

ADVANCED SCHOOL OF SYSTEMS AND DATA STUDIES

DEPARTMENT OF COMPUTER SCIENCE

SOFTWARE ENGINEERING PRACTICE

CS402

**TOPIC/TITTLE**:

VIDO MICRO FINANCE-A MOBILE BANKING SOLUTION

**MOHAMMED MUBAREK ADS21B00260Y**

**DAVID SEBGEFIA ADS22A00038Y**

**DANIEL OCRAN ADS22B00249Y**

DATE: (14th, May 2023)

LECTURER: (MR.FERDINAND HIAGBE)

**FUNCTIONAL REQUIREMENTS**

* **The system should allow all users to authenticate**

Obviously, this system should include a feature that allows mobile bankers, tellers, and mobile bankers to gain access to the system using unique credentials. More so there should be user roles, differentiation a customer account from that of a staff, managers, etc. The procedure should involve having to input credentials, to be granted access automatically by the system after a quick verification against the database, and then for each user login, using Json web tokens to delineate between their privileges. For new users, the feature should allow them to enter their desired credentials and stores the credential in the database, the entered credentials would be unique identifiers that will let the system grant or deny login requests. It’s worth noting, sensitive credentials like passwords should be made hidden to others, while a user is inputting them. Again, all passwords should be hashed from the front end before they are saved in the database. Such feature is required as it control access to the platform and keep the software safe from unauthorized access and potential security breaches.

* **Allow customers verify their personal identity**

The system should be able to take in potential users’ biometric details; in the form or National ID’s, passport, birth certificate, etc. for identity verification. The system must allow electronic submission of relevant documents required to grant account opening requests, loans, etc. OCR (optical recognition) libraries should be used to enable the system recognize texts, signatures and logos from PDFs and Jpeg images, allowing the system verify the validity of the submitted documents applicants. This is emphasized by the need of the system to be able to administer banking services while adhering to the security protocols governing all banks and financial institutions.

* . **They system should allow electronic payment and transactions**

Inherently the vido Microfinance system, just like another other fintech (financial technology) system, should integrate with payment gateways and services to the tellers move funds in and out of the platform. This will also allow the mobile bankers make deposits, withdraw cash, receive and repay loans from or on behalf of the customers, via mobile money, E-zwich, and or other electronic means.

* **The system should allow staff communicate with each other**

There system should utilize like web sockets to allow tellers, managers, and mobile bankers to send real time messages to each other so they work in coordination, in order to able to successfully render the organization’s services effectively. By virtue of this, the mobile banker can quickly reach the branch manager and vice versa, as well as the teller.

* **The system should route requests to the respective staff**

The system should be able to take request from various staff roles and redirect them to the designated destination for further processing. For instance, the system should be able transfer all incoming loan or withdraw requests from a mobile banker and it to a branch manager for further processing.

* **The system should have an in informative and interactive dashboard**

The system should present managers with an interface that allows them to easily check customers’ account details, all pending loans request, outstanding loan balance, deposit and withdraw history, etc. This allows transparency and consequently enables managers effectively monitor the operations of the organization.

* **The system should allow managers grant or deny loan requests**

Pursuant to the operational paradigm of Vido Microfinance, the system should allow managers receive loan applications requests and other relevant requests, and be able to deny or approve these requests upon evaluation. This could be achieved by delegating managers as super-admins on the system with privileged roles.

* **The system must be able to generate periodic reports**

The system in its entirety should be able to sporadically extrapolate data and provide analytical and statistical reports from inference that will help the authorities of the organization in general, to monitor the its performance and deduce initiatives. This consequently would help top level stakeholders of Vido Microfinance, make informative decisions and exploit potential opportunities.

**Ernest Put all the use cases and actors here : label it and delete this guide text when you finish**

**Ernest Put use case diagram for vido microfinance here and actors here : label it and delete this guide text when you finish. NB: not use case diagram for vido micro finance mobile , take not if it.**

**Ernest Put all the sequence diagrams here : label it and delete this guide text when you finish . NB: it should be in this order , depositing funds, withdraw, request loan , process loan**

**Ernest Put the use case diagram for vido mobile app here : label it and delete this guide text when you finish.**

**NON-FUNCTIONAL REQUIREMENTS**

* **Scalability**

Scalability subsumes the tendency of the system to handle rapid changes to workload and user depends. This is pertinent to the architecture of the system, including the hardware, software, technology, and procedures used in building the networks, applications and procedures that constitute the entire system. unequivocally, the adopted server architecture for the imminent system must be able to add resources and scale up to seamlessly handle increased customer demands and vast workload. Such requirement is plausible, as there are millions of potential customers and vehicles that could patronize the services of Vido Microfinance, and hence there is the need of a system that would not collapse when it is being rescaled to a larger volume, in order to meet such high demands.

* **Maintainability**

This requirement emphasizes on the time required for a feature of the system that needs to be fixed, upgraded to increase performance, or adapt to a changing environment. In the case of failure, the Vido Micro finance system must be able to be fixed and mounted back with the most minimum delay possible.

* **Reliability**

The imminent system must again be reliable thus, it should maintain the same efficiency even after extensive use. The system Reliability stresses on the probability of the system performing its intended function for long periods of time with no failure incurred, essentially the users of the system should always be satisfied with the operation of the system. In other words, the probability of failure should be absolute zero and such a requirement does not need much emphasis as it is evident that, potential end users of system would want satisfactory results with the use of a digital Microfinance system.

* **Usability**

Most importantly, the system should be easy to navigate, without extensive need for technical training. For what it is worth, it cannot be assumed each user of the proposed would technology inclined, the target audience must be highly considered. it is literally impossible to implement a system without first defining who the intended users of the system, the tasks such users will perform with the system, and the characteristics of the social environment in which it will be used. It also worth noting that, public perception of the system is crucial to its patronage, which is a primary goal of the Vido microfinance.

* **Compatibility**

As a nonfunctional requirement, system compatibility is used to ensure that, the system will run properly on different browsers, versions, OS, and networks successfully. Thus, the propensity of the proposed system to co-exist with another system in the same environment. As an example, cross browsing, cross platform is a common standard for most web-applications

In reference to the above, the potential system must be able to interact with hardware and software of different sources, without discrepancy and it should be proficient enough to function accurately in different browsers, databases, hardware, operating system, mobile devices, and networks. This requirement is inherent, regarding the fact that the system would be accessed by individuals with varying technology devices.

* **Security**

The security requirement must focus on the two areas of security that includes data security and system security. With the data security requirement focusing on the need for proposed system to incorporate protocols that will ensure that the information being kept by Vido Microfinance is not accidentally deleted, modified, stolen or otherwise misused. Similarly, the system security requirement stresses on the fact that, the proposed system must have safeguards, encryption protocols that ensure the systems is safe from malicious intrusion. Having to ensure the privacy of the data of clients whose data are being stored, protecting sensitive information and subsequently data monopoly thus warrants the need for a system that is robustly secure.

This would also eradicate the concern for the malware attacks, Backdoor attacks, Denial of service (DoS) attacks etc., that the organization ‘s new System could potentially face.

* **Availability**

The availability requirement of the proposed system accentuates on the need of the system to be to be available to perform its designated functions under normal conditions. In brief, the system must be highly resistant to failures. The architecture to implement such a system should be constructed such that the system maintains its operation, despite the failure of some of the individual components that form the system. Most significantly, a fault tolerant system has no service interruption and thus most desirable for a system with no room for errors like a digitalized car registration system.

A single error could lead to catastrophic results for the DVLA and accordingly, the proposed system should be void of any operational error.

* **Performance**

In todays’ digitally connected world, people are always expecting consumer and enterprise software at their fingertips to operate optimally in real time to enhance productivity. The performance of the system is therefore the measurement that relies on the set of metrics used to quantify both the efficiency and effectiveness of the implemented system.

In most cases, this metric explains how long a user must wait before a target operation (page renders, a request is processed or how quick documents are verified). In accordance, the metrics of the proposed system like its’s throughput, minimal, turnaround and response time should be efficient enough to conclude that the system passes the performance assessment. For instance, the loan-request processing page for 9,0000 concurrent users per hour must provide 4 seconds or less response time in a chrome desktop browser, including rendering of text, images over an LTE connection.

* **Capacity**

Here, we consider the volume of storage, more specifically, the amount of data and information the system can hold and whether it can accommodate expansion of data at any applicable point in time. The potential system must be equipped with copious amount of storage so it can keep up with growth in data, due to potential increase in the number of customers of Vido Microfinance

* **Localization**

Localization is that attribute of the system, which specifies, that the elements of the potential system, must align with the context of the potential operational environment. The elements of the Vido Microfinance system must align with the contexts (local languages, currencies, cultures, etc.) of the target audience. This is to ensure familiarity with the intended system by the potential users.

**THE TESTING STRATEGY**

Following the successful development of the Vido Microfinance system, the most quintessential thing to do find is to find out if there is any default or error present in the software so they are eliminated and thus increase the quality of the system. A comprehensive testing approach is crucial to ensure the reliability, functionality, and performance of the system. Specifically, we are going to test all individual constituent components of the system, a method most often referred to unit testing.

**Unit Testing**

By virtue of this strategy, we test the individual units or components of the system. We begin with testing the authentication of the system. A rudimentary method is to try to login in to the system with incorrect credentials and observe the feedback. The expected feedback should be an authentication error, and otherwise, is an indicative of an authentication malfunctioning on our application server. we again test to see if indeed various. From the way our authentication is implemented, once logged in the server should issue JSON web tokens that is logged in session to handle JWT authentication for remembering logged in sessions. So, to test this added on authentication also, we will close the system and reopen it, if the system remembers us and does not require us to login, then it indeed the implementation worked, and an otherwise feedback, would mean imply there is a lingering issue with our JWT authentication. Again, we will test the authorization of all the privileges for all roles: customer, manager, teller. We make sure for each successful login, there is server is able to correctly decode the tokens in the request headers and use it to correctly differentiate all roles and grant the respective privileges.

We will proceed to attempt to open a bank account without submit the required details to see if the system would it. For extensive scrutiny, we will submit invalid user identification documents to see if the system indeed rejects it or not

Next, we will legitimately create an account and attempt to deposit some funds into, our Vido Microfinance account, to analyze the functionality of this unit also. We will attempt to make deposits using all third-party payment gateways and verify to see if indeed they are successfully handling the transactions as supposed to and also, to determine if all continuous deposits are actually incrementing on the customer’s account dashboard. Similarly, we will try to make withdrawals, first by attempting valid withdrawals and then invalid withdrawals on the system, followed by loan request and all other activities.

Concurrently we will login as other roles to determine if all customer requests are indeed being routed to the respective Vido Microfinance’s staff and then further attempt to process these customer requests to determine if indeed, everything is working as expected by Vido Microfinance.

**POTENTIAL ISSUES AND CHANGES THAT COULD ARISE WITH THE SYSTEM**

The most salient issue that perhaps we could think of is potential increase in number of customers of Vido Microfinance and thus resulting in exponential increase in the amount of traffic that our application sever has to handle and perhaps causing a bottleneck To be able to mitigate this potential issue, we are going to use highly scalable backend , thus Java Spring boot for the backend of our server , and then further configure our backend architecture as a microservices architecture , such that when the requests tend to be overwhelming for our application’s backend , we can add more backends and distribute the traffic load across the added backends . Our system could then have two or more backends configured in one microservices, such that lets say the withdraw functionality is handled by one backend, the deposit functionality is handled by another backend and so on. This would ensure our system can withstand future increase in loads.

Similarly, future increase in users would constitute a corpus amount of data, and hence our hosting server should have a high storage and to accommodate this, we would select the most expensive hosting tier that is equipped with large volumes of storage and thus be able to meet the systems future storage requirement.

Again, there is the possibility of the Vido Micro finance transitioning into a fully operational bank, therefore offering full bank services being: bank transfers, receiving salary, AMT services, etc. This would mean our system should be able to handle all these services and as prudent measure, our entire codebase would be version controlled, using Git and implementing a continuous integration, continuous deployment (CI/CD) pipeline, which makes it very quick and easy for us to add new feature onto the system and easily deploy it instantly. The aforementioned measure would ensure our system is able to handle all needed features in future.

**TYPES OF RISKS THAT COULD AFFECT OUR POJECT**

Analyzing risk is an important part of the project planning process, having a clear understanding of potential risk our system could face helps in deducing ways to mitigate it. Just as any other software, our imminent system is subject to a variety of risks, which includes not but not limited to:

**Project Management and User Risks**

Poor project planning, or delays in development can negatively impact the success of the project. Our developing team not being in sync with each other and others leaving the team could threaten the survival of the project. Again, both the staff and customers of Vido Micro finance might be hesitant to adopt the new technology, especially if they lack digital literacy or find the app unintuitive.

**Cost Risk**

Our system could be subject to high unanticipated costs, which stems from unrealistic or non-detailed budgeting in our project planning phase. We might have been of the assumption, the system would be completed under budget, however, but midway through the development process, we could find out we have exhausted our budget and hence endangering the successful completion of the system.

**Security Risks**

For a finance system like ours, financial data, including loan details, customer information, and transaction history, is highly sensitive and thus tenuous security measures can lead to breaches, exposing customer information and putting them at risk of cyber theft and fraud. More so, Hackers could exploit vulnerabilities in the app or web platform to gain unauthorized access to sensitive data or manipulate financial transactions.

**Operational Risks**

Technical glitches or disruptions can cause outages in our system an outage which hinders customer access to financial services and potentially disrupting the services of Vido Micro finance. There might be potential bugs in the software that could lead to inaccurate calculations, incorrect transactions, or unintended consequences for customers' accounts. The is the possibility of the project not performing well as initially expected

Integrating our system with existing core banking systems and other payment gateways might be complex and lead to compatibility issues or data inconsistencies.