Research Article
An Experimental Validation of Public Cloud Mobile Banking

[1]Olawande Daramola, [2]Folake Olajide, [1]Adewole Adewumi and [1]Charles Ayo
[1]Department of Computer and Information Sciences, Covenant University, Ota,
[2]Eco-Bank Nigeria Ltd., Nigeria

Abstract: Currently, financial institutions incur huge expenditure to implement and maintain mobile banking (m-Banking) solutions and this cost is bound to rise significantly, as more customers subscribe to m-Banking services. Cloud computing has potential to facilitate reduced cost, high scalability and a variable cost structure that could guarantee cheaper, reliable and sustainable m-Banking in the long term. While the adoption of organizational private clouds seems natural for banks because of the sensitive nature of banking transactions, some have argued for the adoption of public clouds as a better alternative, despite concerns on issues such as trust, security and privacy. However, there is lack of sufficient empirical evidence in the literature on the suitability of public clouds for m-Banking. Hence, this study presents an investigation of the use of public cloud for m-Banking. A prototype cloud-based m-Banking application was developed using a public platform-as-a-service (Paas) cloud model, which was evaluated for usability and robustness in a controlled experiment. The evaluation result shows that m-Banking on public cloud is viable, if the cloud-based application is sufficiently robust and usable. The result also indicates that m-Banking services on public cloud are suitable for adoption by the banking industry.

Keywords: Cloud computing, mobile cloud banking, mobile computing, robustness, usability

INTRODUCTION

Mobile banking (m-Banking) entails the use of mobile devices to access banking services. It offers a new and more convenient way for customers to carry out banking transactions, by enabling customers to have pervasive access to essential front office banking services such as money transfer, bills payments and account status information. However, the cost of implementing and maintaining m-Banking solutions by financial institutions, remain high and this cost is bound to increase significantly, as more persons subscribe to m-Banking. There will be need to expand the hardware infrastructure (cost of data centre equipment-Data Servers, Network routers etc.) as the number of subscribers increase. Particularly, the cost of m-Banking can grow astronomically in highly populous developing countries-such as Nigeria, Pakistan, Bangladesh-where there is increasingly high mobile phone penetration and multiple millions of people are likely to subscribe to mobile banking services with time. The implication is that banks will have to incur more expenditure in order to ensure scalability and maintenance of system infrastructure for the sustainability of mobile banking initiatives in the long-term. A smart way to avoid this impending scenario is for banks to pursue cloud-based m-Banking solutions that can lead to reduction in cost of implementation and maintenance of m-Banking services in the long term.

Cloud computing is a utility-based model of computing that enables the provisioning of software and hardware services on the Internet that can be engaged by customers on demand on a pay-as-you-use basis. Cloud computing has potential to facilitate cheaper m-Banking solutions that is efficient, reliable and sustainable in the long term. It will ensure that banks:

- Incur lesser cost to implement m-Banking-cost of hardware and software infrastructure.
- Ensure high scalability as the subscriber base continues to grow.
- Leverage on the reliability and security of well-known cloud-based platforms in order to deliver robust mobile banking services to their customers.

In order to embrace cloud computing, banks have two options. First is the use of private cloud, where the organization bears the full cost of data centre infrastructure and the servers are resident within the organization. Given the nature of banking transactions, ordinarily this option appears to offer the most security and inspire confidence of all stakeholders, but it is more expensive. The second alternative, is for banks to host their m-Banking services using a public cloud

Corresponding Author: Olawande Daramola, Department of Computer and Information Sciences, Covenant University, Ota, Nigeria

infrastructure owned by a third party-service provider-which is cheaper, could be very reliable, but more risky, given the sensitive nature of bank transactions. The merits of using public cloud by financial institutions for banking have been advocated in the literature (Viale, 2011). According to Crossman (2010), the use of public clouds will ensure significant reduction in the cost of ownership of m-Banking services and allow leveraging on the reliability, security and high scalability of the hardware and software infrastructure provided by the public cloud service provider to ensure sustainable m-Banking.

However, although the possibility of public cloud m-Banking has been envisioned in the literature, its viability is yet to be empirically validated. Hence, the key motivation for this study is to investigate the viability of public cloud m-Banking and to determine the basis for wider adoption and acceptability, particularly in a developing World context using Nigeria as a case study. To do this, a Platform-as-a-service (Paas) cloud model-Google App Engine-was used to develop a prototype m-Banking application that affords basic m-Banking services. The prototype m-Banking application was evaluated using a controlled experiment where a group of users tested the robustness and usability of the system and assessed its readiness for m-Banking. The result of the evaluation indicated that the application was considered usable, sufficiently robust and potentially suitable for adoption by industry operators.

LITERATURE REVIEW

Mobile cloud computing integrates cloud computing and mobile technology to enable cloud services to be accessed via mobile devices (Niroshinie *et al*., 2013). Hence, mobile cloud banking is one of the emerging applications of mobile cloud computing. In this section, we present a background on relevant issues such as to mobile banking, cloud computing and a report of related works on mobile cloud banking in the literature.

Mobile banking: Mobile banking (viz. m-Banking) is the model of banking that thrives on the use of mobile devices to access financial information. Mobile banking allows customers to undertake typical front-office transactions such as checking account balance, fund transfer and bills payment, in a ubiquitous, convenient and interactive way via a mobile device. The different models of m-Banking that exist include SMS-banking, WAP-enabled or Java-enabled banking on mobile devices (Mallat *et al*., 2004; Ibikunle and Mayo, 2012). SMS-banking involves the use of GSM technology's Short Message Services (SMS) to query the data server of a bank for specific banking operations. Typically, a user sends a SMS to the bank's server and the server accepts the message, processes it and sends result back to the user's phone, tablet or PDA. WAP or Java

enabled banking on mobile devices entails the use of a micro-browser to pull relevant information from a bank's Website using a WAP gateway to display on a user's phone. Typically, the user sends a request through a mobile client application and gets a response with page content returned to his phone using GPRS technology (Mallat *et al*., 2004). Banks in many parts of the World are today, offering m-Banking services. For example, Scandinavian Nordea bank provides users with a wireless banking application that is based on the same infrastructure as its Internet bank (Mallat *et al*., 2004). Prominent m-Banking initiatives in the developing World include M-PESA-Kenya (Hughes and Lonie, 2007), Sri Lanka (Wijetunge *et al*., 2008), Bangladesh (Alam *et al*., 2010), Nigeria (UNCTAD, 2007; Ayo *et al*., 2012), G-cash and Smart Money-Philippines (Owens and Bantug-Herrera, 2006).

Cloud computing: According to Mell and Grance (2011), Cloud computing enables on-demand access to a shared pool of computing resources (e.g., network, servers, storage, applications and services) that can be dynamically provisioned and released over the Internet with minimal effort by the cloud services provider. The key characteristics of cloud computing includes (Mell and Grance, 2011):

Broad network access: This means it affords access to services via the Internet or other extensive networks using a standard protocol such as HTTP, WAP etc., that is independent of the end user device.

On-demand provisioning: This means that customers are automatically provided what they need as requested without manual intervention.

Elasticity: This means the services are scalable, growing or shrinking dynamically based on the emerging customer resource requirements.

Metering: This means that the services rendered to customers are measured and the customers pay provided services according to their level of consumption.

Resource pooling: This means computing resources are made available to multiple consumers using technologies such as virtualization and multi-tenancy resulting in cost saving for the consumers.

The fundamental service models of cloud computing are:

Software as a Service (SaaS): A is service model where the provider makes software applications to the customers to use on a pay-as-you-use basis. Examples of SaaS include Salesforce, Cloud9 Analytics, CloudSwitch and Google.

Platform as a Service (PaaS): A is service model where the provider allows users to create Web

applications easily without the burden of buying and maintaining the underlying software and cloud infsrastructure. Examples of PaaS include Apprenda, Google Apps Engine, Engine Yard, Microsoft Azure and Cloudscale.

Infrastructure as a Service (IaaS): is a service model where the provider provisions hardware resources such as processing, storage, networks and other fundamental computing resources to enable end user tasks. Examples of IaaS include Amazon EC2, GoGrid and Rackspace.

There also exist four deployment models of cloud, which are:

Private cloud: This is when the cloud solution is hosted and managed within the same organization without involving an external party.

Community cloud: This is when the cloud solution serves the exclusive interest of a community. Members of the community or a third party may provide the service.

Public cloud: This is when a single provider makes cloud solutions available for public use. Resource pooling, measured services and scalability are core characteristics of this kind of model.

Hybrid cloud: This is when the model of deployment combines the advantages of any of private, public and community cloud. For example, confidential data can reside in a private cloud, whereas less sensitive data can be available on the public cloud.

Related work: Public cloud m-Banking is gaining prominence around the World and a few of such initiatives have been reported in the literature. M-PESA is a public cloud mobile banking initiative in Kenya that allows users to carry out banking transactions on their mobile devices (Hughes and Lonie, 2007). The service currently hosted on Rackspace cloud platform. M-PESA is also operational in countries like South Africa, Tanzania, Afghanistan and Fiji. Monet is a Pakistan-based mobile banking enabler that facilitates banking transaction via mobile devices. Monet, which was launched in 2012, provides opportunities for a wide range of financial stakeholders-banking agents and mobile network operators-to undertake a variety of financial transactions through their mobile phones. Monet is hosted on IBM SmartCloud technology solution in order to enhance service efficiency and expansive coverage across the Pakistan (IBM News, 2012).

OVERVIEW OF METHODOLOGY

Based on interaction with relevant stakeholders in the banking sector (customers, bankers, information technology personnel in banks) and a study of existing
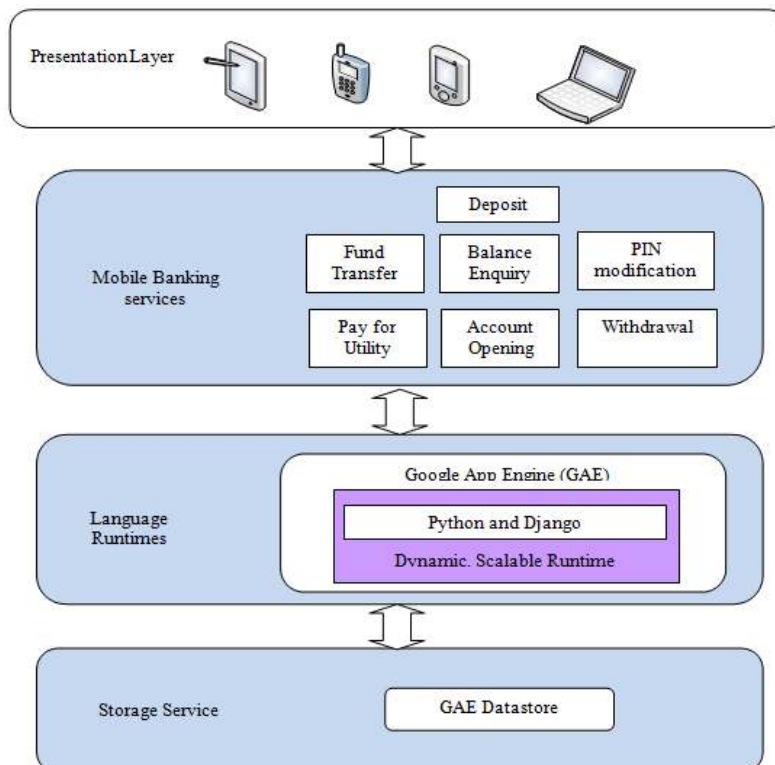

Fig. 1: Architecture of the mobile banking application

mobile banking platforms, we identified the following requirements as core for mobile cloud banking:

- An account creation module
- Deposit account module-by admin
- A funds withdrawal module
- A balance checking module
- A modify PIN module
- A funds transfer; as well as a utility payment module

Moreover, the second author involved in the study is an IT-biased practising banking professional. The layered architecture of the proposed mobile banking application is as shown in Fig. 1.

The first layer is the presentation layer where different devices such as smart phones, tablet computers, laptops and PDAs access the mobile banking application over the Internet.

The second layer is the mobile banking services layer where the mobile banking application runs. It leverages on the Cloud Computing Services layer and the Support Services layer of Google Apps Engine platform.

The third layer is the language runtime layer that allows developers to use familiar development tools and environments to build their applications. This includes the Python, Java and Go languages. For this study, Python programming language and Django framework was used to develop the mobile banking application.
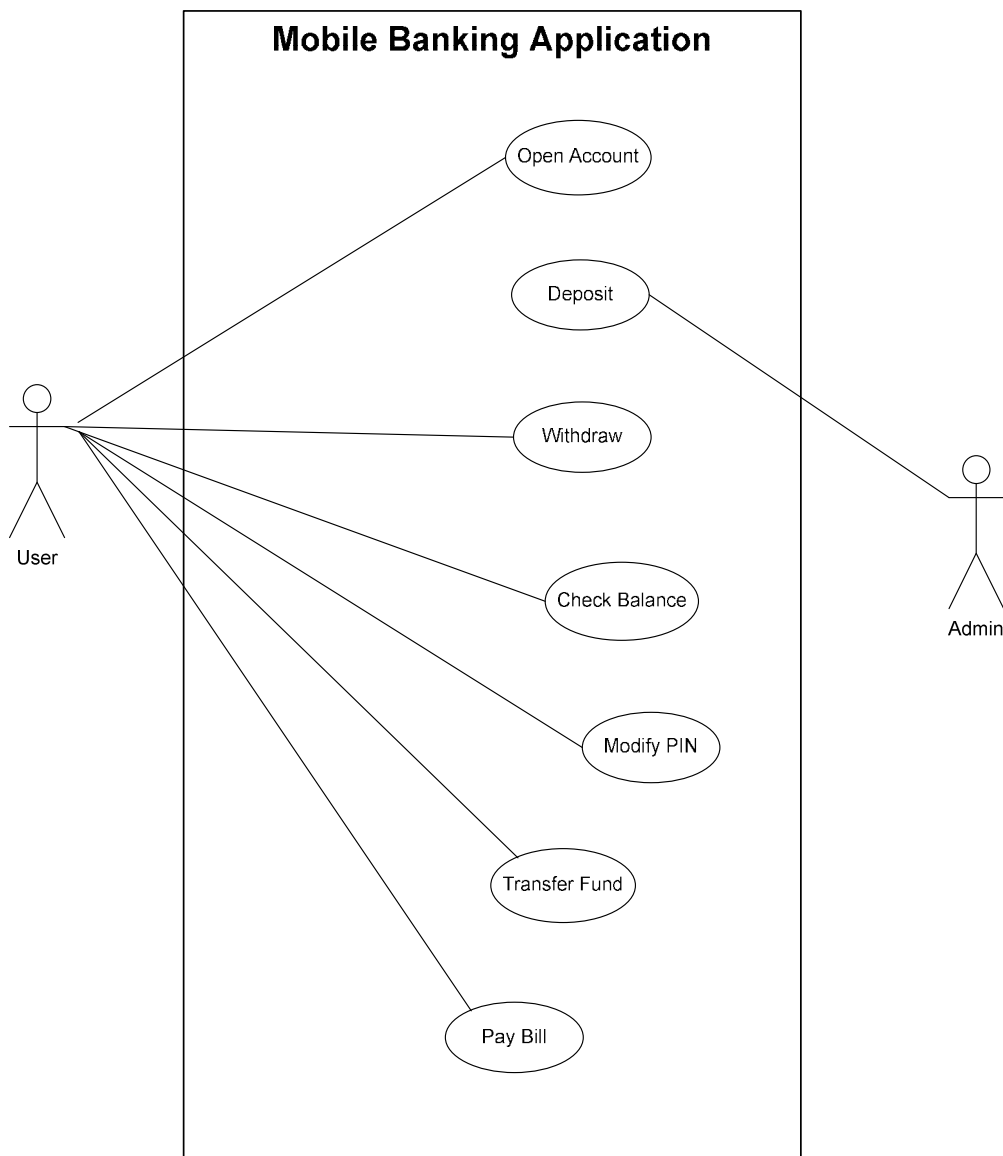

Fig. 2: Use case diagram of mobile banking application

```
          User
-Name        : String   (idl )
-Phone number       : String   (idl )
```

```
          User Account
-Account name        : String   (idl )
-Account number         : String   (idl )
+ Open    ()
+ Deposit    ()
+ Withdrawal     ()
+ Check balance       ()
+ Modify pin      ()
+ Transfer Fund       ()
+ Pay Bill   ()
```

```
          Bank
-Bank name        : String   (idl )
-Bank code      : String   (idl )
```

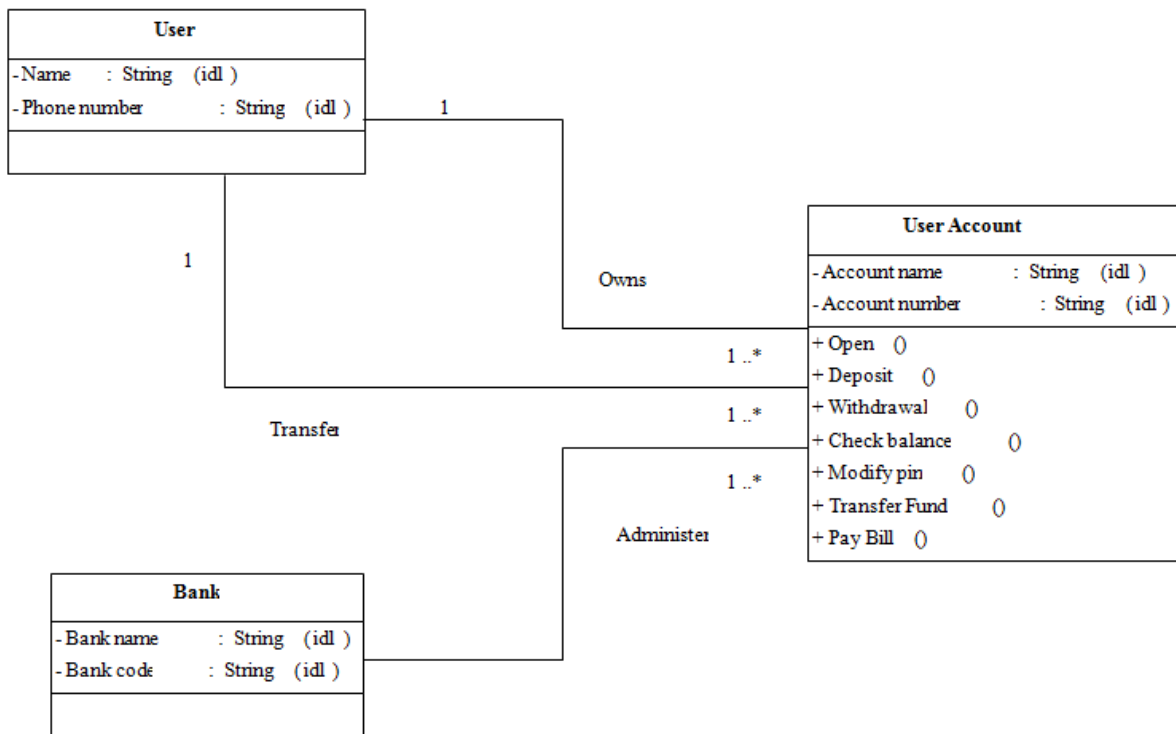Owns   1 ..*

Transfer   1 ..*

Administer   1 ..*

Fig. 3: A view of the class structure of the cloud-based mobile banking application

The fourth layer is the storage service layer that consists of the Google Apps Engine Datastore (https://developers.google.com/appengine/docs/python/datastore/), which is a schema-less object data store providing robust, scalable storage for Web/mobile applications. It provides a rich data modelling API and SQL-like query language called GQL.

DESIGN AND IMPLEMENTATION OF THE MOBILE BANKING APPLICATION

A use case model was used to capture the expected basic functions of the mobile banking application. The use case diagram in Fig. 2 shows the various functions that can be performed by a user that is interacting with the mobile banking application. The user-centric functions include the opening of an account; withdrawal from an account; user's changing of the PIN; checking an account balance; transferring funds from one account to another, payment of utility bills. It is also possible for the admin personnel to credit an account with deposit from a customer.

Google Apps Engine was used as the cloud infrastructure. The m-Banking application was implemented using Python programming language on the Google Apps Engine. Python is an interpreted, general purpose and a high level programming language, whose design philosophy emphasizes code readability. Python also supports multiple programming paradigms, primarily but not limited to object-oriented, imperative and to a lesser extent, functional programming styles. The relative simplicity, sufficiently rich features and better orthogonality of Python when compared to Java makes it our preferred choice of language among the three options afforded by the Google Apps Engine platform.

Figure 3 shows the first screen a user will encounter when logged on to the application at http://mbankapp.appspot.com. There are seven buttons on this main menu (Fig. 4) that serve unique functions such as Open Account, which redirects a user to the open new account form (Fig. 5); Check Balance-which allows the user to get his/her current account balance (Fig. 6); Fund Transfer, which allows a user to transfer money from his own account to another (Fig. 7).

Evaluation of the system: A mobile banking application is a mission-critical software that must be efficient and highly available for ubiquitous access by users. This means that the software must be sufficiently robust. Robustness is the measure of fault tolerance of system that represents its capacity to withstand rough handling. A controlled experiment was performed in order to evaluate the robustness and usability of the m-Banking application. In the controlled experiment, the participants comprising of undergraduate and postgraduate students of the Department of Computer and Information Sciences of Covenant University,
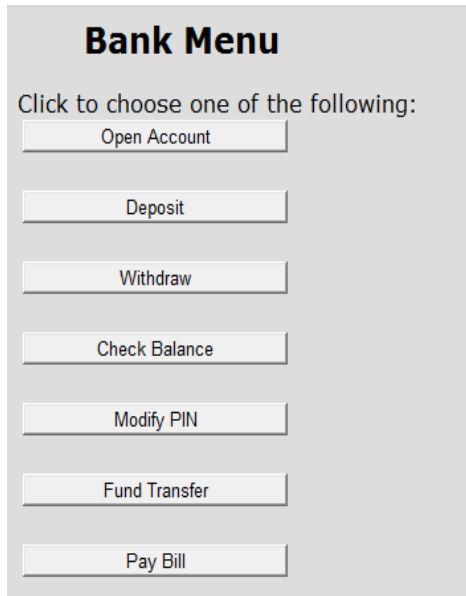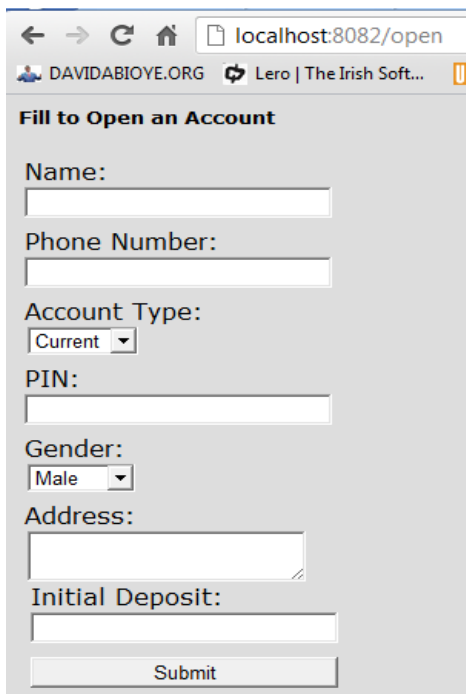
Fig. 4: Main menu

Fig. 5: Form for account opening

Nigeria, were presented with eight usage scenarios. Four of the use cases were right uses of the software while four were wrong uses (misuse) of the software as presented in Table 1.

The participants were instructed to engage the mobile application based on the description of the 8 scenarios under a controlled setting and thereafter fill out a post-experiment questionnaire. A pre-experiment questionnaire was also administered in order to gain
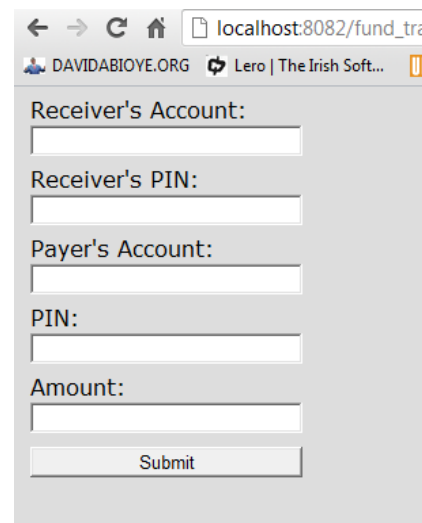

Fig. 6: Form for checking account

Fig. 7: Funds transfer form

understanding of the background of the participants, in terms of their knowledge of mobile banking, mobile devices, fund transfer and cloud computing. The data in the post-experiment questionnaire were collected using a five point Likert-scale:

- Strongly disagree
- Disagree
- Neutral
- Agree
- Strongly agree

In all, there were 18 participants. The participants evaluated the system for robustness and general usability. The robustness attributes considered are data validation, transaction integrity, fault tolerance and security control. The usability attributes that were considered are ease of use, satisfaction, intention to use, completeness and usefulness. The participants' rating of the usability and robustness attributes of the system as collected from the respondents on a Likert scale of (1-5).

Table 1: Use/Misuse cases of four core features of the mobile banking application

Feature	Use case	Misuse case
Creation of a new account	1. Choose the open account option from the Bank Menu 2. Fill the necessary account details (Name, Phone Number, Account type, PIN, Gender Address and Initial deposit) 3. Submit the form by clicking the Submit button	1. Choose the open account option from the Bank Menu 2. Fill the necessary account details but this time omit the initial deposit and either of or both the PIN and phone number 3. Submit the form clicking the Submit button [observe system's response] 4. Input a zero balance or an amount below the bank's minimum (N500) in the initial deposit field 5. Submit the form by clicking the Submit button [observe system's response]
Cash deposit	1. The Admin personnel choose the deposit option from the Bank Menu 2. Admin fill in the required details-account name, pin and amount to be deposited 3. Admin submit the form to complete the task	1. He chooses the deposit option from the Bank Menu 2. He fills in the required details but this time fill in an account name that you did not create and pay into the account 3. He submits the form by clicking the Submit button [observe system's response]
Cash withdrawal	1. Choose the Check Balance option from the Bank Menu 2. Fill in your account details to verify the amount in your account 3. Click 'Home' hyperlink to return to the bank menu 4. Select the withdraw option from the Bank Menu 5. Fill in the required details-account name, pin and amount to be withdrawn with minimum balance of N500 6. Submit the form to complete the task	1. Choose the Check Balance option from the Bank Menu 2. Fill in your account details to verify the amount in your account 3. Click 'Home' hyperlink to return to the bank menu 4. Select the withdraw option from the Bank Menu 5. Fill in the required details-account name, pin and amount to be withdrawn above the account balance 6. Submit the form by clicking the Submit button [observe system's response]
Funds transfer	1. Choose the fund transfer option from the Bank Menu 2. Fill in the required details-payer's account, payer's pin and amount being paid as well as the receiver's account and phone number. 3. Submit the form to complete the task	1. Choose the fund transfer option from the Bank Menu 2. Fill in the required details-payer's account, payer's pin and amount but omit the receiver's account and phone number 3. Submit the form by clicking the Submit button [observe system's response] 4. Fill in the required details and transfer amount exceeding the payer's account balance 5. Submit the form by clicking the Submit button [observe system's response]

Table 2: Participants' background

	Never heard about it %	Read about it %	Have little experience %	Have sufficient knowledge %	Expert %
Mobile banking	5.60	33.3	22.2	38.9	0.00
Mobile devices	0.00	0.00	5.60	50.0	44.4
Fund transfer through m-Banking	0.00	16.7	50.0	22.2	11.1
Cloud computing	16.7	44.4	22.2	11.1	0.00

RESULTS

The data collected from the post-experiment questionnaire was analysed by using the Statistical Package for Social Sciences (SPSS) to generate the mean, frequency distribution and standard deviations values that pertain to the metrics of robustness and usability used to evaluate the mobile banking application. The result of the pre-experiment questionnaire, which reveals the background knowledge of participants, is shown in Table 2. The result show that most of the participants have sufficient background knowledge in key areas relevant to the evaluation such as mobile banking, mobile devices. Most are knowledgeable about mobile banking services (94.4%) although not experts. A total of 77.7% of the participants have some of knowledge of cloud computing, although they do not consider themselves as expert. All participants have varying degree of knowledge about mobile devices and funds transfer.

Table 3: Rating of system's robustness attributes by users

Metrics	Mean	SD
Data validation	4.53	0.72
Transaction integrity	3.56	1.19
Fault tolerance	3.44	0.86
Security control	3.06	1.880

SD: Standard Deviation

Table 4: Rating of system's usability attributes by users

Metrics	Mean	SD
Ease of use	4.56	0.511
Satisfaction	3.67	0.970
Intention to use	3.18	0.808
Completeness	3.61	0.530
Usefulness	3.61	0.916

SD: Standard Deviation

Table 3 and 4 show the results obtained from the data analysis of the post-experiment questionnaire for the robustness and usability metrics respectively. The result obtained from the participants rating of the cloud mobile banking application showed that it obtained a rating of above 3.0 out of a maximum of 5.0 in all

attributes of robustness and usability used to assess the system. For robustness, the m-Banking application was rated highest for its data validation attribute with a score of 4.56 and the lowest in security control with a score of 3.06. In terms of usability, the system obtained the highest rating in the aspect of ease of use with a sore of 4.56, while it had the lowest in intension to use with a score of 3.18.

DISCUSSION

In this section, we shall discuss the observations from the results of the experiment.

Robustness rating of the system: The robustness of a system is the measure of its capacity and capability to withstand adverse situation (Sheard and Mostashari, 2008). The attributes that were used to assess the robustness of the m-Banking application are: fault tolerance, data validation, transaction integrity and security control:

- **Fault tolerance:** The mean rating for the fault tolerance attribute is 3.44. Fault tolerance is the ability of a system to handle adverse treatment by the user. It is also the ability to continue to function in spite of abnormalities in input. The mean score of 3.44 out of 5.0 shows that majority of the participants agree that the m-Banking application is sufficiently fault tolerant.
- **Data validation:** The mean rating for data validation attribute is 4.53 out of 5.0. Data validation is the ability of a system to disallow the input of a wrong type of data by the user at any given instance. Effective data validation will remove the tendency for a system to crash due to wrong input. The high score of 4.53 indicates that most participants strongly agree that the data validation attribute of the cloud-based mobile banking application is satisfactory.
- **Transaction integrity:** The mean rating for the transaction integrity attribute is 3.56 out of 5.0. Transaction integrity is the assurance that the results achieved by a system are as expected. An important implication is that appreciable number of participants believes that the platform can handle sensitive transaction such as funds transfer.
- **Security control:** The mean rating for the security control attribute was 3.06 out of 5.0. Security control refers to safeguards or countermeasures built into a system in order to help it avoid, counteract or minimize security risks. The rating by participants indicates that they agree that the application possesses adequate access control. However, many will prefer a stronger security control in the application.

The usability rating of the system: The usability of a software system is the measure of its potential to satisfy

the needs of a user. The attributes that were used to assess the usability of the cloud-based mobile banking application are completeness, intention to use, ease of use, satisfaction and usefulness:

- **Completeness:** The mean rating for completeness attribute is 3.61 out of 5.0. Completeness refers to the ability of a system to start and finish a given task. The rating indicates that the participants believe that system completes tasks well.
- **Intention to use:** The mean rating for intention to use attribute is 3.18 out of 5.0. Intention to use refers to the willingness of a would-be user to make use of newly developed/introduced software. The rating indicate a general disposition to use the software for mobile banking, however a significant number of participants are still apprehensive.
- **Ease of use:** The mean rating for ease of use attribute was 4.56 out of 5.0. Ease of use refers to how user-friendly a system is. The ratings of participants indicate that they find the application very interactive, easy to use and uncomplicated. This is a vital attribute, which an application that is designed for use by all and sundry should have in order to be effective.
- **Satisfaction:** The mean rating for the satisfaction attribute is 3.67 out of 5.0. Satisfaction refers to the opinion of a user on how well a system meets his expectation. The rating indicates that most of the participants were satisfied with the operations provided by the software.
- **Usefulness:** The mean rating for this attribute is 3.61 out of 5.0. Usefulness is the extent to which a user finds a system beneficial. The rating show that majority of the participants agree that the cloud-based software would be useful as a mobile banking application.

From the result of the evaluation, we can conclude that the m-Banking application obtained generally good rating from users in terms of usability and robustness and those users are quite disposed to using the application for m-Banking services even though it is resident on a public cloud platform.

Threats to validity: In this section, we briefly examine the threats to validity of the outcome of the controlled experiment used to evaluate the mobile cloud banking application based on the categories defined in Wohlin *et al.* (2000). Each threat is considered in a short section before giving a summary of validity claims.

Conclusion validity: Conclusion validity is concerned with the ability to draw the right conclusions about the relationship between the treatment and the outcome of the performed controlled experiment:

- **Low statistical power:** In a sophisticated IT-based domain such as cloud mobile banking, having a

large pool of participants is not necessarily an advantage as compared to having persons with the requisite knowledge and skill competencies. The number of participants used in the experiment (18) is considered sufficient for a first proof-of-concept evaluation in the mobile cloud banking domain. Evidence in the literature show that a minimum of 5 participants are sufficient to get valid opinion on the usability of a tool (Nielsen and Landauer, 1993; Sauro and Kindlund, 2005; Turner *et al.*, 2006; Lewis, 2001; Lewis, 2006).

- **Reliability of measure:** The descriptive statistics measure (mean and standard deviation) used for assessing the rating of the cloud-based mobile banking application by the participants are regularly used in empirical research. Therefore, misunderstanding or misinterpretation of the measures is not likely.
- **Reliability of treatment implementation:** All participants were provided with a relevant introduction and same instruction set for the experiment, containing all relevant uses cases before the experiment. There was no reported case of where the participants had misunderstood the instructions.

Internal validity: Internal validity is concerned with whether something else other than the treatment influenced the outcome of the experiment. Two issues are important here:

- **Instrumentation:** threats due to quality of information provided by the experiment in terms of descriptions, forms and diagrams etc., the 8 usage scenarios (4 right uses and 4 misuses) were adequately described in the instruction note given to the participants. Details included a step-by-step definition of what to do to execute specific scenarios.
- **Selection:** threats due to the way participants were selected: the major criterion for selection was IT knowledge since mobile banking requires hands on knowledge of use of mobile devices to browse the Web. The participants in the experiment were undergraduate and postgraduate students of Computer Science of the Covenant University, who have basic training and practical knowledge in relevant area such as software development, mobile computing, cloud computing and Internet technology. Thus the outcome of the experiment was only determined based on the treatment.

Construct validity: Construct validity is concerned with whether the experiment is realistic i.e., does it reflect the real world phenomena that are to be observed:

- **Instructions to the participants:** All participants got the same introduction for the experiment. Therefore, they had the same starting point, none

was at a disadvantage and hence the experiment results obtained from the participants was due to their interpretation of the same information for all of them.
- **Mono-method:** The only specified tasks to perform are to follow a list of steps to realise specific right uses and wrong uses of the m-Banking application. Hence, the feedback obtained depends only on the two types of tasks (2 variables-usability, robustness). In addition, the assessment of the differences in treatment is based on the user's perception of the two software quality metrics-usability and robustness. Thus, the mono-method bias is not relevant to the experiment.

External validity: External validity is concerned with generalization-where and when are the conclusions applicable and can we generalize from the experiments to industrial practice? The key issue here is whether the results can be generalized to the Nigerian banking industry. For the evaluation experiment, we used 18 participants from Covenant University. However, the participants used for the controlled experiment are quite knowledgeable in the relevant areas such as software development, mobile computing and Internet technology since they have taken courses in these areas as Computer Science and Information Systems majors. This makes them suitable as reasonable substitutes for real experts in a first evaluation (Runeson, 2003; Berander, 2004; Svahnberg *et al.*, 2008).

Based on the discussions in this section, it was assumed that there were no serious threats to validity for our conclusions on evaluation experiment that was performed. In addition, the relatively good rating of the mobile cloud banking application by the participants indicates that application has sufficient merit to be adopted for mobile banking.

Lessons learnt: The following summarises the lessons learnt from the experimental validation of public cloud mobile banking that was performed in this study:

- Only essential front-office mobile banking operations (such as withdraw, checking of balance and fund transfer) are suitable for the cloud. The use of cloud technology does not warrant extending mobile banking services beyond its current scope.
- Although it is possible to have a robust application hosted on the cloud, security will still be a serious concerns for cloud-based implementations. Hence, adequate security control measures must be implemented to ensure optimal security on cloud-based mobile banking platforms.
- Public cloud mobile banking has the potential to reduce the cost of mobile banking, offering reliable and secure mobile banking services, thereby reducing the long-term cost of mobile banking compared to the cost of other alternatives.

- Many people are still sceptical about the security and reliability of public cloud m-Banking. Hence, public cloud provider will need to do more to improve the trust of users in public cloud m-Banking services.
- Hybrid cloud will be the most suitable for banks, such that confidential data and operations are handled in a private cloud while the more open operations are implemented on the public cloud.

CONCLUSION

In this study, we have performed an experimental validation of public cloud m-Banking. To do this, A prototype cloud-based m-Banking application was developed using the Google Apps Engine, which was then evaluated in a controlled experiment. The result of the evaluation shows that m-Banking on public cloud is viable, if the cloud-based application is sufficiently robust and usable. The result also indicates security is still a concern for many users and while the level of trust in public cloud systems still need to be improved. It was also revealed that public cloud application that is sufficiently robust and has good usability attribute would be fit for adoption in practice by banks.

REFERENCES

Alam, S.B., H.M.D. Kabir, M.N. Sakib and A.B.M.R. Sazzad, 2010. A secured electronic transaction scheme for mobile banking in Bangladesh incorporating digital watermarking. Proceeding of IEEE International Conference on Information Theory and Information Security (ICITIS, 2010), pp: 98-102.

Ayo, C.K., U.I. Ukpere, A. Oni, U. Omote and D. Akinsiku, 2012. A prototype mobile money implementation in Nigeria. Afr. J. Bus. Manage., 6(6): 2195-2201.

Berander, P., 2004. Using students as subjects in requirements prioritization. Proceedings of the International Symposium on Empirical Software Engineering (ISESE'04), pp: 167-176.

Crossman, P., 2010. Banks take to Cloud Computing, Bank System and Technology Survey. Retrieved form: http://www.banktech.com/architecture-infrastructure/bst-survey-banks-take-to-cloud-computing/226100004.

Hughes, N. and S. Lonie, 2007. M-PESA: Mobile money for the "unbanked" turning cell phones into 24-hour tellers in Kenya. Innov. Technol. Gov. Globaliz., 2(1-2): 63-81.

Ibikunle, F.A. and Z.O. Mayo, 2012. A model for implementing mobile banking in developing countries (e.g. Nigeria). Int. J. Eng. Technol., 2(3), ISSN 2049-3444.

IBM News, 2012. Leading Pakistan Mobile Banking Provider Uses IBM Cloud to Expand Delivery of Secure Financial Services on Mobile Phones. Retrieved form: http://www-03.ibm.com/press/us/en/pressrelease/39290.wss.

Lewis, J.R., 2001. Introduction: Current issues in usability testing. Int. J. Hum-Comput. Int., 13(4): 343-349.

Lewis, J.R., 2006. Sample sizes for usability tests: Mostly math, not magic. Interactions, 13(6): 29-33.

Mallat, N., M. Rossi and V. Tuunainen, 2004. Mobile banking services. Communi. ACM, 47(5): 42-46.

Mell, P. and T. Grance, 2011. The NIST Definition of Cloud Computing. Special Publication 800-145, National Institute of Standards and Technology. Retrieved form: <URL: http://csrc.nist.gov/publications/nistpubs/800-145/sp800-145.pdf>.

Nielsen, J and T. Landauer, 1993. A mathematical model of the finding of usability problems, Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, pp: 206-213.

Niroshinie, F., S. Loke and W. Rahayu, 2013. Mobile cloud computing: A survey. Future Gener Comp. Sy., 29: 84-106.

Runeson, P., 2003. Using students as experiment subjects-an analysis on graduate and freshmen student data. Proceeding of 7th International Conference on Empirical Assessment and Evaluation in Software Engineering (EASE'03), pp: 95-102.

Sauro, J. and E. Kindlund, 2005. A method to standardize usability metrics into a single score. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05), pp: 401-409.

Sheard, S. and A. Mostashari, 2008. A framework for system resilience discussions. Proceedings of the 18th Annual International Symposium of INCOSE. Utrecht, Netherlands.

Svahnberg, M., A. Aurum and C. Wohlin, 2008. Using students as subjects-an empirical evaluation. Proceeding of 2nd International Symposium on Empirical Software Engineering and Measurement, pp: 288-290.

Turner, C.W., J.R. Lewis and J. Nielsen, 2006. Determining Usability Test Sample Size. In: Karwowski W. (Ed.), International Encyclopedia of Ergonomics and Human Factors. CRC Press, Boca Raton, FL, pp: 3084-3088.

UNCTAD, 2007. Science and technology for development: A new paradigm for ICT. Information Economy Report, New York.

Viale, E., 2011. Cloud Computing Changes the Game. Retrieved form: http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Cloud-Computing-Changes-the-Game-for-Banks-Podcast-Transcript.pdf.

Wijetunge, S., P. Wijesinghe and U. Gunawardana, 2008. Social impact of SMS in Sri Lanka. Proceeding of 4th International Conference on Information and Automation for Sustainability (ICIAFS, 2008), pp: 346-350.

Wohlin, C., P. Runeson, M. Höst, M.C Ohlsson, B. Regnell and A. Wesslén, 2000. Experimentation in Software Engineering: An Introduction. Kluwer Academic Publishers, Boston, MA, USA.