ADAMA SCIENCE & TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING & COMPUTING

DEPARTMENT OF COMPUTING

A MASTER'S THESIS

ON

DESIGNING AN INTEROPERABLE CLOUD COMPUTING FRAMEWORK FOR BANKING INDUSTRY (ETHIOPIA AS A MODEL)

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN SOFTWARE ENGINEERING

By:

Sisay Tigistu

February 2017

Adama, Ethiopia

ADAMA SCIENCE & TECHNOLOGY UNIVERSITY

SCHOOL OF ELECTRICAL ENGINEERING & COMPUTING

DEPARTMENT OF COMPUTING

DESIGNING AN INTEROPERABLE CLOUD COMPUTING FRAMEWORK FOR BANKING INDUSTRY (ETHIOPIA AS A MODEL)

A THESIS SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN SOFTWARE ENGINEERING

By: Sisay Tigistu

Advisor: Dr. Derejey Yohannes (Ph.D.)

February 2017

Declaration

I declare that this study is my original work submitted in partial fulfillment of the requirement for the degree of master of science in software engineering and has not been submitted for any Degree or Diploma in any University or conference. To the best of my knowledge, all source of materials used for the study has been duly acknowledged. I have undertaken the study with the guidance and support of the research advisor.

Sisay Tigistu

Signature:_____

Advisor: Dr. Dereje Yohanis

Signature:_____

February 2017

DESIGNING AN INTEROPERABLE CLOUD COMPUTING FRAMEWORK FOR BANKING INDUSTRY (ETHIOPIA AS A MODEL)

Name and signature of members of the examining board

Name	**Signature**	**Date**
_____ Advisor	_____	_____
_____ DGC Chairman	_____	_____
_____ External Examiner	_____	_____
_____ Internal Examiner	_____	_____
_____ Department Head	_____	_____
_____ School Dean	_____	_____

Acknowledgement

This thesis represents the peak point of my studies, so far, and it incorporates knowledge gathered from different sources, academics and professionals alike. First of all, I would like to express my deepest gratitude to the almighty God for his mercy, guidance, support; Glory to God. Next, I would like to express my sincere gratitude to my research advisor Dr. Derje Y. (Ph.D., Assistant professor in Adama science and Technology University) for his advice, patience and constructive criticisms of this work. Thirdly, my thanks goes to all interviewees; Mr. Ermias Debebe (CBE Adama district IT manager), Mr. Dawit Desalegn (System administrator in Birhan international bank), Mr. Samuel (E-banking representative in Birhan international bank), Mr. Kumneger Worku (System Administrator in Debub Global bank) and Managers of United bank in Adama district office for their honesty suggestion, cooperation in interview and their voluntariness in explaining the operating method of the current banking environment.

Finally, I would like to express my thanks to my friend Hadush Mesfin for his support in developing the framework. I am also thankful to my father, mother, sisters, brothers, friends and Samara university computer science staff members for their encouragement and suggestion that helped me to accomplish this thesis.

Tables of contents

List of Figure

List of Table

List of Appendices

List of Acronyms

AAU: Addis Ababa University

ACH: Automatic Clearing House

API: Application Programming Interface

APIs: Application Programming Interfaces

ATM: Automated Teller Machine

BFSI: Bank and Financial Service Institute

BPC: Business Process Communication

CBE: Commercial Bank of Ethiopia

CEO: Chief of Executive

CORE: Centralized Online Real-time Environment

CPU: Central Processing Unit

CRM: Customer Relation Management

CSA: Customer Security Alliance

CSP: Cloud Service Provider

DB: Database

DBs: Database Systems

EBI: Ethiopian Banking Industry

EBICC: Ethiopian Banking Industry Community Cloud

EBIs: Ethiopian Banking Industries

EJB: Enterprise Java Beans

ERP: Enterprise Resource Planning

GTP: Gross Transformation Plan

GUI: Graphical User Interface

HTTP: Hyper Text Transfer Protocol

IaaS: Infrastructure as a Service

IBCC: Indian Banking Community Cloud

IBM: International Business Machine

ICT: Information Communication Technology

ICTs: Information Communication Technologies

ID: Identification

IDRBT: Institute for Development and Research in Banking Technology

IEEE: Institute of Electrical and Electronics Engineers

IMF: International Monetary Fund

ISO: International Standard Organization

IT: Information Technology

JEE: Java Enterprise Edition

JSF: Java Server Faces

JPA: Java Persistence API

MYSQL: My Structured Query Language

NBE: National Bank of Ethiopia

NIST: National Institute of Standards and Technology

PaaS: Platform as a Service

PIN: Personal Identification Number

POS: Point Of Sale

PSS: Primer Switching Solution

QoS: Quality of Service

RTGS: Real Time Gross Settlement

SaaS: Software as a Service

SLA: Service Level Agreement

SME: Small and Medium Enterprise

SMS: Short Messaging Service

SOA: Service Oriented Architecture

SPI: Software-Platform-Infrastructure

UML: Unified Modeling Language

US: United States

VM: Virtual Machine

VMs: Virtual Machines

VPC: Virtual Private Cloud

VPN: Virtual Private Network

Abstract

Banking industries are a catalyst for a certain countries economy as their role in converting surplus to deficit. Due to the effect of globalization and advancement in technology the business environments are highly dynamic and people needs to access their money in the bank just as in their pocket. To answer this fast access need of a customer it is difficult for a single bank because it needs billions of dollars to afford enough banking access channels and to open a new branch office. The current banking architecture in Ethiopia has many redundancies, limited service deliveries, has no advantages of cost from the bank's point of view even from the customers.

To solve the existing problem in banking industry we have proposed to design an interoperable cloud computing framework that serves the banking community and enables to perform inter-banking transaction easily and to minimize the technological costs through interview and literature review on cloud computing as the research methodology. The result obtained from the interview shows that no single bank in Ethiopia can afford enough access channel to serve their rural to urban customers. Additionally, the recent attempt to integrate the banking channel has got a performance challenge due to redundant switching solution.

The proposed framework is based on community cloud deployment model because of its suitability for an organization having similar business agenda and security requirements and has cost advantage over the private cloud and security advantage over the public cloud. The framework also uses SaaS (Software as a Service) and IaaS (Infrastructure as a Service) service delivery model in the way that new organizations signed for a SaaS also request their infrastructure requirements like storage, processor, and RAM (Random Access Memory) requirement. The proposed framework has designed by considering security, interoperability and other external systems. When any banking industry signed up to use EBICC (Ethiopian Banking Industry Community Cloud) the system allocate a virtual server which has a core banking system and other management software. Finally, the prototype of the designed framework has been developed using JEE (Java Enterprise Edition) architecture to show the validity of the framework.

Keyword: Cloud computing, Interoperability, Banking Industry

CHAPTER ONE

1. INTRODUCTION

In this chapter the background of the research and the rationale initiated to do this research has been presented. It presents the overall objective, methodology, significance of the study, scope, and limitation of the thesis and problem that has to be addressed finally the organization of the thesis is presented.

1.1. Background

In now a day processing banking and financial institution data are unthinkable without the involvement of efficient computing model and computer network technology because banks have a large number of customers distributed all over the world and the transaction related to each customer is also huge. Continual improvement and advancement of technology would assist them for harvesting the actual and expected benefits of ICT (Information Communication Technology). Banks are trying to satisfy their customers' to access their money without any restriction by using recent computer technologies like Mobile banking, ATM (Automated Teller Machine) and internet banking. However, in developing countries, providing those services is difficult for a single bank and those technologies also give limited banking services to the user. Hence to alleviate the limitation in ICT usage, banks in developing country would start to use a pooled resource from the cloud because they are performing almost similar activities. Using pooled resources in the cloud also enables to create inter-bank transaction easily by using different banking channels.

Cloud computing is increasingly becoming an integral component of any organization's computing strategy. Organizations including banks now understand that cloud computing offers the possibility of being able to seamlessly change IT (Information Technology) without expending the time and resources in setting up, configuring, and deploying new systems. This technology offers a more efficient use of resources such as storage, memory, processing, applications and bandwidth ensuring high availability, security and quality [3]. In fact, according to the IBM (International Business Machine) 2010 CEO(Chief Of Executive) Study, "Sixty percent of CEOs plan to use Cloud—up from thirty-three percent two years ago and eighty-nine percent of financial services CEOs said their focus over the

next five years would be “getting closer to the customer.” In banking, customer-focused enterprises deliver a superior customer experience by engaging customers in insightful conversations. Banks aren’t just providing a pleasant banking experience. Because of cloud applications, they can now offer a consistent cross-channel experience, much like customers receive from retailers and airlines [4].

By deploying IT infrastructure and services over the network, an organization can purchase these resources on an as-needed basis and avoid the capital costs of software and hardware. With cloud computing, IT capacity can be adjusted quickly and easily to accommodate changes in demand. While remotely hosted, managed services have long been a part of the IT landscape, a heightened interest in cloud computing is being fueled by ubiquitous networks, maturing standards, the rise of hardware and software virtualization, and the push to make IT costs variable and transparent [5]. Cloud computing delivers: Infrastructure as a Service (IaaS), Platform as Service (PaaS), and Software as Services (SaaS), which available as subscription-based services in a pay-as-you-go model and other essential features to customers [1].

Currently, four cloud deployment models are available these are private cloud, community cloud, public cloud and hybrid cloud. By considering cost-effectiveness, security and interoperability in this study community cloud deployment model has been chosen as a deployment model for banking industries in Ethiopia. A community cloud is created when several allied companies or enterprises decide to move to the cloud together. Either the community as a whole decides to create a common infrastructure platform for all to use (common reasons being the ease of sharing information and cost reduction), or one member or sponsor provides the necessary infrastructure that is used by the community. It also offers the options of not going to the cloud at all or considering alternatives to the cloud. This decision (among others) is made when the data or the process is too critical or contains so much sensitive or business-critical data that the risk of going to the cloud outweighs the benefits. Community clouds offer services to support organizations with shared objectives and common security and privacy requirements [2].

1.2. Statement of the Problem

In the 21st century, the business environments are highly dynamic and people needs to access their money in the bank just as in their pocket. To answer this fast access need of a customer it is difficult for a single bank because it needs billions of dollars to afford enough banking access channels and to open a new branch office. As the result of economic growth in the country, the flow of currency is highly increased from time to time. It's known that banking institutions are the mid player and catch the higher position in the economic environment. Ethiopia set a Gross Transformation Plan (GTP) to reach the country to the mid-income country. To succeed this GTP the National bank of Ethiopia (NBE) set a goal on different government and private organizations.

The recent improvement in technological advancement is encouraging because all of the banks in the county deploy the core banking system and linked it with a national switch called Ethswitch. Even though some advancement in using recent technologies there are also some issues which should be addressed. The issues include:

- The inability of the individual bank to provide full banking infrastructure to their customer.
- Delay in the current integrated system: If the card user tries to access the ATM or POS terminal out of their respective bank there is a problem of delay or not working.
- Duplication of effort: It is better to integrate the ATM or POS(Point Of Sell) terminal of the bank directly to the national switch; i, e no intermediate switches are required. Another duplication of effort is an attempt to deploy costly technologies and software systems individually.
- Technological change: The advancement in technology like smartphones, personal computers, and World Wide Web initiate peoples to mobilize their remote information by their own device. Hence these changes enforce financial institutions to change their existing system and deploy up to date system that supports recent technological advancement
- Involvement of foreign bank: Since Ethiopia became the member of world trade organization forces to open her door for foreign banks increased; especially from the IMF (International Monetary Fund) and from the friendship countries. Currently

some of the banks open their office in Addis Ababa. As many economists suggest this situation increases competition among the domestic and the foreign banks. Therefore strengthening the technology power and multiplying their access channel of the domestic bank is a must.

Cloud computing has generated a tremendous amount of interest and excitement in recent years as it gives a new and useful way to address IT challenges across and beyond financial services [1].Cloud computing is an IT-servicing technology built on the idea of exploiting virtualization technology's shared resource pooling, as well as it can solve the problems with interoperability.

Hence, this study will answer the following three research questions.

1. Do we need to make the banking activities in Ethiopia interoperable?
2. Which banking activities are a candidate for a shared service delivery?
3. Could cloud computing be an alternative solution for creating interoperable ICT service provisioning strategy for EBI(Ethiopian Banking Industry)?

1.3. Objective of the Study

1.3.1. General objective

The general objective of the study is to examine existing ICT infrastructure and banking activities which need to be integrated into Ethiopian banking industries and to develop an interoperable cloud computing framework for Ethiopian banking industries .

1.3.2. Specific objective

The specific objectives of the study are:

- Examining the current ICT infrastructure and strategy in Ethiopian banking industries.
- Identifying the banking activities which need to be integrated.
- Review of related literature to trace the intellectual progress of cloud computing in the banking industry.
- Choosing the appropriate deployment and service delivery model by taking interoperability, security and cost effectiveness into account.
- Designing an interoperable cloud computing framework.

- Developing a prototype to show how the framework works.
- Identifying and recommending future research directions for further investigation on Cloud Computing in the banking industry.

1.4. Significance of the study

Cloud Computing has the potential to transform a large part of the information technology industry by fulfilling the dream of computing as a utility. This research enables the Ethiopian banking industries to operate selected banking activities cooperatively to satisfy their customer by increasing their service availability. The customer of the bank can access their money at everywhere and every time, this save the time and cost wasted unnecessarily. This study also cuts infrastructure costs, and to transform bank's business processes and enhance their ability to grow in new sectors or regions without the time and cost burdens involved with establishing a physical presence. It also alleviates the security related problems due to the lack of knowledge of ICT professionals because the cloud system is managed by centrally by higher skilled professionals. The proposed Cloud Computing interoperability framework could also be used as a baseline for Ethiopian banking industries ICT managerial on how the cloud computing would be implemented and used in banking institutions.

1.5. Scope and limitation of the Study

The study mainly covered examining the current ICT service delivery strategies to support Ethiopian banking industries, to show the applicability of cloud computing in the field of banking, designing an interoperable Cloud Computing framework for Ethiopian banking industries and developing a prototype for the framework. This study did not consider other financial institutions and did not consider full implementation of the framework.

1.6. Organization of the Thesis

The rest of the study is organized as follows: Chapter two covered related literatures on the basic concepts of cloud computing: service types, deployment models, cloud computing adoption strategy, cloud challenges and benefits for banking industry; different publications about EBI existing ICT usages; and related works done on cloud-based service frameworks locally and abroad in order to understand how it is implemented or used in banking sectors. Chapter three describes the research methodology and strategy that aims to identify the potential cloud solutions that could be provided to banking industries. It begins by describing exploratory research approaches. Primary data collection and analysis technique were described as information acquisition method. Validity and reliability requirements also identified. Chapter four presents the results of the interview, techniques and technologies required to design the prototype. In Chapter five, backgrounds of a framework for building the proposed Cloud Computing Framework for EBI, the detail elements of the proposed cloud computing framework, what design is better for EBI, why the specific design was selected for EBIs also discussed. In chapter six the implementation of the prototype and tools used to develop the framework have discussed. Finally, in chapter seven conclusions about the research and suggestions for future research direction were presented.

CHAPTER TWO

2. LITERATURE REVIEW

In this chapter we have traced the intellectual progress, major debates on cloud computing technology and the ICT strategy used by Ethiopian banking industry. Additionally, the basic concepts of cloud computing, some recent literature which has been written by applying cloud computing technology to the banking industry have been discussed.

2.1. Issues of Interoperability

Broadly speaking, interoperability can be defined as a measure of the degree to which diverse systems or components can work together successfully. More formally, IEEE(Institute of Electrical and Electronics Engineers) and ISO (International Standard Organization) define interoperability as the ability of two or more systems or applications to exchange information and mutually use the information that has been exchanged. To be more concrete, in the context of cloud computing, interoperability should be viewed as the capability of public clouds, private clouds, and other diverse systems within the enterprise to understand each other's application and service interfaces, configuration, forms of authentication and authorization, data formats etc. in order to cooperate and interoperate with each other [35]. When we are defining Interoperability from the banking point of view it is a situation in which payment instruments belonging to a given scheme used in systems installed by other scheme and require technical compatibility between systems.

Interoperability is mostly associated with selecting or defining kinds of standards in order to achieve a better service provision. The integration of formerly separated processes or databases into a new centralized system is another option; this may be more effective and efficient in terms of operating costs and service quality [36].

There are four layers of interoperability on implementations; these are [36]; Organizational, Semantic, Syntactic and Technical layer Interoperability. Organizational interoperability is related to whether the participating organizations use the shared information in similar contexts of the business functions, services or business process, i.e., the shared information that impacts the business activities related to business services, legal, political, and process interoperability are to be considered. Semantic interoperability is related to the meaning of

the shared information. It is the ability to automatically interpret the information exchanged meaningfully and accurately in order to produce useful results as defined by the end users of both systems. To achieve semantic interoperability, both sides must refer to a common information exchange reference model. The content of the information exchange requests are unambiguously defined: what is sent is the same as what is understood. According to ETSI technical interoperability is usually associated with hardware/software components, systems, and platforms that enable machine-to-machine communication to take place. This kind of interoperability is often centered on (communication) protocols and the infrastructure needed for those protocols to operate. Technical interoperability only guarantees the correct transmission of bits but does not tell anything about the meaning of these bits and what they represent, not even whether it is voice, video, or data. This is the task of standards on the syntactic layer, which define the syntax of particular services. According to ETSI syntactic interoperability is usually associated with data formats. Certainly, the messages transferred by communication protocols need to have a well-defined syntax and encoding, even if it is only in the form of bit tables. However, many protocols carry data or content, and this can be represented using high-level transfer syntaxes such as HTML, XML or ASN.12.

2.2. Virtualization

Virtualization is the representation of real machine using software that provides an operating environment which can run or host a guest operating system. It is an essential technological characteristic of clouds which hides the technological complexity from the user and enables enhanced flexibility (through aggregation, routing, and translation). By creating a virtual version of a device or resource, virtualization can improve computing efficiency, reduce the need for expensive hardware and cut overall IT costs. As new innovations continue to emerge, this technology is promising to change IT infrastructure and streamline management. Virtual machines are created and managed by virtual machine monitor. More concretely, virtualization supports the following features:

 Ease of use: through hiding the complexity of the infrastructure (including management, configuration etc.) virtualization can make it easier for the user to develop new applications, as well as reduces the overhead for controlling the system.

- Infrastructure independence: in principle, virtualization allows for higher interoperability by making the code platform independent.
- Flexibility and Adaptability: by exposing a virtual execution environment, the underlying infrastructure can change more flexible according to different conditions and requirements (assigning more resources, etc.).
- Location independence: services can be accessed independently of the physical location of the user and the resource.

Two basic concepts of virtualization

1. Making multiple physical resources (such as storage devices or servers) appears as a single logical resource.
2. making a single physical resource (such as a server, an operating system, an application, or storage device) appear to function as multiple logical resources

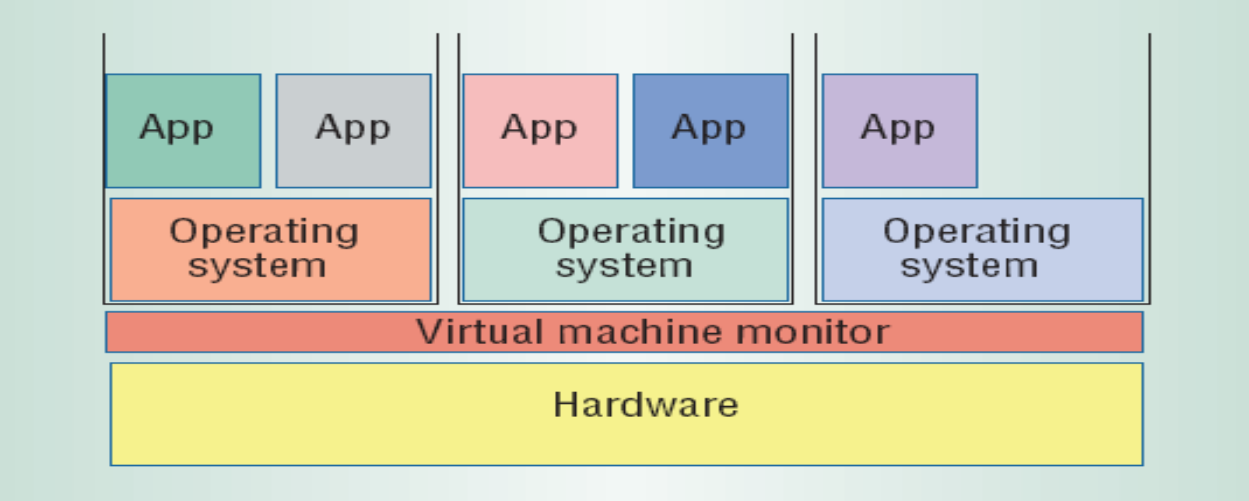

FIGURE 2-1: VIRTUALIZATION ARCHITECTURE

Virtualization has three characteristics that make it ideal for cloud computing [14]:

1. **Partitioning:** In virtualization, many applications and operating systems (OSes) are supported in a single physical system by partitioning (separating) the available resources.
2. **Isolation**: Each virtual machine is isolated from its host physical system and other virtualized machines. Because of this isolation, if one virtual instance crashes, it

doesn't affect the other virtual machines. In addition, data isn't shared between one virtual container and another.

3. **Encapsulation:** A virtual machine can be represented (and even stored) as a single file, so you can identify it easily based on the service it provides. In essence, the encapsulated process could be a business service. This encapsulated virtual machine can be presented to an application as a complete entity. Therefore, encapsulation can protect each application so that it doesn't interfere with another application.

2.3. Cloud Computing

Cloud computing is the recent computing paradigm next to cluster and grid computing technologies and it makes true that the long wish of getting computing resource as a service. Technologies such as cluster, grid, and now, cloud computing have all aimed at allowing access to a large amount of computing power in a fully virtualized manner, by aggregating resources and offering a single system view. In addition, an important aim of these technologies has been delivering computing as a utility. Utility computing describes a business model for on-demand delivery of computing power; consumers pay providers based on usage ("pay-as-you-go"), which is similar to the services obtained from traditional public utility services such as water, electricity, gas, and telephony [1].

Many scholars and organizations working in Cloud computing have attempted to define what "cloud computing "is with respect to its characteristics point of view. According to [9], Cloud computing is a system, where the resources of a data center are shared using virtualization technology, which also provides elastic, on-demand and instant services to its customers and charges customer usage as a utility bill.

(Buyya et al. , 2009) have defined it as follows: "Cloud is a parallel and distributed computing system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers."

Armbrust, et al.,2010 Cloud computing refers to applications and services that run on a distributed network using virtualized resources and accessed by common Internet protocols and networking standards. It is distinguished by the notion that resources are virtual and

limitless and that details of the physical systems on which software runs are abstracted from the user.

The most widely accepted definition of US National Institute of Standards and Technology, [NIST], 2011) defined Cloud computing as a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.

2.3.1. Characteristics of Cloud Computing

According to (NIST, 2011) and (Chaudhry , 2013), cloud computing exhibits its own essential characteristics such as; on demand service, wide network approach, resource pooling, fast elasticity and measured service which demonstrate their relation to, and differences from, traditional computing approaches:

On-demand self-service means a consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider. Cloud computing focuses on IT services which are driven by users' requests. It provides a means of delivering computing services that makes the underlying technology, beyond the user device, almost invisible

Wide network approach permits customers to acquire computing resources IT services anytime, anywhere, and through the user-chosen device (e.g., mobile phones, tablets, laptops, and workstations). Users accessing services via Internet technologies expect a secure, "always-on" computing infrastructure that delivers as easily and reliably as electricity from a wall outlet

When we say resources pooling the provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location-independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth.

A cloud service has characteristics of fast elasticity means that provides the quick and self-regulating raise and reduces to the quantity of present computer transforming, storage and network bandwidth as a request by customer requirement. It is assumed that changes in the resource infrastructure needs are announced first to the middleware manager and then that such changes can be maintained automatically.

Cloud computing resource usage can be measured, controlled, and reported providing transparency for both the provider and consumer of the utilized service. Cloud computing services use a metering capability which enables to control and optimize resource use. This implies that just like airtime, electricity or municipality water IT services are charged per usage metrics – **pay per use**. The more you utilize the higher the bill. Just as utility companies sell power to subscribers, and telephone companies sell voice and data services, IT services such as network security management, data center hosting or even departmental billing can now be easily delivered as a contractual service.

2.3.2. Multi-tenancy

Although not an essential characteristic of Cloud Computing in NIST's model, CSA(Cloud Security alliance) has identified multi-tenancy as an important element of the cloud. A tenant is any application either inside or outside the enterprise that needs its own secure and exclusive virtual computing environment. This environment can encompass all or some select layers of enterprise architecture, from storage to user interface. All interactive applications (or tenants) have to be multi-user in nature [13].

Multi-tenancy is a property of a system where multiple customers, so-called tenants, transparently share the system's resources, such as services, applications, databases, or hardware, with the aim of lowering costs, while still being able to exclusively configure the system to the needs of the tenant [9]. Multi-Tenancy occurs when two or more virtual machines (VMs) belonging to different customers share the same physical machine [11].

The advantage of multi-tenancy are reducing operational costs by dividing hardware and software resources among the different tenants, simplifying the maintenance and management effort. All these benefits of multi-tenancy give result in lower application costs and give the major benefit to small and medium enterprises (SME) [10]. Multi-tenancy service requirements for cloud services providers are policy-driven enforcement,

segmentation, isolation of tenant data, Isolation of the tenant workspace (memory), Isolation of tenant execution, chargeback/billing models for different consumer constituencies ,Tenant-aware security, monitoring, management, reporting and self-service administration, Isolation of tenant customizations and extensions to business logic, tenant-aware version control, Tenant-aware error tracking and recovery.

Multi-tenancy Issues for different types of users can be solved at three different levels in cloud computing. The levels are Infrastructure Level by Virtualization, data center Level by sharing an operating system and Application Level. Using these levels we can develop a model and verify the functional and non-functional behavior of the system [10].

Renting space in a data center and supplying servers, routers, and cables that support multiple customer software requests have been around since Silicon Valley was in its infancy, so datacenter-layer multi-tenancy is well known. This configuration provides the highest level of security requirements if implemented correctly, with firewalls and access controls to meet business requirements as well as defined security access to the physical location of the infrastructure providing the SaaS. In most cases, data center layer multi-tenancy can be a service provider that rents cages to companies that host their hardware, network, and software in the same building [12].

Multi-tenancy at the infrastructure layer is the easiest thought of in software stacks, where one stack is dedicated to a specific customer. This configuration saves costs compared to data center layer multi-tenancy because stacks are deployed based on actual customer accounts. In this case, we can grow hardware requirements based on actual service use. In addition, high-availability may be an option for each customer at the infrastructure layer. Because the stack is known for each consumer, software, and hardware best practices provide options for implementation [12].

Application-layer multi-tenancy requires architectural implementations at both the software layer and the infrastructure layer. Modifications are required for the existing software architecture, including multi-tenant patterns in the application layer. For example, multi-tenant applications require application methods and database tables to access and store data from different user accounts, which compromises on security. If done accurately, however, the benefit is cost savings. For widgets and simple web applications, application-layer multi-

tenancy can be a solution, because a single developer can create software faster and scale more affordable. Disadvantages include more complex application architecture and implementation; rather than the infrastructure dealing with multi-tenancy, the application teams, and architects need to keep these programming patterns scalable, reliable, and future-proof in case the infrastructure architecture changes [12].

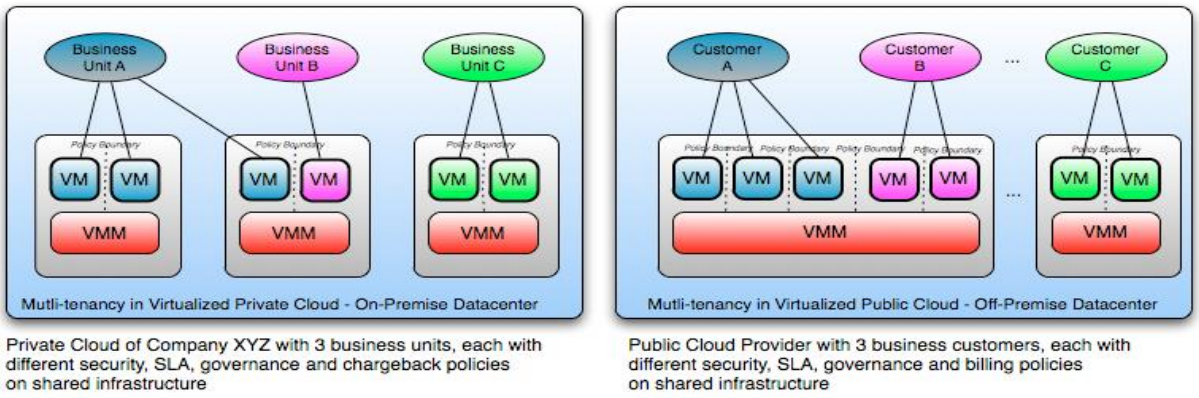

Figure 2-2: Multi-TENANCY [19]

2.3.3. Cloud Computing Service Model

The term services in cloud computing are the concept of being able to use reusable, fine-grained components across a vendor's network. This is widely known as "as a service." Cloud providers are using different service delivery models which can be differentiated based on the level of virtualization they offer, there are mainly three service delivery models; such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). These three models together constitute as SPI Model where S stands for Software, P stands for Platform and I stand for Infrastructure. (Sosinsky, 2011), briefly explain the cloud computing service delivery models in his book, "The cloud computing bible". These models are Infrastructure as - a - service (IaaS), Platform as- a- Service (PaaS) and Software as- a- Service (SaaS).

Infrastructure as a Service (IaaS) is a cloud computing service model in which hardware is virtualized in the cloud. In this particular model, the service vendor owns the equipment: servers, storage, network infrastructure, and so forth. The developer creates virtual hardware on which to develop applications and services. Essentially, an IaaS vendor has created a hardware utility service where the user requested virtual resources as required. When the

client interacts with an IaaS service and requests resources from the virtual systems, those requests are redirected to the real servers that do the actual work.

The Platform as a Service model describes a software environment in which a developer can create customized solutions within the context of the development tools that the platform provides. Platforms can be based on specific types of development languages, application frameworks, or other constructs. A PaaS offering provides the tools and development environment to deploy applications on another vendor's application. Often a PaaS tool is a fully integrated development environment; that is, all the tools and services are part of the PaaS service. To be useful as a cloud computing offering, PaaS systems must offer a way to create user interfaces and thus support standards such as HTML, JavaScript, or other rich media technologies. In a PaaS model, customers may interact with the software to enter and retrieve data, perform actions, get results, and to the degree that the vendor allows it, customize the platform involved. The customer takes no responsibility for maintaining the hardware, the software, or the development of the applications and is responsible only for his interaction with the platform. The vendor is responsible for all the operational aspects of the service, for maintenance, and for managing the product(s) lifecycle.

The difficulty with PaaS is that it locks the developer and the customer into a solution that is dependent upon the platform vendor. An application written in Python against Google's API using the Google App Engine is likely to work only in that environment. There is considerable vendor lock-in associated with a PaaS solution.

A complete cloud computing service model is one in which the computing hardware and software, as well as the solution itself, are provided by a vendor as a complete service offering. It is referred to as the Software as a Service (SaaS) model. SaaS provides the complete infrastructure, software, and solution stack as the service offering. A good way to think about SaaS is that it is the cloud-based equivalent of shrink-wrapped software. Software as a Service

(SaaS) may be succinctly described as software that is deployed on a hosted service and can be accessed globally over the Internet, most often in a browser. With the exception of the user interaction with the software, all other aspects of the service are abstracted away. The table below shows that the summary of service delivery models

TABLE 2-1: SERVICE DELIVERY MODELS [10].

Software as a service	Application Runtime Hardware	**Application and Information Centric** Allows the consumer to use providers applications running of cloud infrastructure Provides various business models and application processes Ex.Amazon Web Services,Salesforce.Com CRM,Google APP Gmail
Platform as a Service	Application Runtime Hardware	**Development Centric** Provides a cloud platform to deploy consumer application created using software development kits and tools supported by the provider Provides Web based runtime development and operating environments Ex.Amazon EC2,Google Apps Engine
Infrastructure as a service	Application Runtime Hardware	**Infrastructure centric** Provides virtual hardware resources such as servers, storage, network, and other fundamental computing resources Ex.Amazon S3, IBM, HP

2.3.4. Cloud Computing Deployment Model

(NIST, 2011) explained four cloud deployment models which are commonly adopted by the cloud providers and government agencies. These are Public cloud services, Private cloud service, Community cloud service and Hybrid Cloud service.

In the public cloud infrastructure is provisioned for open use by the general public. It may be owned, managed, and operated by a business, academic, or government organization, or some combination of them. It exists on the premises of the cloud provider. Public cloud users are typically residential users and connect to the public internet through an internet service provider's network. Google, Amazon, and Microsoft are examples of public cloud vendors who offer their services to the general public. Facebook and Gmail are also examples of services where data is stored in a public cloud. [15] Lists some of the advantages and drawbacks of the public cloud.

The advantages of public cloud computing include:

- Data availability and continuous uptime
- 24/7 technical expertise
- On-demand scalability
- Easy and inexpensive setup
- No wasted resources

Drawbacks of the public cloud:

- Data security
- Privacy

Private cloud service is a cloud deployment model in which the cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers. It may be owned, managed, and operated by the organization, a third party, or some combination of them, and it may exist on or off premises. A private cloud is hosted in the data center of a company and provides its services only to users inside that company or its partners.

A private cloud provides more security than public clouds because in public cloud the user may not know where their data is stored or how it is backed up and whether unauthorized users can get access to it.

The major drawback of private cloud is its higher cost. When comparisons are made with a public cloud; the cost of purchasing equipment, software and staffing often results in higher costs to an organization having their own private cloud.

A community cloud falls between public and private clouds with respect to the target set of consumers. It is somewhat similar to a private cloud, but the infrastructure and computational resources are exclusive to two or more organizations [15]. The cloud infrastructure is provisioned for exclusive use by a specific community of consumers from organizations that have shared concerns (e.g., mission, security requirements, policy, and compliance considerations). It may be owned, managed, and operated by one or more of the organizations in the community, a third party, or some combination of them, and it may exist on or off premises.

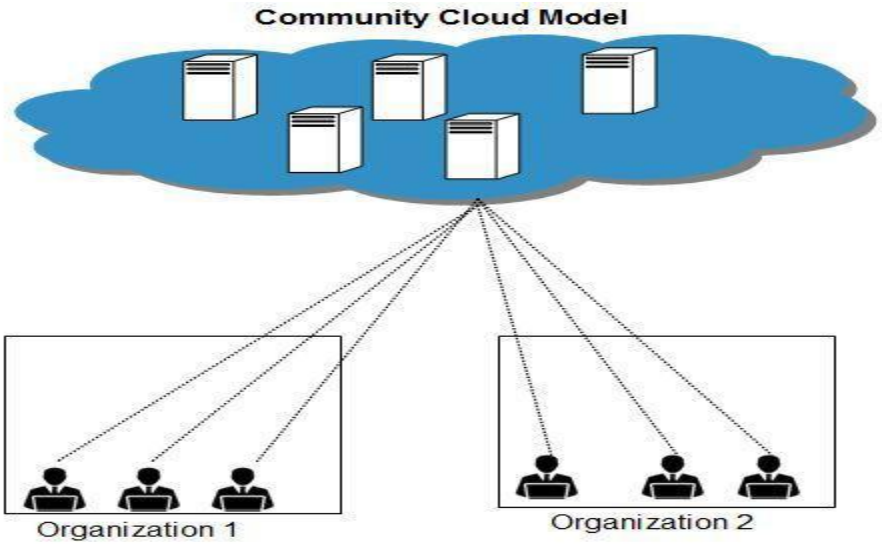

FIGURE 2-3: COMMUNITY CLOUD MODEL[38]

The advantages [15] of community cloud include:

- The cost of setting up a communal cloud versus individual private cloud can be cheaper due to the division of costs among all participants.
- Management of the community cloud can be outsourced to a cloud provider. The advantage here is that the provider would be an impartial third party that is bound by contract and that has no preference to any of the clients involved other than what is contractually mandated.
- Tools residing in the community cloud can be used to leverage the information stored to serve consumers and the supply chain, such as return tracking and just-in-time production and distribution.
- Since it is managed by contract with an impartial third party or by one of or the combined expert of the whole community, it has the advantage of security and privacy over the public cloud.

Drawbacks of community cloud:

- Costs higher than public cloud.

Another cloud deployment model hybrid Cloud the infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g., cloud bursting for load-balancing between clouds). A hybrid cloud is typically offered in one of two ways: a vendor has a private cloud and forms a partnership with a public cloud provider, or a public cloud provider forms a partnership with a vendor that provides private cloud platforms. In a hybrid cloud, an organization provides and manages some resources in-house and some out-house. For example, organizations that have their human resource (HR) and customer relationship management (CRM) data in a public cloud like Saleforces.com but have confidential data in their own private cloud [15]. Hybrid clouds [15] offer the cost and scale benefits of public clouds, while also offering the security and control of private clouds. The advantages of the hybrid cloud include:

- Reduces capital expenses as part of the organization's infrastructure, needs are outsourced to public cloud providers.
- Improves resource allocation for temporary projects at a vastly reduced cost because the use of public cloud removes the need for investments to carry out these projects.
- Helps optimize the infrastructure spending during different stages of the application lifecycle. Public clouds can be tapped for development and testing while private clouds can be used for production. More importantly, public clouds can be used to retire applications, which may be no longer needed because of the move to SaaS, at much lower costs than dedicated on-premise infrastructure.
- Offers both the controls available in a private cloud deployment along with the ability to rapidly scale using the public cloud.
- Supplies support for cloud-bursting.
- Provides drastic improvements in the overall organizational agility, because of the ability to leverage public clouds, leading to increased opportunities.

Drawbacks of the hybrid cloud are:

- As a hybrid cloud extends the IT perimeter outside the organizational boundaries, it opens up a larger surface area for attacks with a section of the hybrid cloud infrastructure under the control of the service provider.
- An easier approach to solving the identity, needs of hybrid clouds is to extend the existing enterprise identity and access management to the public clouds. This opens up concerns about how this approach will affect the enterprise identity and its impact on the organization's security.
- When organizations manage complex hybrid cloud environments using a management tool, either as a part of the cloud platform or as a third-party tool, organizations should consider the security implications of using such a tool. For example, the management tool should be able to handle the identity and enforce security uniformly across hybrid cloud environments.
- A hybrid cloud makes the data flow from a private environment to a public cloud much easier. There are privacy and integrity concerns associated with such data movement because the privacy controls in the public cloud environment vary significantly from the private cloud.
- There are risks associated with the security policies spanning the hybrid cloud environment such as issues with how encryption keys are managed in a public cloud compared to a pure private cloud environment.

2.3.5. Benefits of Cloud Computing for Banks

Cloud computing is that one business solution that every enterprise uses. Many businesses moving to the cloud because cloud computing increases efficiency, help to improve cash flow and offer many more benefits. The benefits of cloud computing for banking industries are listed in the following paragraph.

Fully automated Service provision, monitoring, and management accelerate time-to-market by reducing service provisioning time while reducing the risk of acquiring and implementing new financial services functionality. Once opting for this method of functioning, service providers to focus on managing hardware, software, services as well as end usage outcomes, rather than for businesses to set up extensive and expensive operations

organizations. The entire system can be fully functional in a matter of a few minutes. Of course, the amount of time taken here will depend on the exact kind of technology that is needed for the business.

Highly optimized and virtualized infrastructure enabling scale and cost efficiency; enabling businesses to leverage the right hardware components and manage computing needs as a function of real-time usage, rather than design and run with maximum capacity, redundancy and resiliency upfront. Capital expenditures are reduced because a lot of the load and storage will be shifted over to the service provider who can provide that service at a lower cost. No need to buy a high-performance computer and expensive software applications. Cloud computing makes banks free from maintenance cost; they pay much money every year for maintaining those traditional applications.

Beyond cost savings, the cloud would provide banks with opportunities to build customer centric and active business models to grow their brands more efficiently. Shared Services delivered across trusted domains delivering Security of data, transaction & operations. Cloud computing enable to provide integrated treatment of security across all the operating platforms and for all the services rendered to the end users.

Using cloud computing in banking industries enables to protect the environment by using green technologies. Financial service sectors can use cloud computing to transfer financial services in a virtual environment that will reduce the energy consumption and carbon footprint emanates from setting up of a physical infrastructure.

2.3.6. Challenges for Adopting Cloud Computing

Now a day choosing cloud computing as a computing paradigm is unquestionable because of the benefits it brings to the ICT sector. Despite its benefits, there are challenges which are related to Lack of resources/expertise, coherent regulatory framework guaranteeing transparency, data protection and respect for data integrity, business, technical and organizational challenges.

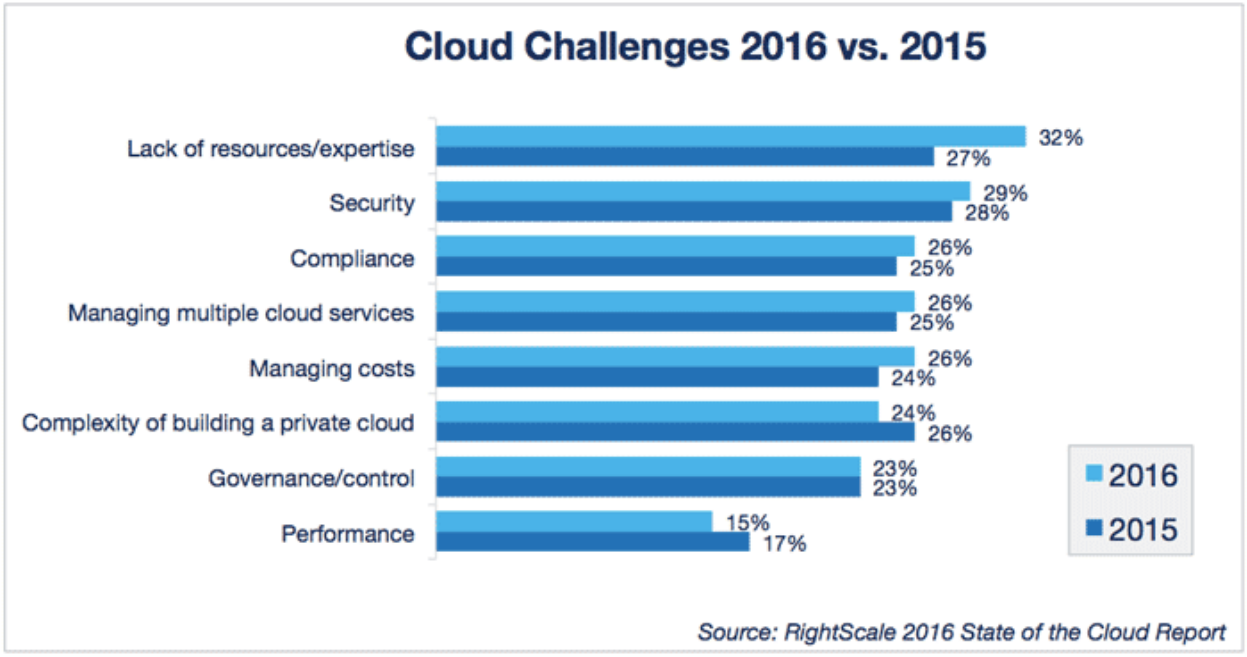

FIGURE 2-4: CLOUD ADOPTION CHALLENGES [26]

Lack of resources/Expertise: The RightScale report on February 2016 shows that since the 2013 State of the Cloud Report, security has been cited as the top challenge in the cloud. Lack of resources/expertise increased from 27 percent last year to 32 percent to supplant security as the largest concern. As more organizations are placing more workloads in the cloud, the need for expertise has grown. Additional training of IT and development staff will be critical to helping address this challenge.

Security: Security is the most challenging barrier to the adoption cloud computing in banking and other financial institutions. Many of the security issues involved in protecting clouds from outside threats are similar to those already facing large data centers. Cloud computing security is an evolving sub-domain of computer security, network security, and, more broadly, information security. It refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing [18]. Some of the common threats to cloud computing security are: data breaches, data loss, Account/service traffic hijacking, insecure interface and APIs, denial of service, malicious insiders, cloud abuse, insufficient due diligence and shared technology vulnerability.

It is obvious that data is the most precious corporate asset, and companies want to know that their data is safe. Companies feel confident when they store data internally because they have full control over it. There is a general agreement among security professionals and expertise that the overall objective of information security is to preserve the availability, integrity, and confidentiality of an organization's information [17]. To protect information and privacy of their customers' cloud providers adopt different security mechanisms like encryption, multi-tenancy, and VLAN. The data at rest is encrypted and stored on the storage environment to ensure that even cloud administrators can't access or modify data. The data in motion is encrypted through the Internet Protocol Security (IPSec) and generic routing encryption (GRE) tunnel, as well as by using a virtual LAN with a software switch available in the Meghdoot stack. Hypervisor's zoning ensures segregation of access rights and privileges between hardware and the virtual environment [20].

Whenever companies decided to move their data to the cloud they should have a clear understanding about who and how to secure their data at a different level, i.e. Security at the network level, security at the host level and security at the application level. Security at the network level means ensuring data confidentiality and integrity of the organization's data in transit to and from the public cloud provider. Application level security is usually the responsibility of both the cloud service provider and the customer. Host level security is protecting attacks to the guest operating system, stealing keys used to access and manage the hosts.

Regulatory and Compliance Restrictions: Besides technological and security issues, there are also regulatory, compliance and legal issues to consider when moving to the cloud. Depending on business requirements, a company's computing infrastructure could be subject to certain compliance regulations. Companies should have a clear list of compliance requirements before considering cloud service providers. If the tenant or cloud customer operates in the United States, Canada or the European Union, they're subject to numerous regulatory requirements. These include Control Objectives for Information and related Technology and Safe Harbor. These laws might relate to where the data is stored or transferred, as well as how well this data is protected from a confidentiality aspect

Performance Bottlenecks: Cloud provisioning is the process of deploying and managing IT resources on cloud infrastructures. Rapid provisioning is a key performance requirement for cloud services, especially when there are a large number of customers requesting resources at the same time. However, it is difficult to determine what factor, or a combination of factors, are the causes of poor provisioning performance because there are no existing tools and methods to trace each status change and execution step in provisioning [21]. Multiple virtual machines (VMs) can share CPUs and main memory surprisingly well in cloud computing, but that network (HTTP protocols and bandwidth issues) and disk input output sharing is more problematic.

Lock-In: Vendor lock-in is a situation in which a customer using a product or service cannot easily transition to a competitor's product or service. Vendor lock-in is usually the result of proprietary technologies that are incompatible with those of competitors. However, it can also be caused by inefficient processes or contract constraints, among other things. Moreover, these types of services dent high switching costs between competing vendors, making customers reluctant or even incapable of transitioning to different vendors. It's the classic case of a tradeoff between good technology and good business. Unfortunately for customers, bad technology and closed standards sometimes make for a more profitable product. The fear of vendor lock-in is often cited as a major impediment to cloud service adoption.

It is not recommended a company become locked-in with a specific cloud provider, because this dependence may limit control over costs and technical cloud hosting options. It is not uncommon for cloud providers to raise service prices or alter their technology offerings [22].

According to [4], one solution to vendor lock-in is to standardize the APIs in such a way that a SaaS developer could deploy services and data across multiple cloud computing providers so that the failure of a single company would not take all copies of customer data with it. Storage expert Arun Taneja offers the following tips for avoiding cloud vendor lock-in:

- Read the fine print of each provider's policies, and if necessary, ask them directly how they facilitate moving customer data out of their cloud storage repository.

- Ask the provider whether they offer data migration tools or services to facilitate the movement of large amounts of data.
- Choose providers that have pledged to support emerging industry standards, such as the cloud data management interface standard created by the storage networking industry association.

SLA Agreements: SLA (Service Level Agreement) is a contract that describes the level of services offered by a cloud provider. In the case of cloud services, SLA could be measured in terms of mean time between failures, mean time to repair the outage and other operational metrics such as network response time and system performance. Companies should perform due diligence to carefully examine a cloud provider's SLA agreements.

2.3.7. Cloud Computing Adoption Strategy for Banks

Cloud computing has created a fundamental shift in how information technology infrastructure is run and managed, changing both the business and technology sides of IT. Transferring enterprise IT to the cloud is a complex task that includes both technical and organizational challenges.

According to (Gartener, 2014) banks are behind other industries when it comes to cloud adoption. One the primary reason adoption has been slow is that banks have received mixed messages related to cloud computing [25]. When banks need to move to the cloud there must have clear and well-designed strategies and phases how to move their data and application otherwise it may lead to unnecessary cost. The path to clouds starts with the architecture that means understanding mission drivers, information under management, existing services under management and core business processes in this case Service Oriented Architecture (SOA) style is advisable in moving to the cloud.

According to [28], the prominent cloud adoptions strategies that are agreed in many experts and organizations have included the following phases as shown in figure 5.

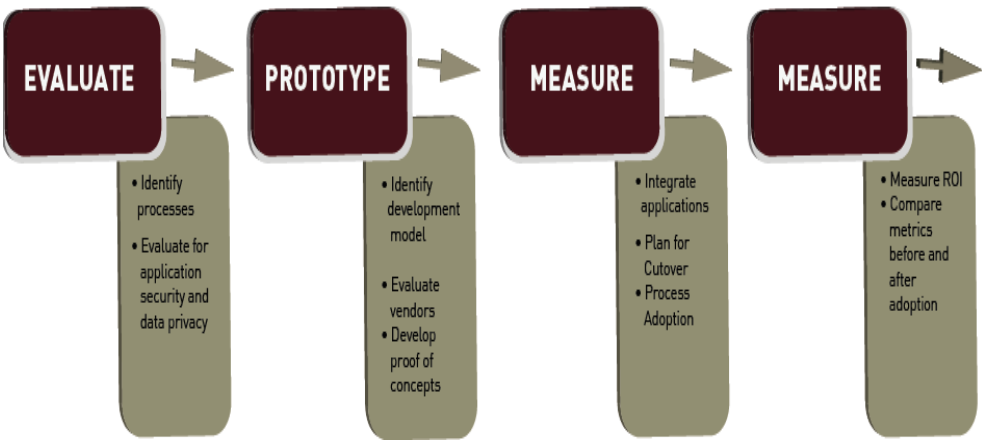

FIGURE 2-5: CLOUD ADOPTION STRATEGIES [28]

Phase I: Evaluate

The development of a successful cloud strategy starts the thorough evaluation of current business processes and applications and identifying those that can be moved to a cloud. The phase consists of developing the knowledge base by participating in seminars, conferences, discussions with the suppliers and consulting the most recent researches in the field. The success of the phase depends on the allocation of sufficient resources for research, for understanding how Cloud Computing functions in different banking industries in the world as a benchmark, the benefits, and risks, policies and the best usage practices of Cloud Computing. The research is conducted by a team formed mainly of IT staff who permanently communicates with the users of the solution regarding the objectives, the progress, costs and benefits of the Cloud Computing solution. The processes and applications should be evaluated from the bank's point of view of the IT needs, structure, and usage. The service oriented architecture represents the base for understanding the data, services, processes and applications that may be migrated or need to be maintained within the banks, so as to observe the security policy. With respect to the IT needs, their structure and usage, the analysis may start from the categories of users who interact with the present IT infrastructure and their necessities. Additionally, privacy requirements when the information is stored on a cloud, peak load hours, architecture constraints and such legal requirements as the physical location of hardware, which will decide applicable legal jurisdiction and laws of country, and other important points, should be identified in this phase.

Phase II: Prototype

In the Prototype phase, particular processes are selected and the type of deployment model, including public, private or a hybrid cloud, is decided. Plus, strategies for storing data with different security requirements and complexities are developed. Some of the key decisions focus on where the most sensitive data should be located and how less sensitive data will be processed. This phase also includes the evaluation of cloud vendors based on data and architectural parameters. Proper assessment of cloud vendors with respect to their focus on security, data confidentiality and availability are completed. Choosing the right vendor involves understanding what each one can offer and how their offerings align with the firm's requirements.

Phase III: Implement

Once the deployment model is decided and cloud vendors are chosen, the implementation phase begins. In this phase, applications are deployed on the cloud. Several factors are kept in mind during this phase, including migration and cutover planning, as well as the adoption and operational management of the new process.

Phase IV: Measure

Finally, an organization should spend considerable time measuring the return on investment achieved and fine-tuning the adoption process. In this phase, return on investment is measured considering the objectives and feedback is collected from end users. The output of this phase is fed back into the evaluation process to fine-tune the next roll-out.

2.4. Related Works

As a recent computing paradigm shift, many scholars and organizations have been conducted a research to adopt cloud computing to the banking industry and other financial institutions. Depending on their need they propose to use some cloud providers' cloud and they follow different cloud computing deployment and service delivery model. Among those researchs three of them which have closely related to this study have been discussed below.

As [37] proposed a new architectural model that supports interoperability payment card system in relation to the country's current organizational and environmental uniqueness. The paper basically focused on the organizational and semantic interoperability layers from the implementation point of view.

This work addressed the problem of resource sharing and interoperability of terminals on ATMs as well as POS terminals for which the cardholder of one bank can be able to use the terminals of others. Since the authors proposed common EFTSwitch, it can solve the issue of interoperability and resource sharing using the switching and routing of transactions initiated from one bank terminals to the base systems and/or terminals. The author designs to architectural models; one for high level layered architecture and one for high-level logical network architecture as shown below.

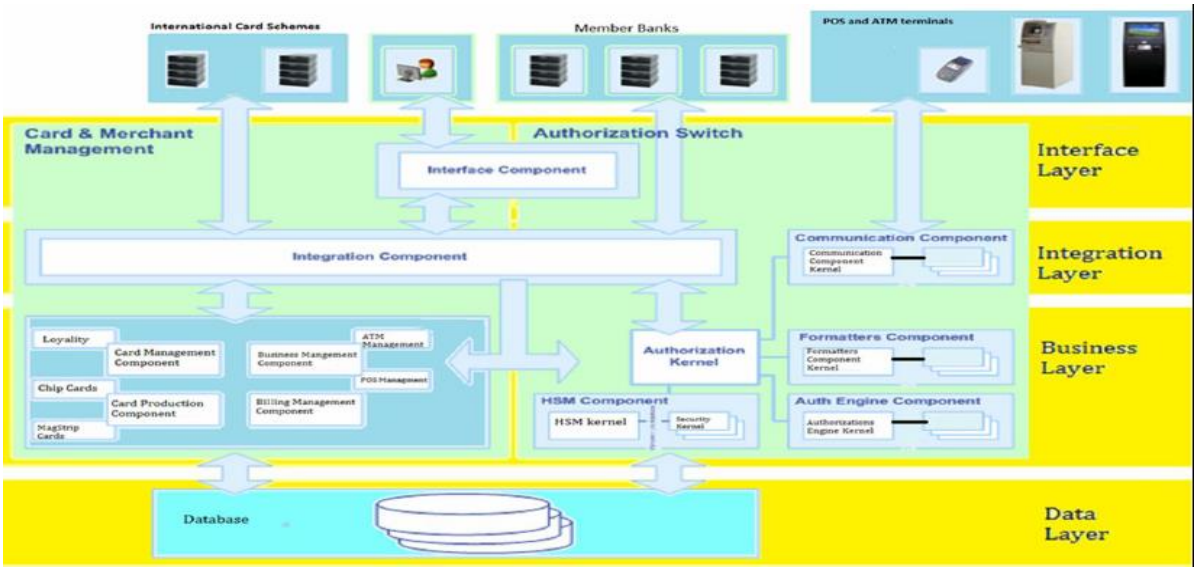

FIGURE 2-6: HIGH-LEVEL LAYERED ARCHITECTURE[37]

This high level layered architecture contains four layers; interface layer, integration layer, business layer and data layer. In interface layer the authorization and authentication component of EFTSwitch and the card and business management interface component Interact with international card schemes for not-on-us transactions. The integration layer facilitates the communication between the interface and business logic layer network lies in the integration layer. The business Layer comprises the business logic of the different interactions between member banks for billing, ATM and POS management, transactions and reconciliations, card management, and loyalty components. The bottom layer, Data layer is the last layer for which all the transactions stored and retrieved for settlement, reconciliation as well as dispute management functionalities

This framework basically solves card and POS related interoperability whereas to make it more economical it is better to make other banking channels and systems interoperable. Additionally, to minimize IT management cost and to make banking system accessible by any customer device broader system interoperability platform is required.

The initiative by the Institute for Development and Research in Banking Technology in Hyderabad, India provides cloud-based services exclusively for Indian banks. The Indian Banking Community Cloud (IBCC) offer services to support banking organizations with shared objectives and common security and privacy requirements. IBCC initiative aims to help banks reduce costs and save resources while providing efficient and secure services. To provision new IT services rapidly and get them to market faster.

BFSI organizations might prefer a private cloud setup, which would protect their existing infrastructure investments while also offering benefits from a cloud-like deployment. Although a private cloud setup is secure in nature, it might be infeasible for some BFSI organizations, especially small and medium-sized ones. Also, the public cloud has many advantages, including lower service costs due to multi-tenancy and resource sharing, tiered service consumption, and lower up-front investments. However, the public cloud's security issue remains unsolved. Consequently, the IDRBT looked to develop a community cloud in which the cloud services targeted organizations with common objectives and security controls. The IBCC provides IaaS, PaaS, and SaaS for banks in India.

The architectural layers of the IBCC are built on the Meghdoot stack which was adopted from the eucalyptus community edition. To ensure both confidence in cloud services and a steady approach for cloud deployment, the IBCC team recommends moving less critical and internal facing applications to the cloud. The following diagram shows the general architecture of IBCC. However, the IBCC need to develop SaaS for common banking transaction processing

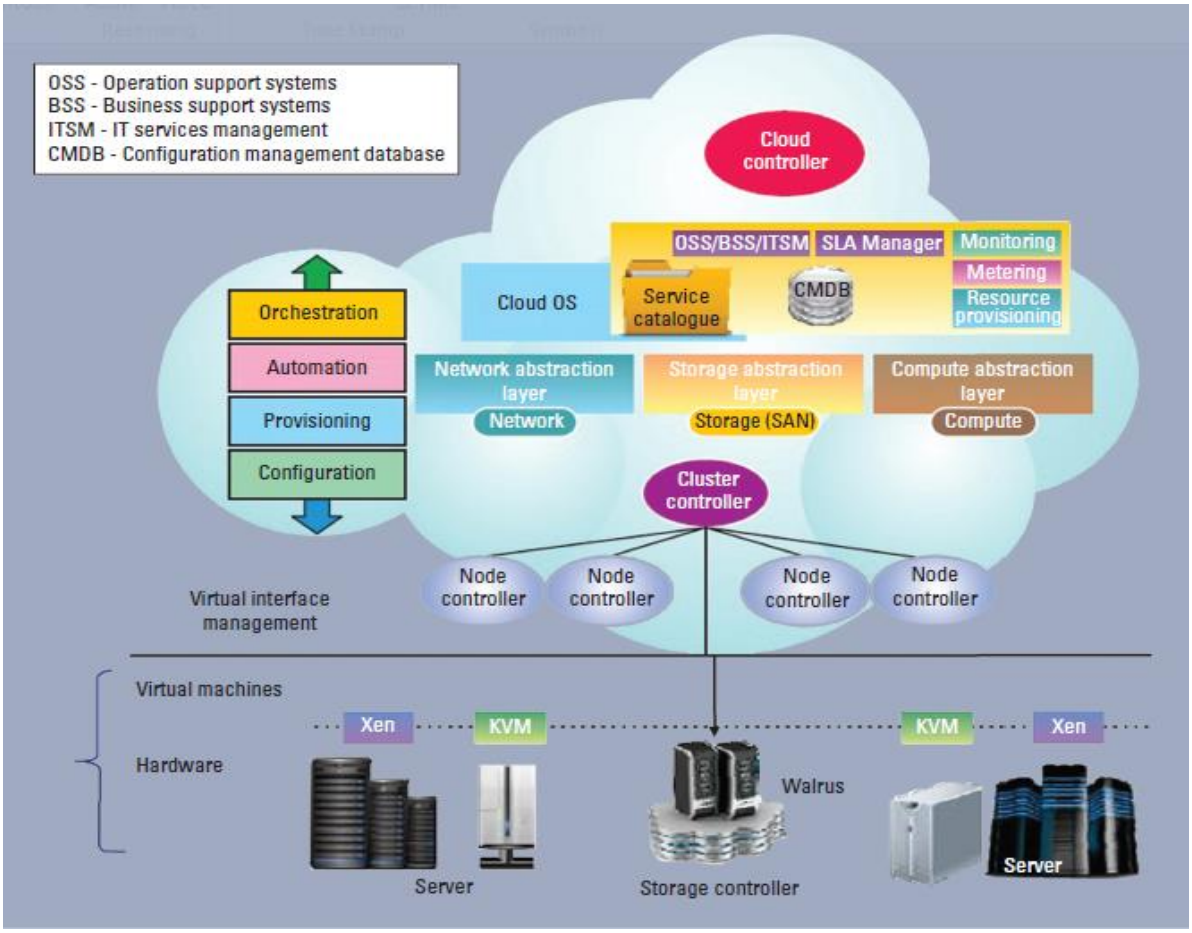

FIGURE 2-7: INDIAN BANKING COMMUNITY CLOUD ARCHITECTURE [20]

Alemayehu (2014) proposes a cloud computing framework for the banking industry in Ethiopia as master's thesis in Addis Ababa University. He analyzed the current ICT structure in Ethiopian banking industry and cloud computing concepts to develop the framework. The result by Alemayehu indicated that Ethiopian banking industries are working in a very highly restricted regulatory policy. Processing financial data abroad the country is a difficult issue because of these regulatory risks and security concerns brings from using external public cloud service providers. Banks also would not be willing to

locate their data on the external public cloud. The fact that customer records and information must be secure and confidential, therefore, banks need to protect their customers' records against any anticipated threats or hazards as well as unauthorized access that could cause substantial harm or inconvenience to the affected customer. Thus the need to find an alternative local cloud service provider that could provide cloud-based solutions for EBIs than external cloud service providers were mandatory.

As a solution to the existing banking problems, the author proposed a hybrid cloud framework which contains five cloud service layers such as; interface, channel service, Business service, platform service and Infrastructure service.

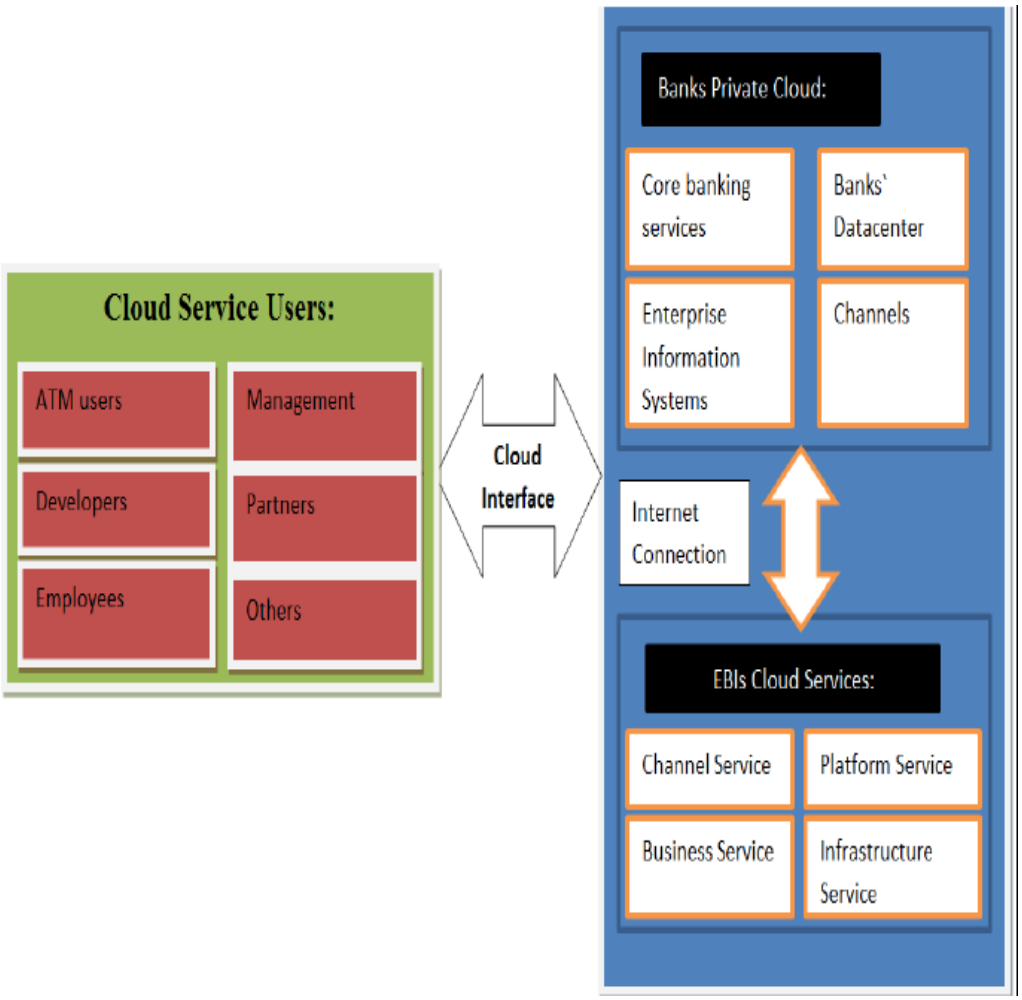

FIGURE 2-8: A CLOUD COMPUTING FRAMEWORK FOR ETHIOPIAN BANKING INDUSTRY [27]

All the reviewed literature agreed on that cloud computing is a major focus of business organizations around the world and it is an IT-servicing technology built on the idea of exploiting virtualization technology's shared resource pooling, as well as its scalability and elasticity. Whereas the deployment model they follow had been different.

In conclusion, Abebe and Fantahun (2012) proposes a technique for making ATM and POS interoperability. In fact to achieve better cost and accessibility advantage, making all banking channel interoperable preferable. In this proposed framework we have integrated all banking access channels. IBCC uses the community cloud for some of the banking functionalities exclusively for Indian banking. Our proposed framework includes all the banking systems and exclusively serving Ethiopian banking community. Alemayehu (2014) propose hybrid cloud deployment model for banking industries, but this framework is more generic and didn't consider the bank which refused to use the proposed system. It mostly the conceptual discussion didn't show the framework from the implementation point of view in detail. We have agreed with (IBCC) uses a community cloud as a deployment model for banking industries because the argument raised by (IBCC) is sound for banking industries. Hybrid cloud computing model is using private and public cloud, in this case it also exposed to security and privacy issues than community cloud. Our proposed framework differ from Alemayehu's work by cloud deployment model used, service delivery model used, the inclusion of the bank who refuse to use our proposed framework, the prototype developed to show the workability of the framework and the inclusion of the switching solution operating within a country, the inclusion of the SaaS and IaaS management tools and the detail explanation and listing of banking activities in each layer of the framework.

CHAPTER THREE

3. RESEARCH METHODOLOGY

In order to achieve the specific and general objectives of the study and to answer the research questions, different data gathering and analysis strategies have been used.

3.1. Research design

The intent of this research is developing a solution to the problem followed by detail analysis of the existing system environment. Hence the research type is a constructive research. Constructive research method implies building of an artifact (practical, theoretical or both) that solves a domain specific problem in order to create knowledge about how the problem can be solved (or understood, explained or modeled) in principle. To give a solution to the problem, the existing system were analyzed by using interview question which helps to study the system.

A qualitative approach was used in this study to conduct the subjective assessment of attitudes, behaviors, and opinions of target data sources (respondents). This approach was selected because of this study aims at to investigate existing IT infrastructure, understand what is happening, and find an alternative solution from an operational point of view.

3.2. Process model

In this research we follow the method of Problem identification, Define the objectives for a solution, Design and prototype development and Evaluation.

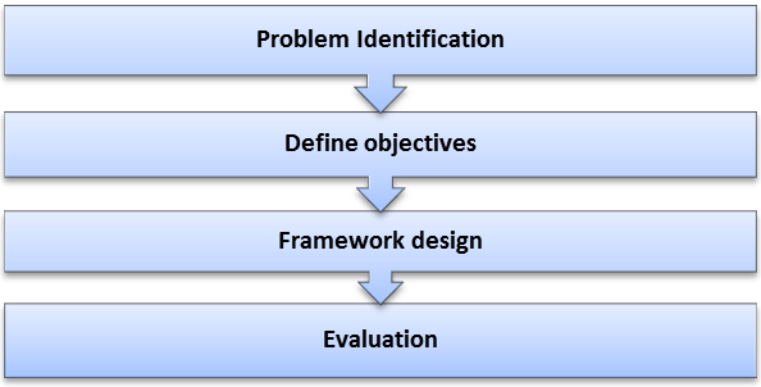

FIGURE 3-1: THE RESEARCH PROCESS

3.2.1. Problem identification and motivation

This phase includes defining the knowledge of the state of the problem and the importance of its solution. We have begun the research by defining the specific research problem and justify the value of a solution by conducting an interview and rigorous analysis of the literature on cloud computing adoption in the banking sector. Since based on the problem definition the researchers atomize the problem conceptually to capture its complexity for developing the proposed cloud computing framework that can effectively provide a solution. The value of the solution also justified motivating the researcher and the audience of the research to pursue the solution and to accept the results and to understand the reasoning associated with the researcher's understanding of the problem.

3.2.2. Define the objectives for a solution

This phase aims at to identify or drive the requirements for developing the proposed cloud computing framework from knowledge of the state of problems and current solutions. The researchers infer the objectives of a solution from the problem definition and knowledge of what is possible and feasible to align with the bank's requirements. The qualitative objectives are used, which described how a proposed framework is expected to support solutions to problems. The researcher used the conceptual framework to inferred objectives rationally from the problem specification. In order to develop the study conceptual framework, we have started with the literature review to gather relevant requirements and aspects of existing Cloud frameworks.

3.2.3. Framework Design

Conceptually, a design research artifact can be any designed object (models, frameworks or instantiations) in which a research contribution is embedded in the design [27]. This phase includes determining the framework's desired functionality and its architecture and then creating the actual artifact. The knowledge of theory that can be brought to bear in a solution is required to move from the objective of a solution to design and development of the proposed framework. Based on the required knowledge of a solution identified in phase one and two with the literature review and expert interviews analysis to develop the proposed framework: particular processes are selected; the type of deployment model is decided; strategies for storing data; the cloud vendor. Finally, the proposed framework has been designed.

3.2.4. Evaluation

This phase aims to Observe and measure how well the framework supports a solution to the problem. Depending on the nature of the problem venue and the solution proposed, this study took the qualitative evaluation analysis techniques which include.

3.3. Data Source

Both primary and secondary data source were used during data collection. Qualitative data were collected from primary sources; such as ICT administrator in selected banks, documentations like NBE directives and different machines observation like ATM, POS, mobile banking.

We have also used Secondary data source like previous researchers work, annual reports of different banks and publications on cloud computing paradigm and banking, books, journal articles, and the like for especially to deal with the solution domain.

3.4. Data Collection Techniques

In the previous section, it's mentioned that different data sources which have been used to achieve the objective of the research. In this section, we have explained that the way how and where to gather this information and why we choose those techniques.

Semi-structure interview method was selected for this study and ICT experts from the commercial bank of Ethiopia, United bank, Birhan international bank, Debub global bank and national bank of Ethiopia were interviewed. Semi-structured interviews give the researcher flexibility to establish the own style of conversation depending on the direction of the interview. To gather information about the current system in banking industries interview questions were organized in the way that enables to get information related to security, individual banks capability to afford enough channel to the customer, customer satisfaction, the reaction of the ICT expert about the existing system, availability of integrated system and reaction about the proposed solution. Interviews were conducted between April 15 and May 25, 2016, and responses were obtained from all selected informants. The objectives and concepts of the study were briefly explained for interviewees. Each informant was interviewed individually by the researcher.

We have also referred documents like directives of NBE to gather the overall regulatory and process structure of the existing banking environment, annual reports and their websites of different banks' to identify their position of ICT usage. Different service access channels like ATM, POS and mobile banking were observed to assess the capability of each channel to give various services.

Secondary sources have a major value in this thesis especially in dealing with the solution stack. Research which has been done previously in the banking sector and cloud computing especially available in websites of (ASTU) Adama Science and Technology University and AAU(Addis Ababa University) were used.

3.5. Sampling Techniques

To study the current situation in banking industry we have used a mixed sampling technique such as purposive sampling and random sampling. Five of the banks among nineteen banks have selected purposively based on their year of establishment, private or public, their infrastructure capabilities. The random selection method is applied to choose banks among purposively selected one, for example, Debub Global bank was selected randomly among recently opened banks. The purposes were to identify the willingness of the banks for integrated ICT service delivery strategy. The best suitable respondents, ICT experts were purposively chosen from selected banks in order to understand banking activities or phenomenon because they understood the technological environment better than the other.

TABLE 3-1: SAMPLE SELECTION

Purpose of selection	No. of selected banks	Name of the bank	No. of respondent
Recently opened	2	DGB	1
		BIB	2
Government bank	1	CBE	2
More infrastructure	1	CBE	2
Privet Bank	1	UB	2

3.6. Analysis Technique

This research aimed to answer three research questions the first two questions enable to understand the problem domain and the last one for the solution stack. To answer these questions using different data gathering tools like an interview, observation and literature review. These data were organized, analyzed and summarized for a better understanding of the current ICT capability, structure, and usage.

As mentioned in the previous section purposive sampling technique has used to understand the willingness of different banks for integrated service delivery model and to check individual banks' capability for providing sufficient ICT service to their customers. The result obtained from the interview have been compared and contrasted with the relevant literature review findings. The analysis made based on the finding of the literature and the data collected from the interview for presenting the requirements to constructing the framework intended in this study.

Literatur on the concepts of cloud computing including and related studies was reviewed. Accordingly, interview and review of the relevant literature are employed to develop the knowledge bases required for developing the proposed cloud computing framework.

3.7. The Development of the Prototype

To implement the prototype of the proposed framework we have used different technologies and architectural models listed below.

3.7.1. Java/JEE

JEE stands for Java Platform Enterprise Edition. The java platform enterprise edition defines a sample standard that applies to all aspects of architecting, developing, deploying and managing multitier server based application. JEE defines a standard architecture composed of an application model, a platform for hosting applications, a compatibility test suite and reference implementation. The JEE platform uses a multi-tiered distributed application model for enterprise applications

EJB stands for Enterprise Java Beans. EJB is an essential part of a JEE platform. EE platform has the component based architecture to provide multi-tiered, distributed and highly transactional features to enterprise level applications. EJB provides an architecture to

develop and deploy component-based enterprise applications considering robustness, high scalability, and high performance. An EJB application can be deployed on any of the application server compliant with the J2EE 1.3 standard specification [33].

3.7.2. OpenNebula

OpenNebula is a cloud computing platform for managing heterogeneous distributed data center infrastructures. The OpenNebula platform manages a data center's virtual infrastructure to build private, public and hybrid implementations of infrastructure as a service. OpenNebula is free and open source software subject to the requirements of the Apache License version 2.

3.8. Issues of Reliability and Validity

Interview questions were organized in the way helps to answer the research question and respondents who have better domain knowledge were chosen for an interview. The result was organized in a group and compared with an other researchers result. Readily available literature was carefully selected from the authenticated sources. The Scientific research method was applied to the interpretation and explanation of findings. Finally, the proposed framework has evaluated by the developing the prototype and the reaction of ICT experts of the bank to test its' validity and acceptability.

CHAPTER FOUR

4. Assessment of the current banking system in Ethiopia

In this chapter we have presented the current banking system in Ethiopia using the information gathered from Interview, observation, literature review. Interview questions were organized in the way that helps to gain understanding about the existing bank's capability to serve their customer, to identify the banking activities best candidate for integration and the respondents' reaction for a change. Nine interviewees from NBE, CBE, United bank, Debub global bank and Birhan international bank were interviewed. Accordingly, results are discussed in this chapter by grouping the issues and comparing the interview findings from relevant literature findings.

Related literature which is written in the adaptation of cloud computing paradigm to the banking industries were reviewed for the intent of choosing a cloud service delivery model, deployment model and for other issues like data center management and virtualization issue was discussed in this chapter.

4.1. Historical Background of Banking in Ethiopia

As the economist agreed that banks are a catalyst for a nations' economic development. In their traditional role as financial intermediaries, banks ensure the transmission of funds from surplus to deficit units and serve to meet the demand of those who need funding. In the country's economy banks basically perform the operation of the payment system, mobilization of saving, and allocation of savings to investment projects.

Most of records and researchers indicated that Ethiopians has practices of the use of modern money more than 2000 years. This had flourished in what is called the Axumite era which can stretch from 1000BC to around 975 AD [24]. The history of modern banking begun in Ethiopia since the agreement that was reached in 1905 between Emperor Minilik II and Mr.MaGillivray, representative of the British owned National Bank of Egypt. Following the agreement, the first bank called Bank of Abysinia was inaugurated in Feb.16, 1906 by the Emperor. The Bank was totally managed by the Egyptian National Bank with specified rights and concessions that were agreed upon the establishment of Bank of Abyssinia. The new Bank, Bank of Ethiopia, was a purely Ethiopian institution and was the first indigenous

bank in Africa and established by an official decree on August 29, 1931, with a capital of £750,000. Bank of Ethiopia took over the commercial activities of the Bank of Abysinia and was authorized to issue notes and coins.

The National Bank of Ethiopia with more power and duties started its operation in January 1964. Following the incorporation as a share company on December 16, 1963 as per proclamation No.207/1955 of October 1963, Commercial Bank of Ethiopia took over the commercial banking activities of the former State Bank of Ethiopia. It started operation on January 1, 1964, with a capital of Eth. Birr 20 million.

Following the declaration of socialism in 1974 the government extended its control over the whole economy and nationalized all large corporations. Organizational setups were taken in order to create stronger institutions by merging those that perform similar functions. The financial sector the socialist-oriented government left behind constituted only 3 banks and each enjoying monopoly in its respective market. The following was the structure of the sector at the end of the era. The National Bank of Ethiopia (NBE), the Commercial Bank of Ethiopia (CBE) and Agricultural and Industrial Development Bank (AIDB).

Following the demise of the Dergue regime in 1991, Monetary and Banking proclamation No.83/1994 and the Licensing and Supervision of Banking Business No.84/1994 laid down the legal basis for investment in the banking sector. Consequently, shortly after the proclamation the first private bank, Awash International Bank was established in 1994 by 486 shareholders and by 1998 the authorized capital of the Bank reached Birr 50.0 million. Dashen Bank was established on September 20, 1995, as a share company with an authorized and subscribed capital of Birr 50.0 million. 131 shareholders with subscribed and authorized capital of 25.0 million and 50 million founded a bank of Abysinia. Wegagen Bank with an authorized capital of Birr 60.0 million started operation in 1997. The fifth private bank, United Bank was established on 10th September 1998 by 335 shareholders.Nib International Bank that started operation on May 26, 1999, with an authorized capital of Birr 150.0 million. Cooperative Bank of Oromia was established on October 29, 2004, with an authorized capital of Birr 22.0 million. Lion International Bank with an authorized capital of Birr 108 million started operation on October 02, 2006. Zemen bank started operation on June 17, 2008, with an authorized capital of Birr 87.0 million. Oromia International Bank

started operation on September 18, 2008, with an authorized capital of Birr 91 million, Buna International bank and Birhan bank in 2009, Abay bank in 2010, Addis International bank in 2011, Debub Global bank and Enat bank in 2012 were established [23].

Currently, there are 16 private commercial banks and the central bank (NBE) and two government banks in the Ethiopian banking industry. Even if there is an allotment of a foreign bank in the county's economy still now there are only two foreign banks; Citi Bank from America and Commerzbank from Germany open their branch in the Addis Ababa. Despite a rapid increase in the number of local financial institutions since financial liberalization, the Ethiopian banking system is still underdeveloped compared to the rest of the world and cash is still the most dominant medium of exchange.

4.2. The Current Banking Activities in Ethiopia

Banks operate their functions based on the regulatory and supervisory measures taken by their respective country and the political economy agendas followed by the government. In the cases of Ethiopia, National Bank of Ethiopia (NBE) is responsible for monetary management and regulation and supervision of banks. NBE set the reserve requirement directive as one of the important monetary policy instruments and prudential regulation tools. According to reserve requirement-6th replacement directives No.SBB/55/2013 banks operating in Ethiopia shall open two separate birr accounts such as reserve account which is used to maintain the reserve balance and payment and settlement account to carry out day to day transactions through the national bank.

In accordance with the NBE regulation banking organizations in Ethiopia are performing the activities like keeping money safe and allowing withdrawals, issuance of checkbooks, credit cards and other payment instruments; providing personal loans, commercial loans, mortgage loans and overdraft agreements for the temporary advancement of the Bank's own money to meet monthly spending commitments of a customer in their current account; notary service for financial and other documents. Banks also perform another activities; like foreign exchange services, wealth and asset management.

The advancement in technology has played a crucial role in improving service delivery standards in the banking industry. ICT investment also facilitates revenue growth through

new value propositions, new market, and sales channels, and improves management of life cycle. Higher customer satisfaction leads to higher loyalty, which ultimately leads to higher revenue growth. ICT can help firms reduce operational costs, general and administration costs and marketing costs (Yalew, 2015).

According to the Growth and Transformation Plan (GTP) of NBE directives mandated banks operate in Ethiopia to have core-banking and modern payment systems. Now a day all banks in Ethiopia have already used the digitized and networked banking services by applying CORE ("centralized online real-time environment") banking system to link all of their branches. The core banking solutions enable banks to link their branches to the central database of the bank and also to perform an inter-bank transaction through the ET-Switch. Even if there is networking problem and high rate of illiteracy in the country customers of the banks may access their money from any of the member branch office/agents/computing devices by accessing the application from centralized data centers. This means that the deposits made are reflected immediately on the bank's servers and the customer can withdraw the deposited money from any of the bank's branches throughout the country. Thus, using core banking technology bank`s provide banking services (including cash withdrawals, bill payments, fund transfer, balance inquiry, and etc) available across multiple inter-connected channels like ATMs, Internet banking, and branches.

To share infrastructure and to give more electronic access channel to their customer, six (6) private commercial banks - Awash International Bank S.C., Nib International Bank S.C, United Bank S.C., Birhan International Bank S.C., Addis International Bank S.C. and Cooperative Bank of Oromia share a network called Primer Switch Solution (PSS) ATM network. PSS is an entity that was eventually established under consortium in 2009. They are the first certified Third Party Payment Processor by the regulatory party, National Bank of Ethiopia. PSS has went operational in July 2012 with its three founder member banks and latter three banks joined taking the advantage of opportunity availed by PSS. Customers of any of the six banks use their ATM card to withdraw money from an ATM and POS terminal of any of the Premier Switch members. The following transaction types are supported by ATMs connected to PSS network; bill payment, money sends, money to receive, fast cash, mini statement, fund Transfer, PIN change, balance inquiry, cash withdrawal.

On the other side through collaboration between Zemen Bank and Dashen Bank under the "Q-Link" network, the customers of Zemen and Dashen Banks are both able to benefit from a total network of more than 200 ATMs at locations throughout Addis Ababa and several regional cities. This collaboration marked as a significant cooperation between competing banks in Ethiopia, which others should be encouraged to follow (Zemen Bank, 2015).

The National Bank of Ethiopia set a strategy to have modern and up to the standard payment system in Ethiopian banking industry. The much anticipated Eth-Switch, which is expected to centralize all banks' online real-time and electronic (CORE) banking solutions, is to commence its pilot project on May 28, 2015 and its official launch on May 12, 2016.The system connects all the banks in the country so that customers can access their respective bank account from any given bank.

Eth-Switch was formed in 2011, through shareholding of all licensed banks in Ethiopia, and the National Bank of Ethiopia (NBE), to provide a national switch system for all financial institutions. Eth-Switch has selected the SmartVista solution from BPC Banking Technologies, a provider of Open System payment solutions, to power its national switching operations in Ethiopia.

SmartVista will connect all banks to a central transaction switching platform that will provide customers with access to their money and other financial services via ATM, mobile and Internet channels, and POS devices, regardless of their home bank. The solution will also allow Eth-Switch to interface with international payments schemes, such as Visa and MasterCard and enhance local acceptance of these globally branded cards. The system will be linked to the national automated transfer system (comprising ACH and RTGS) to facilitate efficient clearing and settlement of all transactions.

Currently, different banking service access channel technologies are available these are ATMs, POS, Internet banking, Mobile banking, branch offices, agent banking and others technologies. Integrating banking service access channels which are held by different banks save cost and increases accessibility.

4.3. The Current Ethiopian Banking System Architecture

As explained in the previous section according to the NBE directive all banks in the country deployed a core banking system and connect all of their branches and access channels through it. As shown in the figure below most of the banks extend their access channel more than the previous dates and in day to day they are approaching to their customer for better service delivery and to give more customer-centric service.

Before the deployment of ET-Switch, PSS-Switch and Q-link switch were deployed in Ethiopia. Now a day these two switches are connected to the ET-switch to connect their member bank to the national payment system. The figure below shows the high-level physical architecture of the existing system from the customer point of view.

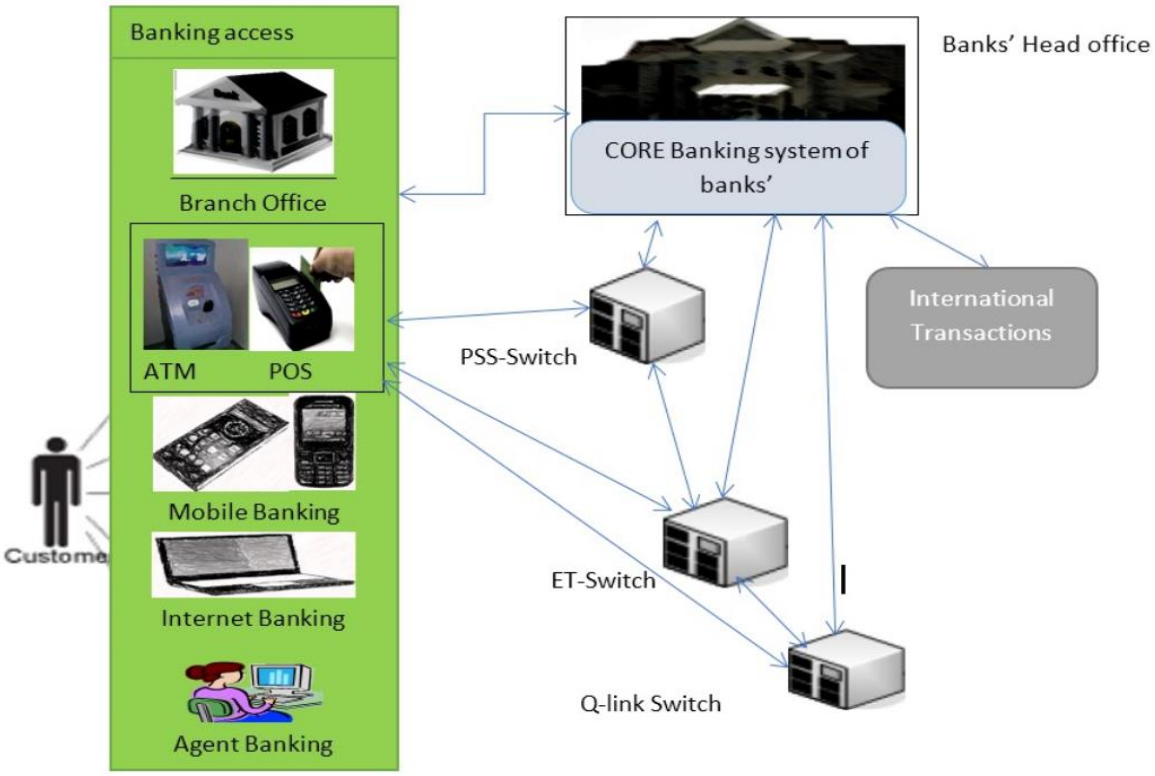

FIGURE 4-1 : the current banking system high-level architecture

Scenario of the current banking system architecture

1. If customers of any bank want to use their respective bank's channel, then the transaction is directly responded by the core banking system of their respective bank.
2. If customers of the PSS or Q-link member bank want to use any PSS or Q-link member bank's ATM or POS terminal, then the transaction is directly responded by

the PSS-switch or the Q-link switch and then the switch links to the core banking system of their respective bank

3. If customers of the PSS or Q-link member bank want to use any bank's ATM or POS terminal in Ethiopia, then the transaction is directly responded by the Eth-Switch which is directly connected with the PSS-switch or the Q-link switch and then the switch links to the core banking system of their respective bank.
4. If customers of any bank out of PSS-Switch and Q-link switch member bank want to access POS or ATM terminal of other banks, then the transaction is directly responded by the Eth-switch which is connected with their respective bank's core banking system.
5. If customers of any bank out of PSS-Switch and Q-link switch member bank want to access POS or ATM terminal of PSS or Q-link member banks', then the transaction is directly responded by the PSS-switch or Q-link switch which is connected with Eth-switch and then their respective bank's core banking system.
6. If any user wants to get a service in relation to foreign transaction, he/she can access through the external system which interfaces with the banks' system

As shown from the above diagram the current banking architecture has many redundancies that mean switching solutions, limited service deliveries that mean integration effort only in ATMs and POS machines, has fewer advantages of cost from the bank's point of view because integrating the privately owned data center . This proposed system alleviates the problem in the current system and enable the banks to share a core banking software and different channel integration and security systems based on their request of the infrastructure.

4.4. Presentation of the Interview Result

In this section, the data collected from the sample organization were presented and accordingly the appropriate interpretations were made

Bank to channel relation

TABLE 4-1: CUSTOMER CHANNEL RELATION

Bank Name	No. of branch offices	No. of ATM	No. of customer	Ratio of customer to channel
CBE	1274	648	14000000	1: 7284.08
UB	130	75	70000	1: 341.46
BIB	91	10	130000	1: 1287.13
DGB	28	5	36615	1: 1109.55

Table 4-1 shows that customer to channel relationship in the selected banks' ATM service and branch offices.as shown from the result largest government-owned bank CBE have one ATM or branch office for every 7284 customers and other privately owned banks UB 1 to 341, BIB 1 to 1287 and DGB 1 to 1109 customer to bank relationship. In this relation, we can say UB has a better customer to channel performance.

System security of the bank

TABLE 4-2: SYSTEM SECURITY STATUS OF THE BANK

Metrics	Number of organization
Very good	1
good	1
Satisfactory	2

Among the four interviewed organization 1 responded to a good security level a 1 very good and other satisfactory. It is not the only case to check a security level of the bank we have also seen that banks are still in attempt to increase their security by signing agreement with the foreign and inland technology companies

Sufficiency of banking channel to achieve organizational mission and vision

TABLE 4-3: CURRENT CHANNEL SUFFICIENCY RESPONSE

Do you think that the banking access channel owned by your organization are sufficient to achieve organizational mission and vision?	Number of "Yes"	Number of "No"
	0	4

The question asked to get the reaction of the respondent whether their banking channel is sufficient or not. As shown in table 5 above the entire respondent agreed on the insufficiency of their access channel to meet their organizational mission and vision in satisfying their customers. All the respondents say "NO".

Availability of integrated service delivery

TABLE 4-4: AVAILABILITY OF INTEGRATED SYSTEM

Number of "Yes"	**1**
Number of "no"	3

The question asked about the availability of an integrated system with other bank has been asked 3 of the bank say "Yes" and 1 "NO". It was true two switching solutions are available and some of the banks are the member of them and one currently begun Eth-Switch. These switching solutions integrate the ATMs and POS machines. This attempt of integrating banking channel clearly shows that a single bank cannot afford sufficient banking access channel to their customer.

4.5. ICT service delivery Capacity of individual banks

During the interview, questions which enable to assess the banks' capacity to afford enough access channels to their customer; their security level; availability of any integrated system and the reaction of ICT experts to change the current system were raised. The respondents also asked to answer the number of customer in their organization, the number of ATMs, POSs and branch offices. They were agreed that due to high investment cost on IT infrastructures, other related factors they can't deliver enough service to their customers. Even the largest government-owned bank CBE has 1274 branch offices and 648 ATMs to serve around 14 million customers. This observation indicates that banks should have look another technological option to deliver service that satisfies their customer.

Even though there are some current attempt to integrate the ICT service but most of the banks invest millions of dollar for fulfilling the changing computing needs to deliver the banking services consistently in the market at enterprise level individually. One example to this is the recent agreement between the Commercial bank of Ethiopia (CBE) and Igloo to build a security center which protects hackers and an attempt to cyber-attack to its infrastructure costs up to 30 million birr. It's argued that what if all banks need this kind of security system, it costs 30 million times 18 banks which are 540 million birr that is unnecessary wastage of the country's economy. Also, one more example is that recently CBE wants to deploy the switching system which will integrate the transaction made via ATM and POS which will estimate to cost four to five million dollars. Since all the banks need this system it is better to collaborate for the better achievement and cost efficient IT strategy following saving the country from a shortage of foreign currency deficiency.

4.6. Identification of Banking Activities for Integration

We have observed that the forms to perform banking activities in the banks taken as a case study are similar that is a fertile ground for developing SaaS to the banking industries. To perform a banking transaction like payment, deposit, transfer, account management, auditing and report generation it's possible to use a shared pool of resources accordingly increase service availability, interoperability and reducing cost. It is also possible to share some of the management systems like CRM(Customer Relationship management), HRM(Human Resource Management), Finance control and case management system.

4.7. Selection of Cloud Services Provider

Before dealing with the cloud service user organization should have to analyze the cloud providers' characteristics like the location of their data center, the service they provide, Service Level Agreement (SLA) and billing method. Amazon, Google, Microsoft, SalesForce, and IBM are among the major cloud providers. In this research, they are not selected as a candidate cloud provider because of the unsolved security and privacy issue in the use public cloud for the banking and other financial institutions.

As one of the respondent during interview time explained that Google cloud providers came to Ethiopia and had given a seminar on their cloud service for a financial institution and request EBI to use their cloud service but EBI were unwilling to use Google's cloud service because of security and privacy issue.

In this research, we have proposed to build the new cloud provider which is operating within a country and managed by Eth-switch. Therefore, Eth-switch is the candidate and trusted the provider to serve cloud computing solution for EBI.

4.8. Selection of cloud computing service model

Cloud providers basically use three cloud computing service delivery models such as IaaS, PaaS and SaaS. Among these service delivery models, SaaS has been chosen as a service delivery model in this research. The SaaS customer is an end-user of complete applications running on a cloud infrastructure and offered on a platform on-demand. The applications are typically accessible through a thin client interface, such as a web browser. The customer does not control either the underlying infrastructure or platform, other than application parameters for specific user settings. Even if we choose a certain model to deliver a service to the customer there are some synergies among the model. For example, if an infrastructure as a service provider needs some application form to their customer and also we choose SaaS for the banking user they also use the platform and infrastructure of the provider. The following three diagram shows the possible three software-as- a- service architectural styles.

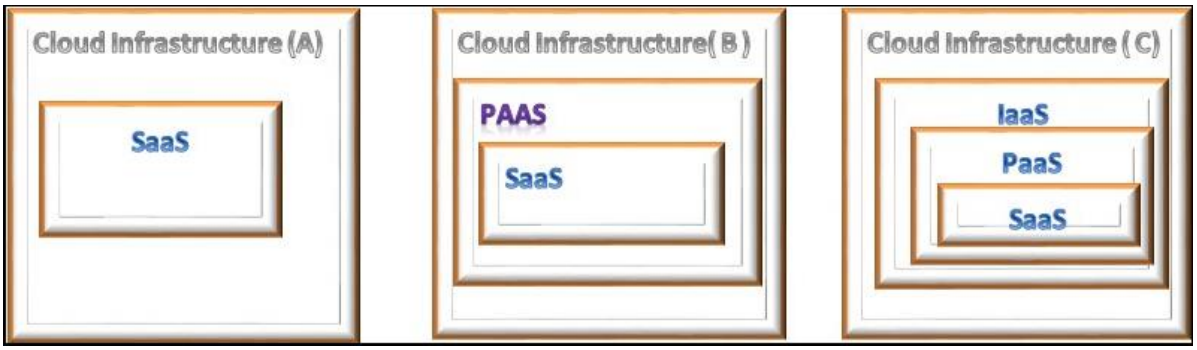

FIGURE 4.2: DIFFERENT SAAS ARCHITECTURE

As shown in the above diagram there are three different architectural styles even in using SaaS. We have chosen architectural style shown on fig 12 C. Hence any customer banks of EBI can get software as a service with a subscription base also uses the platform and infrastructure. However by considering that big capability variation among the banks and to control denial of service attack the interested organization requests a virtual infrastructure according to their requirements such as network bandwidth, storage, and computational resources.

4.9. Selection of cloud computing Deployment model

In literature review section the advantages and disadvantages of cloud computing deployment models have been assessed. Based on the assessment private cloud is more secure than the other while it is very costly whereas public cloud is going with many security and privacy issues. Consequently, we have looked to design a community cloud in which the cloud services targeted organizations with common objectives and security controls. This model is more secure than the public cloud and it takes the cost advantage of private cloud

As mentioned in NIST, 2013 the cloud computing roadmap any of the four service delivery models can be deployed on any of the three cloud deployment models. Hence the proposed system provides a core banking system over the community cloud deployment model as shown below.

FIGURE 4. 1: THE SERVICE DELIVERY MODEL ON COMMUNITY CLOUD

4.10. The data center networking issue

Ethiopian community cloud has expected to have multiple distributed data centers which are managed centrally by using the OpenNebula cloud infrastructure management tool. In each data center the application server, the database server, and the virtual machine manager are necessary to be installed.

CHAPTER FIVE

5. PROPOSED INTEROPERABLE CLOUD COMPUTING FRAMEWORK

5.1. Backgrounds of the Framework

This research has aimed to design an interoperable cloud computing framework that serves the banking industry in Ethiopia. To do this, Firstly, we have studied how the current banking operations are performed and the banking activities which are possible for the integrated service delivery. Secondly, the cloud computing technology which helps to develop the framework has been identified.

In literature review and result discussion section, we have presented the requirements to develop the proposed solution. Based on that assessment the community cloud deployment model is chosen as the deployment model and infrastructure as a service and software as service models are chosen as a cloud service delivery models. When any banking industry signed up to use EBICC, the system allocates a virtual server which has a core banking system and other management and security systems. In this proposed system we have used multiple data center architecture to protect the entire site/application from being negatively affected by some type of network/power failure, lack of available resources, or service outage that's specific to a particular datacenter.

5.2. Building the Framework

The proposed framework considers the current banking architecture which contains individual bank's data center, the ETH-switch, Q-link, PSS, NBE and other global system interfaces. After analyzing each component of the current banking architecture we have chopped out some of the redundant components which are the causes for delay of the transaction, and difficult for interoperability then we have added a community cloud which replaces core banking and other management software which are managed by individual banks' data center.

To make the framework more clear two architectures have been designed. The first one shows the high-level component architecture which is designed from the end user point of

view and the second one shows the process model of the system which is also designed from the user point of view. It is known that currently different banking channels are available and customers of the bank use any of these access channels whether owned by their respective bank or any other banks. The figure below shows the high-level component architecture of the proposed framework.

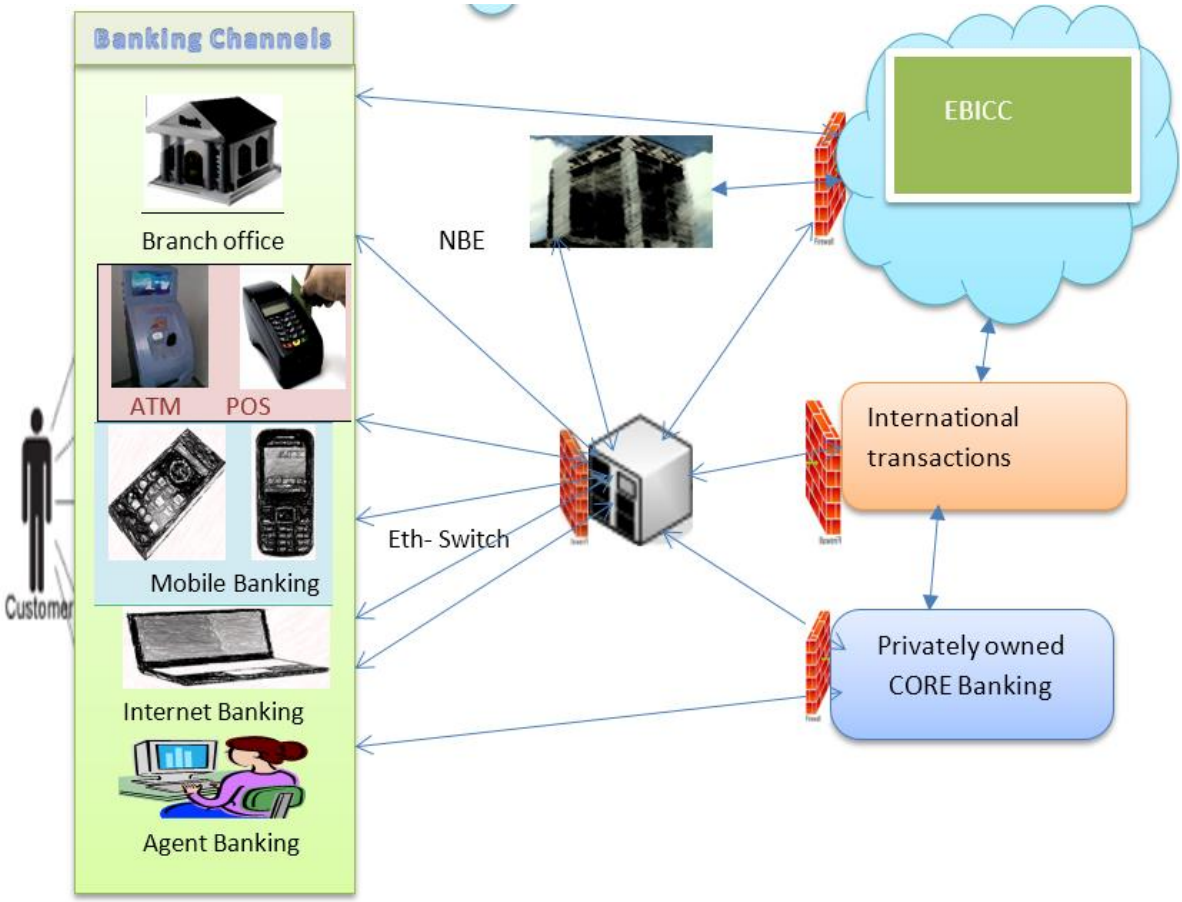

FIGURE 5-1: HIGH-LEVEL COMPONENT ARCHITECTURE OF PROPOSED EBICC

The high-level architecture basically contains Customers, access channels, NBE, ETHswitch, EBICC, global transaction and privately owned core banking system. The customer is any users of the banking system whether a bank customer or administrators who access the system by using any banking system access channel. Access channels are devices which enable users of the banking system to perform different transactions. These banking access channels use internet network or mobile cellular network to access the system. NBE is the controller of banking operation in Ethiopia which controls Clearing and settlement issues, banking service availability to the customer and banks' financial capability. Hence it's

included through ETH-switch which is the national switch that enables the inter-bank transaction under the control of NBE.NBE also access EBICC directly to control the transaction among the members. In this proposed framework we have designed ETH-switch to handle transactions between EBICC and privately owned core banking system and also for global transactions. It is not expected that all banks in Ethiopia join EBICC immediately that is why we include a privately owned core banking system to the framework. EBICC is proposed community cloud model which deliver a core banking solution and other banking systems to the banking industries in Ethiopia. Customers of EBICC member bank directly access the system by using any of member banks channel whereas customers of privately owned data center bank access the EBICC through the ETH-switch and their bank's data center directly.

The scenario about the high level proposed framework architecture:

1. If customers of a bank want to use their respective bank's channel, then the transaction is responded by EBICC if the bank is the member of EBICC else responded by privately owned core banking
2. If customers of the member bank of EBICC want to use any member bank's access channel, then the transaction is directly responded by the EBICC.
3. If customers of the member bank of EBICC or customers of a bank with privately owned core banking system want to use any other bank's access channel, then the transaction is checked by the ETH-Switch then responded by EBICC if the bank is the member of EBICC else responded by the privately owned core banking system .
4. If any user wants to get a service in relation to the foreign transaction, then they can access the system through the external system interfaces of EBICC or privately owned core banking system.
5. National Bank of Ethiopia (NBE) can control the banks through Eth-Switch and also can process the clearing and settlement information.

The flow chart diagram of the high-level architecture

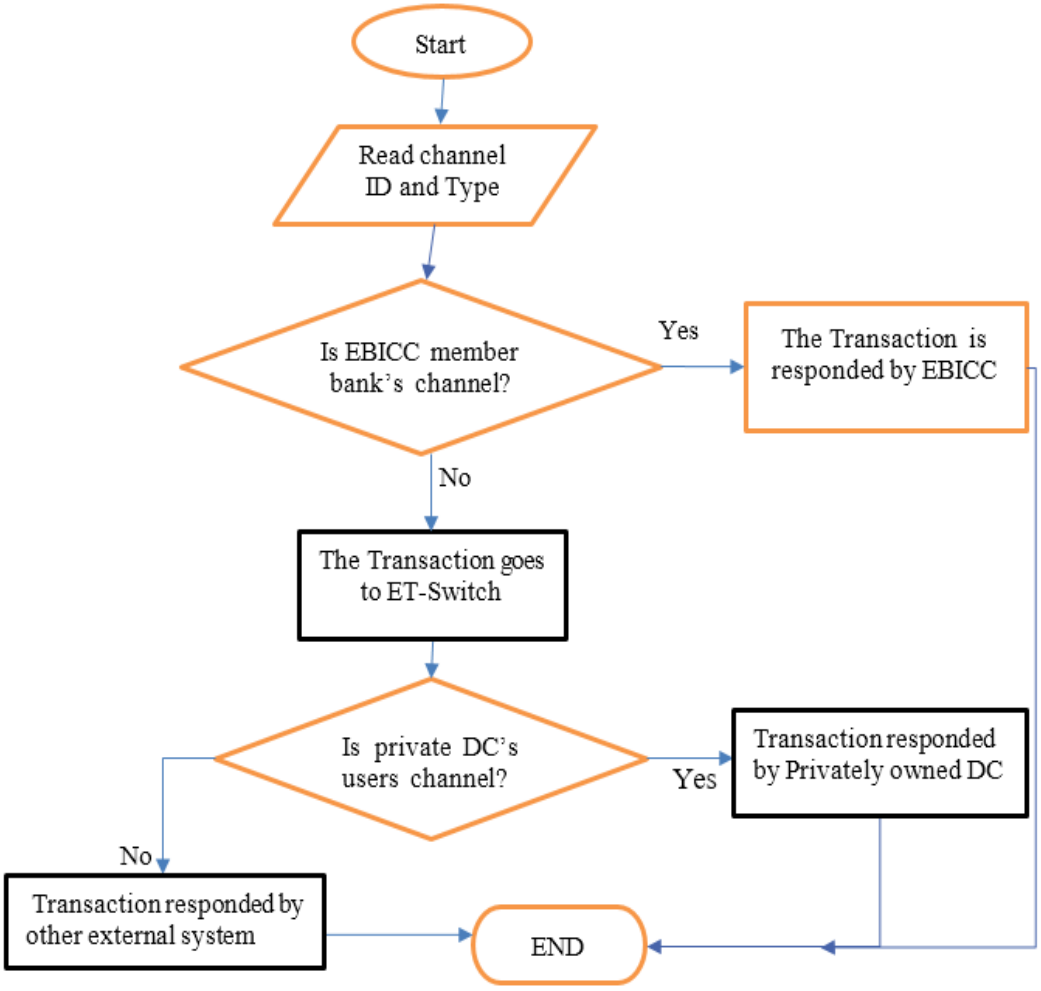

FIGURE 5-2: FLOW CHART FOR HIGH-LEVEL ARCHITECTURE THE PROPOSED FRAMEWORK

Based on the background described above the process model of the framework is designed in layered architecture as shown below.

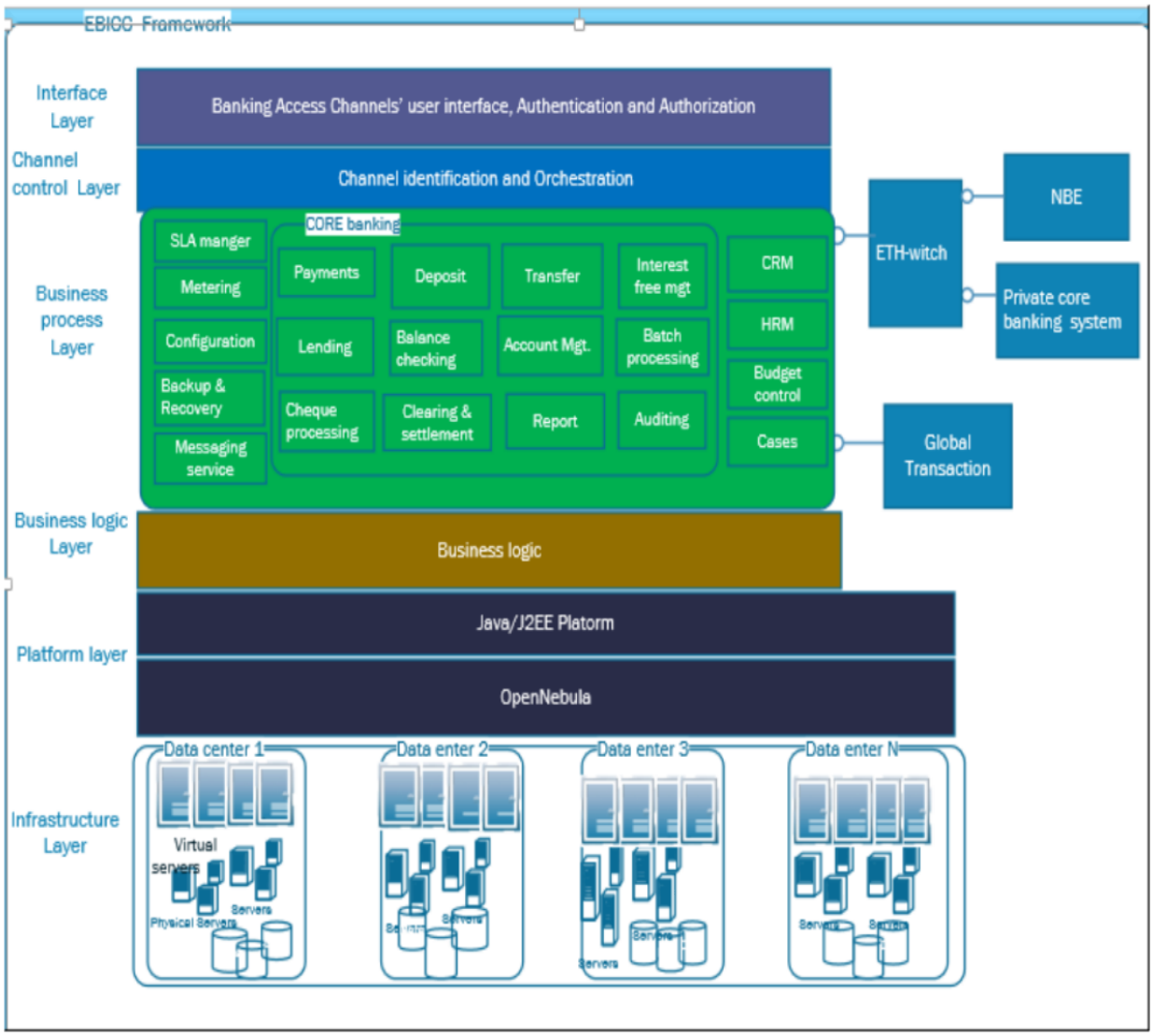

FIGURE 5-3: THE PROPOSED INTEROPERABLE COMMUNITY CLOUD FRAMEWORK FOR EBI

The proposed framework in the above figure contains six different layers (interface, channel control, Business process, business logic, Infrastructure management and Infrastructure service).

5.2.1. Interface layer

This layer is an interface between the user and the proposed EBICC. The layer contains security credentials and new user sign up form which enables to access each cloud services through security-rich environment. If the user the correct username and password then based

on the role assigned and the type of device used can process the respective transaction. The tenants are identified by their credential and their respective page displayed at that session.

5.2.2. Channel control layer

As discussed in chapter four, there are different kinds banking access channels such as ATM, POS, mobile, agents and branch banking. To display the correct banking service list to the user and to control the banking transaction the channel should be identified by their ID and orchestrated. Additionally to facilitate the interbank transactions these banking channels owned by different banks need to be integrated.

5.2.3. Business process layer

Once the user and its channel have been identified then the business services lists are offered. Accordingly, the user fills the required fields and can process the transaction. In this layer different hosted applications are listed such as core banking system and other management software.

5.2.4. Business logic layer

This component is where basic business logic scripts were written and connected to the query processing to manipulate tenant data and perform application specific operations.

5.2.5. Platform layer

In this layer, both the J2EE and the cloud virtual infrastructure controller OpenNebula are used. J2EE used to manage the application layer and OpenNebula for managing the virtual infrastructure. The J2EE Application Programming Model is the standard programming model used to facilitate the development of multi-tier, thin client distributed application.

OpenNebula has been chosen as an infrastructure management tool because of its capability enables to manage heterogeneity and complexity of the data center infrastructure, cluster management and interoperability with the legacy system. In addition, OpenNebula orchestrates storage, network, virtualization, monitoring, and security technologies to deploy multi-tier services (e.g. compute clusters) as virtual machines on distributed infrastructures, combining both data center resources and remote cloud resources, according to allocation policies.

5.2.6. Infrastructure layer

In this layer, the physical and the virtual servers are available in different data centers. The layer offers infrastructure services including servers, storage, and network bandwidth on the cloud environment. The users can consume the virtual services from the proposed EBIs infrastructure service provider. In this layer, the virtual server installed with the core banking system software and other management system software is provisioned to the banking industry based on their processing and storage resource request.

5.2.7. Another external system interfaces

EBICC framework is designed to work with other external interfaces such as ETH-switch, NBE, privately owned core banking system and global transactions. NBE and banks which are not a member of EBICC interact with the system using ETH-switch whereas global transactions directly linked using the allowed interface.

It's known that those banking industries are very sensitive to security, hence we are not expected them to move to their system to the cloud immediately. But fewer infrastructure banks, low capital and newly opened banks will be the premier candidates in using the proposed system if it will be implemented.

CHAPTER SIX

6. PROTOTYPE DEVELOPMENT FOR THE PROPOSED FRAMEWORK AND VALIDATION OF RESULTS

6.1. Overview of the Prototype

In the previous chapter the design of the proposed framework which contain six different layers and which is based on community cloud computing deployment model and SaaS and IaaS service delivery models have been presented. In this chapter, we have discussed how the proposed framework can be implemented and become usable to the banking community. We put here, the highest level view of the system which looks like and how it works if it will be implemented and the programming tools to implement the framework.

6.2. The Development Environment

To implement the prototype the component of java NetBeans 8.0.2 IDE called JEE (Java Enterprise Edition) platform which is built on the top of java standard edition have been used. The java platform enterprise edition defines a sample standard that applies to all aspects of architecting , developing, deploying and managing multitier server based application. The aim of the JEE platform is to provide developers with a powerful set of APIs while shortening development time, reducing application complexity, and improving application performance [39]. JEE enables us to develop a multi-tier, model-view-controller architecture. Typically, multi-tiered applications have a client tier, a middle tier, and a data tier. The client tier consists of a client program that makes requests to the middle tier. The middle tier's business functions handle client requests and process application data. The data tier functions handle long-term storing of information for middle tier[40].

In this prototype, we especially using JEE technologies like; JSF (Java Server Faces) for presentation tier, JPA (Java Persistence API) for persistence and EJB for business logic. We have also used glassfish server as application server and MySQL as a database server.

EJB (Enterprise Java Bean) is an essential part of a JEE platform that provides an API and runtime environment for developing and running large-scale, multi-tiered, scalable, reliable, and secure network applications. EJB offers enterprises the advantage of being able to

control change at the server rather than having to update each individual computer with a client whenever a new program component is changed or added. Enterprise beans are managed components that encapsulate the core functionality of an application. EJB component, or enterprise bean, is a body of code that has fields and methods to implement modules of business logic.

JPA is An API for accessing data in underlying data stores and mapping that data to Java programming language objects. The Java Persistence API is a much higher-level API than JDBC and hides the complexity of JDBC from the user.

JSF a set of APIs for representing user interface components and managing their state, handling events and input validation, defining page navigation, and supporting internationalization and accessibility. JSF is designed to be flexible, leverages existing standard user interface and web-tier concepts without limiting developers to a particular mark-up language, protocol, or client device. In this case, we have integrated prime face components as a templet and bootstrap for cascading style sheet.

GlassFish Server Open Source Edition provides a server for the development and deployment of Java Platform, Enterprise Edition (Java EE platform) applications and web technologies based on Java technology. MySQL is used as a database server because it is suitable to integrate it with the Glassfish server.

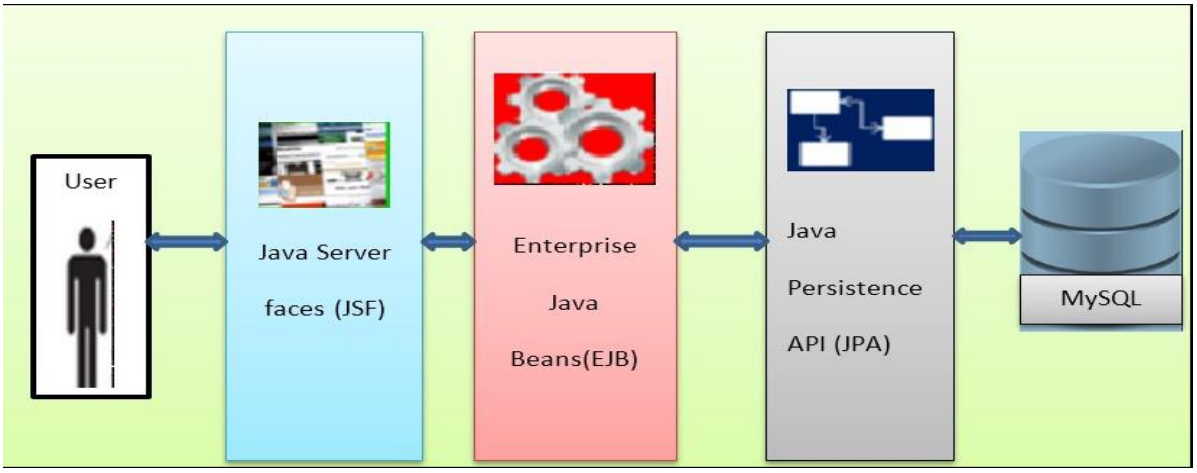

FIGURE 6-1: PROGRAMING MODEL ARCHITECTURE

6.3. The Prototype Modeling

A model is an abstraction describing system or a subset of a system. Under this topic we have used different kinds of UML(Unified Modeling Language) diagrams to model the functionality and Sequence of activities of the system. Functionality of the system has described by using use case diagrams which describes the functionality of the system from the user's point of view and Sequence of activities of the system by using sequence diagrams which describes sequence of messages exchanged among a set of objects.

6.3.1. Use Case Diagram

Use cases focus on the behavior of the system from an external point of view and describe a function provided by the system that yields a visible result for an actor.

Actors and their related use case identification

. We can identifty the following use cases and related to each actor:-

- **Actor name**: Cloud administrator

  Related use cases: Register bank
- **Actor name**: Bank system administrator

  Related use cases:
  - ✓ Manage branches
  - ✓ Manage user
- **Actor name**: Branch officer
  - ✓ Register customer
  - ✓ Manage account
  - ✓ Transfer money
- **Actor name** :Customer
  - ✓ Transfer money
  - ✓ View account statement

Use case diagram for the proposed framework prototype

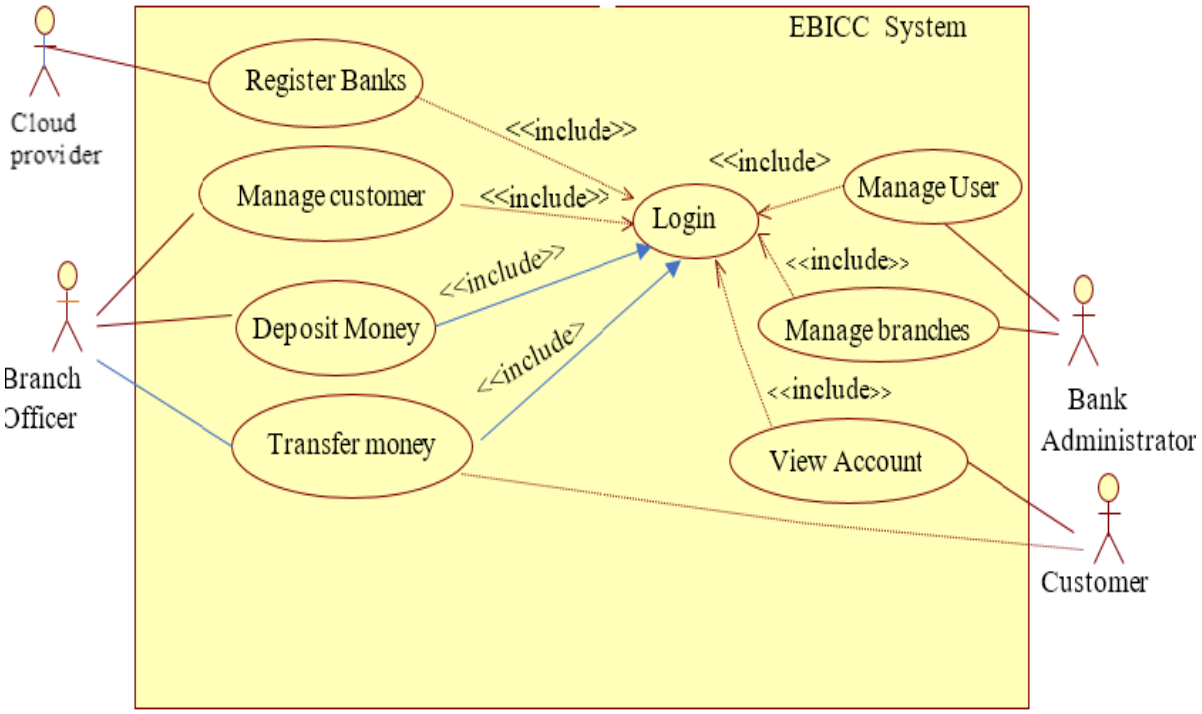

FIGURE 6-2 : USE CASE DIAGRAM FOR THE PROPOSED FRAMEWORK PROTOTYPE

Use case Description

Use case Name: Login

Use case ID: UC1

Actor: Cloud provider, Bank system Administrator, Branch officer, Customer

Description: - initiated when the EBICC users want to login

Precondition: The user should have a valid username and password

Main flow of event

1. The system displays authentication and sign-up link of the system.
2. A user who has already signed up Inserts its' organization name. The role, username, and password
3. Click login button
4. The system verifies that the login form was filled correctly
5. Use case ends

Alternative flow: Access not granted for the user

A-5. The system verifies that the user credential was incorrect

A-6. The system displays an error message that shows the user the user credential was incorrect

A-7. The use case resume from step 2 of main flow of events.

Post condition: The system displays the respective page of the user

Use case name: Register bank

Use case ID: UC2

Actor: Cloud provider

Description: is activated when the cloud provider logged in and wants to add banks.

Precondition: user must be log in to the system

Basic flow of events:

1. The cloud provider choses the bank registration link
2. The cloud provider fills the bank name and bank code
3. Click on register button
4. The system verifies that the bank registration form was filled correctly
5. Use case ends

Alternative flow: A System unable to register bank.

A-4. The system verified that the User filled the bank registration form incorrectly

A-5. The system displays an error message that shows the user filled the form incorrect

A-6. Use case resumes stepping 2 of main flow of events

Post condition: The system displays the successfully registered message

Use case name: **Manage user**

Use case ID: UC3

Actor: Bank system administrator

Description: activated when the bank system admin logged in and wants to add, remove, view or edit users

Pre-condition: user must be log in to the system

Basic flow of events:

1. The system administrator choses the user management link
2. The system administrator choses create user link
3. Then fill the user information

4. Click on register button
5. The system verifies that the user registration form is filled correctly
6. Use case ends

Alternative flow: A <u>System unable to register the user.</u>

A-5. The system verified that the User filled the user registration form incorrectly

A-5. The system displays an error message that shows the user filled the form incorrect

A-6. Use case resumes stepping 3 of main flow of events

Post condition: The system displays the user registered successfully message

Use case name: **<u>Manage customer</u>**

Use case ID: UC4

Actor: Branch officer

Description: is activated when the branch officer logged in and wants to add, remove, view or edit customers

Pre-condition: user must be log in to the system

Basic flow of events:

1. The branch officer choses the customer management link
2. The branch officer choses create customer link
3. Then fill the customer information
4. Click on register button
5. The system verifies that the customer registration form was filled correctly
6. Use case ends

Alternative flow: A <u>System unable to register the customer.</u>

A-5. The system verified that the User filled the user registration form incorrectly

A-5. The system displays an error message that shows the officer filled the form incorrect

A-6. Use case resume from step 3 of main flow of events

Post condition: The system displays the customer registered successfully message

Use case name: **<u>Transfer money</u>**

Use case ID: UC5

Actor: Branch officer, Customer

Description: is activated when the branch officer or customer logged in and wants to transfer money

Pre-condition: user must be log in to the system and the customer should have an account if he/she wants to transfer from their account

Basic flow of events:

1. The branch officer or the customer choses the transfer link
2. The branch officer or the customer fills the transfer form
3. Click on transfer button
4. The system verifies that the transfer form was filled correctly
5. Use case ends

Alternative flow: A <u>System unable to transfer money.</u>

A-5. The system verified that the User filled the transfer information incorrectly

A-6. The system displays an error message that shows the user filled the form incorrectly

A-7. Use case resume to step 2 of main flow of events

Post condition: The system displays the money transferred successfully message and discount the customer's account if transferred from their account

Use case name: **<u>Deposit money</u>**

Use case ID: UC6

Actor: Branch officer

Description: is activated when the customer wants to deposit money and the branch officer log in to process deposit

Pre-condition: user must be log in to the system and the customer should have an account

Basic flow of events:

1. The branch officer choses the core banking link
2. Then the branch officer choses the deposit link
3. fill the deposit form
4. Click on deposit button
5. The system verifies that the deposit form was filled correctly
6. Use case ends

Alternative flow: A System unable to deposit money.

A-5. The system verified that the User filled the information incorrectly

A-6. The system displays an error message that shows the user filled the form incorrectly

A-7. Use case resume from step 3 of main flow of events

Post condition: The system displays the money deposited successfully message and increment in the customer's account balance

6.3.2. Sequence diagram

Sequence diagram is a dynamic model of use cases, showing the interaction among classes during a specified time period. We have used a sequence diagram to formalize the behavior of the system and to visualize the communication among objects. This diagram describes the flow of messages, events, actions between objects and show time sequences that are not easily depicted in other diagrams.

User login sequence diagram

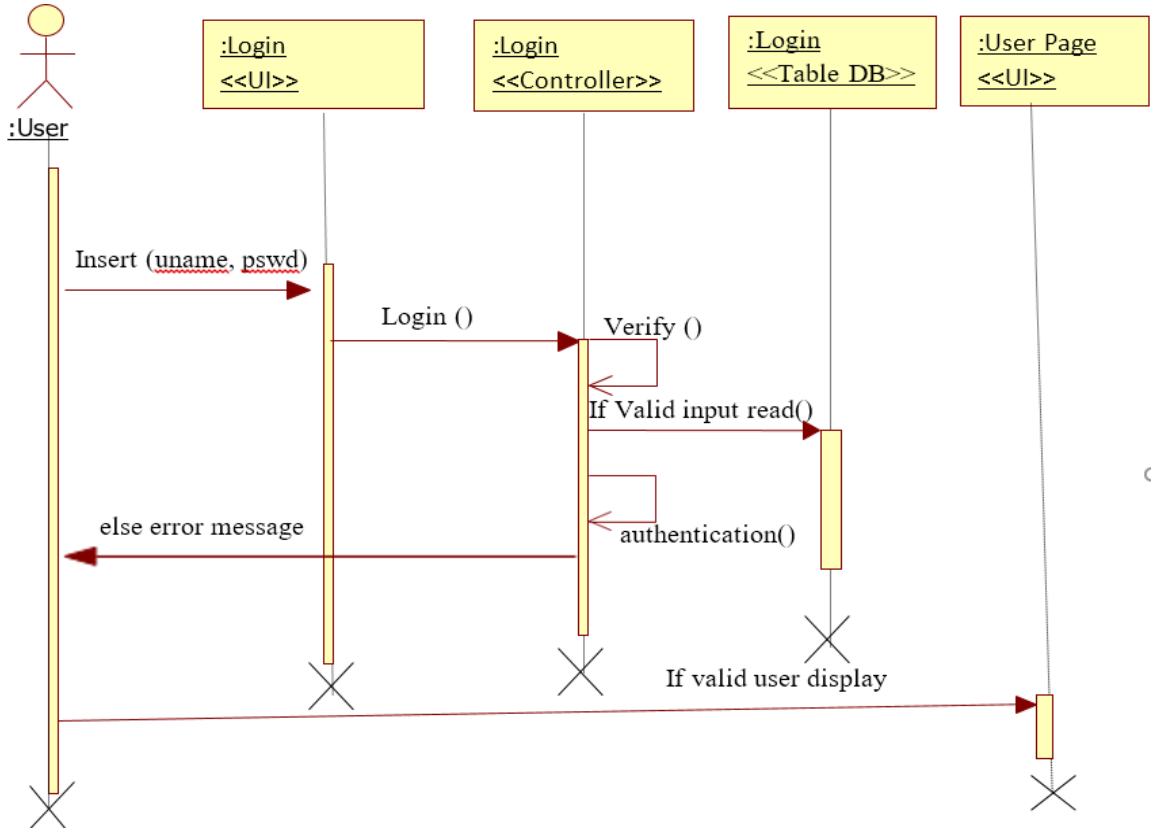

FIGURE 6-3: USER LOGIN SEQUENCE DIAGRAM

Description for User login sequence diagram

1. When a registered user wants to login the system displays authentication and sign-up link of the system.
2. A user inserts their organization name. role, username and password
3. Click login button
4. The system verifies that the login form was filled correctly
5. If the user fills the login page incorrectly then system displays error message and begin from step 2
6. Else the user filed the required credentials correctly then the system authenticate user input from the database
7. If the user input match with the user information in DB then the system display error message and begin from step 2
8. Else the user input match with the user information in DB then the respective page of the user will be displayed
9. The sequence ends

User registration sequence diagram

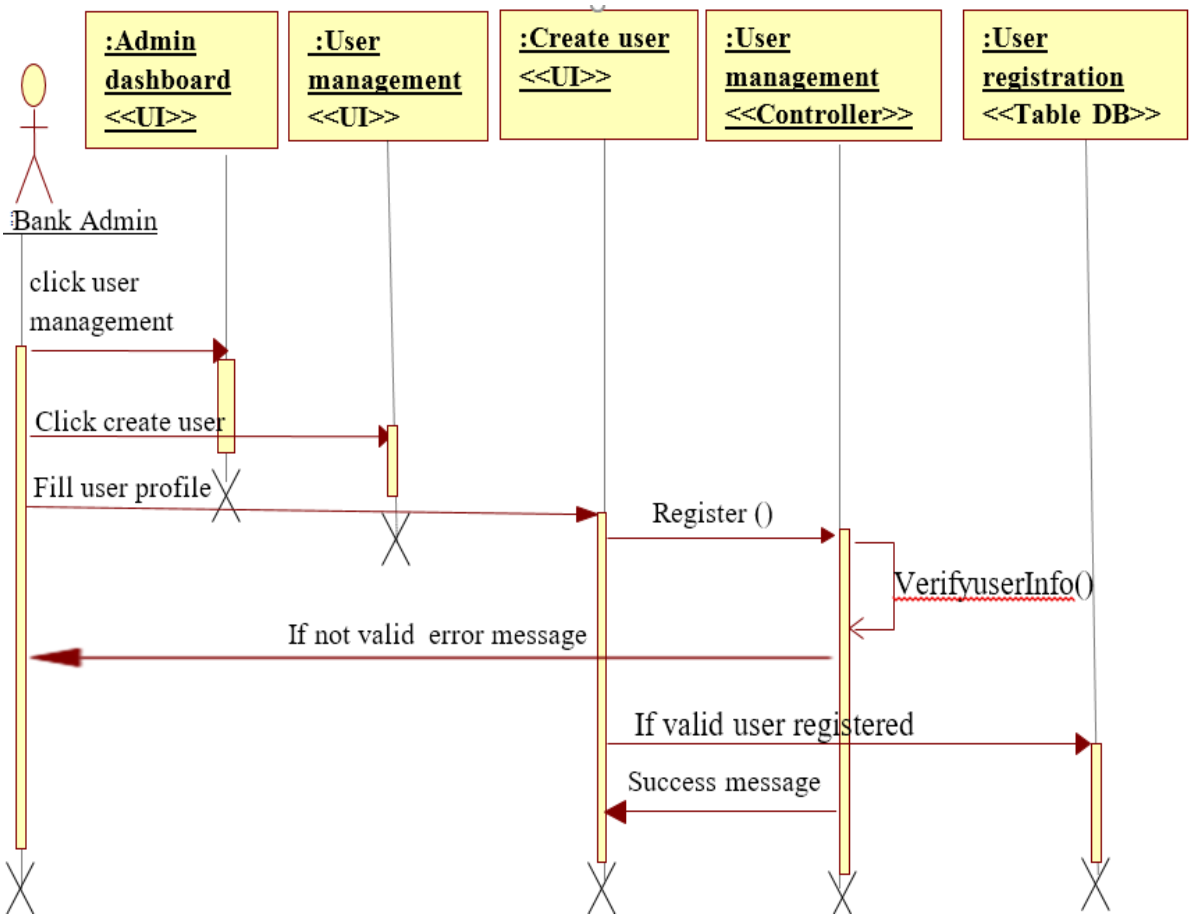

FIGURE 6-4:USER REGISTRATION SEQUENCE DIAGRAM

Description of user registration sequence diagram

1. When a bank admin wants to register a user they should login to the system
2. Then the system displays admin dashboard
3. The system administrator choses create user link
4. The system displays the user registration form
5. Then the system admin fill the user information
6. The user management controller verifies the user information
7. If not filled correctly the system displays error message
8. Else registered user information in database and displays the user registered successfully message then the sequence ends

Money transfer sequence diagram

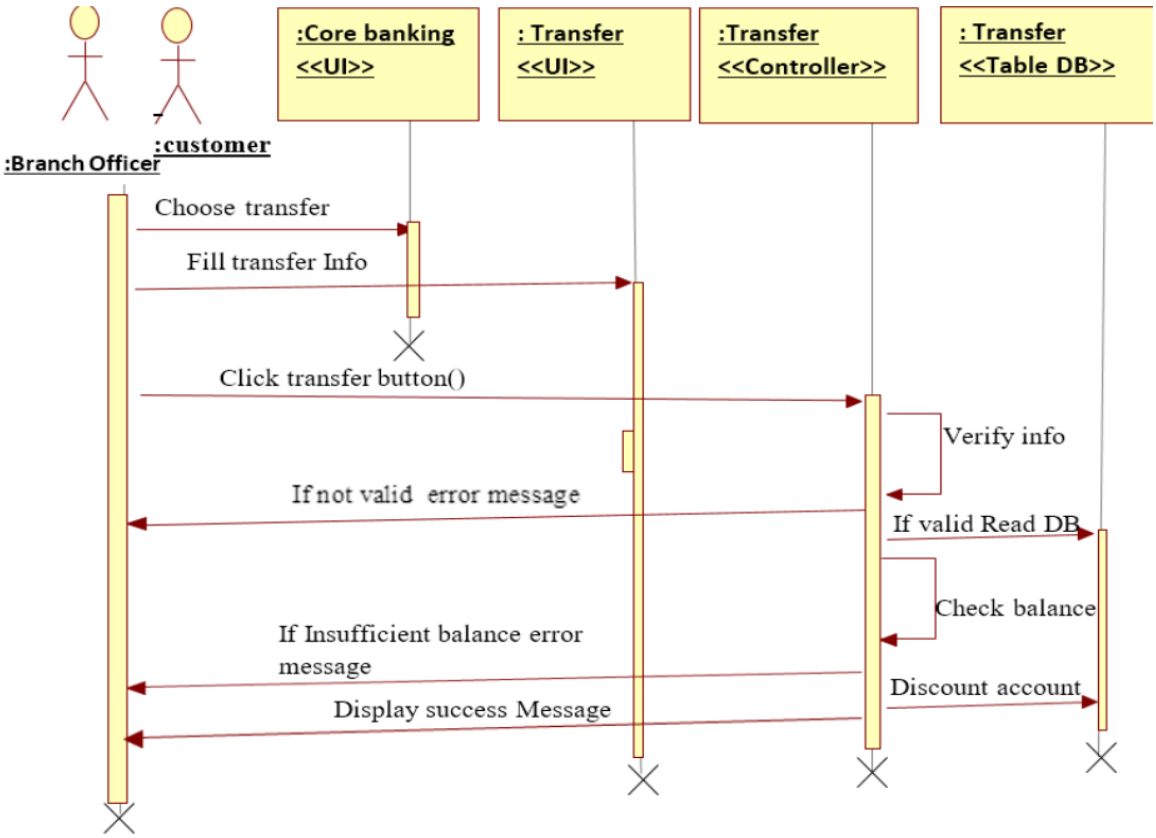

FIGURE 6-5: MONEY TRANSFER SEQUENCE DIAGRAM

Description transfer money

Pre-condition: user must be log in to the system and the customer should have an account if he/she wants to transfer from their account

Basic flow of events:

1. When the branch officer or customer wants to transfer money they should login
2. The system displays the core banking dashboard
3. Then the branch officer or the customer click on transfer link
4. The system displays the transfer form
5. The branch officer or the customer fills the transfer form
6. Click on transfer button
7. The system verifies that the transfer form was filled correctly
8. If the User filled the transfer information incorrectly
9. Then the system displays an error message and begin from step 5

10. Else read the data base to check sufficient amount in the person's account
11. If the person account is not sufficient
12. Then the system displays account insufficient message and begin from step 5
13. Else discount the person's account and display the success message
14. Finally the sequence ends

6.3.3. Activity diagram

Activity diagram displays a special state diagram where most of the states are action states and most of the transitions are triggered by completion of the actions in the source states. This diagram provide a task-centric view of the behavior of an object. They can be used, for example, to describe sequencing constraints among use cases, sequential activities among a group of objects, or the tasks of a project.

Money deposite activity diagram diagram

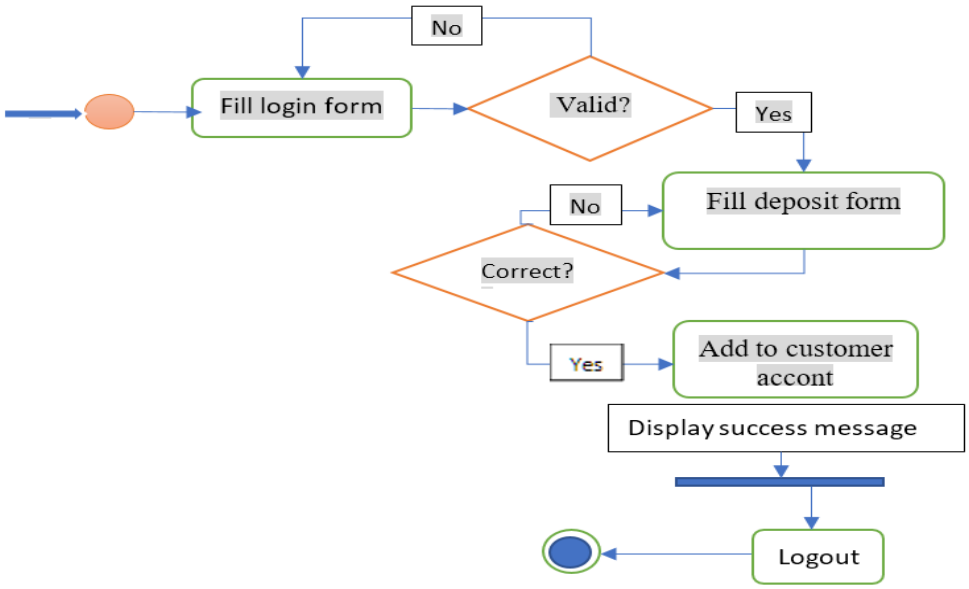

FIGURE 6-6: MONEY DEPOSIT ACTIVITY DIAGRAM

Description of Money deposit activity diagram: Money deposit activity include login. To make a system secure before processing deposit transaction the user must be authenticated and authorized. If the branch officer enters the correct user credential then they can access the deposit form. If any error made by the user during deposit the system notifies the kind of error and enable them to re-enter the deposit information.

Money transfer activity diagram diagram

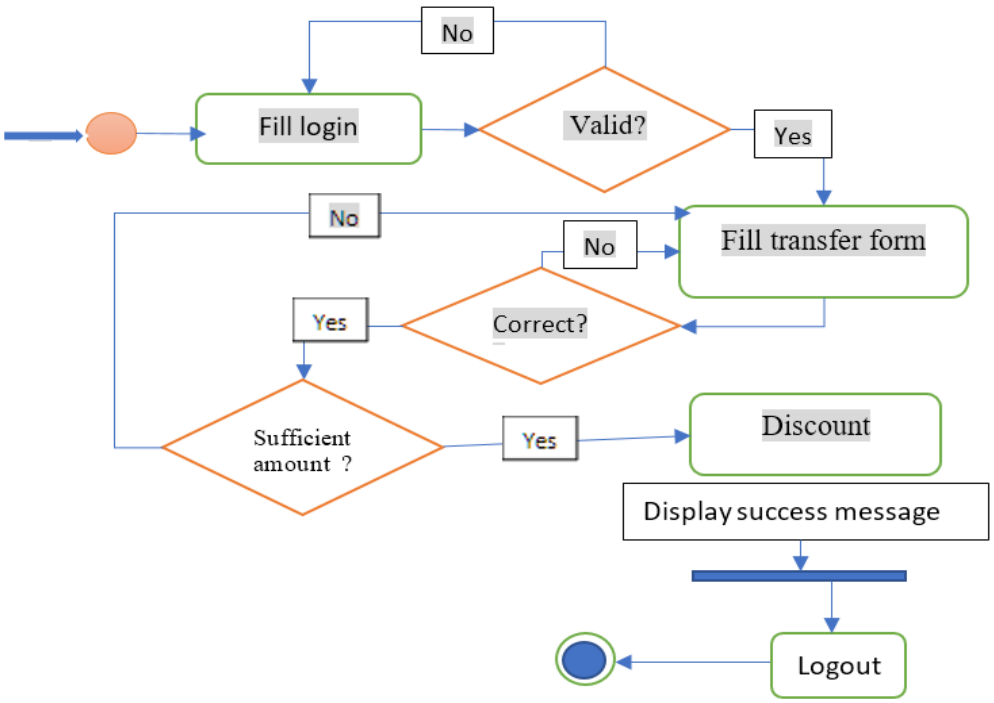

FIGURE 6-7 : MONEY TRANSFER ACTIVITY

Description of money transfer activity diagram: Money transfer activity also includes login. To make a system secure before processing deposit transaction the user must be authenticated and authorized. If the branch officer or customer enters the correct user credential then they can access the transfer form. If any error made by the user during the transfer system notifies the kind of error and enable them to re-enter the transfer information. This activity enables the user to transfer the money from their account and their respective bank to any banks branch and to any ones account who is already a member of EBICC. This activity also works in account to account transfer, local transfer or cash to account transfer.

Customer registration

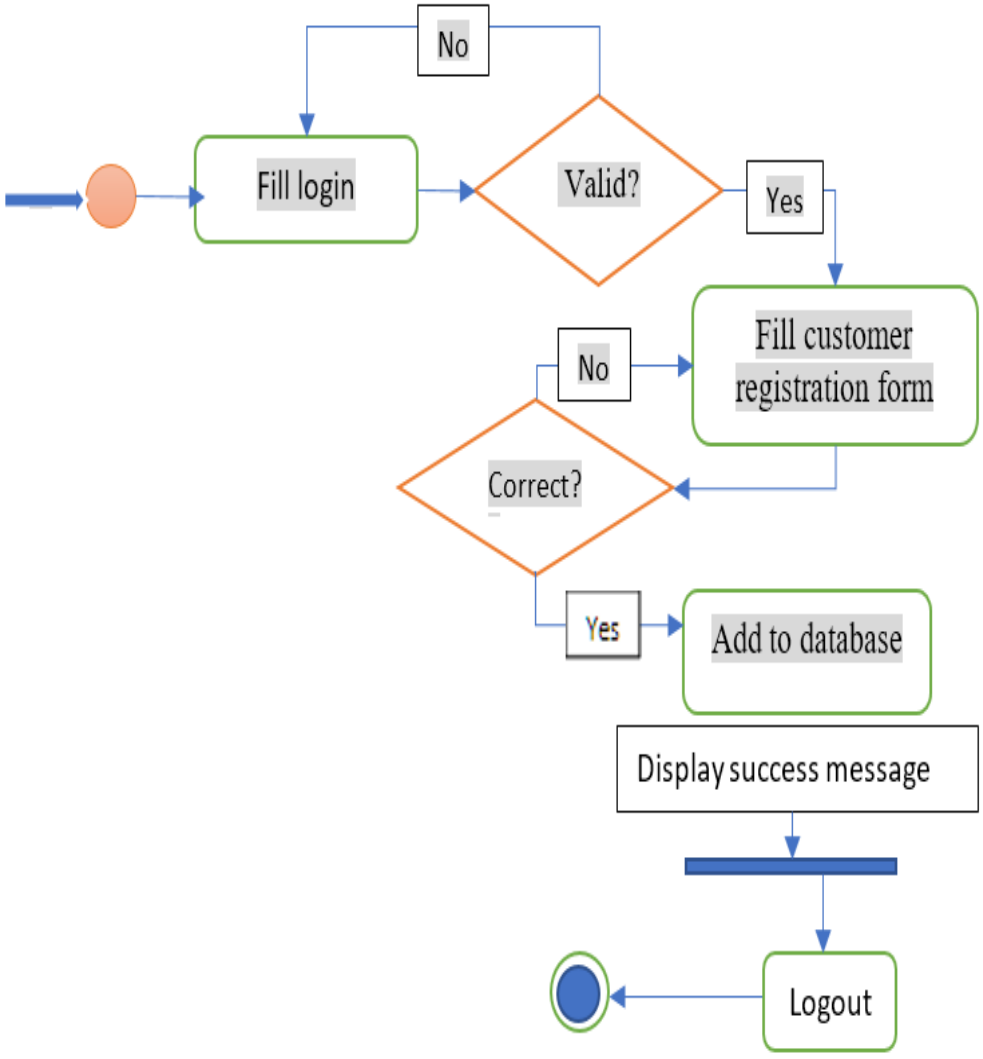

FIGURE 6-8 : CUSTOMER REGISTRATION ACTIVITY DIAGRAM

Description of Customer registration activity diagram: Once the branch officer log in them can initiate the activity of customer management in this case they can register customer, update, and view operation on customer. The customer account can be any times but a customer should be only one time in a certain bank. If the branch officer fills the customer profile correctly the user will be registered in their organization and the customers also can create a login account for their own

6.4. The screen snapshot of the prototype

In this section, some of the sample application interfaces snapshots have been included.

6.4.2. The signUp Page

When the banking organization decided to use the EBICC the system administrator signed up and the EBICC allocate a virtual machine which is installed with a banking system software and other management software. When the At the time of sign up the admin selects his/her role as “administrator” the system displays additional text boxes to fill the required amount of processor, memory (RAM) and storage. Once the administrator signed up for the cloud service he/she creates a user for their organization such as the management staff and the branch officers and performs other administrative functions.

When a customer of a bank wants to sign up to the EBICC first they select their role as a “customer ” then the system displays one additional textbox to fill their account number and based on that account number the system checks weather they are a customer of the member bank of EBICC.

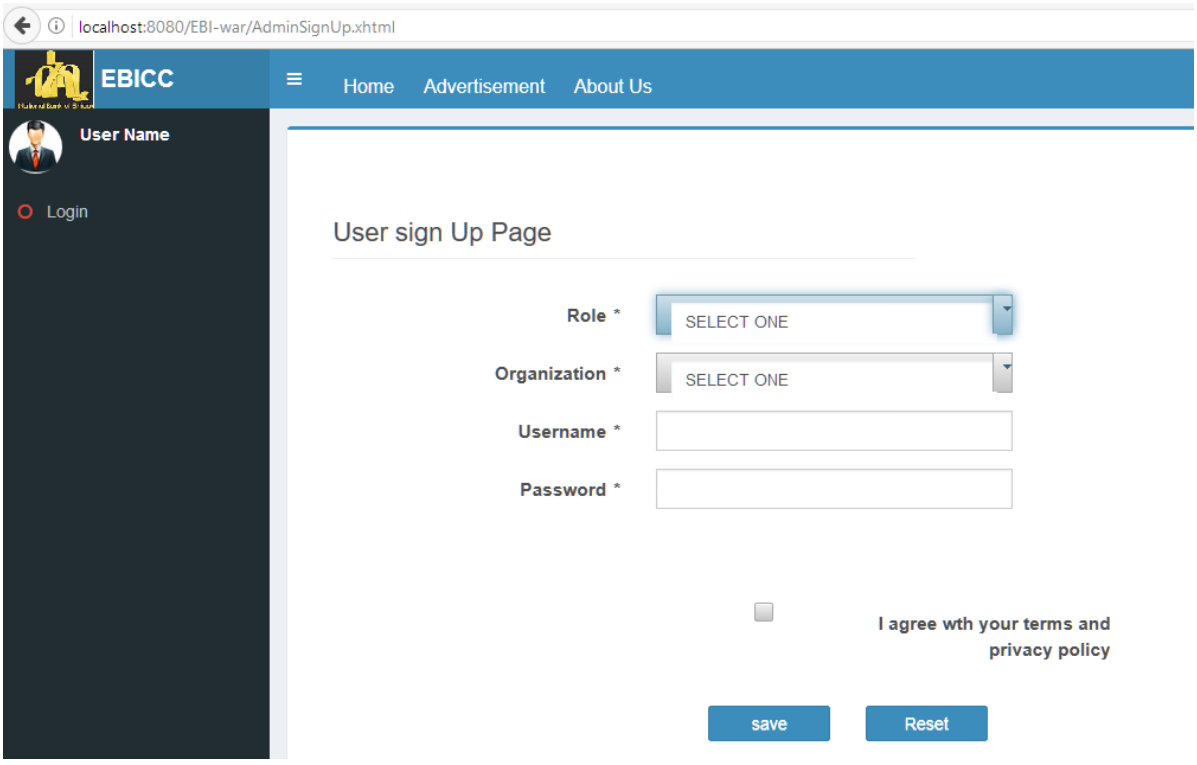

FIGURE 6-9 THE USER SIGNUP PAGE

6.4.3. Login Page

Login page enables to authenticate and authorizes the user based on their role,organization , user name and password. In this page, the user who fill the correct user name and password can log into their respective page and perform a banking transaction.

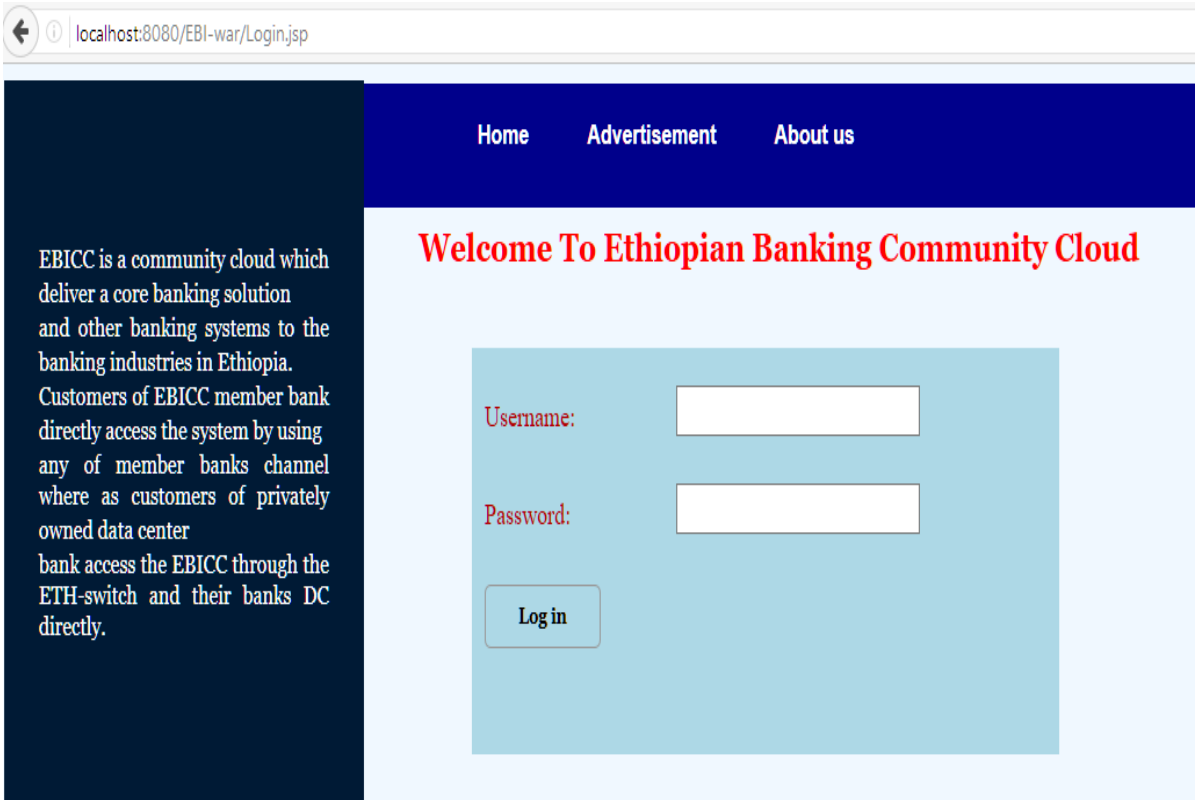

FIGURE 6-10 : LOGIN PAGE

6.4.4. CBE Admin DashBoard

When the user of the bank were identified as a CBE system administrator, then the CBE administrator dashboard will be displayed. This dashboard contains the metering, backup, service level agreement, configuration, Branch management and user management services.

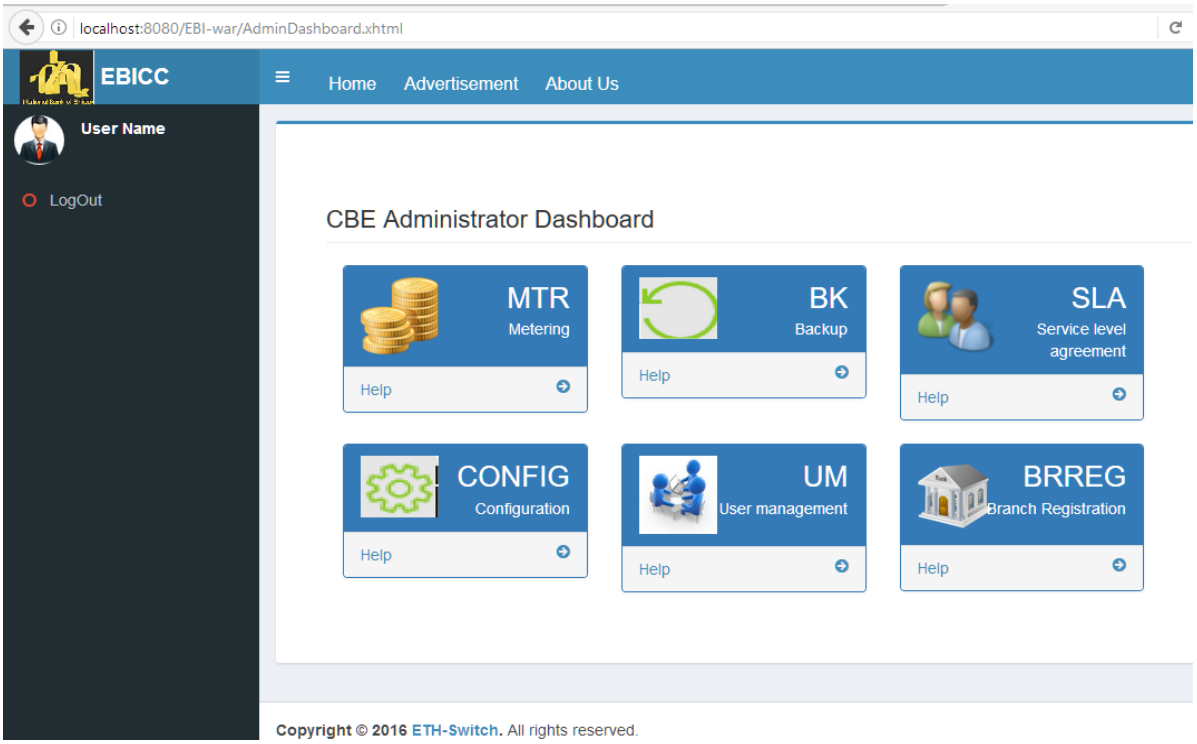

FIGURE 6-11 : ADMINISTRATOR DASHBOARD

When the admin chooses the user management module, then they can create, view, update, search the user information.

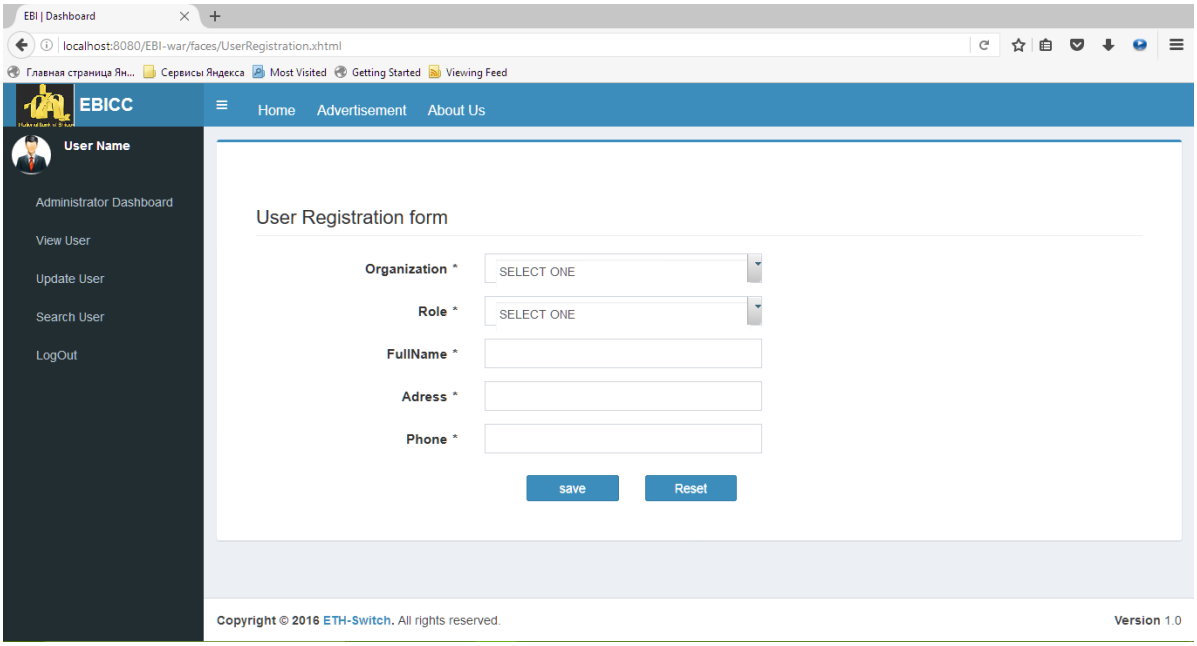

FIGURE 6-12: USER REGISTRATION

6.4.5. Management Dashboard

When the user were identified as a management staff, then the system displays the management page and they can perform the activities related to HRMS, FMS, CRM, case management reporting and messaging according to their role.

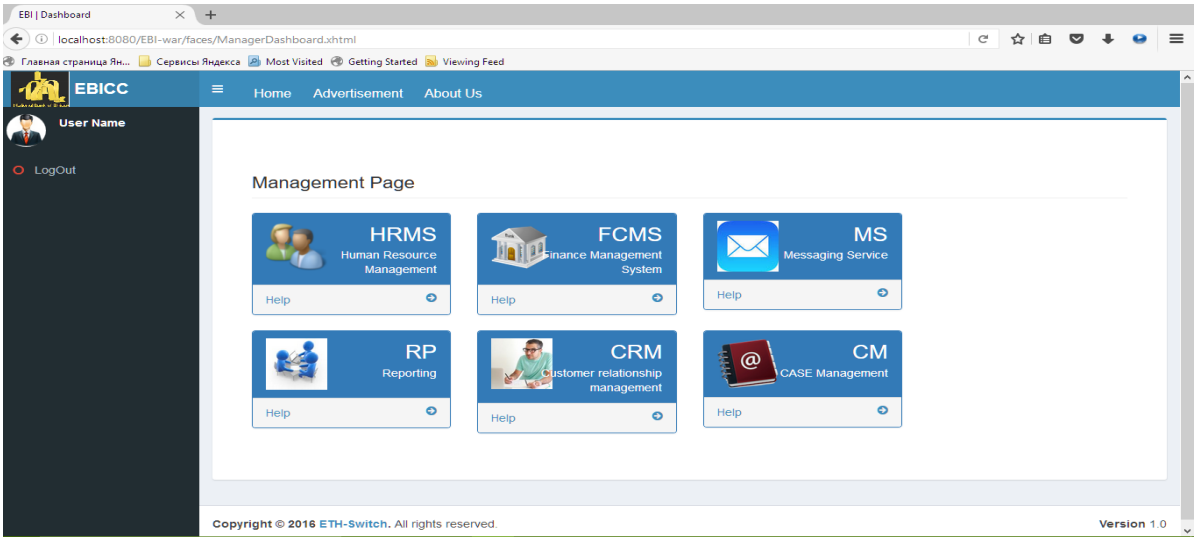

FIGURE 6-13: THE MANAGEMENT DASHBOARD

6.4.6. BIB Branch Officer Dashboard

When the user was identified as a BIB branch officer , then the system displays "BIB branch officer dashboard". The BIB branch officer dashboard contains the link of core banking system, customer registration, auditing, reporting, cheque processing and messaging services.

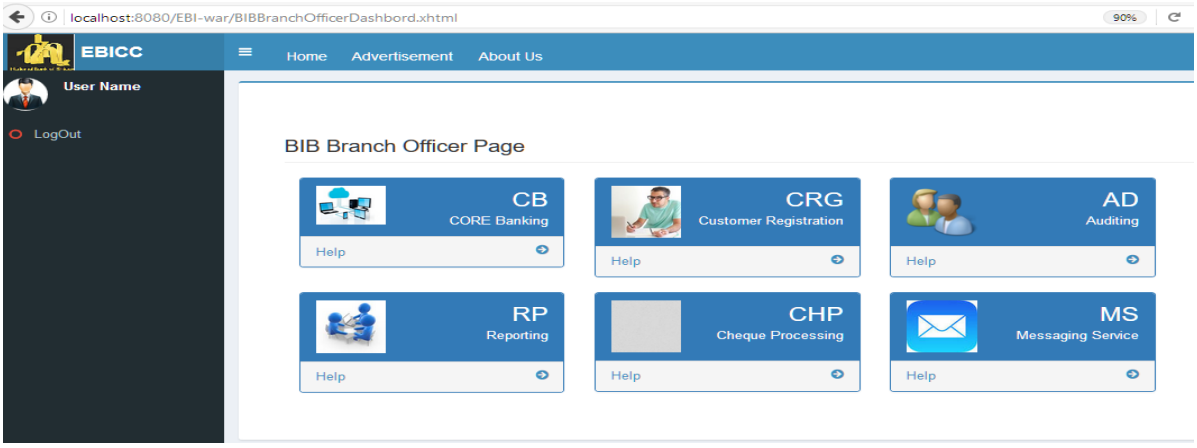

FIGURE 6.15: BIB BRANCH OFFICER DASHBOARD

When the branch officer clicks on the link of customer registration the page shown as below will be displayed. This form enables to register a customer only once at a time but allow to register customer account more than once.

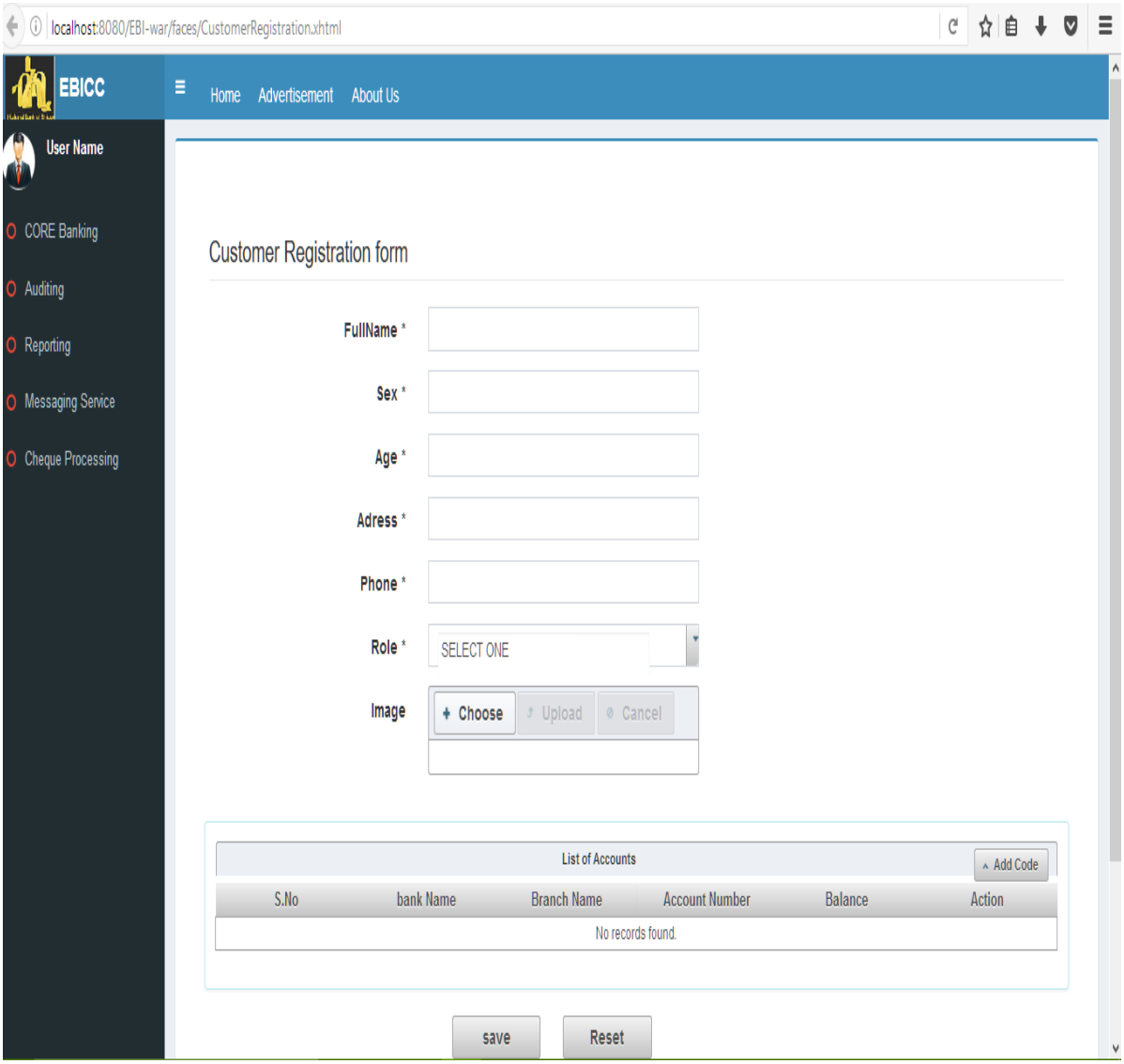

FIGURE 6-14: THE CUSTOMER REGISTRATION PAGE

6.4.7. ATM Customer Service

When the user of the bank identified as a customer, then he/she can access a service based on the device they used. The form in figure 6.17 shows that the ATM user services.

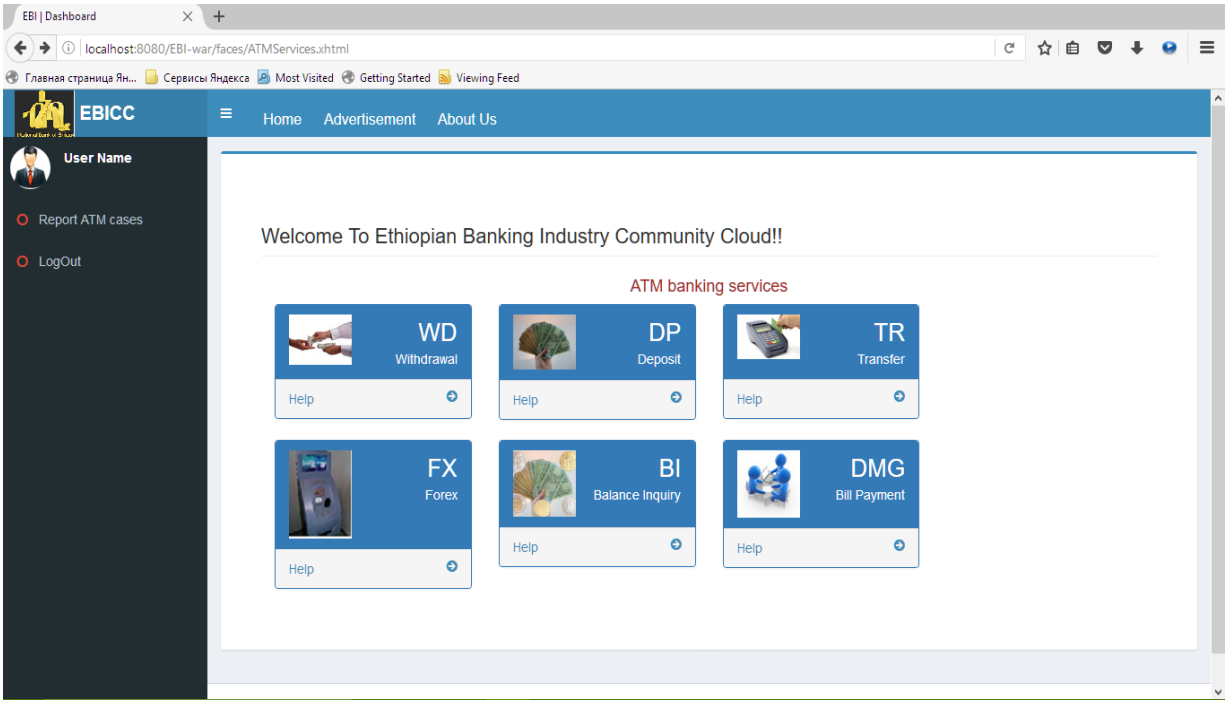

FIGURE 6-15: ATM SERVICES

6.4.8. Money Transfer

A customer of a bank or branch officer can perform one of a money transfer transaction i.e, account to account transfer and local transfer by using the following page.

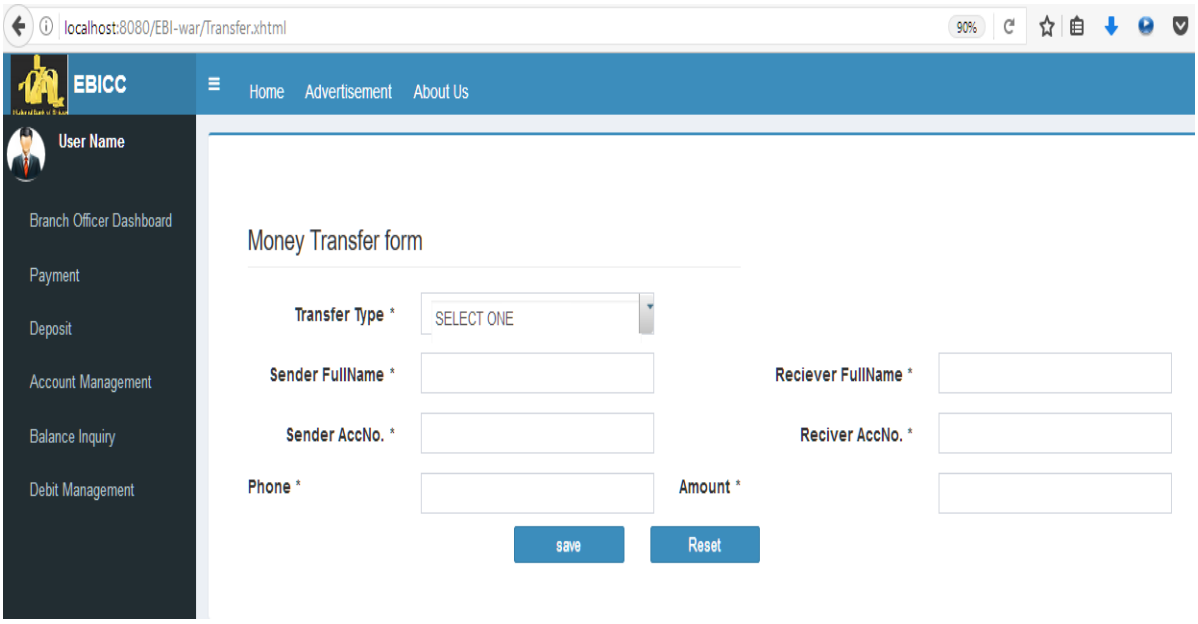

FIGURE 6-16: MONEY TRANSFER

6.5. Validation of Results

To clearly define the problem in Ethiopian banking industry we have followed the scientific research method and we have prepared the interview question which enables to understand the banking environment. The interview includes the best respondent of IT workers having a clear idea about the current working process. The result obtained from the study shows that no single bank in Ethiopia can afford enough access channel to serve their rural to urban customers. This result is agreed with previous researchers' result [27]. This result also supported by the current attempt made by banking industries to integrate their ATM and POS channel to afford enough service to their customers. Whereas this attempt also got performance problem related to redundant switching solution and separate banking data center. One can easily observe this performance problem by inserting their ATM card in other banks ATM machine and they perform a transaction with slow system performance or not working at all.

As a solution to the problem, we have proposed an interoperable cloud computing framework which removes those redundant switches and merging separate data centers which are the causes of performance degradation. The framework also enables the bank to use a banking channel owned by one bank can be used by another bank this interoperability of channel solves the economic limitation of the bank to afford enough access channel to their customer. As we have observed in current banking system software some of the banks use a different version of the same software. Hence, this proposed framework enables them to share a banking system software and other management system software accordingly they save the unnecessary cost of purchasing and maintain the software purchased from abroad and this also has a value in countries economy. The framework is also valid from the customer's point of view because it enables them to access their money on any device which enables to banking service on anytime on any banks channel just like accessing their pocket. The validity of the framework is also checked by developing the prototype.

The reflection from the bank administrators show that the framework and the prototype solves the current banking problem but they still doubt in security issue, the national bank enforcement policies and strategies and the network bandwidth issues. In this research we have achieved the initial goal hence it is valid in achieving the goal of the research.

CHAPTER SEVEN

7. CONCLUSION AND FUTURE WORKS

7.3. Conclusion

This research has been begun with the intent to seek a solution to the problem that we have observed from the environment. The problem that initiates to do this research is the unavailability of sufficient banking access channel in the bank that we have registered as a customer. We have inspired by this problem and followed a scientific research method by asking three research question which enables to understand how the current banking environments are operating and what other problems are there , and also what techniques and technologies are there to solve the existing problem.

Interview questions are designed in the way that enables to capture the real economic and potential capability of the bank to afford enough infrastructures to their customer and to secure their system and also to seek other problems in the banking industry. After formulating the problem in banking industry we have reviewed a number of journals and books to give a solution to the existing problem. When we have gone deeper into the study we could have seen other forces that make a banking industry in Ethiopia to seek another computing strategy. The increment of computer hazards, technological changes initiates customer to use their device, involvement of redundant switching systems, allotment of a foreign bank to open their branch in Ethiopia are some of the forces.

The result obtained from the study shows that no single bank in Ethiopia can afford enough access channel to serve their rural to urban customers. Due to this insufficiency, there are some attempts to integrate ATMs and POS machines owned by a different organization like ETH-Switch, Q-Link and PSS. In these switching system, we can observe that unnecessary duplication of efforts because ETH-Switch is already incorporate all banks ATMS other switches create additional routing of packets accordingly performance problem occurred. Even if attempts to integrate the banking channels we could observe that there are also individual runs among the banks as mentioned in the result and discussion section CBE as an example. Competition to get technological advantage one bank over the other is wastage of county's economy because as developing country we buy these technologies from

developed countries with an expensive cost that also creates a dollar shortage with our country.

Hence including all the above issues into account and even those integration attempts are still not cost saving to the bank and could not create a satisfied customer. One thing we have also observed that all the banking transaction processing forms used by different banks are almost similar and even some of the banks use a different version of the same software. This similarity indicates that banking industries are the candidate for using a shared service like banking system software and management software.

In this thesis, we have proposed an interoperable cloud computing framework which uses a community cloud deployment model to make access channels owned by different banks interoperable and which shows how to use a single instance of software is shared by all banks. We have basically considering the two layers of interoperability the organizational which is using shared information and the semantic interoperability which is based on the meaning of the shared information. Community cloud deployment model has been chosen because of its suitability for an organization having similar business agenda and security requirements. Additionally community cloud deployment model has cost advantage over the private cloud and security advantage over the public cloud. SaaS service delivery model which is installed on the virtual infrastructure is chosen.

The prototype of the framework has been developed using JEE platform which is powerful a multi-tier, scalable and security-rich platform. To implement the framework we have used a separate database model by considering security.

As many design research, the clear contribution of this thesis is a new framework which not ever been developed by other researcher and that enables to interoperate a banking channels and to share a banking software. We have presented the existing banking environment both in architecture and in the textual description. We have also shown that how to implement the SaaS-based system to the banking industry.

7.4. Future Works

In this thesis the problem raised and the question asked has been answered and the framework which enables to achieve interoperability has been designed. We forward some research ideas as a future work to make the framework more reliable.

- ✓ Since cloud computing needs fast and secure network connection; hence workability of the framework on current Ethiopian internet connection should be tested.
- ✓ Since some researcher suggests a hybrid cloud deployment model to the banking industry; hence a comparative study between a community cloud and a hybrid cloud is required by deploying both frameworks in test bed environment.
- ✓ Banks are under control by NBE, therefore when the banks implement anything like our solutions, they need strategies, policy, and procedure set by supervisory. Hence research based on national strategies, policies, and procedures will be required to convince NBE and banking industries.

References

1. Buyya, R., Bromberg, J. , & Goscinsiki, A .(Ed). (2011). Cloud computing principles and paradigms. Hoboken, New Jersey: *John Willey & Sons, Inc*.
2. National Institute of Standards and Technology. (2011).The NIST Definition of Cloud Computing. Special Publication 800-145: *U.S. Department of Commerce*
3. Buyya, R. ,Yeo, C. S. , Venugopal, S. , Broberg , J. and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems*.
4. Armbrust, M. , Fox, A., Griffith, R. ,Joseph, A. , Katz, R. , Konwinski, A. , Lee, G. , Patterson, D. , Rabkin, A. , Stoica, I. , Zaharia, M. Above the Clouds: A View of Cloud Computing. *Communications of the ACM* , vol. 53 . April 2010.
5. Chandra, M. S. , Mondal , A. (2011). Identification of a company's suitability for the adoption of cloud computing and modeling its corresponding Return on Investment: *Department of Industrial and Management Engineering, Indian Institute of Technology*, Kanpur, India
6. Chaudhry, S. An Overview On Current Trends, Technologies And Future Scope Of cloud computing: *International Journal of Scientific & Engineering Research*, 4(8), August-20
7. Sosinsky, B. (2011). Cloud Computing Bible. Indianapolis, Indiana: *Wiley Publishing, Inc.*
8. Kabbedijk, J., Bezemer, C., Jansena, S., & Zaidman, A. Defining Multi-Tenancy: A Systematic Mapping Study on the Academic and the Industrial Perspective. *Journal of Software and Systems.* October 16, 2014.
9. Md. Tanzim Khorshed, A.B.M. Shawkat Ali, and Saleh, A. Wasimi. (2012). "A Survey on gaps, threat remediation challenges and some thoughts for proactive attack detection in cloud computing*," Future Generation Computer Systems*.
10. Saraswathi, M. and Bhuvaneswari, T. Multitenancy in Cloud Software as a Service Application: *International Journal of Advanced Research in Computer Science and Software Engineering, 3(11),* November 2013.
11. AlJahdali, H., Albatli, A., Garraghan, p., Townend, p., Lau, l., & Xu, j. (2014). Multi-Tenancy in Cloud Computing. Leeds, United Kingdom*: White rose.*

12. Petersson, J. (2011, July 5). Best practices for cloud computing multi-tenancy: Building a scalable, available cloud computing multi-tenancy architecture. Retrieved on April 18, 2016, from, *http://www.ibm.com/developerworks/cloud*.
13. Kajeepeta, S. (2010, April 12). Multi-tenancy in the cloud: Why it matters. Retrieved on April 18, 2016, from, *http://www.computerworld.com/article/2517005*.
14. Sareen, P. Cloud Computing: Types, architecture, applications, concerns, virtualization and role of IT governance in the cloud. *International Journal of Advanced Research in Computer Science and Software Engineering, 3(3), March 2013.*
15. Goyal, S. Public vs private vs hybrid vs community - cloud computing: A critical review. *I.J. computer network and information security, 3, 20-29, February 2014.*
16. Suresh, M. C. Cloud computing strategic considerations for banking and financial services institutions. *TATA consultancy service, Retrieved on April 18, 2016, from http://www.tcs.com/rss_feeds* .
17. Meskerem, A. and Abrehet, O. Cloud computing conceptual security framework for the banking industry. *Journal of Emerging Trends in Computing and Information sciences, 5(12), December 2014.*
18. Wikipedia,free encyclopedia, https://en.wikipedia.org Retrieved on May 7, 2016.
19. Cloud Security Alliance (CSA). Security guidance for critical areas of focus in cloud computing V2.1. December 2009
20. The Indian banking community cloud. http://www.infoq.com/articles/indian-banking-community-cloud, R*etrieved on January 5, 2016.*
21. Chen, X. W. Find provisioning performance bottlenecks in the cloud. *IBM, developerWorks, September 7, 2012.*
22. Gorelik, E. (2013, January). Cloud computing models: Comparison of cloud computing service and deployment models. Working paper, *Massachusetts Institute of Technology*
23. The National Bank of Ethiopia (NBE). *http://nbe.gov.et/aboutus/index.html*, *Retrieved on May 20, 2016*

24. Alemayehu, G. (2004). The structure and Performance of Ethiopia's financial sector in the pre and post-Reform period: With a special focus on banking. *Retrieved on May 20, 2016, from, http://alemayehu.com/On%20Going%20Research/*
25. Deville, W. H. (2014). Bank cloud adoption for financial institutions: Best practices and advantages of moving your financial institution to the cloud. *Retrieved on May 20, 2016, from www.dh.com*
26. RightScale 2016 state *of the cloud report.* Hybrid Cloud Adoption Ramps as Cloud Users and Cloud Providers Mature. *Retrieved on May 25, 2016 from www.rightscale.com*
27. Alemayehu, A. (June 2014). A cloud computing framework for Ethiopian banking industry. *Addis Ababa University,Ethiopia*.
28. Garg, A. (2011). Cloud computing for the financial services industry. *Sapient Global Markets*
29. Zemene bank (2015). Annual report 2015. Addis Ababa, Ethiopia. *Retrieved on July 30, 2016, from http://www.zemenbank.com*
30. National bank of Ethiopia (2013). Licensing and supervision of the business of financial institutions regulation of mobile and agent banking services directives no. fis /01/2012. Addis Ababa, Ethiopia: *Author*.
31. Sewagegnehu, D. Powers of agent banking. *Fortune, 16(811), Nov 16, 2015.*
32. Zuo Z. J2EE architecture and pattern in an enterprise system. *University of Tampere, Finland*, May 2004
33. Tutorialspoint (2015). Enterprise Java Bean (EJB). *Retrieved on December 20, 2016, from http://www.tutorialspoint.com*
34. Lewis, G.A. The Role of Standards in Cloud-Computing Interoperability. *Software engineering institute, Carnegie Mellon University*, October 2012.
35. Cloud standards customer council (November 2014). Interoperability and Portability for cloud computing: A Guide. Retrieved from http://www.cloud-council.org/CSCC
36. Kubicek, H, Cimander, R. and Scholl, H.J.(2011). Organizational Interoperability in E-Government, Springer-Verlag Berlin Heidelberg.

37. Abebe, G. and Fantahun, A. Interoperable payment systems: Requirements driven architecture for Ethiopian banking sector. *HiLCoE journal of computer science and technology 1(1), December 2012.*
38. Tutorialspoint. Cloud computing tutorial. *Retrieved on January 18, 2017 from* *www.tutorialspoint.com/cloud_computing/*
39. Jendrock, E., Navarro, R. C., Evans, I., Haase, K. and Markito, W. (2014). Java platform enterprise edition. *Oracle.*
40. Siebenstich, E. Overview of implementation of Java EE in JBoss AS 7. Comenius *University, Bratislava, Jun 2013.*

Appendix

A. Interview Questions

These interview questions are designed to gather information about current ICT service and technology used by the banking industry of Ethiopia in order to identify problems and to design a better computing framework. All responses will be used to conduct a study on the thesis for the Partial Fulfillment of the Requirement for the Degree of Master of Science in software engineering. I will not be used your responses for any other purposes other than the intended purpose. Besides, this interview should only take about 10-20 minutes of your time. I am thankful for your cooperation in the interview questions:

1. Could you tell me the technologies and techniques used by your organization which facilitates the banking activities?
2. What kinds of software and database system do you use to manage these technologies?
3. Do you know the number of branch offices, ATM and POS machines that your organization has?
4. Do you think that these branch offices, ATM and POS machines are enough to achieve organizational mission and vision? Yes ☐ No ☐
5. Could you tell me how much of the customer uses ATM card and how much of the customer uses mobile banking? If possible express in percent.
6. Do you remember the common feedbacks which are raised by your customer in relation with the accessibility of their money?
7. Do you think that your customers are satisfied by your banks ICT service? Yes ☐ No ☐
8. Could you tell me how much of the annual budget of your bank is allocated for ICT? If it is possible to explain in percent.
9. What is the estimated amount of budget needed for ICT in opening a new branch office?
10. Could you estimate how much of money is needed to install an ATM machine?

11. What do you say about the security of your bank's system?

Very good ☐ Good ☐ Satisfactory ☐ Poor ☐

12. What do you say about the system security expert of your bank?

Very good and can protect any kind of attack ☐

Good and protect some Kind of attack ☐

Satisfactory and need to upgrade their skill of security ☐

Poor security experts and need to be replaced ☐

13. Could you mention the limitations of the current ICT delivery strategies of your bank?

14. What do you say about the current ICT service used by the bank need to be? Changed ☐ Modified ☐ As it is ☐

15. Is there any integrated service delivery model followed by your bank with other banks? Yes ☐ No ☐

16. If you say **yes** for question 13, which banking activities are integrated? And mention the names of the bank.

17. If you say yes for question 13, could you tell me how do you manage that integrated service?

18. Could you mention some other banking activities which are possible to be integrated?

19. Do you observe that any limitation in those integrated service delivery?

Yes ☐ No ☐

20. If you say **no** for question 13, could you mention some banking activities which are possible to be integrated to give ICT service in common?

21. As an ICT expert what will be your reaction, if a community cloud computing infrastructure is to be built in EBI that could be used by all banks in Ethiopia to provide ICT services?

Thank you!!!

B. The login View from EBI-war

```
<?xml version='1.0' encoding='UTF-8' ?>
<!DOCTYPE composition PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
  <ui:composition
   xmlns="http://www.w3.org/1999/xhtml"
   xmlns:ui="http://java.sun.com/jsf/facelets"
   xmlns:h="http://java.sun.com/jsf/html"
   template="resources/templates/template.xhtml"
   xmlns:p="http://primefaces.org/ui"
   xmlns:f="http://xmlns.jcp.org/jsf/core">
   <ui:define name="navigaion_content">
  <li><a href="AdminSignUp.xhtml"><i class="fa fa-circle-o text-red"></i> <span>Admin
Sign Up</span></a></li>
  <li><a href="CustomerSignUp.xhtml"><i class="fa fa-circle-o text-red"></i> <span>User
Sign Up</span></a></li>
     </ui:define>
   <ui:define name="contents">
       <div class="box box-primary">
         <div class="box-body">
            <div class="row">
               <div class="col-md-12">
                  <div class="box-body ">
                     <h:form class="form-horizontal" id="frmSystem">
                      <h:panelGroup style="border: 0px; background-color:transparent;"
id="search" >
    <p:growl id="growl" showDetail="true" autoUpdate="true"
severity="information,fatal"/>
                           </h:panelGroup>  </h:form>   </div>
               </div>   </div>    <div class="row">  <div class="col-md-12">
                  <div class="box-body ">
```

```
                <h:form id="mainPageForm">
            <h:panelGroup style="border: 0px; background-color: transparent" id="mainPage"
                                >
                            <div class="box-body">
        <p:growl id="growl" showDetail="true" autoUpdate="true" severity="information"/>
                                <div class="form-horizontal  col-sm-12">
                                   <div class="row">
                                      <div class="col-lg-12">
                                         <h2 class="page-header">Login Page</h2>
                                      </div>
                                      <!-- /.col-lg-12 -->
                                   </div>
                                 <div class="form-group"><p:outputLabel for="txtOrganization"
class="control-label col-sm-3" value="Organization"/>
                                      <div class="col-s<p:selectOneMenu style="width: 100%"
id="txtOrganization" class="form-control" required="true" >
                                 <f:selectItem itemValue="#{null}" itemLabel="SELECT ONE"/>
                                       <f:selectItem itemValue="CBE" itemLabel="CBE"/>
                                       <f:selectItem itemValue="BIB" itemLabel="BIB"/>
                                       <f:selectItem itemValue="UB" itemLabel="UB"/>
                                    </p:selectOneMenu>
        <p:message for="txtOrganization" class="col-sm-3 control-label" display="icon"/>
         </div>   </div>
                                <div class="form-group">
   <p:outputLabel for="txtUserName" class="control-label col-sm-3" value="User Name"/>
                                      <div class="col-sm-4">
     <p:inputText id="txtUserName" value="#{loginController.userSignup.username}"
title="UserName" class="form-control"
style="background-color: white; width: 100%;" required="true" />
 <p:message for="txtUserName" class="col-sm-3 control-label" display="icon"/>
</div>     </div>
```

```
                                    <div class="form-group">
      <p:outputLabel for="txtPassword" class="control-label col-sm-3" value="Password"/>
                                    <div class="col-sm-4">
                                       <p:password id="txtPassword"
value="#{loginController.userSignup.password}"
                                                title="Password" class="form-control"
      style="background-color: white; width: 100%;" required="true" />
<p:message for="txtPassword" class="col-sm-3 control-label"
display="icon"></p:message>    </div> </div>  </div>   </div>
                        </h:panelGroup>
                        <div class="form-group col-sm-10">
     <div class="form-horizontal col-sm-8" style="padding-bottom: 10px; right: -290px" >
                            <div class="col-sm-3"> <p:commandButton id="glbtnsave"
value="Ligin" class="btn  btn-block btn-primary col-lg-4"
                                    action="#{loginController.search()}"
                                    process="@this,:mainPageForm"
                                  update="@this,:mainPageForm,:frmSystem"  >
                                </p:commandButton>     </div>
                            <div class="col-sm-3">
   <p:commandButton value="Reset" class="btn  btn-block btn-primary col-lg-4"
    action="#{loginController.search()}"   id="reset"
process="@this" update="@this,:frmSystem,:mainPageForm," />
</div>   </div>
   <div class="col-lg-12"  <div style=" text-align: center; font-size: 16px; color: blue">Did
you forget password?</div>
            <a href="PasswordRecovery.xhtml" style="color: #fff;">Recovery </a>
                        </div> </div>   </h:form>
            </div> </div> </div> </div></div>
   </ui:define> </ui:composition>
```

C. The login controller

```
package controller;
import bussiness_logic.LoginSesionBeanLocal;
import entity.Admininfo;
import entity.UserSignup;
import java.io.Serializable;
import java.util.ArrayList;
import java.util.List;
import javax.ejb.EJB;
import javax.faces.view.ViewScoped;
import javax.inject.Inject;
import javax.inject.Named;
@Named
@ViewScoped
public class LoginController implements Serializable{
   @EJB
   private LoginSesionBeanLocal loginSesionBean;
   @Inject
   UserSignup userSignup;
   @Inject
   Admininfo admininfo;
   public UserSignup getUserSignup() {
      return userSignup;
   }
   public void setUserSignup(UserSignup userSignup) {
      this.userSignup = userSignup;
   }
   public Admininfo getAdmininfo() {
      return admininfo;
   }
   public void setAdmininfo(Admininfo admininfo) {
```

```
        this.admininfo = admininfo;
    }
    List<UserSignup>LserSignupList;
    public List<UserSignup> getLserSignupList() {
        return LserSignupList;
    }
    public void setLserSignupList(List<UserSignup> LserSignupList) {
        this.LserSignupList = LserSignupList;
    }
    String AdminData="CustomerSignUp.xhtml?faces-redirect=true";
          public String search(){
        if(userSignup.getUsername()!=null && userSignup.getRoleed()!=null &&
userSignup.getPassword()!=null ){
            if(userSignup.getRoleed().equals("Administrator")){
            userSignup= loginSesionBean.login(userSignup);
            return "AdminDashboard.xhtml?faces-redirect=true";
        }
        else if(userSignup.getRoleed().equals("BranchOfficer")){
            userSignup= loginSesionBean.login(userSignup);
            return "BranchOfficerDashbord.xhtml?faces-redirect=true";
        }
      else if(userSignup.getRoleed().equals("Manager")){
            userSignup= loginSesionBean.login(userSignup);
            return "ManagerDashboard.xhtml?faces-redirect=true";
        }
              else if(userSignup.getRoleed().equals("Customer")){
            userSignup= loginSesionBean.login(userSignup);
            return "ATMServices.xhtml?faces-redirect=true";
        }    }    return null;    }}
```

D. The Admin signup business logic from UserSignupFacade.java

```
package mapper;
import entity.Admininfo;
import entity.UserSignup;
import et.gov.eep.commonApplications.JsfUtil;
import java.util.ArrayList;
import javax.ejb.Stateless;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.Query;
@Stateless
public class UserSignupFacade extends AbstractFacade<UserSignup> {
   @PersistenceContext(unitName = "EBI-ejbPU")
   private EntityManager em;
   @Override
   protected EntityManager getEntityManager() {
      return em;     }
   public UserSignupFacade() {
      super(UserSignup.class);  }
      public UserSignup getInformationLoginData(UserSignup adminLogin) {
     //accessing e
      Query query = em.createNamedQuery("UserSignup.findByLogin");
     // query.setParameter("organization", adminLogin.getOrganization());
      query.setParameter("username", adminLogin.getUsername());
      query.setParameter("password", adminLogin.getPassword());
      query.setParameter("roleed", adminLogin.getRoleed());
           System.out.println("----------"+query.getResultList().size());
      try {
         UserSignup admin = (UserSignup)query.getResultList().get(0);
         if(query.getResultList().isEmpty()){
            JsfUtil.addErrorMessage("incorrect user name or pasword or role");
```

```
        }
        else if(query.getResultList().size()>0){
        return admin;       }
        return null;
    } catch (Exception ex) {
        ex.printStackTrace();
        return null;     }    }
 public ArrayList<UserSignup> loginData(UserSignup adminLogin) {
    Query query = em.createNamedQuery("UserSignup.findByLogin");
   // query.setParameter("organization", adminLogin.getOrganization());
    query.setParameter("username", adminLogin.getUsername());
    query.setParameter("password", adminLogin.getPassword());
    query.setParameter("roleed", adminLogin.getRoleed());
    try {
ArrayList<UserSignup> userSignupList = new ArrayList(query.getResultList());
            if(query.getResultList().isEmpty()){
           JsfUtil.addSuccessMessage("no record found ");
        }
        else if(query.getResultList().size()>0){
        return userSignupList;
        }
        return null;
    } catch (Exception ex) {
        ex.printStackTrace();
        return null;   }    } }
```