Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7 – No. 39, August 2023 – www.caeaccess.org

Cloud-Enabled Digital Cash Transaction System

Adeyinka Festus Osuolale
Department of Computer Science
School of Computing
The Federal University of Technology, Akure, Nigeria

David Ayanlola
Department of Computer Science
School of Computing
The Federal University of Technology, Akure, Nigeria

ABSTRACT
The act of transferring products and services is a time-honored and successful endeavor, just as this digital cash transaction system. Man's inability to meet all of his wants independently has made commerce necessary. Trading has developed from its early beginnings to a more sophisticated and refined form. Bartering is one of the earliest types of trade. This project's goal is to create a cloud-enabled digital cash transaction system model and put it into practice using several important development tools. The logical part is implemented using a hypertext preprocessor, and the user interface, which is created on a web server, is developed using a hypertext markup language and cascading style sheet. The cloud infrastructure provides the computing power and storage necessary to process and store transaction data securely. The digital wallet is an electronic account that stores digital cash, which can be used for making transactions. However, the growth trend is toward offline digital cash (such as ATM, Mastercard and Visa International), where withdrawals are done using cards (metal or plastic) on machines designed specifically for the purpose.

General Terms
Cloud computing, Financial Transactions, Digital system

Keywords
Commerce, cloud-enabled, Digital Wallet, Transaction, Mastercard, ATM, Hypertext preprocessor.

1. INTRODUCTION
The performance of exchanging products and services is a long-standing and successful endeavour. The need trade emerged as a result of man's inability to meet his own needs on his own. Trading has progressed from its earliest phases to a more sophisticated and refined form. Barter trade is one of the oldest kinds of commerce. Barter trade was regarded to some extent as a required answer to the problem exchange; yet, it came with a slew of flaws.

A cloud-enabled digital cash transaction system is a financial technology solution that facilitates digital transactions of money or other assets between parties, using cloud computing infrastructure. It combines the convenience of digital transactions with the benefits of cloud computing, including scalability, flexibility, and security. The problem of mutual desire overlap, among other things, is a key impediment to the system's development. Paper cash, which has particular features (homogeneity, portability, malleability), is backed by the government, and is a legal tender for exchange, was introduced as a great breakthrough in the system of exchange (trade). The paper cash system did, in fact, greatly contribute to the ease with which transaction could be conducted. According to Wikipedia, a research paper by David Chaum introduced the idea of digital cash in 1989, he founded DigiCash, an electronic cash company, in Amsterdam to commercialize the ideas in his research.
Most of the systems that have been presented so far, whether online or offline, lack the capacity to transfer coins. In these payment systems, a coin's lifetime corresponds to the lifetime of the transaction it is a part of. In contrast, paper money only changes hands and maintains its worth over a number of transactions. Transferable money has the obvious benefit of avoiding the requirement for new coins to be produced for each transaction. The negative burden that would be placed on the central bank or coin-issuing body to verify the legitimacy of the coin is one drawback of online systems that has been frequently noted. Since each coin in a transaction needs to be confirmed by a central server, this is unquestionably a bottleneck. In the suggested method, a central validating server and distribute the workload across other entities was eliminated. It is also recommended that the electronic payment system have a mechanism in place for addressing disputes. Given the fact that the payer and payee do not even know each other's' real identities as they transact over the Internet, the payment system should be able to give guarantees to both the parties in the transaction. As part of the proposed system, the outline of such dispute resolution protocol was included.

2. LITERATURE REVIEW
In the work of [7], "Design and Implementation of Secure Digital Payment System", the increase in the use of computer networks to access and pay for goods and services prompt them to design and implementing a secure Digital Payment System using he security architecture of the system is design by RC5 encryption/decryption algorithm written by Microsoft Visual Basic 6.0. In the secure electronic payment system, the authorized owner can use this system reliably. The transaction can only be performed electronically on a computer system.

In the work of [5], "Digital Wallet (Electronic Financial Transaction System)", all personal banking operations are made easy and faster, which brought about the objective of the study to designing a wallet to solve all the complexity and cashless trading, bringing people under secure transaction and reduce risk. The information gotten from other wallet models (such as PayPal, Skrill) was used in designing the new wallet. The development brings everyone under banking channel and wallet is open 24/7, despite the significant result, the system does not have an app that can run on Android and IOS.

A payment or withdrawal message with a digital signature (pin code) and used as a medium of exchange is referred to as a "digital cash transaction system" (trading). The digital cash transaction system is supported by a third party, typically a bank, who is willing to convert digital currency to actual currency has value. In order to conduct financial transactions digitally, [2] defines a digital payment as a method of payment supported by

banks and interconnected between customers and banks. Electronic payment systems in which the payment instrument is e-cash. Such systems are classified into two types, online and offline [8]. Online e-payment systems are those in which the transfer of electronic money between the payer and payee takes place in the presence of a third party, usually a bank, which guarantees the authenticity of the coins being transferred. In contrast, in offline systems, the payer and payee are the only parties involved in the transaction. When the payee deposits the coins with a bank, the amount of the transfer is confirmed.

3. METHODOLOGY

The system typically involves three components: the cloud infrastructure, a digital wallet, and a payment gateway. The cloud infrastructure provides the computing power and storage necessary to process and store transaction data securely. The digital wallet is an electronic account that stores digital cash, which can be used for making transactions. The payment gateway acts as an intermediary between the digital wallet and the merchant or vendor receiving the payment. This chapter presents the overall architecture of the system comprising of the components (modules) and interactions that make up the system. The database design for User registration (or User profile), user authentication, Transaction History, and Transactions are designed. Software models such as use-case design, activity design, and sequence design using Unified Modelling Language (UML) are developed. Keeping in mind, The Central Bank of Nigeria Naira Redesign Policy and Revised Cash Withdrawal Limits was announced on January 9, 2023, with some detailed information like.

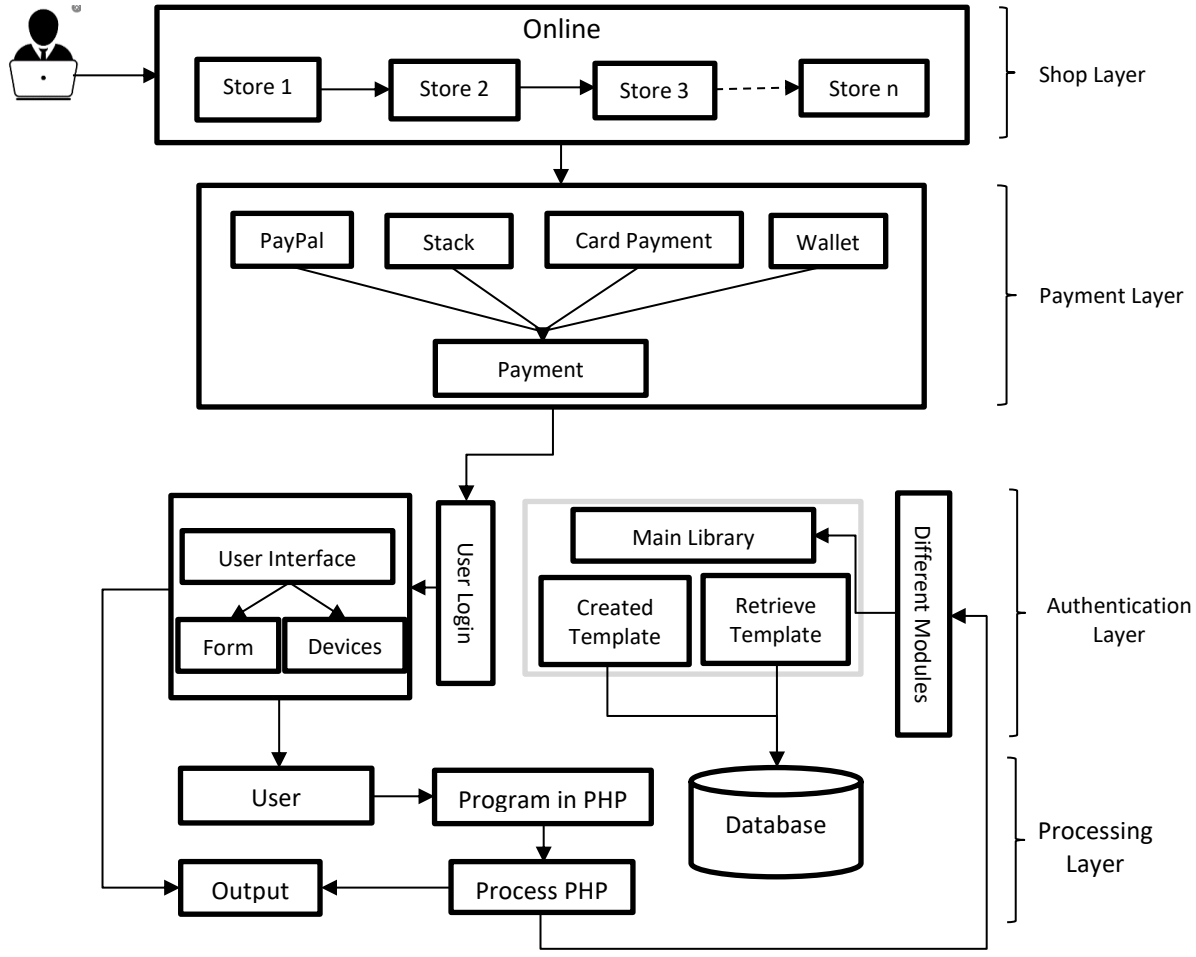

Figure 1: System Architecture

i. The maximum weekly limit for cash withdraws across all channels for personal and corporate organisation shall be N500,000 and N5,000,000 respectively.

ii. In compelling circumstances where cash withdraw above limit is required for legitimate purposes, such request shall be process with extra fee of 3% and 5% for individual and corporate organisation respectively.

iii. Third party cheque above N100,000 shall not be eligible for payment over the counter where the extant limit of N10,000,000 on clearing cheque is subsist.

3.1 System Architecture

The architecture that presents the components of the system and the interaction among them is presented in Figure 1.

3.1.1 The shop layer: The Shop Layer is made up of all online shops that accept electronic payments. Any credit or debit card, electronic wallet, pay-on-delivery service, or Secure Online Electronic Transaction (SOET) may be used. In order to emulate the concept, an online store is deployed.

3.1.2 The transaction layer: The payment layer offers a number of payment options, and users are required to select the most practical one. The study focuses on the SOET option, which features a payment gateway through which the customer must pass in order to pay for the item(s) they have chosen while conducting online shopping.

*3.1.3 The authentication layer***:** This layer is made up of a number of parts that work together to guarantee the user's account is adequately secure during transactions. These components include the User Login, where the user's credentials must match those provided during the enrolment process in order for the user to log on to the system. Providing the correct credentials is essential access will be refused if the user provides a false one.

Through a web browser, forms are provided with which the user communicates with the system appropriately, and a mobile device that assists the user in entering the login credentials in order to gain access to the transaction platform, the user interface allows the customer to interact with the system while concealing the underlying functionalities. The authentication layer's Hypertext Pre-processor Login Authentication Core is made up of a number of modules that guarantee the efficient use of login credential authentication. many dynamic links libraries are used in all of the Extraction and Matching Module's functions that deal with authentication.

The transaction server that executes the transaction after the user's claim has been correctly confirmed is known as the "Processing Layer." This layer is made up of a combination of programs that oversee the seamless execution of the transaction by the online bank. This layer contains a number of software elements, including the Apache web server, the PHP web programming language, and the MySQL database. After the user's claim has been correctly confirmed, the transaction is implemented by the transaction server in the processing layer. This layer is made up of a combination of programs that oversee the seamless execution of the transaction by the online bank. This layer contains a number of software elements, including the Apache web server, the PHP web programming language, and the MySQL database.

3.2 Database Design

Database design is the organization of data according to a database model or schema. Here, the data that must be stored and how the data elements interrelate is determined. With this information, one can begin to fit the data to the database model. In this work, relational database model is employed. The first step to designing a relational database is to define the schema. The schema is a map of where all the data lives in tables - table names, column names. The schema shows how tables relate to each other.

3.2.1 User Registration Table: This table is used to store customer's profile. These profiles include User's ID, User's name or Email, Password and Confirm Password. The user registration schema is presented in Table 1.

Table 1: User Registration Table

Field Name	Data Type	Size	Description
UserId	Long Text	50	Identification of a User
UserName	Long Text	50	Username or user Email
Password	Varchar	50	User Password
ConfirmPassword	Varchar	50	Duplicate of User password

*3.2.2 User Authentication Table***:** This used to secure the transactions going on in system by only allowing authorized users to gain access to the system. Database fields that make up the user authentication table include user id, username, and password. The user authentication table is presented in Table 2.

Table 2: User Authentication Schema

Field Name	Data Type	Size	Description
UserId	Long Text	50	Identification of a User
UserName	Long Text	50	Username or user Email
Password	Varchar	50	User password

*3.2.3. Payment Information Schema Table***:** Payment Information Schema table contains information of the transaction that is been made on the system, this table also contains information to track transactions in the system. The system contains several fields such as UserId, Account Number, Account name, Transaction Id, tracking Id, Payment Amount, Service key, online payment code and Transaction Date. The Payment Information table is presented in Table 3.

*3.2.4. User Transaction List Schema Table***:** This table contains the list of transactions performed by a user in order to keep track of all transaction history by a user. It contains several columns like User ID, Date Created, Transaction code and Paid amount. The schema is presented in Table 4.

Table 3: Payment Information Schema

Field Name	Data Type	Size	Description
UserId	Long Text	50	Identification of a User
AccountNumber	Long Text	50	Username or user Email
AccountName	Varchar	50	User password
TransactionId	Varchar	50	Transaction Identification code
TrackingId	Varchar	50	Transaction tracking code
PaymentAmount.	Decimal	10	Amount that is been Transferred or received
ServiceFee	Decimal	10	Amount charged for transaction
OnlinePaymentCode	Varchar	50	Random number generated for transaction
TransactionDate	DateTime	50	Date and time of transaction

Table 4: User Transaction list Schema

Field Name	Data Type	Size	Description
Userid	Long text	50	Identification of a user
Datecreated	Datetime	50	Date of transaction
Transactioncode	Varchar	50	Transaction code
Paidamount	Decimal	10	Amount transferred

3.3. System Design

Systems are developed to address issues. The systems approach can be viewed as a methodical technique to solving a problem.

The primary focus of the System Analysis and Design (SAD) course is on software development activities in today's dynamic environment. The new system must be designed based on user requirements and a thorough analysis of the current system. The system design phase is now. It is the most important stage of a system's development. The systems analysis's logical system design is transformed into a physical system design. Preliminary or general design and structured or detailed design are the two stages that typically make up the design process. In this study, the Object-Oriented UML design methodology was utilized to create the designs.

3.3.1. UML Diagram

A software-intensive system's components can be seen, specified, built, and documented using the graphical language known as Unified Modelling Language (UML). It provides a common format for drafting a system's blueprints, including conceptual things such as business processes and system functions as well as concrete things such as programming language statements, database schemas, and reusable software components [1}. Furthermore, In the area of object-oriented software engineering, the Unified Modelling Language (UML) is a standardized general-purpose modelling language. Visual models of object-oriented software systems can be created using a set of graphic notation tools included in UML. UML incorporates concepts from data modelling, business modelling, object modelling, and component modelling and may be applied across a variety of implementation technologies and the software development life cycle [5]. For modelling the dynamic components of systems, UML offers five different types of diagrams. Use case diagrams, sequence diagrams, collaboration diagrams, activity diagrams, and status chart diagrams are some examples of these diagrams. Use case diagrams are essential for modelling a system's behaviour.

3.3.2 Elements of the UML Diagram

Use case diagram: A problem domain's collection of all use cases is known as a use case model, and the diagram that represents it is known as a use case diagram, Figure 2. A use case model demonstrates the range of functionality that a system must offer. Looking at the use case model, a user can determine whether or not all needs are met. A use case model is crucial because it provides a broad overview of the system without becoming bogged down in implementation specifics. All of the system's necessary features should be expressed in the use case model. One should be able to determine whether all user requirements are accurately captured and whether all user roles are by looking at the use case model. [6].

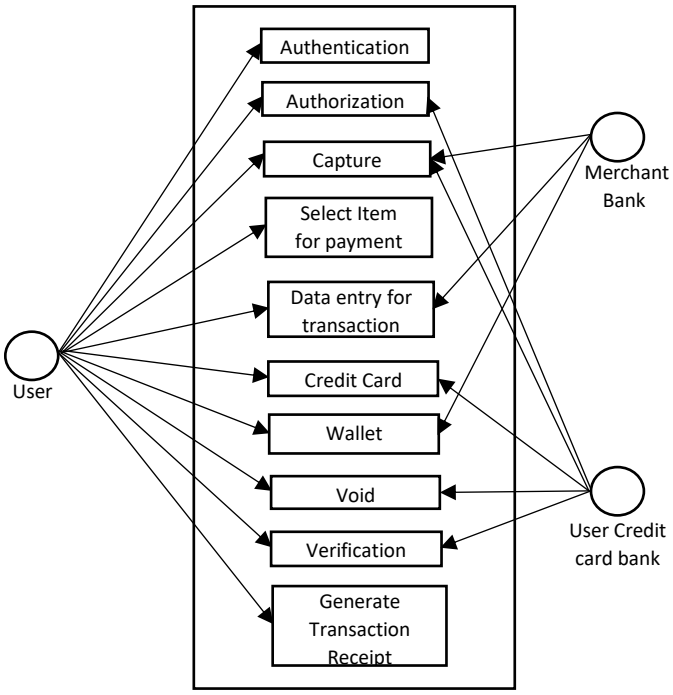

Figure 2: Use Case Diagram

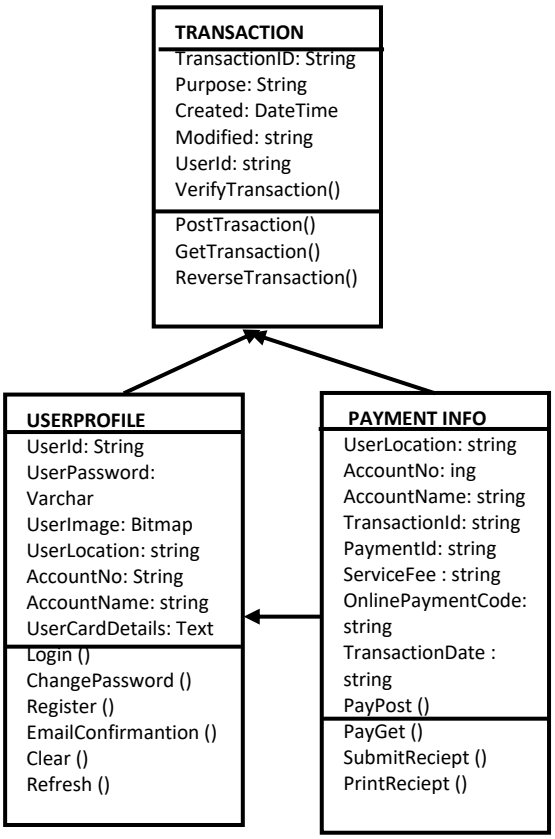

Figure 3: Digital Cash Transaction System Class Diagram

Class diagrams: This describes the kinds of objects found in the application and their inter-relationships as depicted in Figure 3. There are two types of inter-relationships: associations and subtypes. Class diagrams are basically an adaptation of Entity Relation Diagram (EER) diagrams, with some minor differences.

UML class diagrams may model some part of the real world e.g., the world of meetings and scheduling), a design specification (e.g., for a system that does meeting scheduling), or an implementation [3].

Interaction diagrams: This captures interactions among objects. Typically, an interaction diagram models what happens for a single use case [3]. It shows a number of objects and the messages that are passed between them during the execution of the use case. There are two (comparable) types of interaction diagrams: collaboration diagrams and sequence diagrams. Use icons to denote the objects participating in an interaction diagram (sequence or collaboration).

Activity diagrams: These are used to describe the procedures followed in the execution of a use case and to depict the system's control flow. Using activity diagrams, both sequential and concurrent activities are modeled. So, it basically depicts workflows visually using an activity diagram. An activity diagram focuses on condition of flow and the sequence in which it happens. What causes a particular event are also described using an activity diagram. The basic purposes of activity diagrams are similar to other four diagrams. It captures the dynamic behaviour of the system. Other four Message flow from one item to another is depicted using diagrams, but activity diagrams are used to indicate message flow from one activity to another. Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing
thing in the activity diagram (Figure 5) is the message part. It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

A Deployment diagram: This is a UML diagram type that shows the execution architecture of a system, including nodes such as hardware or software execution environments, and the middleware connecting them. Deployment diagrams are typically used to visualize the physical hardware and software of a system. Using it you can understand how the system will be physically deployed on the hardware. Deployment diagrams, Figure 4, help model the hardware topology of a system compared to other UML diagram types which mostly outline the logical components of a system.

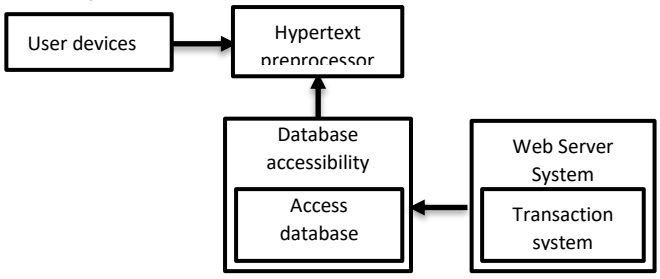

Figure 4: Digital Cash Transaction System Deployment diagram

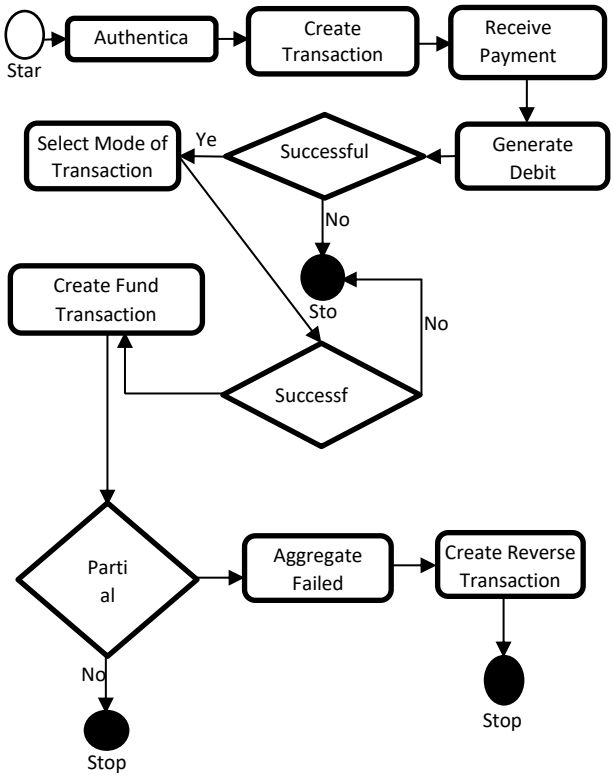

Figure 5: Activity diagram of Digital Cash Transaction System

3.4. Mathematical Model of the Online Transaction System

Markov chain is a discrete Markov manner with time and nation parameters. Because the time discovered through the overall Markov manner is limitless, the time is a non-stop variable, and its fee is likewise non-stop. A limitless partition may be made among adjoining values. Meanwhile, the time parameter is discrete, and the country of the Markov chain is likewise finite. When one nation may be listed, the destiny nation is only associated with the cutting-edge nation, and it has not anything to do with the preceding nation. For instance, recall a discrete-time stochastic manner, $\{X_n, n = 0, 1, 2 \ldots\}$. Here, its fee is a finite or countable set of S, that is the nation area of the manner, and a nation area is the gathering of all viable states that comprise the Markov chain. The finite dimensional distributions of the manner are proven in equations (1) to (6):

$$P\{X^0 = i, \ldots X^n = j\}\ i_0, \ldots j\ \in S, n \geq 0 \qquad (1)$$

Where *P* is the possible values of Xi form a countable set S called the state space of the chain.

Where the set S is the state space of the process and the value Xn ∈ S is the state of the process at the time n. probability distribution uniquely determines the probability of all events in the process., therefore, if the finite-dimensional distributions of two random processes are equal, their distributions are equal:

$$P\{X_{n+1} - j|X_0, \ldots, X_n\} = P\{X_{n+1} = j|X_n\} \qquad (2)$$

The stochastic process on the countable set S and X= $\{X_n: n \geq 0\}$ is a Markov chain, for any $i, j\ \in S, n \geq 0$:

$$P\{X^{n+1} = j | X^n = i\} = P_{ij} \quad (3)$$

Where Pij is the probability of a Markov chain moving from state *i* to state *j*. obviously,

$$P_i \geq 0, \sum_{n=1}^{\infty} P_{ij} = 1, j = 0, i = 1 \quad (4)$$

(i) Condition (1), called the Markov property, means that at any time n, the conditional distribution of the next state X_{n+1} is independent of the past state X_0, ..., X_{n-1}. That is to say, the future state is independent of the past state and is dependent on the present state.

(ii) Condition (2) simply says that the transition probabilities do not depend on the time parameter; the Markov chain is "time homogeneous." If the transition probabilities were functions of time, the process Xn would be a non-time-homogeneous Markov chain. Pij is a matrix that consists of the transition probability. This process is called the transition probability matrix.

$$P = \begin{bmatrix} P_{00} & P_{01} & \dots \\ P_{10} & P_{11} & \dots \\ \vdots & \vdots & \vdots \end{bmatrix} \quad (5)$$

To describe the finite-dimensional distribution of stochastic processes is a basic problem in analyzing the structure of stochastic processes. The finite dimensional distribution of X*n* is determined by its initial probability distribution X0 and transition probability. Assuming that the transition probability and the initial probability of the Markov chain Xn are Pij and $\alpha i = P\{X0 = i\}$ respectively, for any i0, …, in ∈ S and n ≥ 0, the following formula can be deduced:

$$P\{X^0 = i_0, \dots, X^n = i_n\} = \propto_{i0} P_{i0,i1,} \dots P_{in-1,jn} \quad (6)$$

4. SYSTEM IMPLEMENTATION, RESULTS AND EVALUATION

This section presents the implementation of digital cash transaction system. The software and hardware requirements for this system to ensure that the system runs smoothly. The snapshots in Figure 6 to Figure 12 show different system interfaces and development environment

4.1 Software and Hardware Requirements

Implementing the system involve the following software tools. Operation System: Microsoft Windows 10 OS because the software is developed as a window-based application. Database management system (DBMS): Microsoft Access for storing information for reference, reporting and analysis. MySQL database Engine: The Major DBMS used in the application programming interface (API) to enable the interaction of users with database engines. It is very necessary to know about the engines for production databases and it also impacts future development. XAMPP Server: is an abbreviation where X stands for Cross-Platform, A stands for Apache, M stands for MYSQL, and the Ps stand for PHP and Perl. XAMPP is one of the widely used cross-platform web servers, which helps developers to create and test their programs on a local webserver. Microsoft Visual Studio code: Microsoft visual studio code is an integrated development environment (IDE) from Microsoft. It is used to develop both the program interface of the Digital Transaction system and the coding it can as well be used to develop websites, web apps, web services and mobile apps. Visual Studio uses Microsoft software development platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both native code and managed code. Visual Studio does not support any programming language, solution or tool intrinsically; instead, it allows the plugging of functionality coded as a Visual Studio Package. When installed, the functionality is available as a Service. It is cross platform, that is, it works on Microsoft windows, Linux, and Mac operating systems. While Visual Studio code only works on Microsoft windows operating system and Mac operating system. For efficient running of this system, the following hardware specification are required; at least 1.3GHZ of processor speed, 120GB of Hard Disk space and at least 2GB Random Access Memory (RAM).

4.2 System Implementation Snapshot

Snapshots of the web-based application for the Digital transaction system can be found in the following figures.

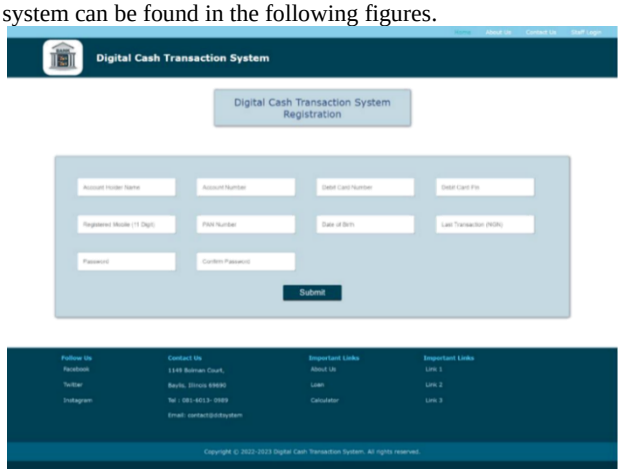

Figure 6: Full registration or Signup page

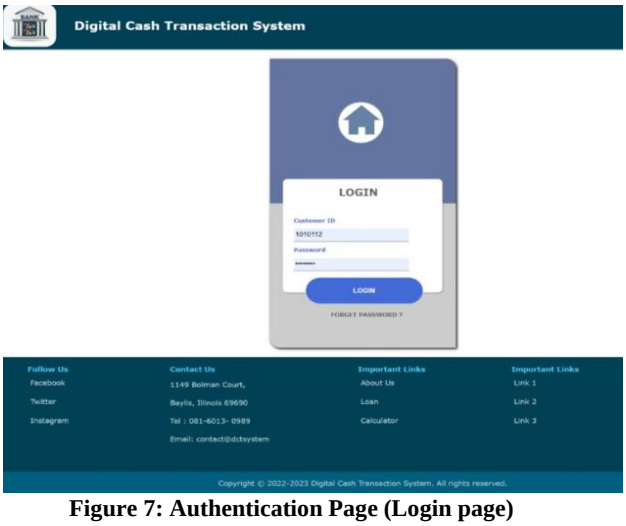

Figure 7: Authentication Page (Login page)

Figures 6 and 7 show the registration and login pages which can also be referred to as authentication pages. These are the pages a user needs to pass through before a transaction could be performed on the system. Furthermore, is the security process that allows users to verify their identities to gain access to their personal accounts on a website.

Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 7 – No. 39, August 2023 – www.caeaccess.org

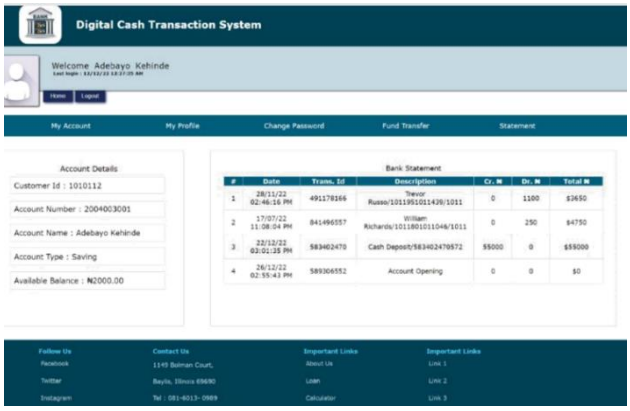

Figure 8: Dashboard Area of the Digital cash transaction system

Figure 8 shows where all necessary activities need to be carried out from update user information, to change display name, fund transfer and review transaction statement. Moreover, this page displays a monthly financial report that summarizes the actions of the account holder.

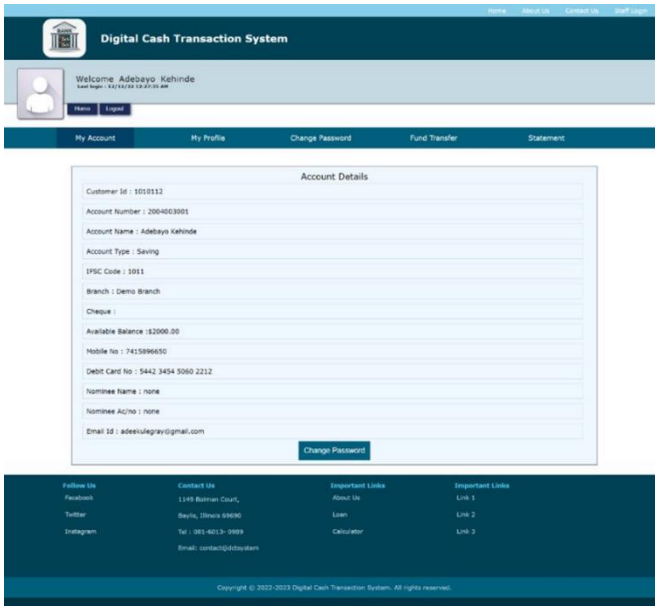

Figure 9: Complete Account display page

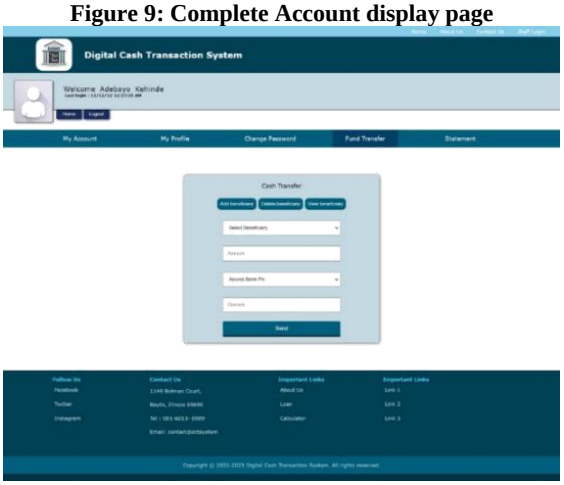

Figure 10: Fund Transfer page

Figure 9 shows the user's complete account display page, this gives user accessibility to view their information and make necessary changes. Figure 10 is the Fund Transfer page. This is one of the technical pages in the system in terms of implementation. This page Provide certain details, such as the account number, destination bank, and comment, to show how a transaction is made in the system. The beneficiary input box is optional and relies on whether the user has beneficiaries saved in the system or not. A confirmation page will appear after the transaction has been completed.

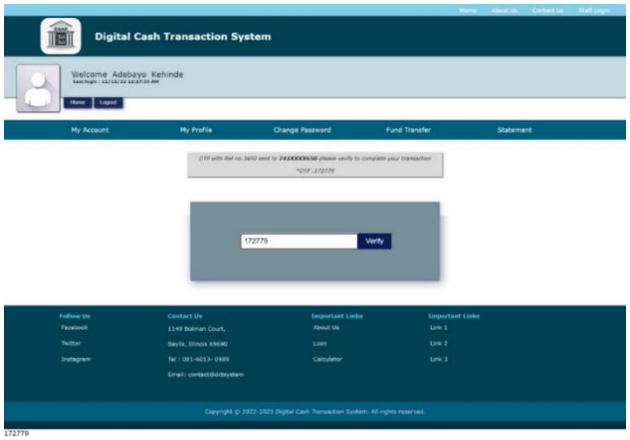

Figure 11: OTP verification Page

Figure 11 depict the verification code page, here the verification code appears for user to process the transaction, every verification code of each transaction is different to ensure a new transaction is initialized and the transaction can be traced afterwards. Figure 12 shows the transaction statement of a particular user. This is list of all transactions for a bank account over a set period, usually monthly. The statement includes deposits, charges, withdrawals, as well as the beginning and ending balance for the period.

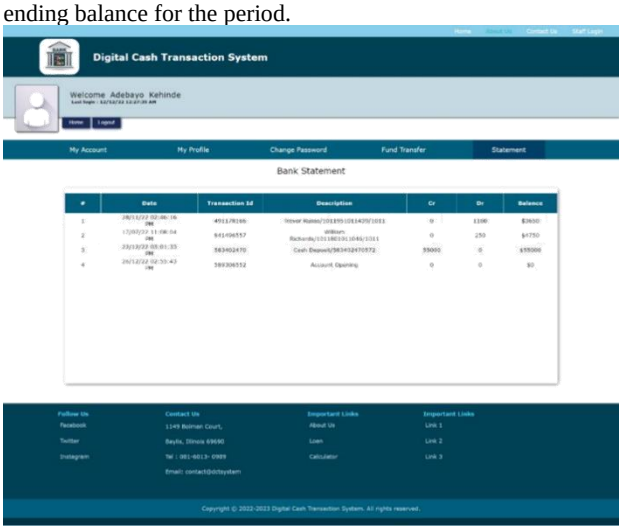

Figure 12: Statement of transaction

4.3 Results and Evaluation

A decision maker-centric method of comparison and selection was adopted. Although it could appear like a tautology, some authors have really used an ad hoc method, which could result in wasted infrastructure investment, irritated potential customers, harm to the organization's reputation, and a loss of sales of goods and services. He gave thought to two qualities for digital payment systems. Low financial risk or default risk is usually cited as desired and, for the agent transactions in which it has a direct

stake. Another trait that is usually mentioned as desirable is monetary value. However, the UMDL only cares about financial value for its own transactions inasmuch as it lowers the risk of default; it is only one of several aspects of a payment mechanism that may affect financial risk. The Table 6 shows are evaluations for some characteristics

Table 6: Evaluation of Digital Cash Transaction System

	First Virtual	NetBill	Ecash	CyberCash
Easily exchangeable	N/A	Yes	Yes	N/A
Locally scalable	Yes	limited	Yes	Yes
Low transactions delay	No	yes	yes	No
Micro transactions	No	No	No	No

The number of parameters on through which assessed is made to different alternatives in this project was reduced. It is suggested that an axiomatic method for choosing a mechanism be employed. Markov Chain selection means that a mechanism will only be considered "acceptable" (by the user) if it meets specific, presumptive requirements. If there are one or more procedures that adequately address all of the factors that matter to the decision-maker, this technique is easy and uncontroversial. The difficult scenario, for which no mechanism demonstrates all of the desired qualities, is the focus of this effort (but there is at least one mechanism that performs well enough overall that the decision maker prefers to choose such a mechanism rather than forego digital payment altogether).

5. CONCLUSION

The acceptance of online cash transaction system or digital payments will be required in the future; thus, it will also be necessary for people's behaviours to shift. In addition to being safer than a cash transaction, becoming cashless also takes less time. Additionally, it aids in keeping track of all completed transactions. As of March 2016, there were more than 22 crore smart phone users in India and more than 100 crore active mobile connections. With higher internet speeds, this number will rise even further. cash transactions are being accepted in more remote places thanks to the expansion of the mobile network, Internet, and electricity. It follows that a cashless transaction system is the way of the future. In addition, this project studies the dynamic optimization of resource scheduling in the queuing model by employing the Markov chain, which provides theory and method support for practical application in the future and maximizes the usage of resources in the model. The significance of online cash transaction system or electronic in commerce in the real world is quite evident from the changing modern trends. Business transactions of all kinds are increasingly conducted online and remotely due to an increased reliance on online technologies. This extraordinary increase in electronic transactions has highlighted the need for a user authentication technique that is faster, more secure, and more practical than passwords and tokens. Since authentication is based on human trait, the framework makes online cash transactions comparable to what is available in the real world. This will go a long way toward making online transactions safer and secure and consequently enhancing user's opinion and patronage in the cashless society.

6. ACKNOWLEDGEMENTS

Thanks to the entire staff members of School of Computing and the Department of Computer Science of the Federal University of Technology, Akure, Nigeria for various support and corporation received during the course of this research work,

6. REFERENCES

[1] Bharath, P. (2012). Principles of Software Engineering. European Journal of Operational Research, 172(1), pp. 255–402.

[2] Briggs, A. & Brooks, L. (2011). Electronic Payment Systems Development in a Developing Country: The Role of Institutional Arrangements. The Electronic Journal on Information Systems in Developing Countries, 49(3), 1-16.

[3] John, T. L. (2014). "UML Diagrams "5th International Conference on machine Learning, Optimization & Data science.

[4] Mahbubul, I. (2018) "Digital Wallet (Electronic Financial Transaction System)". http://dspace.uiu.ac.bd/handle/52243/169

[5] Padmanabhan, B. (2012). Principles of software engineering, "Unified modeling language (UML) Overview" European Journal of Operational Research, 12(1).

[6] Petraq, P. and Panos, P. (2006). UML Diagrams. 5th International Conference on machine Learning, Optimization & Data science.

[7] Raghu, K. and Supriya, S. (2015). "A Secure SMS Protocol for Implementing Digital Cash System". doi:10.1109/icacci.2015.7275893.

[8] Zarrin Kafsh, S. (2015)." Developing Consumer Adoption Model on Mobile Wallet in Canada" (Doctoral dissertation, Université d'Ottawa/University of Ottawa).