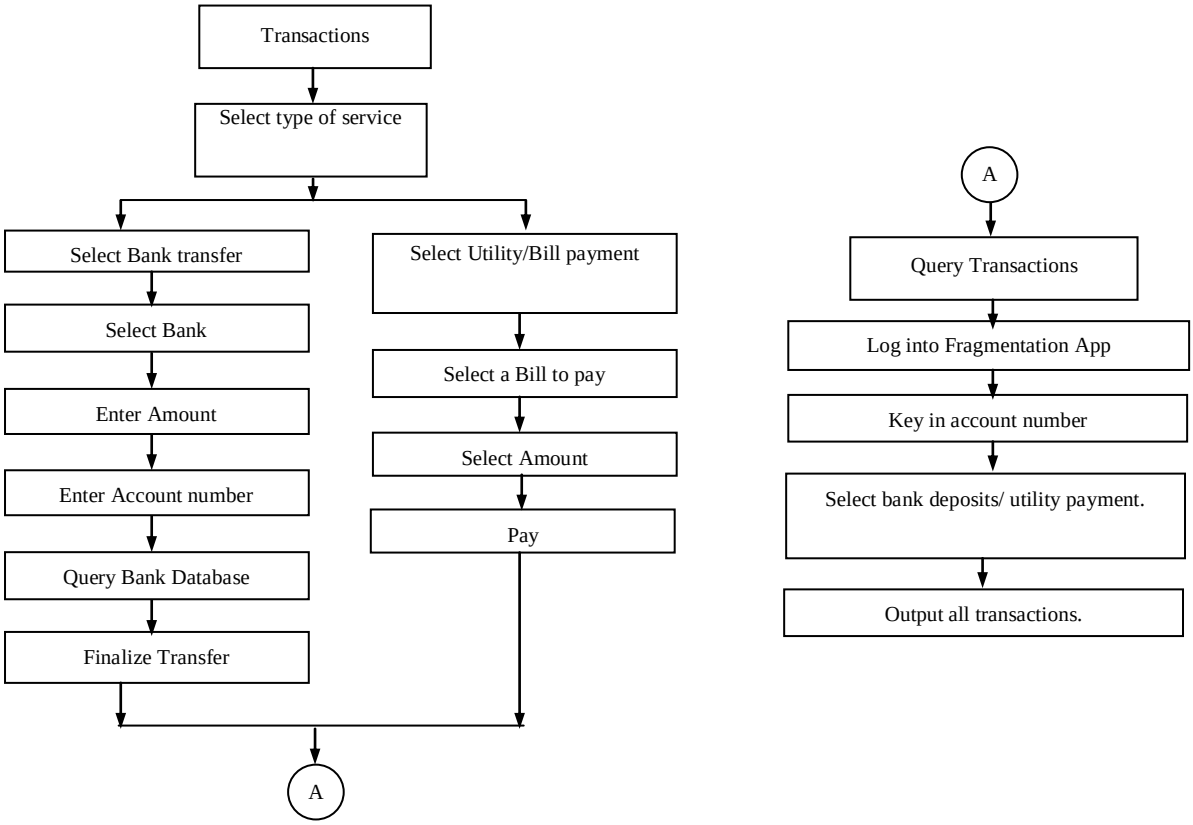


Fig 1: Information Flow Diagram

Design Specification and Algorithms

To solve the problem of taking proper allocation of already fragmented database, genetic algorithm (Algorithm_Genetic and Figure 2) technique was adopted as depicted in Figure 3.

Algorithm_Genetic

1. Generate an initial population, repeating random strings of fixed size.
2. Do the selection, reproduction, crossover and mutation operations of the all population.
3. Replace the old population with the new one.
4. Repeat Steps (2, 3) until number of iterations is finished.
5. Display the best answer found (which has the best-fitness).

Algorithm_Fragment Allocation

Input: K: Number of the last fragment
Rmax: Number of database relations
Nmax: Number of fragments in each relation

Step 1: Set 0 to K
Step 2: Set 1 to R
Step 3: Do steps (4 - 21) until R > Rmax
Step 4: Set 1 to I
Step 5: Do steps (6 - 20) until I > Nmax
Step 6: Set 1 to J
Step 7: Do steps (8-18) until J > Nmax
Step 8:If I ≠ J and ∃ Si ,Sj Є SR
goto step (9)
Else Add 1 to J,
go to step (18)
Step 9: If Si ∩ Sj ≠ Ø
do steps (10)-(17)
Else
Add 1 to J and go to step (19)
Step 10: Add 1 to K
Step 11: Create new fragment Fk = Si ∩ Sj and add it to F
Step 12: Create new fragment Fk+1 = Si - Fk and add it to F
Step 13: Create new fragment Fk+2 = Sj - Fk and add it to F
Step 14: Delete Si
Step 15: Delete Sj
Step 16: Set Nmax + 1 to J
Step 17: End IF
Step 18: End IF
Step 19: Loop
Step 20: Add 1 to I Step 21: Loop
Step 22: Set 1 to I
Step 23 Do steps (24 - 35) until I > Nmax
Step 24: Set 1 to J
Step 25: Do steps (26 - 33) until J > Nmax
Step26:If I ≠ J and ∃ Si,Sj Є SR
goto step(27)
Else Add 1 to J,
go to step (33)
Step 27: If Si ∩ Sj = Ø do steps (28)-(33)

Step 28: Add 1 to K

Step 29: Create new fragment Fk = Rj - UF

Step 30: End IF

Step31:If Fk ≠ Ø Add Fk to the set of F

Step 32: End IF

Step 33: Loop

Step 34: Add 1 to I Step 35: Loop

Step 36: Set 1 to I

Step 37: Do steps (38 - 53) until I > F

Step 38: Set 1 to J

Step 39: Do steps (40 - 51) until J > F

Step 40:If I ≠ J and ∃ Fi,Fj Є FR goto step (41) Else, Add 1 to J and go to step (50)

Step 41: If Fi ∩ Fj ≠ Ø do steps (42)-(49) Else, Add 1 to J and go to step (49)

Step 42: Add 1 to K

Step 43: Create new fragment Fk = Fi ∩ Fj and add it to F

Step 44: Create new fragment Fk+1 = Fi - Fk and add it to F

Step 45: Create new fragment Fk+2 = Fj - Fk and add it to F

Step 46: Delete Fi

Step 47: Delete Fj

Step 48: Set F + 1 to J

Step 49: End IF

Step 50: End IF

Step 51: Loop

Step 52: Add 1 to I

Step 53: Loop

Step 54: Add 1 to R

Step 55: Loop [7]

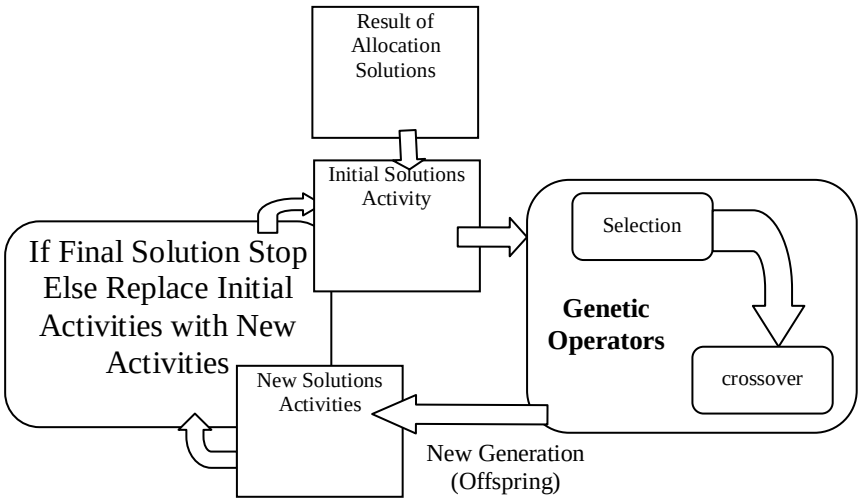


Fig 2: The Basic Genetic Steps

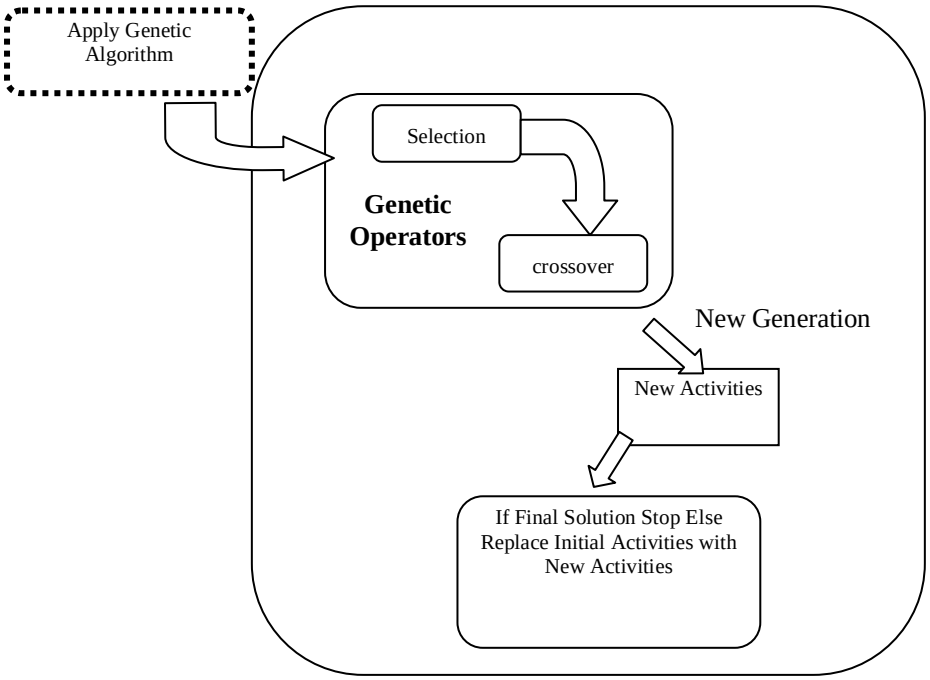


Fig 3: Fragment Allocation Techniques

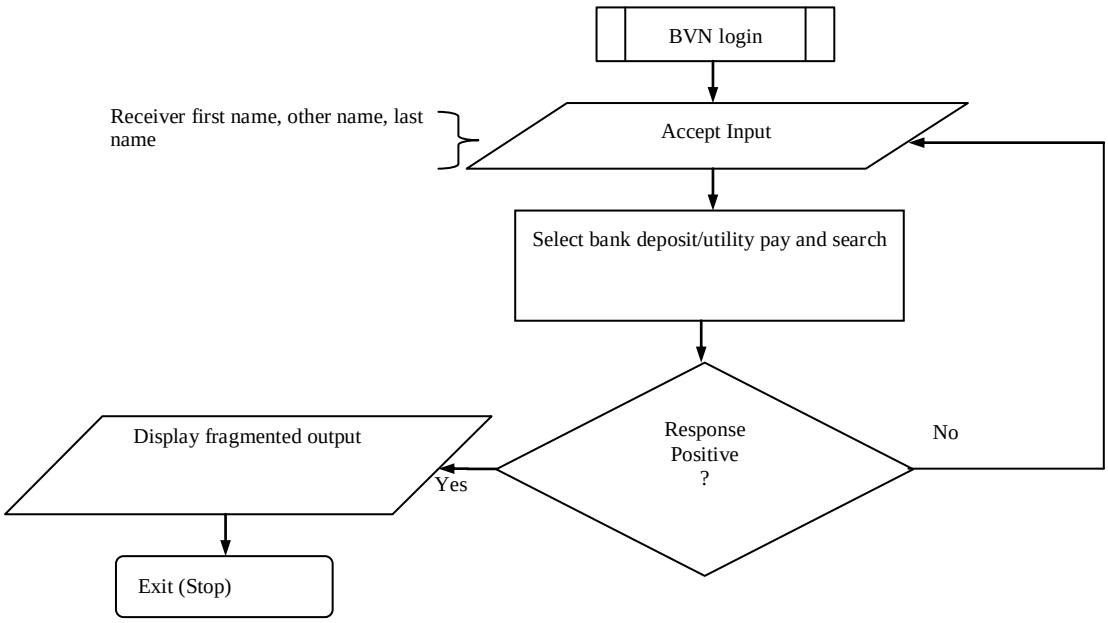


Fig 4: Flowchart on fragmented system

RESULTS AND DISCUSSION

The results of this work are shown in Figure 5 to figure 8. The system is designed to display all the transactions made by an individual within a specified period of time by bringing the transactions closer for easy access (Figure 4). It equally shows the beneficiary bank, transaction mode, date and time, amount, sender and receiver's account number on the fragmented database. The use of genetic technique as a model supports and promotes efficiency of the system in the sense that the data are properly arranged on a row, also displaying the transactions made, payment of utility bills according to the order of transfer. The benefits of this model to the users are reduction in communication cost by reducing the number of servers, easy and fast retrieval of data and closeness of data to the user. The result of this work is as shown in Figure 5 to figure 8.

Admin Menu

Admin menu is made up of Open Account, Search Account and Log Out (Figure 5). Open Account takes the admin to an interface that enables the admin create an account for a new customer picking the customers information from the keyboard (Figure 6) and storing to the database for future use and after which a unique account number is created automatically by the system. Search Account is very important to navigate from one account to another looking at a considerable fact that in an average bank, a million plus customer accounts may all have a link to the same database, searching with name or account number permits ease of transaction after the customers detail and statement have been retrieved from the database. There are a number of functions provided on this level, update of an existing customers account, view and delete accounts in the event "the customer chooses to close their account". Logout, the admin logs out after the admin is through with whatever job for the day. Every admin has a unique login ID, with a login ID, the system keeps record of user's input and how frequent a user was active. This also makes it practically possible for users to be held liable of any theft or wrong input that was entered into the system.

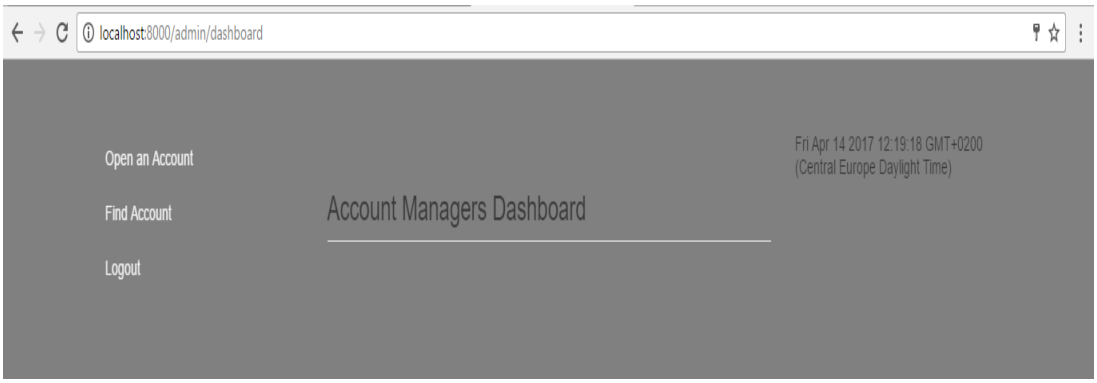


Fig 5: Admin Main Menu

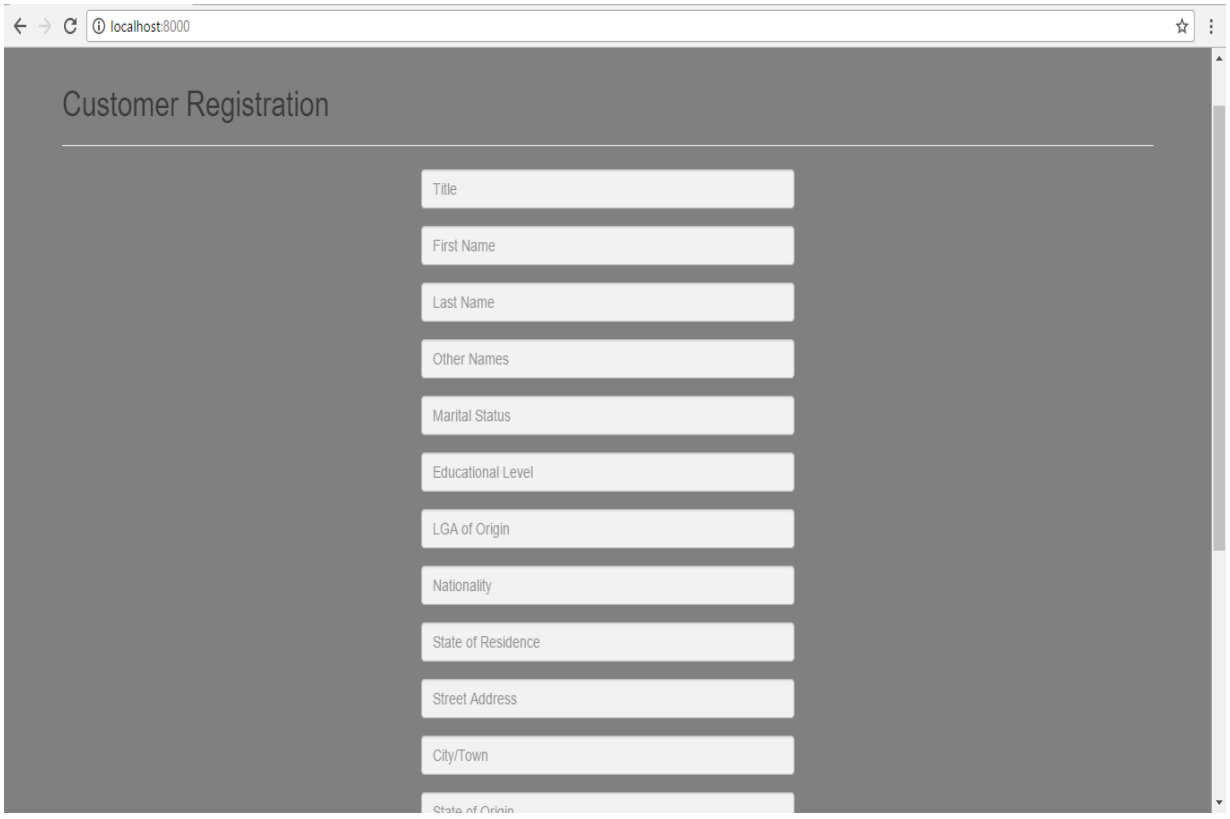


Fig 6: Customer Registration Interface

User Menu

This is made up of Account Balance, Account Activity, Account Statement, Utility/Bills Payment and Bank Transfer (Figure 7). Account Balance directs a user to the interface that contains details of all accounts connected by BVN. The user available balance in all accounts that are connected will be viewed without access to printing. Account Activity aids a user to see the last five activities or transactions. Account Statement displays all transaction on the account within a stipulated period of time; the range in period is being decided by the user of the account. It was designed so that the user will have an option of printing this statement or downloading and saving on their personal device for viewing regardless of the presence of network access at their leisure. Utility/Bills Payment (Figure 8) - A user is expected to make other transactions outside view of statement, checking of account balance and so on. This interface avails a user the opportunity to pay for light bill, water bill and purchase of data or recharge cards with ease from any of the connected bank accounts with the help of BVN fragmentation. Bank transfer equally gives a user the opportunity to transfer funds from any of the user's registered bank accounts to other accounts either to business partner or family member.

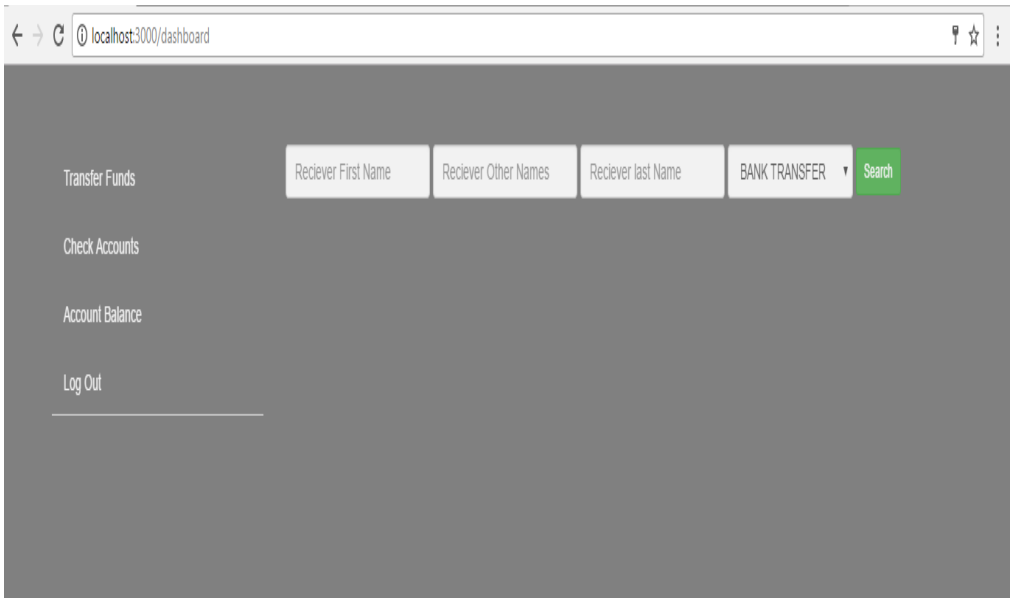


Fig 7: User Main Menu

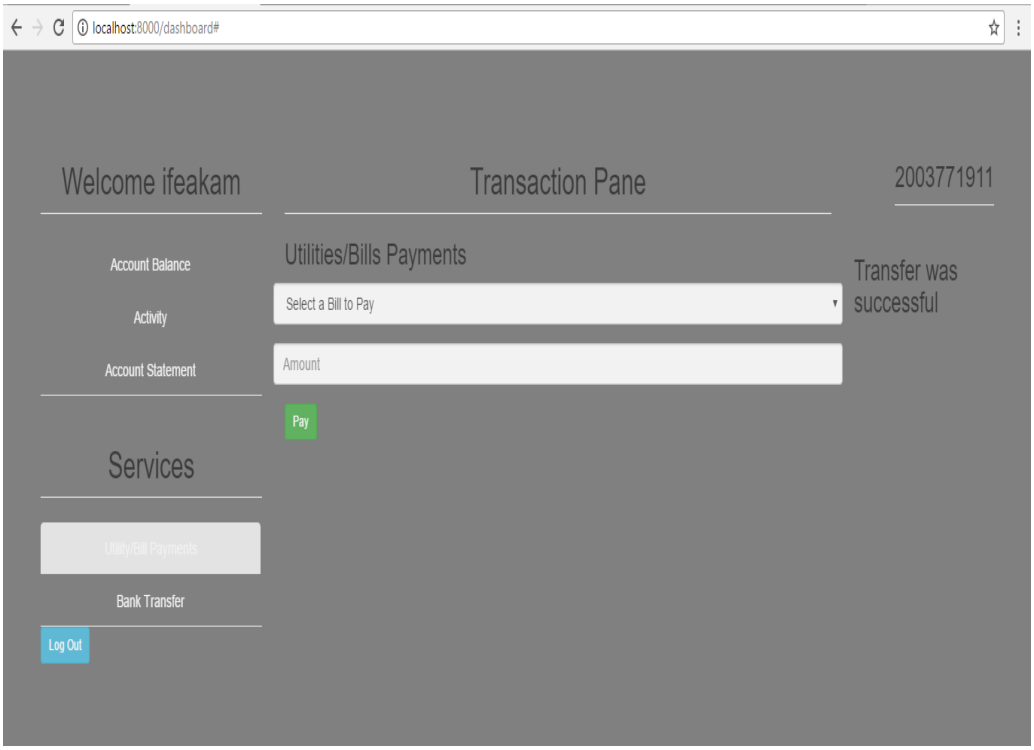
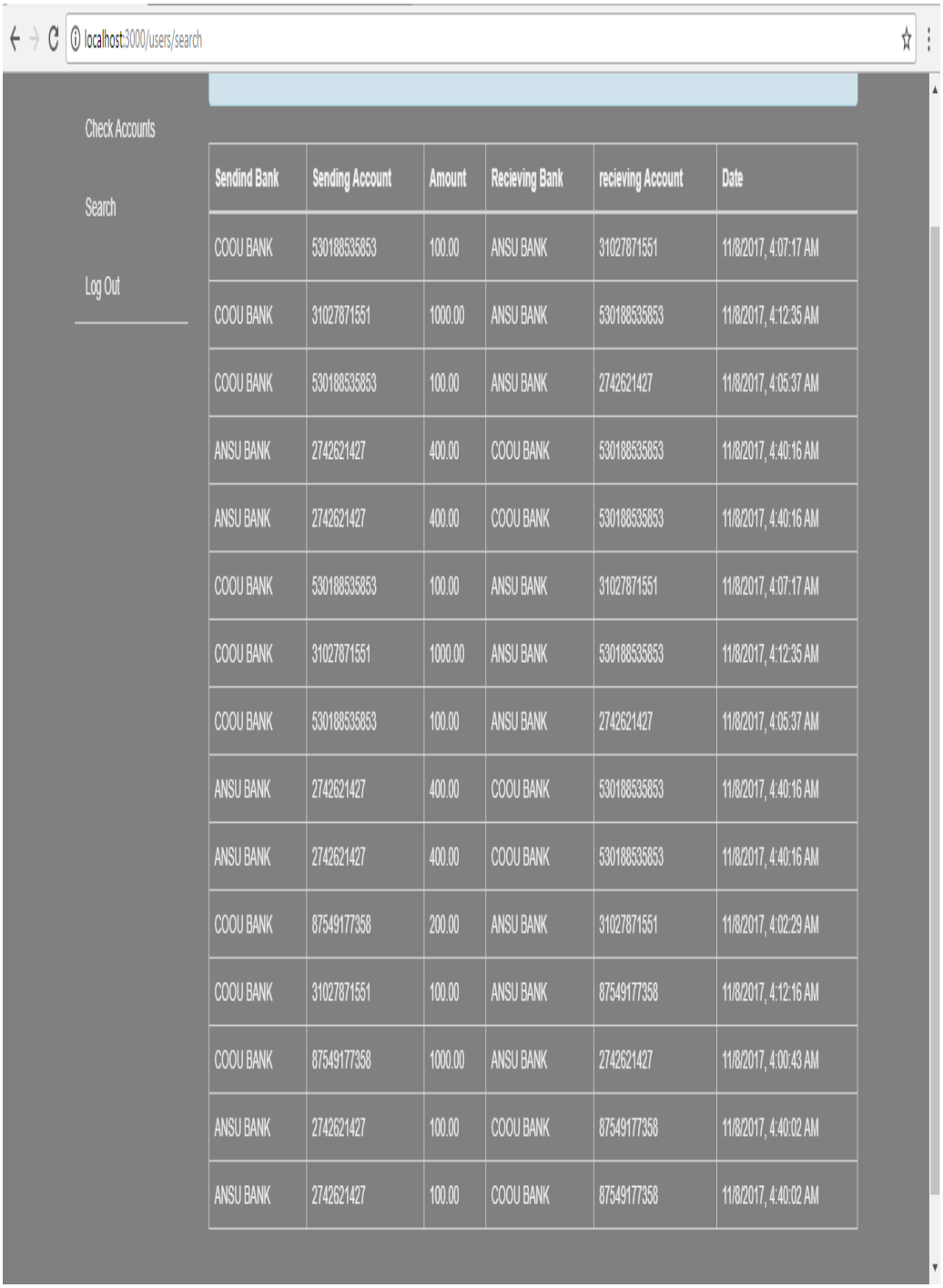


Fig 8: Utility/Bills Payment Interface



Sendind Bank	Sending Account	Amount	Recieving Bank	recieving Account	Date
COOU BANK	530188535853	100.00	ANSU BANK	31027871551	11/8/2017, 4:07:17 AM
COOU BANK	31027871551	1000.00	ANSU BANK	530188535853	11/8/2017, 4:12:35 AM
COOU BANK	530188535853	100.00	ANSU BANK	2742621427	11/8/2017, 4:05:37 AM
ANSU BANK	2742621427	400.00	COOU BANK	530188535853	11/8/2017, 4:40:16 AM
ANSU BANK	2742621427	400.00	COOU BANK	530188535853	11/8/2017, 4:40:16 AM
COOU BANK	530188535853	100.00	ANSU BANK	31027871551	11/8/2017, 4:07:17 AM
COOU BANK	31027871551	1000.00	ANSU BANK	530188535853	11/8/2017, 4:12:35 AM
COOU BANK	530188535853	100.00	ANSU BANK	2742621427	11/8/2017, 4:05:37 AM
ANSU BANK	2742621427	400.00	COOU BANK	530188535853	11/8/2017, 4:40:16 AM
ANSU BANK	2742621427	400.00	COOU BANK	530188535853	11/8/2017, 4:40:16 AM
COOU BANK	87549177358	200.00	ANSU BANK	31027871551	11/8/2017, 4:02:29 AM
COOU BANK	31027871551	100.00	ANSU BANK	87549177358	11/8/2017, 4:12:16 AM
COOU BANK	87549177358	1000.00	ANSU BANK	2742621427	11/8/2017, 4:00:43 AM
ANSU BANK	2742621427	100.00	COOU BANK	87549177358	11/8/2017, 4:40:02 AM
ANSU BANK	2742621427	100.00	COOU BANK	87549177358	11/8/2017, 4:40:02 AM

Fig 9: Output Format of the Fragmented System

ACKNOWLEDGEMENT

I wish to express my sincere gratitude to the Almighty God for giving me the strength to achieve this work. My acknowledgement also goes to my family for their support

CONCLUSION

A modern approach for an improved database fragment allocation technique was proposed in this work. This approach sets in integrating two techniques namely database fragmentation and fragment allocation. These techniques are developed to avoid setback of database fragmentation and data allocation like data redundancy and complexity of data redistribution problem. In addition to that, the innovation to our approach is to satisfy a

specific level of data availability and consistency. The result gotten in this work indicated that our approach significantly improved performance requirement satisfaction in distributed systems. The strength of this system suggests reduction of data transfer between sites, increased security, and event log in is easier as it improves availability, and at the same time control transactions hence lead to safer transactions, efficiency in performing transactions is not ruled out of the robustness of the system. It is not tremendously affected by errors in assumptions about the distribution of sample errors, data that are not required by local applications are not stored locally and it is very well easier to implement in any organization. Keeping inventory of all transactions made from this window for accountability. The system must be informative, robust, responsive, user-friendly and secure. System should be analyzed and designed to allow possible future expansion..

REFERENCES

[1] **Apers, P.M.G. (2008)** *"Redundant allocation of relations in a communication network".* In proc. 5th Berkeley workshop on distributed data management and computernetworks. 245-258.125.

[2] **Chen, J. (2012).** Distributed Database: Fragmentation and Allocation, Journal of Data Mining and Knowledge Discovery ISSN: (3)2229-6662 & ISSN 2229-6670, 2012.

[3] **Cornell and Yu, (2018)** *"Vertical partitioning algorithms for database design". Journal ACM Transactions on Database Systems (TODS). Issue 4, Dec. 1984.(9)680-710*

[4] **Ozsu, M.T. and Valduriez, P. (2007)** *"Distributed and parallel database systems".* In tucker, A., editor, handbook of Computer science and Engineering, 1093-1111. CRC press. 38.

[5] **Ma, Hui** (Massey University, 2007) "*Distribution* design for complex value *databases* : *a* dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Information Systems at Massey University"

[6] **Hawick,K.A.; Leist, A.; Playne,D.P.** ,
https://mro.massey.ac.nz/discover?scope=%2F&query=an+author+proposed+an+optimization+model

[7] https://html.scirp.org/