አዳማ ሳይንስ እና ቴክኖሎጂ ዩኒቨርሲቲ
Adama Science & Technology University

SCHOOL OF ELECTRICAL ENGINEERING & COMPUTING

DEPARTMENT OF COMPUTING

A MASTER'S THESIS

ON

Cloud Based Capacity Building Resource Sharing Architecture for Healthcare Institutions in Ethiopian

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE

OF

MASTER OF SCIENCE IN SOFTWARE ENGINEERING

BY:

YIMER AMEDIE

October, 2016

Adama, Ethiopia

DECLARATION

This thesis is my original work, and it has not been presented for a degree in any other university. All sources of materials used for the thesis have been acknowledged.

____

Yimer Amedie

This thesis has been submitted for examination with my approval as university advisor

____

Dr. Dereje Yohannes (Asst. Professor)

October, 2016

አዳማ ሳይንስ እና ቴክኖሎጂ ዩኒቨርሲቲ
Adama Science & Technology University

SCHOOL OF ELECTRICAL ENGINEERING & COMPUTING

DEPARTMENT OF COMPUTING

Cloud Based Capacity Building Resource Sharing Architecture for Healthcare Institutions in Ethiopian

By:

YIMER AMEDIE

Name and Signature of Members of the Examination Board

DEREJE YOHANNES (phD) Advisor	____________________ Signature
____________________ External examiner's name	____________________ Signature
____________________ Internal examiner's name	____________________ Signature
____________________ Chairperson's name	____________________ Signature

ACKNOWLEDGEMENT

First and above all, I praise God, the almighty for providing me this opportunity and granting me the capability to proceed successfully. This thesis appears in its current form due to the support and guidance of several people.

First I would like to express my sincere gratitude and heartfelt thanks to my advisor Dr. Dereje Yohannes (Asst. professor) for his dedicated help, advice, inspiration, encouragement and continuous support throughout my thesis work. Secondly, I would like to offer my sincere thanks to all of the other people supporting and guiding me with patience.

Lastly, I would like to thank my family for giving birth to me and supporting me spiritually throughout my life. Thank you all my families and friends who support me in different ways.

Abstract

Capacity building program in the HDIs plays a vital role to increase the capacity of the workers and to reduce health inequality by creating awareness and ability. But, there is no means to share and access these CBP resources because the resources are dispersed in different HDIs and is not electronically available. So Practitioners do not consistently use a single site to share their knowledge and resources. Resource sharing is a mode of operation which consists of various collaborative activities between resource centers and other concerned bodies. Its purpose is sharing knowledge and experience very quickly and timely. The introduction of cloud computing has opened up fresh opportunity for better collaboration and sharing. So currently, there are many promising improvements such as collaborative methods to electronic repositories, electronic interactions, establishing content consortiums and other practical experiences for resource sharing purpose.

This study proposes community cloud platform architecture for sharing of capacity building resources in the healthcare institutions. The CCBRSA architecture adapts the CCRA v.2 to exploit all considerations and features of cloud based software architecture. The CCBRSA architecture facilitates fast access to capacity building resources and best practices with low cost and high availability. It teaches or instructs the health workers about the health system technologies in depth. Also some of cloud computing freely available platforms and tools were analyzed and compared. The prototype was implemented using OpenShift online Redhat's platform. The prototype is very interactive and user friendly which have resource sharing and accessing portal. Hence, the CBRS system is used as a common alliance and electronic means of resource sharing, knowledge sharing, collaboration and learning center for HDIs. It reduces health inequality by improving the ability of health workers, because the workers use CBRS system as a tool for self-learning and for knowledge and experience sharing.

Key Words: Capacity building, Cloud computing, Resource sharing, Software architecture

Table of Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS AND ACRONYMS

AHMC: Adama Hospital Medical College

API: application programming interface

ASP: application service provision

AWS: Amazon Web Service

BPaaS: business process as a service

CB: capacity building

CBP: capacity building program

CBRaaS: capacity building resources as a service

CBRS: capacity building resource sharing system

CCBRSA: cloud based capacity building resource sharing architecture

CCRA: cloud computing reference architecture

CLI: command line interface

CSP: Cloud Service Provider

CSS: cascading style sheet

DNS: domain name system

EMR: electronic medical record

FAQ: frequently asked questions

GUI: graphical user interface

HDI: healthcare delivery institution

HIT: healthcare information technology

HTML: hypertext markup language

HTTP: hypertext transfer protocol

IaaS: infrastructure as a service

ICT: information and communication technology

IDE: integrated development environment

IT: information technology

JSON: java script object notation

NIST: National institute of standards and Technology

PaaS: platform as a service

REST: Representational State Transfer

ROI: return on investment

SaaS: Software as a service

SDK: software development kit

SOAP: Simple Object Access Protocol

SSH: secure shell

URL: uniform resource locator

WSDL: web service description language

XML: Extensible Markup Language

Chapter One: Introduction and Background

1.1 Introduction

Today, advancement of technologies has improved the concept of resources sharing. This is relevant in electronic means of sharing documents, services, manpower and tools with low cost and high availability fashion. According to [1], the 21st century Healthcare Information Technology (HIT) has created the ability to electronically stores, maintains, and moves data across the world in a matter of seconds and has the potential to provide healthcare with enormous increasing productivity and quality of services. It provides a way of using information and communication technology (ICT) to share resources and to deliver quality healthcare services. Thus the integration of ICT with healthcare will help any country to share resources and to deliver quality health care services to all citizens. It provides efficient and effective patient information handling and sharing of resources at the right time among medical professionals for clinical decision making and initiative.

Currently web service and cloud computing are the promising technologies to access and share information and resources ubiquitously. Cloud computing has recently been applied to almost every industry from insurance to travel and hospital services. Many researchers from computer science and related fields have studied its application in the healthcare system for saving life by automatically sending emergency health information to the ambulance [4], for remotely monitoring the patient; for using telemedicine technologies like EHR and others that are deployed and installed in one hospital [2]. On the other hand, web service technologies and concepts are at the heart of cloud computing to make its application more accessible, flexible and easy to use for the health industries to share resources and to deliver high quality health services equally for all citizens in the country. But any of the researchers didn’t touch about the use and application of web service and cloud computing technologies for sharing capacity building resources, and for making it available electronically. Capacity building [13] involves assisting people to gain the knowledge and experience that is needed to solve problems, implement change, build effective actions and reach sustainability. So it goes beyond simply training or providing technical assistance. However, even if it is advantageous and have great value to reduce health inequality, to improve the knowledge and technology adaption of the healthcare givers and to build broad clinical skills and experience; projects for capacity building would demand a narrow focus because of the high cost to provide the CBP, but health workers needs have been broader. Also the health workers are not aware of how to use tele-medicine applications because of unavailability of training resources. Also CB tools, resources and approaches are not available electronically [12]. Since cloud computing [5], which is the use of computing resources (hardware and software) that are delivered as a service over the internet, provides an attractive IT platform to reduce the cost of capacity building (CB), it is the promising solution to address this broader need of health workers and to solve other capacity building problems and issues by facilitating sharing of resources. So it enables the health workers to access capacity building resources from anywhere, every time by using their mobile phone, laptops or other electronic devices and it will used to create modern alliance or self-learning environment.

Our study sought to develop cloud based capacity building resource sharing architecture for the healthcare institutions to create better, effective, interactive and collaborative resource sharing center for the health workers in order they develop ability, problem solving skill, knowledge and experience sharing to deliver quality health services.

Generally, by using web service and cloud computing technologies, the whole capacity building resources have been stored in one place in the cloud and available for authorized users ubiquitously through the use of broad network access. The resources are provided and controlled by one HDI or CB provider and the other healthcare institution’s health workers get access to these resources from this HDI which reduces the cost, saves time, accessed in easy and smart way for the health institutions because they would not provide/prepare the resources by themselves. Hence this thesis elaborates the application of web service and cloud computing technology to share capacity building resources in the healthcare area; develops cloud based capacity building resource sharing architecture using web service concepts that supports sharing of capacity building resources; saves resource preparation cost and time; makes the resources more deliverable, accessible and effective, increases the use of cloud-health application and reduces health inequality.

1.2 Statement of the problem

There are considerable health inequalities in Ethiopia and the healthcare service needs to be improved. For this reason capacity building is identified as a key step to promote health equality. This is because of capacity building to improve health is an important element for the health organizations to make the health practitioners and managers more capable and responsible [3]. But most of healthcare institutions have no potential to provide capacity building by their own due to technical and economic problems. That's the high cost for the capacity building program and the required personnel and tools to share and access these capacity building program resources. The capacity building program demands a narrow focus; however the health workers needs are broader in the health institutions. Many healthcare systems in developing countries like Ethiopia do not have the capacity to allocate their financial resources equitably and manage their technical resources optimally due to lack of well-skilled managers and professionals [17]. Furthermore, the health workforce is unequally distributed leading to skill mix problems [6, 7]. Also the usages of tele-medicine applications are low because of refusal of the health practitioners. This is due to unavailability of training resources on medical technologies.

Although the capacity building tools and resources are available in different organizations [12], the problem is, there is no means to access and use them because some organizations restrict dissemination of their tools and other organizations have not published these tools. Even electronically available, capacity building resources are not readily and widely available to many users because existing virtual or physical libraries are not comprehensive. Practitioners do not consistently use a single site to share their materials, successes, and challenges with their peers around the world because there is no noteworthy site. Hence, a knowledge exchange network that creates partnership and collaboration among medical practitioners is clearly needed.

Therefore, this thesis would focus with the development of cloud based capacity building resource sharing architecture that provides sharing of capacity building resources to the health workers in the healthcare institutions which would be cost effective, accessible and deliverable. Also it makes the capacity building resources and tools available electronically; it creates common place or site in which health practitioners can share their materials, knowledge and experience.

1.3 Purpose of the Study

The basic purpose of cloud based capacity building resource sharing architecture is to create healthcare learning environment and collaborations for health Practitioners at remote and district areas and specialists in urban hospitals to build capacity of the health workers by providing consecutive sharing of capacity building resources and experience. This will ultimately results in improvement of health practitioner's potential to deliver quality health service and help to reach rural areas. The design also used to create opportunities for international capacity building agencies to share free capacity building resources hence they can be able to join the system to support developing countries and also it is used to interface with the substantial number of Ethiopian and non-Ethiopian health professionals living outside the country to share their experience and knowledge.

In addition to developing cloud based resource sharing architecture and its prototype, we have tried to address the following issues:

- Conducted a survey to demonstrate potential uses of cloud computing technologies in the healthcare system of Ethiopia especially for sharing capacity building resources.
- Argued and demonstrated that capacity building resources can be shared and delivered using cloud computing paradigm and web service technologies. It also delivered using the existing ICT infrastructures and the future generation of mobile technologies in Ethiopia.

1.4 Objectives of the study

The objectives of the research study were:

- ✓ To conduct the survey of potential uses of cloud computing technologies in the healthcare system of Ethiopia especially for sharing of capacity building resources.
- ✓ To develop cloud based capacity building resource sharing architecture that enables the delivery or sharing of resources from one general hospital to the district healthcare institutions.
- ✓ To argue and demonstrate that capacity building resource sharing can be delivered and accessed using cloud computing paradigm; the existing ICT infrastructures and the future generation of mobile technologies in Ethiopia.

- ✓ To show that sharing of capacity building resources can be integrated into the E-Health like EHR to deliver quality healthcare service by creating healthcare learning system together with health information sharing environment.
- ✓ To develop prototype of the proposed architecture

1.5 Research Questions

This research intends to answer the following questions:

- Would it possible to design cost effective but state of the art architecture for cloud based capacity building resource sharing?
- Would it possible to integrate sharing of capacity building resources into cloud base healthcare system?
- Which mobile technology and existing ICT infrastructure can be used to deliver and access cloud based capacity building resources?
- How sharing of cloud based capacity building resources reduces health inequality and increases awareness?

1.6 Assumptions, Scope and Limitation of the study

In this study we are assuming that capacity building resource sharing can be implemented and delivered using cloud computing technologies by the use of the existing ICT infrastructure and the emerging mobile phones like 5G. We also assuming that the healthcare workers will found the design of cloud based capacity building resource sharing architecture useful to them, and there will be significant difference in cost, availability and worker's ability by the introduction of this cloud based capacity building resource sharing architecture. This study would have the following scope and limitations

- ✓ Capacity building resource sharing architecture based on cloud computing have been developed which is only for the health service delivery organizations and it would not have full implementation. The full implementation of the architecture requires more time, funds or money and additional human resource in order to be workable in the real world.
- ✓ The architecture would not intend to support different language for the online articles and commenting purpose. So this study would focus on the development of

the architecture of cloud based capacity building resource sharing that will support only English language. This is because of it needs translators if the online articles and comments are written by different language which is big project by its own.

- ✓ If multiple hospitals provide the capacity building, the architecture would not consider the integration of these capacity building providers which will be headache to manage for the cloud service providers. But they can share their materials in this common place to make it accessible by other institutions.

1.7 Significance of the study

It is envisaged that the results of this thesis would be used to implement an efficient, cost effective and deliverable capacity building resource sharing system in the health area, which would help to improve the ability of the health workers and balance proper health care delivery by reducing health inequality in the country. The main beneficiaries of the results will be the health practitioners and the public society as a whole. Other beneficiaries include: Medical institutions in general, Researchers, capacity building providers, fundraising organizations, ICT Organizations etc. The main beneficiaries and specific benefits of the result is listed below

Benefits for Medical institutions:

- ✓ The healthcare institution has access to capacity building program over time ubiquitously.
- ✓ It will enable parallel access to both the capacity building resources and telemedicine technologies which is new in the area of healthcare providers.
- ✓ The healthcare institutions will receive the capacity building resources provided by one general hospital or capacity building providers as a service from cloud service providers.
- ✓ It will used for establishing multi-sector partnership by contacting other organization through the system without going to the organization directly whether the organization is private or governmental and local or foreign. Also it will used to find and contact funding organization easily because the architecture of the system will have features that include address of funding and insurance organizations.
- ✓ It will used as initiatives to develop and implement IT functions and to provide supportive supervision to the staff throughout the organizations.

- ✓ It makes the resource sharing cost effective, interactive and deliverable for the healthcare institutions and it leads to sustainable ability to solve problems, to implement changes and to build effective action to reduce health inequalities.

Benefits for Health workers:

- ✓ Enables mobilization of training resources. The health practitioners in the healthcare institutions will access the training materials anytime anywhere using their mobile phone (5G), PDAs, PCs and other electronic devices.
- ✓ Used to create healthcare learning environment in which the health practitioners in the world shares their experience and knowledge for others in the health area and for the community.
- ✓ It provides multiple mode of training with the help of current technology.

Benefits for Fundraising organizations:

- ✓ It creates the opportunities for international healthcare capacity building agencies to develop electronic or web based materials and resource sharing systems.
- ✓ For the funding organization and voluntaries, it will used to show the health outcomes of the capacity building they fund by directly visiting the system output or report measured by different indicators.

Benefits for capacity building providers and for Researchers:

- ✓ It will used for constantly monitoring the capacity building initiatives to make a decision to advance, rework or kill initiatives by the decision makers in the health area. Also it is used to identify and promote good practice by sharing capacity building experience from other healthcare providers.
- ✓ Researchers from medical science and software engineering fields can use the result of this thesis as an input to their study. It will also help them to design a better architecture based on new technologies like by integrating cloud computing with SOA and web services together with other medical technologies.

Benefits for ICT Organization and for the public:

- ✓ ICT organizations will get market opportunity by providing communication services to those institutions that would implement cloud based capacity building resource sharing system.
- ✓ The society would get access to proper medical care and quality health service if the design could be implemented successfully.

1.8 Methodology Overview

To carry out the overall works of this thesis, different methods and techniques was employed. Data collection and analysis techniques, programming and tools techniques, evaluation and testing mechanisms, comprehensive document analysis as well as prototyping as a methodology were used. Primary data from government and private healthcare delivery institutions were collected directly and secondary data through documentary analysis were collected. For the purpose of collecting data, interview technique was used using questionnaire that is because of this study uses qualitative research approach. The questionnaires were distributed to hospitals in Adama and Debre Zeyit using personal contacts. Because of budget constraints, we could not distribute the questionnaire outside Adama and Debre Zeyit. To analyze the data the content analysis method was employed.

In addition, different tools were used to accomplish this thesis. Some of the tools are window 8 operating system Microsoft office for preparing the document; Adobe Photoshop to create images and UML tools for designing the different views of the architecture (ArgoUML and Microsoft Visio). As noted above, UML tools were used due to the following reasons: using standardized notation without sacrificing specialized model data, increased domain and design model reuse, increased customer involvement/understanding of problems translation to product solution. The architecture implementation is built using OpenShift Online platform and open source Drupal content management framework with the help of php 5.4 cartridge, on the base of apache web server. Drupal module based content management framework is also one of the web application cartridge that are found on the OpenShift platform. The prototype that is used to demonstrate the architecture implemented using PHP on OpenShift online platform has been developed and tested.

Php programming language was used to develop the prototype because it is widely available hosting and programming solution and the following benefits.

- Easy to develop, customize and host into OpenShift server
- It is familiar to us

On the database side, OpenShift cloud database for the purpose of storing the data on the cloud and MySQL database engine to develop our web application were used. The comprehensive discussion and analysis of the selection of the platforms and tools is presented in chapter four.

1.9 Organization of the thesis

This thesis work is organized in seven chapters. As we have discussed in this section, the first chapter present introduction and background. The rest of this thesis work is organized as follows: Chapter 2 presents review of literatures on capacity building and cloud computing; and the details of various related works that are related to our work. Methodology of the work is presented in Chapter 3. Chapter 4 discusses selection of the platforms and tools. The design of the proposed architecture is discussed in Chapter 5. Chapter 6 presents the implementation of the prototype as well as the evaluation of the prototype. Finally, in chapter 7, conclusion remarks and future work is discussed.

Chapter Two: Literature Review and Related Works

2.1 Introduction

In the previous chapter an introduction to the study was provided, including purpose, objective and significance of the study. This chapter deals with the review of important documents including different books and articles that are related to our work as well as used in this work in order to identify best practice and analyze the current situations of the area. First, it provides understanding about capacity building; the use of capacity building in the health domain and capacity building program in the health system of Ethiopia. Secondly, it explains about cloud computing; Cloud computing service delivery and deployment models; cloud computing enablers; cloud application architecture and design considerations. Thirdly, some of cloud computing related technologies including virtualization, grid computing, utility computing, autonomic computing and others will be presented. Fourthly, it provides information on the uses of cloud computing technologies to share and access capacity building resources. Lastly, works done by other researchers in the area that are related to our study will be discussed.

2.2 Overview on Capacity Building

Now a days, technologies becomes advanced; everything is operated using ICT and the health practitioners must adapt to these ever changing technologies and problem solving skills. As a result they needs to enhance their capacity to develop confidence to use the current technologies, to have the required professional skills to provide quality health service, to manage the health delivery organization's infrastructures and resources and to give fast clinical decisions.

Capacity building is a familiar term to health areas. Although there is no commonly accepted definition for capacity building [15], it's defined by different researchers in different ways. And the concept is almost similar which touches improvement of ability, knowledge and effectiveness and adapt to change (technological, environmental and logical). In simplest term capacity building means the enhancement of professional skills and ability to provide quality services in any sectors at all levels (individual, group or organizational).

Different scholars from health area and other sectors defines capacity building in their own words. As discussed in [11], capacity building defined in broad terms. It encompasses a multitude of activities like building abilities, relationships and values that will enable organizations, groups and individuals to improve their performance and achieve their development objectives. According to the authors, capacity building includes strengthening the processes, systems and rules that influence collective and individual behaviour and performance in all development endeavours. And it means enhancing people's technical ability and willingness to play new developmental roles and adapt to new demands and situations. Similarly, in the healthcare sectors, "Capacity building is an approach to the development of sustainable skills, organizational structures, resources and commitment to health improvement in health and other sectors, to prolong and multiply health gains many times over." as defined in [3].

2.2.1 Level of capacity building

As described in [3], Capacity building occurs within program or within system. Within health program refers to capacity building activities that takes place in one health delivery organization and within health system refers to the totality of the healthcare system in a country. Different researchers [11, 14] in the area of health classify capacity building at three levels which is individual/health program level, organization level and system level. However, the fourth level of capacity building was included in [10, 18], which is the individual/community level which focus on the beneficiary of the health service. Those individuals contribute their role to improve health service by giving health personnel important information about their previous health problems. Also they can shape the health system by participating in the media to demand that the system respond to their needs and in bringing about desired changes. Generally, the authors discuss all level of capacity building as follows.

Individual/health program personnel capacity building: is the most critical level of capacity building, which includes all individual working in the health system in different technical, managerial and support areas. So capacity of those health workers is required to deliver quality health service and to increase health system performance. This includes clinical judgment and techniques, diagnosis, treatment, sterilization or sanitation practices; management practices, such as written record-keeping and supervision; or money management, technology usage, problem solving, or communications skills. For health personnel, there must be sufficient funds for training, physical space and materials for basic professional education and to conduct training events, and adequate human capital to be trained.

Organization capacity building: this level of capacity building focus on the structures, processes, and management of specific health-related organizations in order to function smoothly and to adapt changing circumstances. This includes communication network, financial management, information management and others. Because capacity building of health personnel takes place in different organizations or institutions, there is a link between these two levels.

System capacity building: this level of capacity building is defined by the boundaries set by the system and aimed to support healthy environment. It encompasses [3] different organizations which is seen as a collection of institutions or organizations, and the health personnel in those organizations working together to deliver health care and/or promote better health. This include design of the overall structure and policies that guide health care delivery; the coordination of different types of organizations (public and private), their managers, providers, and staff; and the allocation, management, and regulation of health system resources as a whole.

2.2.2 Uses of Capacity Building in the Health Domain

Capacity building has a number of advantages for the health system to improve their service and to adapt the changing circumstance. Some of the uses discussed in [3] are listed below.

- The main advantage of capacity building is to share learning and knowledge development across and beyond the collection of organizations that have common purpose or simply it has an important role in expanding the knowledge as well as it ensures program sustainability and enhances problem solving capabilities [18, 19].
- **Used to create responsive systems:** Capacity building involves a focus on the processes that support change within and between organisations. It leads to systems which value critical problem solving and leadership across organisations. Responsive systems are more likely to work in partnership to address health challenges.
- **Used to address inequity:** There is increasing evidence that poorer health is linked to the conditions that arise out of inequity and social exclusion. Capacity building is promoted across government as a mechanism for addressing inequity and building stronger communities through increasing community and civic participation.
- **Used for multiplying health gains**: A focus on capacity building will increase the likelihood that other people and organisations within health and other sectors will also be able to promote health. This will multiply health gains many times over.

- **Used to increase visibility**: A focus on capacity building increases the recognition given to the diverse efforts of practitioners working with others to take on and sustain programs. It gives a 'name' to a large portion of work carried out by practitioners in developing effective programs.
- **Used for taking accountability:** One of the difficulties of working 'invisibly' is that practitioners are not readily accountable for this part of their work. Similarly, managers have lacked clear guidelines for assessing the quality of work purporting to build capacity for health promotion.
- **Used for reorientation of health services:** The message is that along with treating ill health, health services need also to take greater responsibility for improving the health of the communities they serve.

2.3 Capacity Building Program in the Health System of Ethiopia

Ethiopia is a developing country facing different challenges and problems related to delivering quality health care services because of lack of well-trained and well-skilled technological manpower in the health industry to make it fully automated. Capacity building program is provided using manual tools and technologies and no longer available after some years so that permanent infrastructure investments are hard to justify and flexible on-demand services are sought alternatively; if disaster or other system failure occur , no chance of getting the capacity building tools and resources again due to lack of backup.

Currently, Ethiopian Minister of health provides training for the health trainer and distributes them to train another trainer and the actual training will deliver to the required personals. There are three stages or layers in this process. So there is very high cost which is the sum of material cost (like modules copy, print, bind, transport cost and others) and professional fee (for the trainee and trainer). The training time is for two or three weeks which is short or not enough to achieve good medical knowledge and experience to make them capable. After finishing the training, the trainers have only limited modules to refer for another day if they need. And they may not fully implement the knowledge of the training because they may forget it through time. Also the training may not effectively reached to the health professionals as intended because the trainer rises or presents and shows a lot of issues per day to cover or to finish the training on time and budget and to deliver all the planned training. But it may be tedious and difficult to gain the required skill properly.

Moreover, there is no means of making capacity building tools and resources available electronically. That means many organizations in Ethiopia including the health sectors develop tools and prepare training materials to carry out capacity building, but such resources and tools are often not published and not available electronically. In any means making resources and tools available electronically can avoid wasted resources in developing new tools when existing tools may serve the same purpose [15]. Hence using cloud computing technologies to share and deliver capacity building resources is indispensible and will have a great value to increase availability of the resources, tools and materials in the health system of the country.

Generally, as discussed in the literatures and other sources, a lot of tools and approaches to institutional capacity building are available, but the challenge/problem is how to gain access to, selecting and using them, because there is no electronic means of disseminating these capacity building resources, tools and approaches. Thus the best practice for sharing and disseminating one organization's capacity building resources, tools and approaches is to use cloud computing technologies which is the promising solution for the problem noted above.

2.4 Cloud Computing

When it was invented is one of the first questions asked with introduction of a new technology. In that, cloud computing has evolved through a number of phases which include grid and utility computing, application service provision (ASP) and software as a service (SaaS). But the overarching concept of delivering computing resources is rooted in the sixties and has developed along a number of lines, with Web 2.0 being the most recent evolution. So cloud computing is long term dream of grid computing and utility computing. It has started around 1970s when the concept of Virtual Machine (VM) created, but it becomes well known with the arrival of **Salesforce.com** in 1999, which pioneered the concepts of delivering enterprise applications via a simple website. The next development was Amazon Web Services in 2002, which provided a suite of cloud-based services. Then in 2006, Amazon launched its Elastic Compute cloud (EC2) as a commercial web service that allows small companies and individuals to rent computers on which to run their own computer applications. Another big milestone came in 2009, as Web 2.0

hit its stride, and Google and others started to offer browser-based enterprise applications, though services such as Google Apps [20].

Over all, cloud computing is just another step in the history of communication, it is all about as a service. Cloud computing is a buzzword as written by some authors [21, 22]. But some experts and scholars refuses the idea of buzzword because services are not new brand, there have been network services since the invention of the internet [23, 24]. Also cloud computing is defined as anytime, anywhere, with any device, accessing any service from the perspective of customers and different professionals defines cloud computing as IT as Service (ITaaS). Generally, in different scholars, books and cloud experts manuals, the concept of cloud computing has been discussed and defined in a number of ways.

Perhaps the most generally accepted definition has been put forth by the Us Government's National institute of standards and Technology (NIST). Accordingly, "Cloud computing is a model for enabling network users' on-demand access to a shared pool of configurable computing resources that can be rapidly provisioned and released to the client without direct service provider interaction. This model is composed of five essential characteristics, three service models and four deployment models."[5] The five essential characteristics that cloud computing system must offer are:

- **On-demand self-service:** A client can provision computer resources without the need for interaction with cloud service provider personnel.
- **Broad network access:** Access to resources in the cloud is available over the network using standard methods in a manner that provides platform-independent access to clients of all types. This includes a mixture of heterogeneous operating systems, and thick and thin platforms such as laptops, mobile phones, and PDA.
- **Resource pooling:** A cloud service provider creates resources that are pooled together in a system that supports multi-tenant usage. Physical and virtual systems are dynamically allocated or reallocated as needed.
- **Rapid elasticity:** Resources can be rapidly and elastically provisioned. From the standpoint of the client, cloud computing resources should look limitless and can be purchased at any time and in any quantity.

- **Measured service:** The use of cloud system resources is measured, audited, and reported to the customer based on a metered system.

2.4.1 Cloud Computing Service Delivery and Deployment Models

In addition to the five key characteristics, there are seven commonly known models in cloud computing based on its service that are provided to the user and the way of deploying such services (figure 2.1). Three things [25] that provide consumer capabilities are often mentioned in the context of cloud computing. That means Cloud Service Providers (CSPs) offer their services according to three fundamental models. They are Software as a Service (SaaS), Platform as a Service (PaaS), and Infrastructure as a Service (IaaS). Each service is built on top of the other.

In essence, **SaaS** is frequently used model in which an application runs directly on the cloud and the user accesses it via the Internet. It refers to on-demand software which is built on top of cloud's PaaS. It is a web based cloud service model that makes the software available entirely through a thin client interface like web-browser. As a result the users of SaaS software don't care about where the software is hosted, which operating system it uses and by what programing language it developed. In this model, the user does not manage or control the underlying cloud infrastructure including network, servers, operating systems, storage, or even individual application capabilities, with the possible exception of limited user-specific application configuration settings.

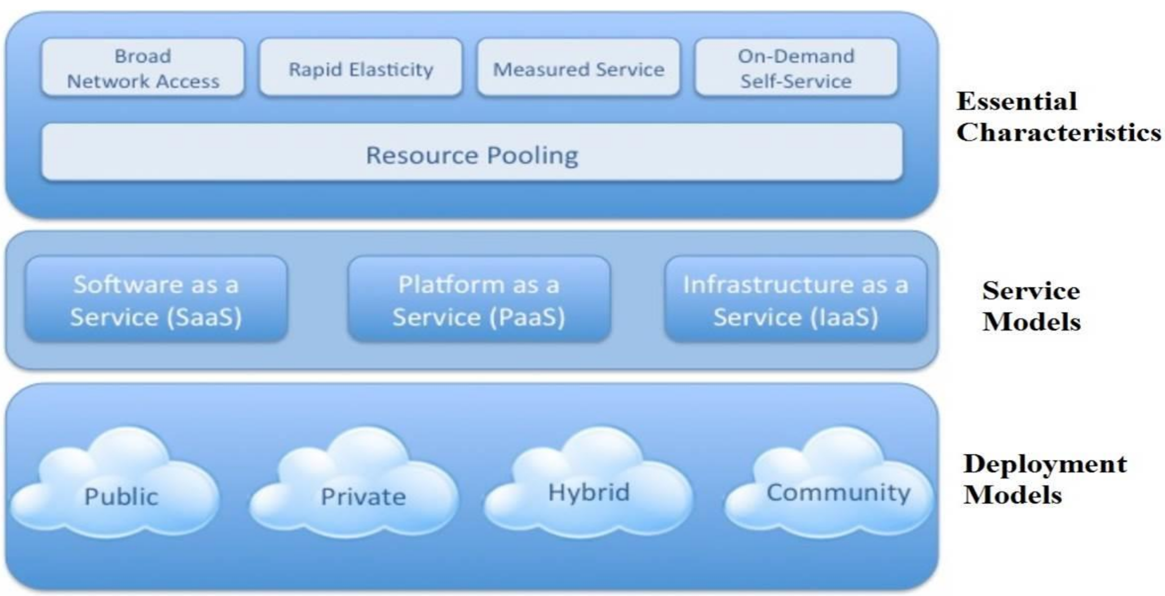


Figure 2.1: NIST definition of cloud computing

Even if it has restrictions on the client's application development freedom, the SaaS model is most popular and holds the majority of the market share compared to other two models, due to ease of adaptability, good return on investment and reduced time to market.

With **PaaS**, services are built on top of the cloud's Infrastructure-as-a-Service (IaaS) and consumers can create and deploy their own applications onto the cloud infrastructure. So that, it provides virtual environments for developing, deploying, monitoring, and managing applications online without the cost of buying and managing the corresponding software tools or hardware. The users have the freedom to develop their own applications using the language of their choice without reliance on a web-based console. The caveat with this model is the interoperability of the application between different development environments.

IaaS provides the underlying infrastructure which enables consumers to access processing, storage, networks and other fundamental computing resources in order to deploy and run arbitrary software, including operating systems and applications. But IaaS doesn't provide applications to the consumers; it simply offers hardware to them in order to put whatever they want on it.

The four ways (models) of deploying cloud computing are: Private Cloud, Public Cloud, Community Cloud, and Hybrid Cloud, which define the implementation of the cloud infrastructure within an organization.

Private cloud is the most expensive model among peers. The cloud infrastructure is provisioned for exclusive use by a single organization comprising multiple consumers (e.g., business units). It means the cloud serves only its designated organization and is not shared among others. The users are given privilege access using highly secure keys and often use external clients like VPN to log into the cloud. This also provides the client with complete access to the cloud infrastructure thereby enabling them to customize the platform to their specific needs. 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.

Community cloud is the most cost effective cloud model. 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. The cost is amortized over all these different organizations.

Public cloud is the most popular and serves a large number of users. The 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. In this model, the customer shares the cloud infrastructure with other customers. The operating and maintenance costs of the cloud are the responsibility of the cloud service provider. The cloud-hosting provider charges a fee, using a license model of pay per user. This benefits the organization monetarily as it eliminates the need for spending on operational IT costs. This model suits small businesses, which require load balancing and management, hosting SaaS applications, using interim infrastructure for development and testing of their applications. Most service providers are transparent with the physical locations of the datacenters. They provide the user with various levels of security packages to the public cloud users, based on subscription models.

Hybrid cloud is a combination of two or more of the above-mentioned models. Means that, the cloud 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). This enables organizations to partition their data based on the sensitivity of the information and host them on various clouds, striking a perfect balance between privacy and cost effectiveness.

2.4.2 Cloud computing enablers

Cloud computing is increasing in its use and importance in health and other sectors [27]. It sparked a significant amount of interest amongst IT industry and research [28]. So it becomes the promising solution in almost all sectors including the health sector in order they use ICT to provide quality services. Having this, it is better to know the key technological breakthroughs for realizing the promise of cloud computing which is called cloud enablers. As defined in [29], Cloud enabler refers to the technologies and manufacturers that serve as the backbone for all cloud computing products and services. A broad term incorporating technology vendors and solutions, a cloud enabler allows an organization to build, deploy, integrate and deliver cloud computing solutions. They are primarily IT firms that develop hardware, software, storage, networking and other related product serving as a cloud environment component.

Four enablers of cloud computing technology have been explained in [30], these are virtualization, multi-tenancy, web services and cost effective hardware which contribute to the development of effective and efficient cloud computing. Similarly, Yogesh K. and Dwivedi N. [31], presents six technology enablers for realizing the promise of Cloud Computing. These are Low-cost access and computing devices (LCACDs), Parallel programming, communication networks, Open Source Softwares (OSS), Cloud access to high performance computing and green computing

Virtualization technology is a key enabler and the foundation of cloud computing which is a method of running multiple independent virtual systems on a less physical resource, makes one computer act as many, sharing the resources of those computer hosts across multiple environments. It enables the division of the physical resources of the machine based on the application requirement and allows for dynamic reconfiguration of resources as required. The main benefit of using virtualization is the portability associated with the creation of n-numbers of virtual machines between different servers. This improves efficiency in the use of existing hardware as well as creating potential for rapid scaling up of resources on the fly and it makes the cloud function more effectively.

Multi-tenancy is the way in which access is given to several clients, who simultaneously use the same resource without intruding on each other's space or data stores. It is the key common attribute of both public and private clouds, and it applies to all three layers of a cloud such as IaaS, PaaS and SaaS. Often, multiple clients use a single application on the server and artificial software authentication protocols are set up to ensure privacy. This leads to optimal use of physical computational resources. Depending on the deployment model of the cloud, there are varying levels in the implementation of multi-tenancy concepts.

Web services cloud computing uses web services for connection, which is a software system that enables communication between two machines over a network. It provides a standardized platform to clients, allowing them to access applications in the server through a network. It uses XML for data tagging, SOAP for data transfer, WSDL for service description and UDDI for available service. By default, this system does not have any GUI features. However, most cloud service providers integrate this system to a web page with a GUI based application to offer specific functionalities.

Low-cost access and computing devices (LCACDs) are other enablers which are expected to play a pivotal role in the adoption of cloud computing among the masses, especially in the developing counties. Also **Parallel programming** enhances user experience by increasing effective use of multi-core processor. It is known that cloud applications are fully internet based. So high speed **communication networks** are required and the authors described as an essential cloud enabler.

Another key enabler of cloud computing is **Open Source Softwares (OSS)**. It provides users full access to source code, which can be modified to suit specific user requirements and reduces the cost pertaining to SaaS. **Cloud access to high performance computing** is used to increase adaptability by E-researchers in order to use E-infrastructures for their scientific enquiry which is a long time dreams of grid computing. The last but not the list cloud computing enabler is **green computing** to make efficient use of electricity in cloud computing infrastructures that is deployed in a very large data centers.

2.4.3 Cloud Application Architecture and Design Considerations

Understanding of which application architectures are fit/best appropriate to operate in the cloud as well as the kinds of applications and data that run well in cloud environments, data backup needs and system workloads are the basic concepts to build application to run in a cloud environment. As described by Bernard Golden, there are three architectural choices for cloud environments: Traditional architecture; asynchronous application architecture; and Synchronous application architecture. As the author discusses, because of not all cloud environments are equal, we need to research and design our application appropriately for the particular environment offered by our cloud provider. In addition, we need to determine which applications and data sources used by the application will run in the cloud. Also we need data backup to ensure recoverability in case of crashes [32].

Traditional architecture can be used for cloud application only if there is stable demand rather than tolerate huge variations in system load. A traditional architecture calls for one or more Web server systems interacting through a middle-tier software framework, ultimately interacting with a database. Infrastructure as a Service cloud providers; such as Amazon Web Services (AWS) as well as GoGrid and Rackspace accommodate this architecture.

In **synchronous application architecture**, end-user interaction is the primary factor in which large numbers of users may hit the system in a short duration and potentially it creates poor performance. To avoid this problem providing enough Web servers to handle total traffic; providing enough middleware to manage demand and providing a data tier that scales is required and must be included in the architecture.

Asynchronous application architecture does not support end-user interaction but focused on information processing like an extract from a database or a set of files. So this application architecture is used if there is no need of user interaction.

After selecting appropriate application architecture and platform, the applications will be designed. When designing applications for the cloud, we have often found it useful to know the most important part of the design of any application. This section describes **design considerations** [33] to overcome challenges to rapidly deliver reliable, manageable, feasible, scalable and economical cloud solutions for varying business requirements.

The **feasibility** which is the ability to deliver and maintain the system, within budgetary and time constraints like SLA must include under consideration. **Availability** describes the ability of the solution to operate in a manner useful to the consumer in spite of transient and enduring faults in the application and underlying operating system, network and hardware dependencies. So we need to consider uptime guarantee, Replication and failover, disaster recovery, performance and security. For gaining high availability, multiple zones, Graceful Degradation, global data replication, Distributed data management platforms and Asynchronous Messaging must be under consideration in cloud based applications.

Scalability is focus on any requirement to add additional capacity to the application and related services to handle increases in load and demand. So we need to consider capacity, platform/data and the load to avoid contention issues and bottlenecks. There are two kinds of applications that need the scalability of cloud environments: user-facing or synchronous (i.e., a user interacts with the system and waits for an answer) and asynchronous (i.e., data is input, processed, and eventually concludes being worked up, with no one sitting around waiting for the results). Also manageability which describes ability to understand the health and performance of the live system and manage site operations is one of the design considerations. So we need to consider monitoring and deployment concepts to make the application more manageable.

2.4.4 Cloud Computing Related Terms and Technologies

A familiarity with cloud computing terminology will help anyone to follow the industry's developments and to have better understanding about the new language that has emerged in the area of cloud computing. There are hundreds of terms related to cloud computing. However, in this study presents some of the terms [34] and technologies [35] related to cloud computing that anyone should know. The related technologies are presented as follows.

Cloud computing typically has characteristics of Grid computing, Virtualization, Utility Computing and Autonomic computing [35].

Grid Computing: involves a network of computers that are utilized together to gain large supercomputing type computing resources. Using this network of computers large and complex computing operations can be performed.

Virtualization: it is the act of creating a virtual rather than a physical version of a computing environment, including computer hardware, operating system, storage devices, and so forth. It introduces a layer between Hardware and operating system. It also forms the foundation of cloud technology and helps in improving server utilization.

Utility computing: is a provisioning model in which services are available as needed, and users are charged for specific usage, in a manner similar to municipal utilities such as electricity or water. It helps in reducing initial investment. As the computing requirements for an individual or an organization changes, the billing changes accordingly, without incurring any additional cost. If the usage has reduced, then billing will also reduce accordingly.

Autonomic computing: is an initiative started by IBM in 2001. Autonomic means "self-managing" computers. In Autonomic computing, computers can automatically correct themselves without human intervention. Hence the goal of autonomic computing is for computing infrastructure to self-correct itself in unforeseen situations.

2.5 Role of Cloud Computing to Share Capacity Building Resources

The emergence of cloud computing technologies leads to new developments for diverse application domains. This is particularly true for healthcare with its unlimited importance in today's health workers. So the use of cloud computing technologies with in the healthcare systems and the integration of capacity building resource sharing into E-Health to create worldwide healthcare learning environment will improve capacity of health workers as well as the healthcare delivery organizations.

Thus cloud computing includes, but not limited to the following roles to share and access capacity building resources for the healthcare practitioners and the organization itself.

- ✓ It is used in educating healthcare professionals in providing access to medical knowledge, applications and literature, while serving as a source for public and formal education for health [9].
- ✓ Also cloud computing provides the ability of mobilization of capacity building tools, resources and knowledge.
- ✓ Cloud computing is cost effective and works via the Web, with access to services by means of a simple browser and, increasingly, with free operating systems using any device which have internet access. Thus its implementation will benefit the healthcare institutions to get access to the capacity building resources from one general hospital or capacity building provider (that have full training tools and best approach) as a service.
- ✓ It increases disaster protection: Disaster is an event that makes the continuation of normal functions impossible, when the users are accessing and sharing capacity building resources on the Internet. Disaster recovery is necessary in capacity building system services. Disaster recovery programs in clouds provide more options than traditional disaster recovery models for organizations to restore information very quickly and effectively [26].

Clearly, sharing of capacity building resources has a great role to promote health especially for the smallest healthcare institutions. Hence the health system must focus on capacity building to multiplying health gain, to address health inequality, to build responsive system, for reorientation of health service and increases the recognition given to the diverse efforts of practitioners working with others to take on and sustain programs. And cloud computing will contributes high opportunities to make the capacity building resources more cost effective, available, deliverable and accessible by the healthcare institutions. As a result it will benefit the health care system of Ethiopia to address health inequality and to promote health.

2.6 Related Works

At present, both industries and academics have done a lot of significant works in the area of cloud computing based healthcare resource and service delivery. However, we are not familiar with any work that studies about capacity building architecture in the healthcare area that are based on cloud computing concepts. As a consequence, we address, in turn, work related to cloud based healthcare applications and architecture in general. As a result, sharing of information, creating collaboration and supporting communication among medical practitioners is the most core idea of different articles that have been done by relating cloud computing and healthcare resource and service delivery issues. So that accessing patient healthcare information everywhere, with any device and every time as needed is the most important concept [1, 2, 4, 9, 36] to provide quality health and to save patients life. Several papers include the idea that cloud computing might be used to support sharing of patient information and medical related soft wares; collaboration among stakeholders in the health area and enable patients in managing their health status.

In that, the work of He et al. which is the development of six-layer private cloud platform architecture with message Queue (MQ) as a cloud engine, focuses on self-monitoring of health status and enabling of citizens to manage their own health. They have seen component as a service which inherited SaaS NIST model, to develop a private healthcare cloud architecture which should provide early warning of diseases and instant reports for individuals [36]. Siddiqui et al. describe the concept of a Telecare Medical Information System (TMIS) which deals with delivery of remote healthcare solution to the patient in their home and includes different medical services for patients and medical professionals such as real-time diagnosis and remote monitoring of physiological signals. In this work, Patients and medical practitioners use their smart phone as their complete identity over the cloud. So they should connect to the TMIS by using their smartphone. And their smartphone needs to be equipped with authentication possibilities to ensure data privacy and data security. For this purpose, the authors propose a three-factor authentication (3FA) based on a dynamic cloud computing environment to enable remote user authentication [37]. Similarly, Van Gorp and Comuzzi discuss cloud-based personal health record system called MyPHRMachines which is an academic prototype, where a cloud is used to deploy health-related data and the application software to view and analyze it in a personal health record system. After uploading their medical data to MyPHRMachines, patients can access them again from remote virtual machines that contain the right software to visualize and analyze them without any need for conversion. As the authors describes, the patients can also build their records and can share their remote virtual machine session with selected caregivers by using this MyPHRMachine [38]. Furthermore, Takeuchi et al. [39] present a prototypic cloud-based system to store personal health and lifestyle data using mobile devices. In a cloud infrastructure they claim to have implemented data-mining technologies to extract individually important information such as lifestyle patterns. Although other persons like dietitians should have the possibility to add comments into the system, it is not explained how data access in the cloud will be managed.

Yao et al. [2] presents Cloud-based hospital information system as a service for grassroots healthcare institutions, the authors created a community cloud based medical service delivery framework (CMSDF). The author’s purpose is to reduce technical and economic barriers (like high upfront investment costs and the perceived lack of return on investment (ROI) and challenges of implementation, maintenance, and renewal) to deliver quality healthcare service by enabling sharing of softwares and tools from large general hospital to its associated smaller health institutions including community health service centers or rural clinics. In the prototype CMSDF, a cloud-based Virtual Desktop Infrastructure is owned and managed by a large hospital which is able to share its medical software as SaaS with the smallest healthcare institutions. According to the author for the 34 cooperative sanatoriums that participated, 89.9% of investment and maintenance cost were saved because the smaller facilities had not to buy and host expensive software on their own.

According to Ratnam and Ramayah [40], A structural equation modeling approach for the adoption of cloud computing to enhance the Malaysian healthcare sector, cloud computing could be the method of choice to reduce the escalating costs of data storing and sharing. Even if the authors did not describe this cloud system in detail, they have mentioned that a cloud platform using Microsoft Windows Azure was used as prototype architecture. In contrast, Hsieh and Hsu have presented a very comprehensive and detailed description of a 12-lead ECG telemedicine service based on cloud computing [41]. They have clearly described how the processing, visualization, management and eLearning services are deployed within the commercial Microsoft Azure cloud platform. They have further presented the reasons for adopting the Azure platform and the financial background of the implementation, based on the Azure pricing model with monthly costs directly related to CPU hours and GB storage used.

Moreover, Hiden et al. [42] described the development of e-Science Central (a platform as a service which itself was built on an infrastructure as a service environment) which allows scientists to store, analyze and share data in the cloud. Their article clearly illustrates not only the set of cloud services provided, which cover data storage services, but also service execution, workflow enactment and security. Finally as one of three case studies, they present a medical pilot investigation (the MOVEeCloud project) where medical specialists assess the physical activity of patients based on data uploaded to the e-Science Central cloud by wearable accelerometers.

To summarize, any of the researchers didn't touch the problem/lack of healthcare knowledge exchange network to enable sharing of capacity building resources, tools and approaches to create collaboration and to develop knowledge, ability, experience, technology adaption and problem solving skill. Also none of the researchers develops a framework, architecture or model for sharing and accessing capacity building resources using cloud computing. Therefore, we have tried to discuss these related works done by other researchers which include telemedicine related works, patient's self-management works, medical imaging and clinical information system works as a broad domain. And by considering untouched gaps discussed above, we propose to design cloud based capacity building resource sharing architecture which improves the ability of the health workers by incorporating sharing of capacity building resources for health workers in the existing health system (E-Health). Certainly, we believe that our work will contribute a lot to the study area and it creates the idea of sharing and accessing capacity building resources as a service (CBRaaS) which is not touched by any of the works explained above.

Chapter Three: Methodology

In this study, the first step after deciding the area of interest, was performing literature review to get previous works about cloud based capacity building architecture and the implementation of healthcare capacity building system in the cloud computing. The next step was conducting interview with medical practitioners and capacity building trainers from different healthcare organizations including public and private hospitals and clinics to collect data about the current capacity building program; and to know their expectations and indispensableness on the cloud based capacity building resource sharing system. The last step was selection of tools to design the proposed architecture and to implement its prototype for the purpose of proof of concept.

Based on these steps, a design of cloud based capacity building resource sharing architecture and its prototype was developed to address the problem (Fig 3.1).

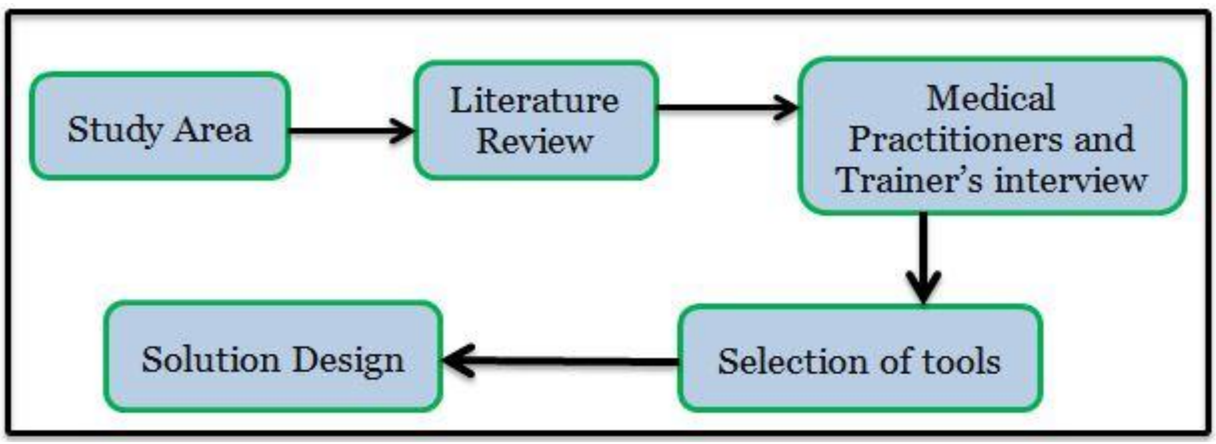


Figure 3.1: Research Methodology

3.1 Research Design

For any research studies, there are four commonly used approaches which are quantitative, qualitative, design oriented and mixed method [43]. The quantitative research approach uses standardized measures, numerical values, have large sample size and analyze data using statistical program. It is used when one starts with a hypothesis and tests for validation and invalidation of that hypothesis. It is applicable to phenomena that can be expressed in terms of quantity or number [44]. On the other hand, qualitative research approach is focused on describing situations, phenomenon, problem or event; tends to be more in-depth and have smaller sample size [45]. A mixed methods research design is a procedure for collecting, analyzing, and mixing both quantitative and qualitative research and methods in a single study to understand a research problem [46]. It is mostly used when one type of research is not enough to address the research problem and when practically multiple view points; biased and unbiased; subjective and objective ideas occurred [55]. Design-Oriented Research creates an artifact by artificial object and its basic objective is producing new knowledge during this process. It is mostly focus on knowledge flows and process steps that is problem awareness, suggestion, development, evaluation and conclusion; eventually aims to get result or output [47]. In this study, a combination of qualitative and design oriented research method were used for data manipulation and to evaluate the functionality of the prototype. Due to qualitative approach focuses on involving quality and helps in having insight in to problems, more tasks in the study were qualitative.

To show the usefulness of the proposed cloud based capacity building resource sharing architecture for healthcare organizations and their workers, design-oriented research methodology was used in this study. Design-oriented research method is basically a problem-solving model [48]. The five steps of design oriented method followed in this thesis work are identification of problem, architecture design, architecture evaluation, revaluation and improvement of architecture and communication and discussion of research.

3.1.1 Identification of problem and Requirement

Currently, CB resources and tools are dispersed in different HDIs [12] but these tools and resources are not electronically available, not shared and widely used by many users. Requirement or data collection was the basic step to design the proposed architecture after identification of the problem. In this study, the required data and the user requirement was collected using different methods which includes in depth interview with healthcare practitioners and capacity building trainers, observation with the study area and distributing questionnaires that contains questions related to capacity building program in the health area and problems encountered when they employed the program. Then the collected data was organized and analyzed to use for the design of the architecture.

3.1.2 Architecture design

To achieve the objectives stated under the given constraints and limitations by using the specific approach, the design of CCBRSA were focused on the elements or components of capacity building program. To design the proposed architecture, reference architectures and other internet resources was used. In addition, the 4+1 view model of software architecture and UML has been chosen. This is because the 4+1 view model of software architecture is the best model to design and analyze the different views of the architecture and the UML is used to represent these views using standardized symbols and icons. So UML based solutions for designing, analyzing, managing and building the architecture was employed. The conceptual architecture is presented in chapter five and the different view of the architecture is presented in chapter six of this thesis.

3.1.3 Architecture evaluation

In the software development process, architecture can be evaluated at the early stage or later stage. This is used to compare and identify gaps in various alternatives during the early design stages. There are different Software architecture evaluation methods which include experience-based, simulation-based or prototyping, mathematical modeling based and scenario based [49]. In this study the early stage evaluation and prototyping is good because the cost to fix an error found during requirements or early design phases is orders of magnitudes less to correct than the same error found during testing [50]. Evaluating and improving CCBRSA architectures related to quality attribute which is set as goals was checked using expert on software engineers, health professionals, capacity building trainers and selected other professionals who have cloud technology skills, which is presented in chapter six.

3.1.4 Revaluation and improvement of architecture

After examining the performance and its workability, comment and suggestion have been acquired from the expertise and users to add more feature and functionalities for improvements of the proposed architecture.

3.1.5 Communication and Discussion of research

To address how the problem can be solved and how the system works, communication contributes an essential role. Thus to answer these and other related questions, help manual and orientation were prepared for the end users. While quality attributes of this work depends much more on the software architecture than on code related issues, architecture should be clearly presented to all users. The discussion of this work concentrates on the architecture quality attributes to be secure, scalable, interoperable, portable, and reliable.

3.2 Process Model

In software Engineering, process model is a description of the sequence of activities carried out in the software project and the relative order of these activities. There are several process models which include prototyping, unified process, incremental and agile. We can rarely combine process models or use together in software development process [51]. In this study, prototyping and Unified process models were used together to accomplish the overall objectives. Prototyping is about creating some parts of the system being developed. It is not a standalone and complete development methodology rather it is a model to test specific features in the situation of complete methodology [52] like unified process.

Hence**, Prototyping** is a process model that offers a solution to the problem and it is the practice of building an early version of a system which shows particular features of the actual system. There are two types of prototyping which are throwaway and evolutionary prototyping. In **throwaway prototyping** the different parts of the system are developed and then will be ignored after demonstrating the basic functions that the system will carry out. Thus the final deliverable system will differ from the prototype. But **evolutionary prototyping** technique implements the prototype of the system that demonstrates its basic features and finally it will be part of the final deliverable system.

The **unified process** (UP) or rational unified process (RUP) is the most recent software development process model which provides overall plan and strategy to be used as part of it. It has two purposes, the first one is meeting user's need by the use of techniques like use cases, iteration and emphasis on the architecture of the software. The second purpose is accommodating risks by making use of iterative technique. UP consists of four phases: inception, elaboration, construction and transition. **Inception phase** consists of a feasibility study to establish whether the proposed system is worthwhile. In **elaboration phase**, the creation of general architecture of the system and requirements lists is completed. In **construction phase**, the development of the actual system, in this case the prototype of the actual system is performed. The **transition phase** involves making the system in to use by installing and running this system. Generally, the unified process model importantly uses use cases, iteration and software architecture techniques; and the prototyping model is concerned with the development of incomplete version of the system. Therefore these are the reasons to select these models for this study.

3.3 Data Source

The data which is used as an input for this work is acquired from the following sources:-

- Stakeholders from Government and Private Hospitals which includes medical practitioners, trainers and ICT professionals working in the HDIs. In this way three government HDIs were selected namely Adama Hospital Medical College (AHMC) and Bishoftu general hospital. From private, two hospitals were selected namely Adama General Hospital and Medical College (AGHMC) and Sister Akilesiya Memorial Hospital (Sr. AMH).
- The second data source in this work is comprehensive literature review and document analysis from the internet. So internet is one of the sources of data as explained in the methodology overview section of chapter one.
- The other source of data in this work was observation. During this study observation were used as a means to collect the required data specially to find health practitioners problem related to lack of training.

3.4 Sampling Techniques

Sampling involves the selection of individuals, units, and/or settings from a defined study area. There are different sampling techniques in qualitative research to sample the research participants. This includes purposive sampling, snowball sampling, quota sampling, typical case sampling, convenience sampling and others [53]. In this study, purposive sampling technique was used to collect the qualitative data. So data collection was started by identifying the participants using purposive sampling technique. Purposive also known as judgment or selective sampling technique is mostly used technique in qualitative research due to the qualities the informant possesses. Healthcare practitioners from different healthcare delivery institutions were chosen directly through personal contact. The healthcare practitioners and other concerned bodies were selected based on appropriateness for the study; the knowledge they have and technology used, because this technique uses appropriateness principles in place of random sampling [55].

3.5 Data Collection and Analysis

This study was carried out in two different parts of the country Adama and Debre Zeyit. The interviewed professionals were both from ICT professions working in the healthcare system of the HDIs and the medical practitioners of the HDIs. Also the interviewees were from both private and public HDIs. This enables us to collect the required data correctly as well as it makes the data collection procedure more easy.

This data collection task involves two phases: (1) in-depth interviews with 41 health workers and ICT professionals from both private and public HDIs concerning their ideas about capacity building program, its indispensableness, current situations, barriers and solutions to provide it consecutively and about cloud computing as well as their interest if it is cloud based; (2) observation based on work place study about health worker's technology adaption, experience and problem solving skills and their medical knowledge. Although the need of consecutive capacity building program for health workers is clear, efforts were made to get the correct information about the health worker's ability and experience in their work place from different patients. This is done through different techniques but getting the right information was extremely difficult as observed.

The basic focus where data were collected concerning capacity building and cloud computing were, understanding about capacity building, CBP importance, way of delivering CBP, problems the faced when practicing CBP, CBP effectiveness in the current situations, the use of ICT in their day to day activities, the know how about cloud computing and its effect related to cost, availability of resources and ability of workers, solution suggestion to solve the current problem and the interest of health workers if it will be cloud based. We collected data about these concepts because it helps us to build the baseline for the architectural design.

Specific areas covered to gather relevant data about CBP were Adama Hospital Medical College, Adama; Bishoftu General Hospital, Debre Zeyit; Adama General Hospital and Medical College, Adama and Sister Akilesiya Memorial Hospital, Adama.

Generally, the survey of different HDIs was conducted using questionnaire, interview and observation. Using the questionnaire, we have investigated the CBP concepts, way of delivery, problems they faced and possible solutions they suggested as well as their use of ICT and other technologies in their activities specially related with the CBP activities. Then we have included questions to know their interest if the CBP will be cloud based and supported by the current technologies to share experience, knowledge, training resource and tools. After that, the questionnaire was distributed to two government hospitals and two private hospitals, practicing in Adama and Debre Zeyit.

The interview mainly focused on importance of CBP for the health practitioners, how currently provided to them and what efforts were done to provide it consecutively to them. Also how the health practitioners get access to the capacity building resources and materials, in terms of technology used and collaboration and knowledge environment reserved were emphasized.

Rough understanding and situation about capacity building and related issues were found from health practitioners employed in Adama General Hospital and Medical College and Sister Akilesiya Memorial Hospital. As they explained, even if they got more training per year, it is not enough to them to be able to handle any challenges related to their profession. This is because the CBP needs high cost to provide it consecutively. In addition, the lack of advanced training tools, resources, approaches and well-trained manpower who train other trainers and the trainees are the obstacles to provide the training. However, they noticed that making the capacity building program available for all health practitioners and the community will contribute a great role to reduce health inequality and to deliver quality healthcare services for all societies.

From another HDI, Adama Hospital Medical College, the health practitioners and ICT professionals forwards their idea about accessibility and limitations of the current CBP approach. Also they explained, looking forward with cloud based CBRS system as an alternative means for accessing the capacity building resources will benefit them highly and will used to get E-experience sharing. The health practitioners from Bishoftu General hospital clarifies that, over time, the changing technologies regarding to medical areas may affects their effectiveness in their work. As they discussed, the need of knowledge and experience sharing network is apparent and required for them.

From these HDIs, 41 individual health and ICT practitioners were interviewed. When the practitioners asked about the importance of CBP in the healthcare area, all of them argued as it is very important. According to the practitioners, even if it is important, the CBP is not provided to them as they need over time. This is due to the high cost of the CBP, lack of well skilled and experienced trainer who adapts the current technology, lack of ICT support and lack of sharing of resources, tools and best practices. Also it is not that much effective and efficient because of the above mentioned reasons. But they strongly argued that, the integration of ICT with the healthcare industries is required and helpful.

The next point discussed was about the cloud computing technologies and its role for accessing and sharing resources of capacity building program. Despite, most of them did not know about cloud computing, they put their idea that, cloud computing may increase the availability of the training and decrease its cost. At the end, they suggest the following suitable mechanisms to provide the CBP for all health practitioners at any time they need.

- Modernizing the resource accessing and sharing mechanism for easy access by different electronic devices including smart phones ubiquitously
- Making the CBP cloud based for high availability and agility
- Create broad healthcare network and healthcare learning environment for collaboration, resource and experience sharing

The main purpose of this work was, looking forward to use the current technology for sharing and accessing capacity building resources in healthcare institutions to solve the problem of lack of healthcare learning, knowledge sharing and networking and collaboration center. Cloud computing based platform architecture was proposed to solve the problem through sharing of all capacity building resources to all healthcare workers at all level. Cloud computing has a solution for storing and processing large amount of capacity building resources, for sharing and accessing these resources with cost effective, flexible, scalable and multi-tenant way. This will help the healthcare workers to develop ability, experience and knowledge, problem solving skill and technology adaption in their profession by accessing the shared resources using their mobile phone and other existing ICT infrastructures ubiquitously.

Chapter Four: Selection of Cloud Platform and Service Provider

4.1 Introduction

This chapter provides an overview on the five selected cloud platform service providers and more in detail about Redhat's OpenShift online platform for this work. For the purpose of developing prototype of the proposed architecture, we have started to find the best cloud platform as a service provider. As a result, we have got a lot of available tools by conducting advanced search and we have selected only the five tools for comparison and analysis to use the best one for our intension.

At first, a relevance selection based on availability (open source or commercial) was conducted to remove obviously commercial tools because only freely available tools were needed for this study as an academic purpose. The tools selection process is shown in Fig 4.1.

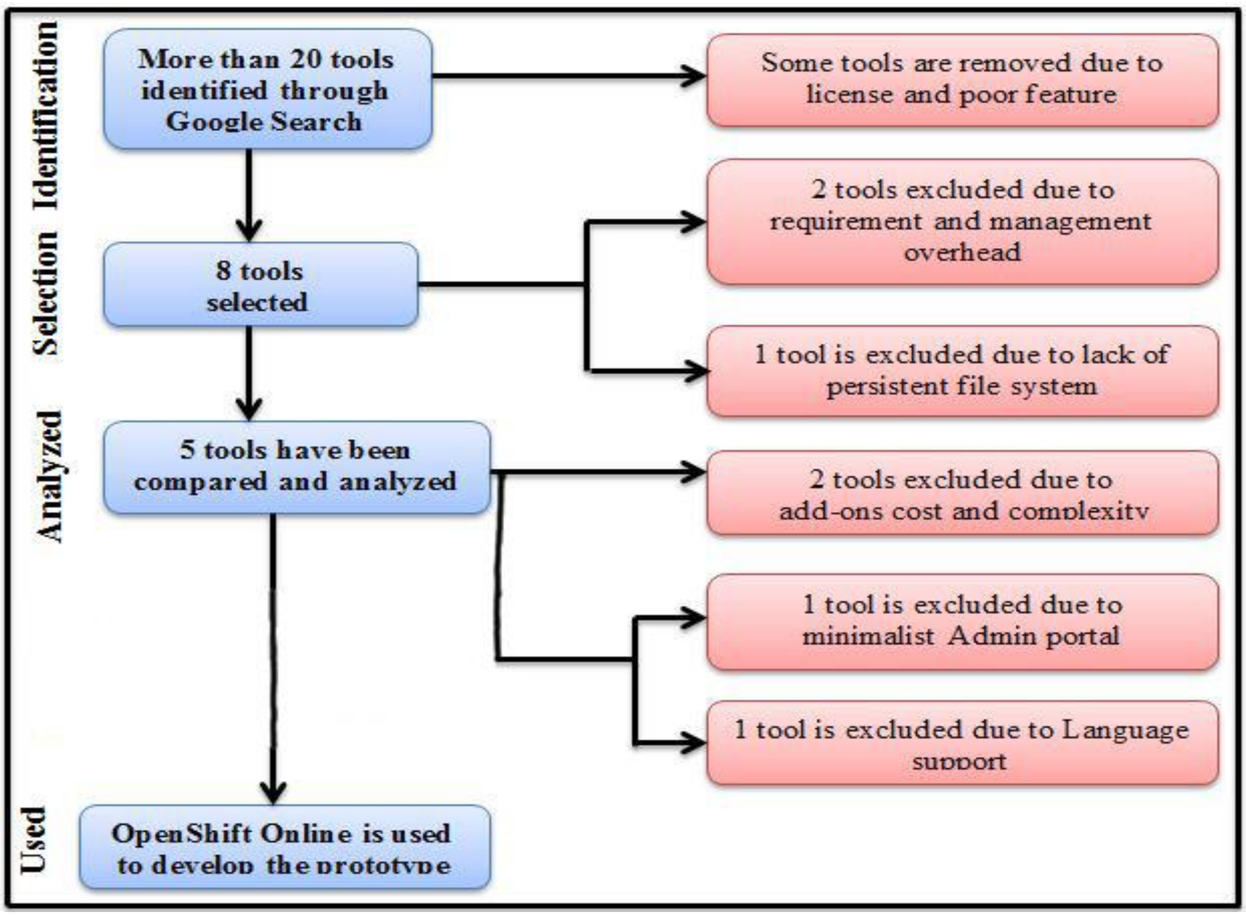


Figure 4.1: selection of tools flowchart

4.2 Analysis and comparison of cloud application development and hosting platforms

Platform as a Service is a cloud model designed for software developers to develop web and mobile applications. In this study we focus on the platforms that are used for web application development and hosting. There are a lot of tools available for the application developers in the cloud environment and we try to describe the most popular platform providers and their development and hosting tools. At first we have identified around twenty three tools, after rejecting some tools due to license and poor feature, we have selected the eight tools such as OpenShift, Microsoft Azure, Amazon AWS (Amazon Web Service), Cloud foundry, Heroku, Google App Engine, EngineYard and Appfog. Out of these tools, we are picked only the five tools for comparison and analysis.

When comparing the platform providers and their tools explained below, we have used seven criteria such as GUI support, programing language they support, middleware technologies, software framework, core native services as a part of offering, pricing model and availability of the tools. The list of cloud application development and hosting tools that we have compared are: OpenShift, Microsoft Azure, Cloud Foundry, Heroku and Google App Engine. A brief comparison among these cloud application development and hosting tools is shown in table 4.1.

4.2.1 RedHat – OpenShift

OpenShift is Redhat's cloud platform which offers an application development environment for developers and provides two options for developers to interact with OpenShift such as through a web console or the command line client tools as well as developers can interact with OpenShift through an integrated development environment. So this platform is highly customizable and offered in three forms: OpenShift Online (a public cloud-based hosting service), OpenShift Enterprise (a private PaaS that runs in your data center) and OpenShift Origin (the open source application hosting platform). In OpenShift, an application is a combination of code, configurations and application components called **cartridges** which include databases as well as languages and frameworks, and QuickStarts, which are applications with code and libraries already configured. So developers work on applications and components. Multiple cartridges can run on the same server because cartridges are logically isolated from one another and it has gears and nodes to show the abstract structures that isolate components. With this, nodes hold gears that contain one or more cartridges. Even though, OpenShift have the ability of automating workflows which helps developers to scale automatically in order to handle peak workloads and to administer tasks like virtual server provisioning, configuration and scaling and supports git repositories for code management; it Works well with Git, but non-Git deployments might require additional steps [54, 56].

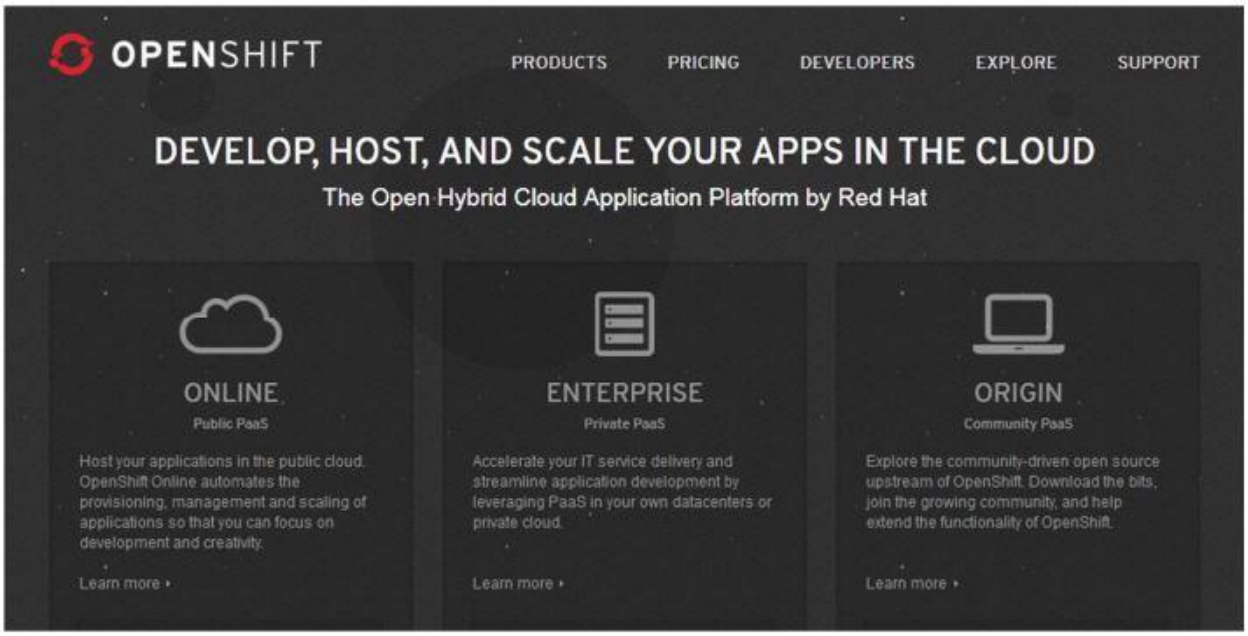


Figure 4.2: The three OpenShift offerings

4.2.2 Microsoft – Azure

Azure is Microsoft’s cloud platform that offers the combination of IaaS and PaaS at one to suit the needs of customers, which is a foundation for running applications and storing data in the cloud. It enables developers to use any language, framework, or tool to build applications. So that Microsoft azure supports any operating system, language, tool and framework. In addition, developers can use Visual Studio for creating and deploying applications which support local debugging of application code, a trace facility and storage account diagnostics, as well as other troubleshooting features. Microsoft azure features and services are exposed using open REST protocols. In this platform, Applications are administered through the Windows Azure dashboard or through a command line interface [54, 57].

4.2.3 Pivotal - Cloud Foundry

Cloud Foundry is an open source cloud platform as a service (PaaS) originally developed by VMware. It is now owned by Pivotal Software, which is a joint venture made up of VMware, EMC, and General Electric. Also a commercial version called Pivotal Cloud Foundry (Pivotal CF) is available from Pivotal which provides extra tools for installation and administration. The Pivotal CF benefits from the use of Cloud Foundry technology and will run on Hybrid solutions as well as private and public cloud solutions. Cloud Foundry is available as a stand-alone software package. It supports the full lifecycle, from initial development, through all testing stages, to deployment. Applications deployed to Cloud Foundry access external resources via services. And a load-balancing router sits at the front of Cloud Foundry to route incoming requests to the correct application essentially to one of the containers where the application is running. The best feature of Cloud foundry is: it offers support for all major programming languages and database systems; the software allows developers to run and manage multiple applications at the same time and the dynamic routing it offers enhances the efficiency of applications. Also Cloud Foundry offers other important services which include high performance dynamic routing, application centric lifecycle API, buildpack which provide a framework and runtime support for the applications, data and web services brokers, container management, application health management, integrated real time logging API and configuration management tool called BOSH. Although it supports these features and many languages and frameworks, including Java, js, Go, PHP, Python, and Ruby, not all applications will make a good fit. Hence, it does not offer support for arbitrary languages and frameworks [54, 58].

4.2.4 Salesforce - Heroku

Heroku is the early cloud Platform as a Service (PaaS) provider which is ideal for quick deployments and fits a wide range of distributed applications. Heroku has been around for a while, it started with Ruby programming language, currently it has gone on to develop support for Java, Node.js, Scala, Clojure, Python and PHP. It works best with applications that fit well into the Twelve Factor App methodology. It provides **dynos** (web dynos which respond to HTTP requests or worker dynos which respond to task requests in a queue) which are abstract computing environments and virtualized containers that run processes in isolated environments.

Each dyno can run a variety of different types of processes, including web apps that serve pages, processes that serve API requests, Time-based batch or background workers and One-off programs. Each dyno is allocated 512MB of memory, which is isolated from other dynos and processes to ensure that any memory leaks that occur do not affect other apps running on the same hardware. That means each process in Heroku runs in isolation from one another, despite the fact that a number of different processes may be running on a single virtual instance. Heroku ensures that each dyno has its own independent file system and that memory and system resources are evenly distributed. Isolation also ensures that overloaded apps cannot compromise other apps running on the same virtual machine or server, and even isolates processes from the underlying operating system itself. It also provides protection against violations of memory safety introduced by events such as buffer overflows, thereby minimizing security risks and vulnerabilities for apps run on the platform. Also add-ons which are third party applications are available as services within the Heroku platform. Due to the number of add-ons vary and loads vary across application components, this leads to cost fluctuations which can make it difficult to plan ahead [54, 59].

4.2.5 Google - App Engine

The App Engine is Google's platform designed for web applications and supports different languages such as Java, Python, Go and PHP and allows for integration to other technologies such as Hadoop, MongoDB and others. It uses a sandbox model which isolates processes thereby reducing the risk that a rogue process on a physical server will disrupt operations of other processes on that server. Google App Engine like other platforms manages many of the implementation details about maintaining virtual servers, operating systems and development tools. But it is restricted to four programming languages which is its limitation [54, 60].

Table 4.1: Comparison among the five cloud application development and hosting tools

Criteria	OpenShift	Azure	App Engine	Foundry	Heroku
GUI Support	Web console interface and command(rhc client tool)	Windows Azure dashboard and command line	Command line interface	CF command line and BOSH tool	Heroku Command Line tool
Language support	Java, Ruby, PHP, Python and Perl	.NET, Java, php, Node.js , Python and Ruby	Java, Python, Go and PHP	Java, Ruby and Node.js	Ruby, Python, Java, Scala, php, Cloture and Node.js
Middleware	Zender server Tomcat and jboss	Tomcat java 7	--------	Tomcat java 6, 7	-----------
Framework	Drupal(php),Switchyard(java0.8),vert.x(java2.1), Django	Cakephp, Django , etc	Django, Webapp2	Spring, play Rails, Sinatra and Grails	Play, grails Django, flask, rails
Service	Data storage, Messaging, backup development and mobile services	No native, databases, messaging queues and caching	Storage and development services	Data storage, messaging, development and mobile services	Development, storage and message queuing
Pricing	Based on the number and size of Gears, (monthly bill/ fixed usage or free usage)	Based on size of instances running	Based on usage, monthly bills or free usage	Based on upon request or free usage	Based on number & size of dynos, Postgres DB and add-ons used
Availability	Open source: http://www.redhat.com	Open source: http://azure.microsoft.com	Open source: http://www.cloud.google.com/appengine	Open source: http://www.cloudfoundry.org	Open source: http://www.heroku.com
Provider	RedHat	Microsoft	Google	Pivotal	Salesforce

Based on the above comparison and analysis, we have used Redhat's OpenShift for our study to implement the prototype of the proposed architecture. In OpenShift, Basic systems administration tasks such as virtual server provisioning, configuration, and scaling are automated. In addition, OpenShift allows a wide variety of choice of environments (UI, command line, IDE), flexibility and accessibility of different programming languages including PHP will benefit us greatly. As a result, this is the best platform for us to spend more of our time on code and less time on setting up development environments such as configuring operating systems and installing libraries and packages. As we have discussed in section 4.2.1 of this chapter, OpenShift is offered in three forms or versions and our focus is on OpenShift online development platform which is public platform and the version used by startup program freely. It creates a file system layout and provides disk space, CPU resources, memory, network connectivity, and an Apache or JBoss server to create, deploy, and manage applications in the cloud. Hence, we can easily and quickly develop our capacity building training system prototype, we can push the code to build and run on OpenShift through the git (software version control solution), then OpenShift online orchestrates where application containers will run and manages our application to ensure its availability for end users, that means it creates universal and unique URL for our application in order it to be accessible globally.

4.3 OpenShift Online

4.3.1 Overview

OpenShift Online by Red Hat is a Platform as a Service (PaaS) that provides developers and IT organizations with an auto-scaling, cloud application platform for deploying new applications on secure, scalable resources with minimal configuration and management overhead. Also it supports a wide selection of programming languages and frameworks [61].

Table 4.2: the features and benefits of OpenShift Online [62]

Features	Benefits
Self-Service and On-Demand Application Stacks	By enabling Developers with the ability to quickly and easily deploy application stacks, the OpenShift platform can increase productivity and foster innovation in application design and delivery.
Standardized Developer Workflows	With OpenShift, the Application Development organization can standardize the Developer workflow and create repeatable processes for application delivery in order to streamline the entire process.

Polyglot – Choice of PL and frameworks	Developers can choose between Java, Ruby, Python, PHP, Perl and other languages on the OpenShift platform allows them to choose the right tool for the job, and make a different choice for each project as needed.
Built-in Database Services	Choice of database technologies available in OpenShift automatically connected into the application stacks as needed, relational and modern NoSQL data stores.
Extensible cartridge	Developers can add other language, database, or middleware components that they need via the customizable OpenShift Cartridge system.
Multiple interaction modes	Provides multiple interaction models with the PaaS enabling developers to use a rich set of command-line tools, a powerful multi-device web console, or an Eclipse-based Integrated Development Environment.
One Click Deploy	Deploying to the OpenShift platform is as easy a clicking a button or entering a "Git push" command.
Automatic Application Scaling	The OpenShift platform enables cloud elasticity by providing automatic horizontal application scaling as application load increases.
Source Code Version Management	The OpenShift platform includes the Git distributed version control and source code management system.

4.3.2 Gears and Cartridges

OpenShift Online have two basic components called **Gears** and **Cartridges** [61]. Table 4.3 describes the basic system components of OpenShift Online. A **gear** is a server container with a set of resources that allows developers to run their applications and it run on OpenShift in the cloud. Currently, there are three gear sizes available with OpenShift Online: small, medium, and large. Each size provides 1 GB of disk space. The large gear has 2 GB of RAM, the medium gear has 1 GB of RAM, and the small and small gears have 512 MB of RAM. By default, there are three small gears available with a total of 1.5GB of RAM and 3GB of disk space. OpenShift Online can assign these three gears to a single application and its cartridges.

Cartridges are the plug-ins that houses the framework or components that can be used to create and run an application and used to support various programming languages, databases, monitoring services, and management. One or more cartridges run on each gear, and the same cartridge can run on many gears for clustering or scaling. There are three kinds of cartridges: web/standalone cartridges, Add-on/embedded cartridges and downloadable cartridges.

Web/standalone cartridges are the languages, frameworks or application servers that are set up to serve the web content, such as JBoss, Tomcat, Php, Python, or Node.js. Having one of these cartridges is sufficient to run an application. **Add-on/embedded cartridges** provides extra functionality and capabilities to application, such as a database like MySQL, MongoDB and PostgreSQL and Monitoring and Management like PhpMyAdmin, HAProxy, Cron, Switch Yard, RockMongo and Jenkins Client, but cannot be used on its own to create an application. **Downloadable cartridges** are available for new and existing applications along with the supported standard OpenShift Online cartridges. These are custom cartridges created by users, or available from the OpenShift community. These cartridges are downloaded and installed using the URL to the manifest of the hosted downloadable cartridge.

Table 4.3: OpenShift Online system components [61]

System Component	Description
Gears	Resource-constrained containers for application code where cartridges run. Gears determine the amount of RAM and disk space available to a cartridge.
Cartridges	Cartridges provide the functionality to run applications. Numerous cartridges are currently available to support languages such as Perl, PHP, and Ruby, as well as many database cartridges, such as PostgreSQL and MySQL.

Table 4.4: OpenShift Online application components [61]

Component	**Description**
Domain	The domain provides a unique group identifier for all the applications of a specific user. The domain is not directly related to DNS; instead, it is appended to the application name to form a final application URL of the form http://*App_Name-domain.example.com*
Application Name	The name of the application is selected by a user. The final URL to access the application is of the form http://*App_Name-domain.example.com*
Alias	DNS names can be provided for the application by registering an *alias* with OpenShift Online and pointing the DNS entry to the OpenShift Online servers.
Git repository	A Git repository is used to modify application code locally. After the code is applied, the **git push** command is required to deploy the revised code.

4.3.3 Developer Interaction

As outlined in section 4.2.1 of this chapter, there are two ways to interact with OpenShift: the Management Console or the Client Tools [63]. The OpenShift Online Management Console is a graphical interface accessed with a web browser and is best suited for:

- Setting up
- administering and managing accounts
- Launching new applications
- Managing and monitoring applications.

While, the OpenShift Online CLI tools/rhc client tools are used to manage a cloud environment using a command line interface, and provide features that are not currently available in the management Console. The client tools are best suited for:

- Coding
- Debugging
- Advanced application management.

To work with OpenShift online client tool, we must install the rhc client tool in our machine and because we have developed this thesis work using windows operating systems; we have described the steps of installing OpenShift Online client tools on Windows operating systems. There are three general steps, these are: step1: installing Ruby programming language step2: installing Git software and step3: installing the client tool. The first two steps are pre-request softwares to install the rhc client tool. So before going to the client tool installation, we must install the two required softwares Ruby and Git (**Appendix II**).

Chapter Five: Design of proposed architecture and related issues

5.1 Underling Concepts

This chapter describes the architecture of healthcare capacity building resource sharing system. In order to design this architecture to provide sharing of capacity building resources as a service, we have taken inspiration or concepts from IBM CCRA [64], NIST cloud reference architecture [65] and available literatures on software architecture approaches for cloud based systems. In the CCRA reference architecture, cloud business models and corresponding architectures have been categorized based on seven cloud adaption patterns. It identifies common architectural patterns for each cloud adaption pattern which describes the technology that underlie each type of cloud implementations. The reason why we have used these two reference architecture is: 1) it shows clear way of how we construct the architecture across all deployment models by providing the steps we follow and the required components which saves time. 2) It meets necessary cloud requirements like elasticity, self-service and flexible sourcing. 3) It exploits comprehensive architectural principles to ensure design scalability for efficiency, security, service management and virtualization as well as to reduce errors throughout the entire development process.

Also we have used graphical elements or symbolic representations of icons for cloud applications from the work of Christophe F. et al. [66], which presents a collection of patterns to describe cloud computing, its service models, its deployment models and its architectural patterns in general. As a result, they have mentioned three types of cloud application architectural patterns such as elasticity pattern, availability pattern and multi-tenancy patterns. They have clearly discussed their context, challenges of each type of patterns and they have put the solution for each type of pattern with proof of concept by: sketching these patterns; analyzing their results and relating with other patterns which affect them.

Furthermore, our work intends to introduce cloud based capacity building resource sharing architecture with two essential parts: the front-end and back-end; that are usually connected via internet. The first part is to illustrate the different health delivery institutions and client’s infrastructures like computers, laptops, mobile phones as well as the network that is used for accessing the cloud capacity building resource sharing system. Simply, the first part consists of the visible interface and applications (example web browser like chrome, IE or Mozilla Firefox) required to access capacity building resource sharing system from the cloud providers. The second part is to illustrate the healthcare capacity building program basic components and services which is the cloud part of the architecture. It consists of all the required resources to access and share capacity building resources to the end users. So it includes various servers, middleware, storages, virtual machines, programming languages, platforms, the capacity building resource sharing system itself, security and management mechanisms, deployment models and other pool of resources. The more details about the cloud part of the architecture are presented in section 5.3 of this chapter. The first part which is the front/client end of the architecture is connected with the cloud part of the architecture through the internet (Fig 5.3).

5.2 Design Goals and Constraints

The overall goals of cloud based capacity building resource sharing architecture design is to provide resource sharing system which meets the needs of healthcare workers related to building their capacity. As a result, the architecture should be flexible and easy to use and learn with low cost, increased productivity, well-defined interface and user-friendliness. In this study, we have given concerns on some of quality requirements described in section 5.3.2 of this chapter which includes scalability, cost and well-defined interface as a design goal.

Constraints are a limiting factor and severely restrict options for making design decisions. In software architecture design, there are two types of constraints **technical** and **non-technical or business** constraints [67]. In this study the following technical and non-technical constraints were considered. Technical constraints include: **Programming language: -** we have used php programming language to develop the prototype of the architecture due to the following reason. Easy to code and debug and more templates are available freely to use as a starting resource also we are more familiar with php than other languages. **Operating system or platforms supported: -** this study works properly on windows and Android operating systems with the support of OpenShift Online Platform. **Use of a specific library or framework: -** we have used Drupal framework to develop the prototype. Non-technical constraints include: **Schedule: -** this is the time frame for final delivery of the thesis work, so we have stuck with our schedule during this work. **Budget: -** this is the money provided to accomplish the work which is fixed in this level, so it restricts us from collecting the required data and tools extensively. **Software licensing restrictions: -** we are restricted to use freely available softwares and tools to conduct this thesis work which lacks full features. **Internet-based: -** all cloud based applications are working properly on dedicated internet connection; due to this, slow internet connection may affect the accessibility or availability of the system.

5.3 Design Requirements

5.3.1 CCBRSA Features as a Requirement

These are the functional requirements of the capacity building resource sharing system which are related to functional aspect of the system. As a result, they define functions and functionality within and from the prototype system.

- It should register health practitioners and other users.
- It should support capacity building resource sharing based on cloud computing.
- It should use as healthcare learning environment.
- It should create collaboration among hospitals, capacity building providers and health workers.
- It should support sharing of experience electronically which include best practices, challenges and their solutions, professional ethics, problem solving mechanisms etc.
- It should support uploading and downloading of CB materials and tools.
- It should support watching videos online related to different issues and technologies
- It should support searching of the available CB resources, tools and approaches.
- It should provide FAQs, their answers, online discussion and activities.

5.3.2 CCBRSA Quality Requirements

These are the non-functional requirements of the capacity building resource sharing system. They are intended to support the functional requirements concerning to user interface, scalability, availability/accessibility, cost, performance and others.

- User interface is the only way for users to perceive the capacity building system; hence, it must be attractive, easy, simple, clear, consistent and responsive and provide help information as well as have purposeful layout, color and text. Otherwise the functionalities of software system cannot be used in convenient way.
- In cloud based system, scalability is the basic characteristics it must have, which is the ability of a system to either handle the load increase without impacting the performance

of the system or the ability to be readily enlarged. Hence, the cloud based capacity building system will be scalable in its features. When the user of the system increases, it increases the RAM and storage space automatically.

- Availability in cloud computing, defines the proportion of time that the system is functional and working. It can be measured as a percentage of the total system downtime over a predefined period. In our case, it will be available every time and everywhere using any device.
- Cost is one of the quality attributes that is, cost of the system with respect to time to market, expected project lifetime & utilization of legacy. So cloud is cost effective because there is no initial investment cost to build and run the system which is done by cloud providers.
- Performance is the indication of the responsiveness of a system to execute any action within a given time interval. With this regard, cloud based capacity building system is highly responsive because the resources are put on distributed and high performance servers. So the client will receive the resources in fastest way.

5.4 General cloud framework for the CCBRSA

The core function of designing cloud based capacity building resource sharing architecture is to create global healthcare knowledge sharing network to have one capacity building resource sharing center for all health workers and at all level. Hence it facilitates the access of capacity building tools, approaches and resources that are dispersed in different healthcare delivery institutions and capacity building providers from one place. In addition the process of designing cloud based capacity building resource sharing architecture will be progressive. Hence, according to the present condition of capacity building program, cloud computing expertize and OpenShift online, the design of cloud based capacity building resource sharing architecture can familiarize the following generic cloud framework shown in fig 5.1. It contains the below mentioned components.

Physical resource pool: it includes hardware resources such as network, storage equipment and virtual server. These dispersed resources are connected together through network virtualization with the intension of creating cloud centric healthcare knowledge sharing network and shared capacity building tools and materials.

Infrastructure as a service (IaaS): The healthcare delivery organizations and end users can share full hardware resources by virtualizing physical resources. Therefore the virtualization technology increases the user need by dynamically allocating resources to the active need so it avoids the usage of fixed storage space. It is fully managed by the cloud provider and it manages resource and environment of the physical infrastructures like server, storage and networking to the virtual machine.

Platform as a service (PaaS): It contains the OpenShift platform which will control the development environment to enable creation and hosting of applications and programs with the ability to access public clouds resources and platforms.

Software as a Service (SaaS): The end users orders application software service that are deployed by the platform service providers through the network. The service providers gather payments based on used services. And different services such as sharing of workplace training resources, research capability training resources, shared experience, shared tools and materials management and others can be provided on this level.

OpenShift Online: As we have described in chapter four, it is Redhat's OpenShift online offering. This platform is used to create and deploy capacity building resource sharing system. On the capacity building provider's (General hospital) center, there are individual level, community/beneficiary level, organization level and system level capacity building components which will adapt different approaches and tools.

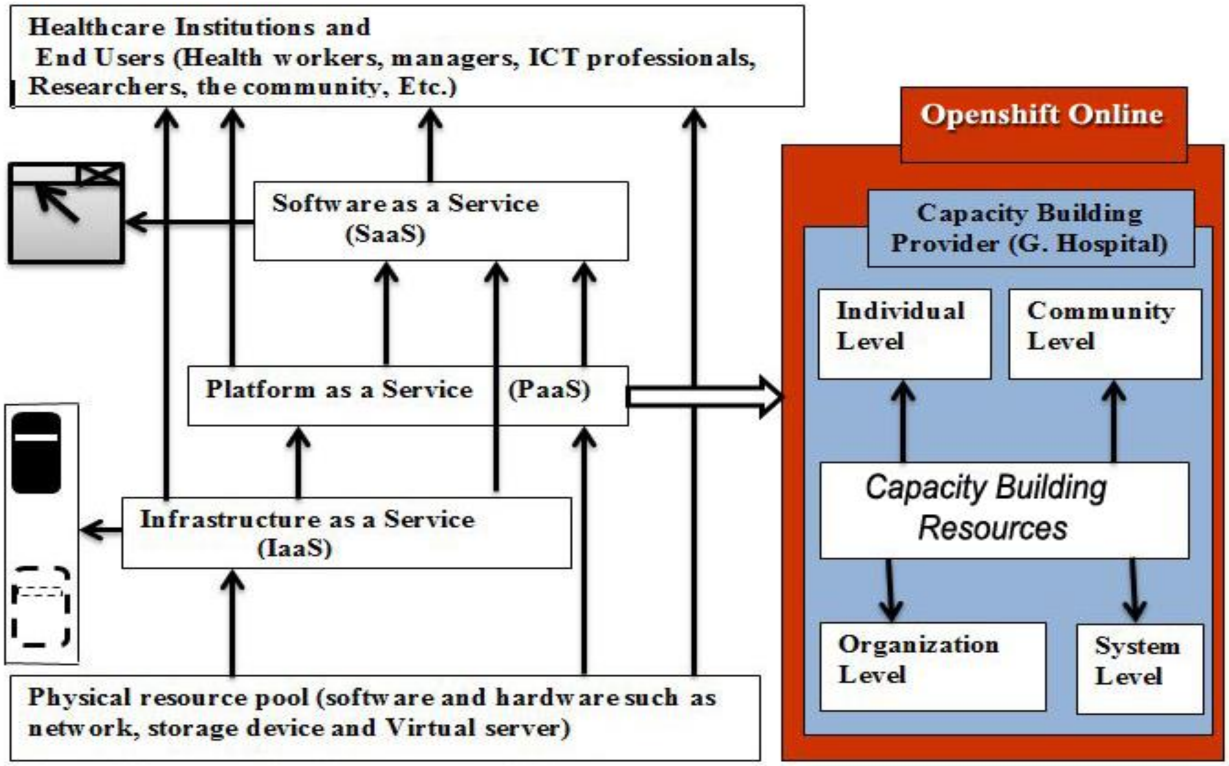


Figure 5.1: General cloud framework of CBP program

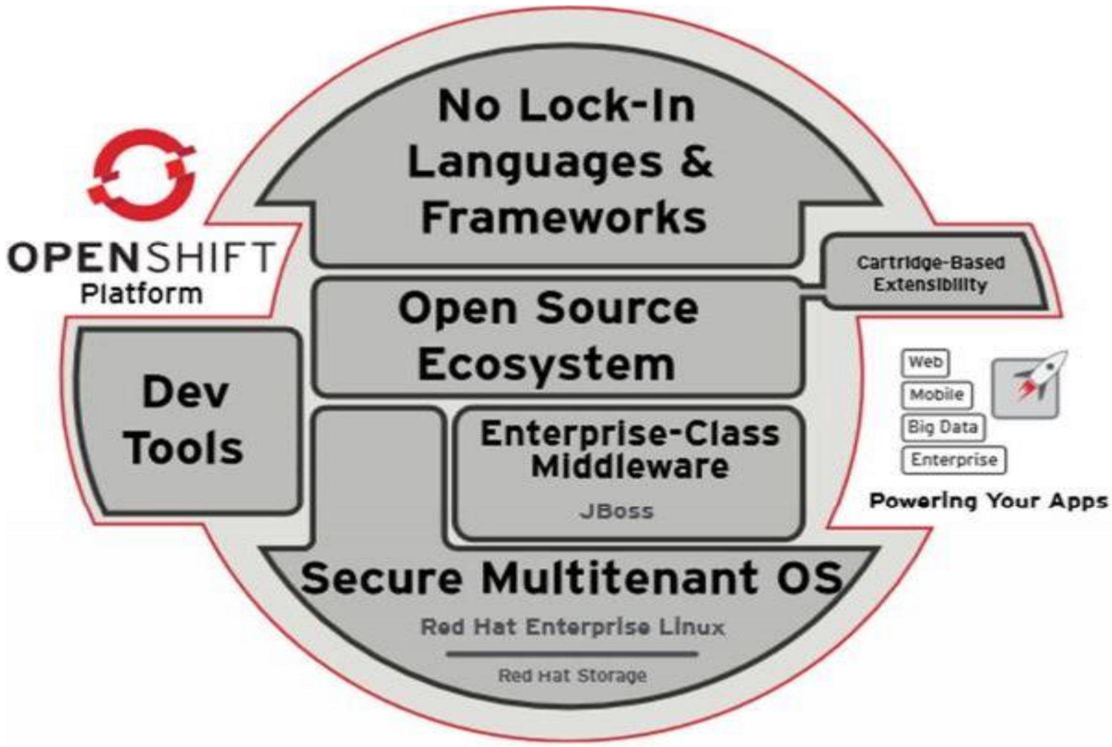


Figure 5.2: OpenShift framework architecture (source: [73])

5.5 The suggested Architecture

Based on the above stated underlie concepts, the description of client and cloud side parts and the general framework shown in Fig 5.1, this study suggests the following architecture to implement cloud computing as a tool to provide, access and share capacity building resources. Furthermore, it is designed to realize the capacity building resources and tools sharing and to provide worldwide resource sharing services for the healthcare practitioners working in the different health sector of Ethiopia.

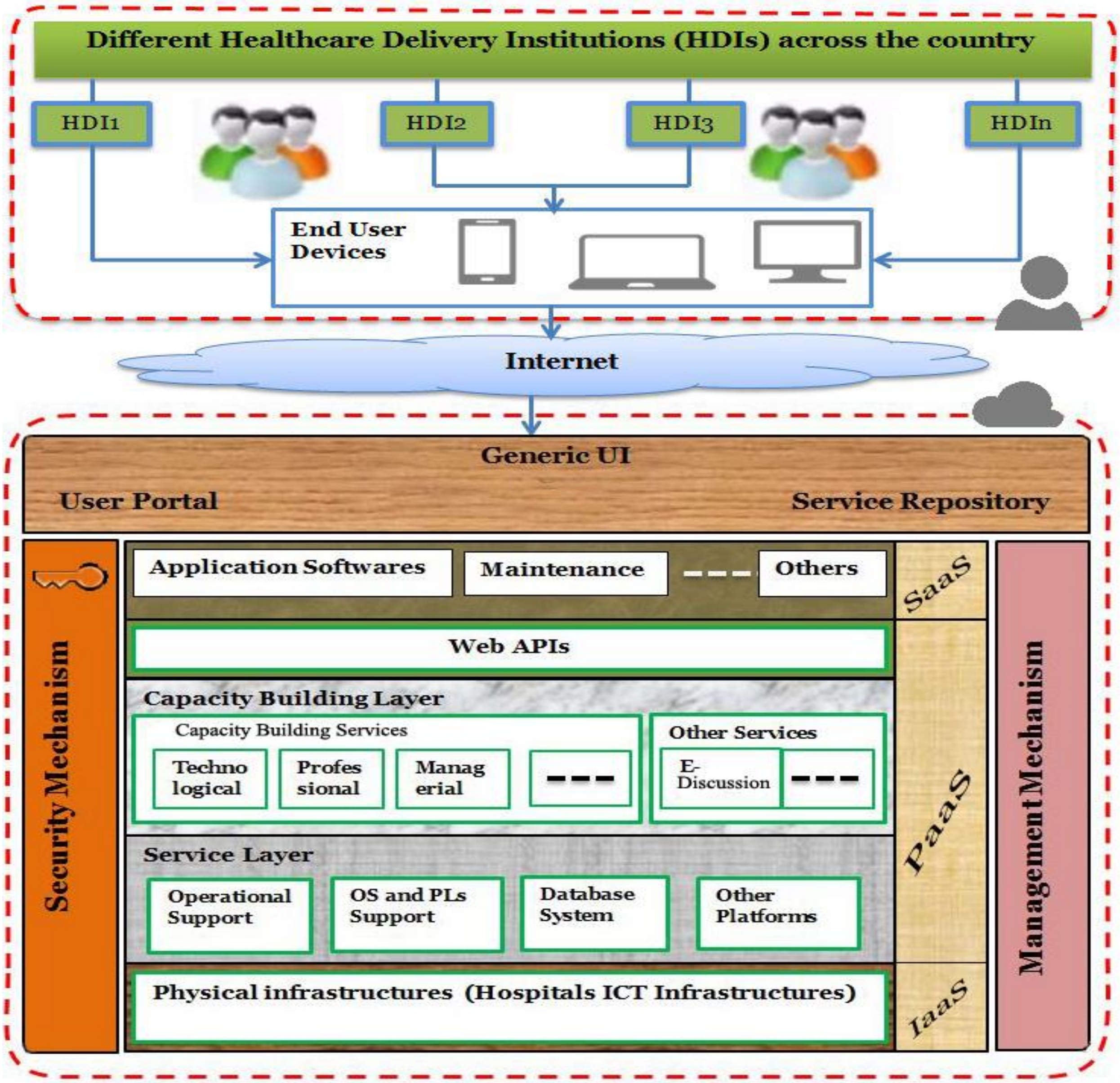


Figure 5.3: Conceptual CCBRSA for healthcare CBP framework (Source: Own Work)

The proposed architecture integrates healthcare delivery institutions and cloud using dedicated internet connection as shown in fig 5.2 and it contains four layers and two modules to manage the user access and service delivery. In the community cloud CCBRSA, the users can request any capacity building services from the cloud provider. The details of CCBRSA layers are described below.

5.5.1 Generic User Interface Layer

The generic user interface layer is the top layer of the CCBRSA architecture which is connected with the front-end part of the architecture via internet and act as an interface between the end users and the cloud services. It provides support for the capacity building system to handle request from multiple types of end user devices, tracking requests to the correct end users and adjusting capacity building materials and resources according to the specific need of the users. So it is also used for authentication and authorization mechanism. It contains two sub-components. **User Log** which is used to store user information and their access level such as user name, password, user type and others. Also it manages and supports the resource utilization by keeping notes about the resource requirement and their status; **Service Repository** which contains different services and their detail.

5.5.2 Software as a Service (SaaS) Layer

SaaS layer is the second layer of the architecture below the generic user interface layer which allows access to the hosted capacity building system and any third-party application software to run on the cloud platform. Services provided in the PaaS layer can be utilized by sending request messages through Web APIs. So healthcare institutions and users can access capacity building resource sharing system and other third-party softwares through web interface.

5.5.3 Platform as a Service (PaaS) Layer

PaaS layer is the third layer of the architecture which is deployed on cloud provider that can support requirement of the designed platform. It consists two sub-layers called Capacity building layer and service layer.

5.5.3.1 Capacity Building Layer

It is the main layer of the platform for capacity building services which includes resource sharing services and other services. The resource sharing services are : (1) professional resource sharing which includes Laboratorial training resources, Pharmaceutical training resources, emergency related training resources, Midwifery related training resources, patient care training resources etc., (2) technological resource sharing which includes EMR-related technology training resources, specific medical equipment training resources, computer skill training resources etc., (3) managerial resource sharing which includes budgeting and financing related training resources, hospital management and problem solving related training resources, policy and planning related training resources, procurement and resource management related training resources, quality service delivery related training resources etc., (4) community resource sharing which includes child and maternal health related training resources, prevention and treatment of epidemics related training resources, proper usage of medicine related training resources, modern life style, food and vaccination to be healthy related training resources etc., (5) shared tools and approaches and (6) shared-experience. Other service includes E-Discussion, Research-Capability training resources etc.

5.5.3.2 Service Layer

It provides basic services such as development environment like operating system and programming language, operational support, database management system, user management, middleware and other platforms.

5.5.4 Infrastructure as a Service (IaaS) Layer

IaaS Layer is the bottom layer of the architecture in which the cloud based capacity building resource sharing system is implemented. It contains the physical and virtualized infrastructure including healthcare delivery institution’s ICT Infrastructures.

5.5.5 Security and Management Mechanisms

Security is the capability of a system to prevent accidental actions outside of the designed usages which is still the big issue. In cloud computing there are a lot of security mechanisms to protect data and the whole system holding and processing it. Out of these mechanisms the following are very common and used almost in all cloud service providers in addition to their own security methods [68]. The mechanisms are Encryption, Hashing, Digital Signature, Public Key Infrastructure (PKI), Identity and Access Management (IAM) and Single Sign-On (SSO). In this study the IAM mechanism was used. IAM controls and tracks user identities and access privileges for capacity building resources, the knowledge sharing environment and the capacity building system. It encompasses components and polices such as authentication, authorization, user management and credential management which are already provisioned for IaaS environment. Also remote administration, resource management, SLA (Service Level Agreement) management and billing management are used as management mechanism in cloud computing. These mechanisms usually provide unified APIs and can be offered as single result, custom applications, merged into different output sets or multi-function applications [69].

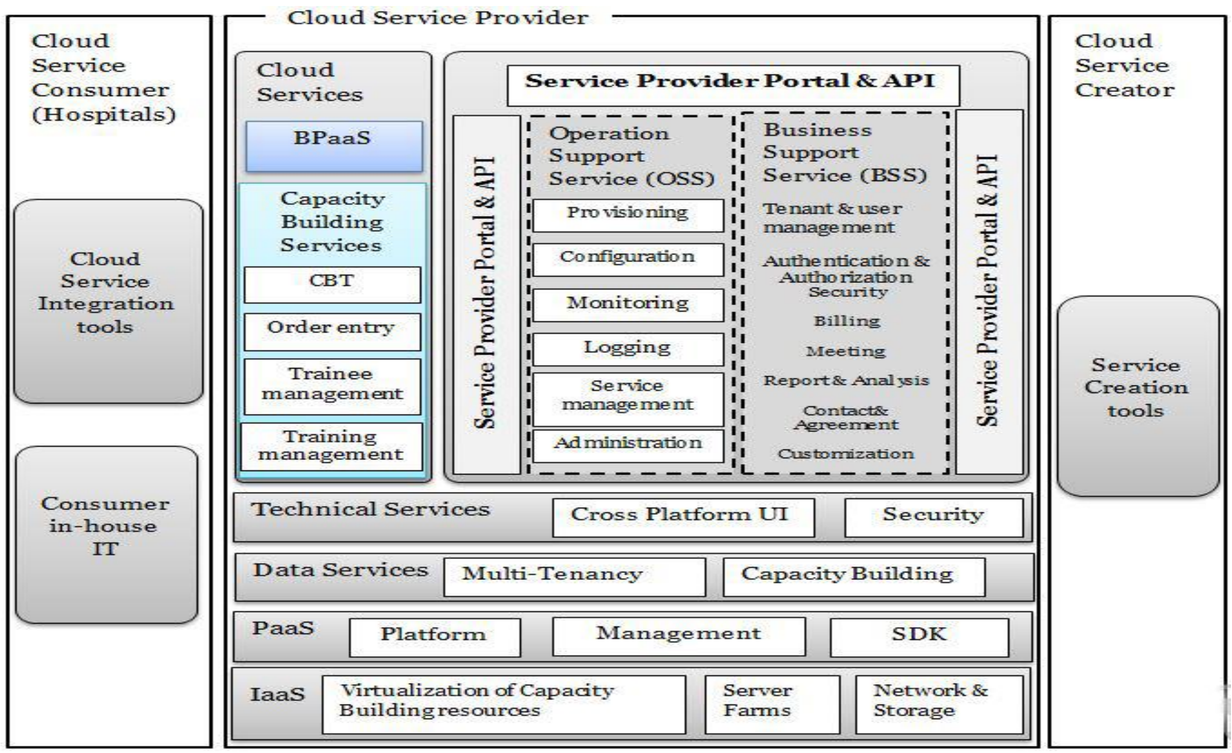


Figure 5.4: overview of functional view of CCBRSA (Source: adapted from [64])

As shown in the above diagram (Figure 5.4), the cloud based CBP system will work by logging into the service provider’s portal and by ordering capacity building services through the hospital. This service has been created by the cloud service provider or simply by OpenShift online and it is a web based application accessible using web browser. Then the OpenShift Online which is a cloud service provider in this case, will validate client request through the BSS, and if the validation is okay, it will provision the request through the OSS. In this case, the

OSS is used to provide the service; the BSS is used to validate the request; and the two components found in the left and right side of the diagram which are cloud service consumer and cloud service creator, will provide interfaces and tools for the end users and service creators. The infrastructure where the cloud based CBP will be implemented will support the different types of cloud (IaaS, PaaS, SaaS, BPaaS).

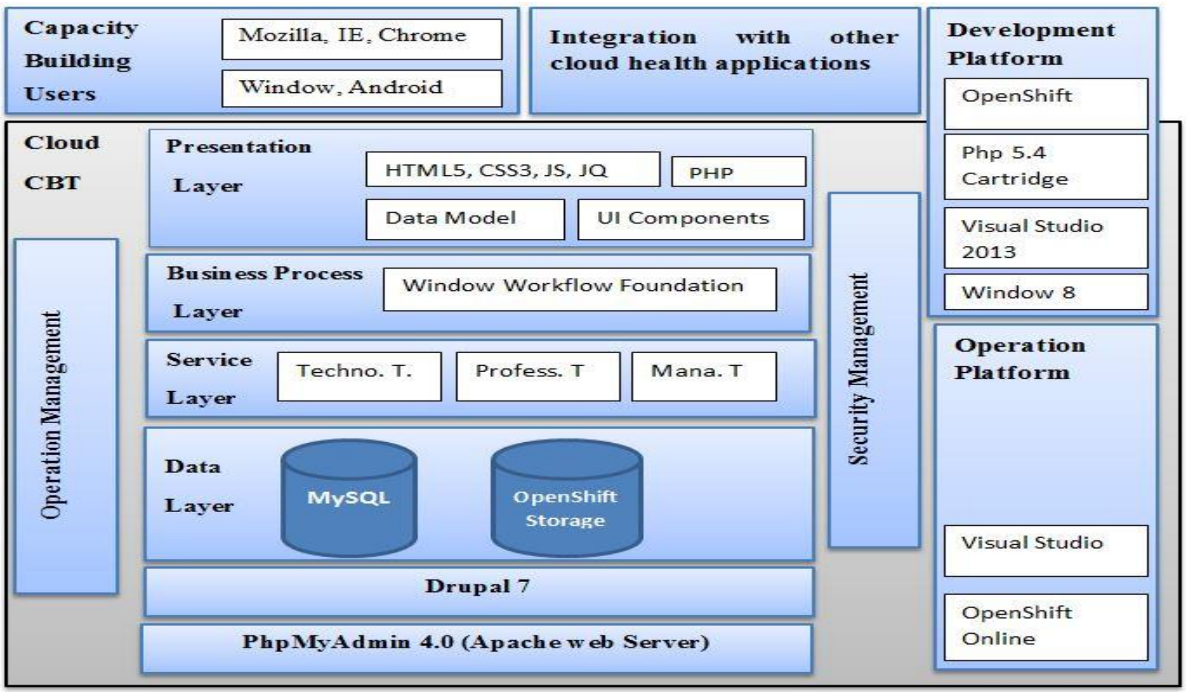


Figure 5.5: overview of development view of CCBRSA (Source: adapted from [64])

Figure 5.5 illustrates development view of the proposed architecture. In the CCBRSA, the capacity building users can access the system using different operating systems and web browsers. The UI of cloud based CBP system was created using HTML, CSS, JS, IQ and php as a main language, we have considered window workflow foundation for business process management. The integration of this CBRS system with other current healthcare system can be operated using JSON or other programing technologies. Different services including technological training resources, professional training resources, managerial training resources and other capacity building resources was included as a service. For the purpose of managing and storing the capacity building resources, we have used MySQL database, OpenShift storage, Drupal web content management and apache server. The development platform is OpenShift online with php5.4 cartridge; the different views of the architecture was developed using visual studio 2013 and ArgoUML.

Chapter Six: Development of the Prototype

In this chapter, we provide overview of the prototype implementation of the suggested architecture elaborated in chapter five. With this end, we are focusing on the implementation of some parts of the capacity building resource sharing system; that is the front end web application interface development on OpenShift online cloud platform, smart way of sharing resources, searching and accessing of these shared resources to improve medical practitioners ability, research and patient care, the accessibility of this web based interfaces from the OpenShift providers using URL of the system, workability of the linkage in between each menus, the portability or mobile and tablet device interface looking and desktop or laptop layout of the developed prototype and other important features for proof of concept . To implement the prototype of the architecture and to elaborate implementation details, the 4+1 view model of software architecture as an architecture description approach, which have presented by Kruchten [70] are used.

In the first sub portion of this chapter, we have described architecture description approach followed by use case of the system. Then we have described different view of the healthcare capacity building resource sharing system according to 4+1 view model of software architecture [70]. Finally, we have explained implementation details, testing and findings and discussions.

6.1 Architecture Description Approach

Software architecture description is the collection of ways for expressing, communicating and analyzing the architecture, often arranged into various views of the architecture. Using these views to describe the architecture of software-intensive systems, Kruchten have created the 4+1 view model of software architecture [70]. The views are logical, development, process and deployment and the scenario or use case that indicates the plus one dimension which are briefly explained as follows.

Logical View: This view represents the functionality of the system which is provided to the users as a service. It can be presented by sequence diagrams, class diagrams and collaboration diagrams.

Development View: This view denotes the details and organization of modules and subsystems or library that makes the system and concerned with software management. It illustrates the system from programmer point of view which is represented by component diagrams and package diagrams.

Process View: This view considers some non-functional requirements to describe the dynamic behavior of the system by elaborating activities of the system using UML activity diagrams.

Deployment View: This view represents the software to hardware mapping which is the physical topology of the system and represented by UML deployment diagrams.

Use Cases: This dimension is denoted using UML use case diagrams and represents sequence of interaction between system's features and their actors. We have chosen specific parts of the capacity building training architecture to provide implementation details of different services of the system and interaction between applications and services that are provisioned using the system.

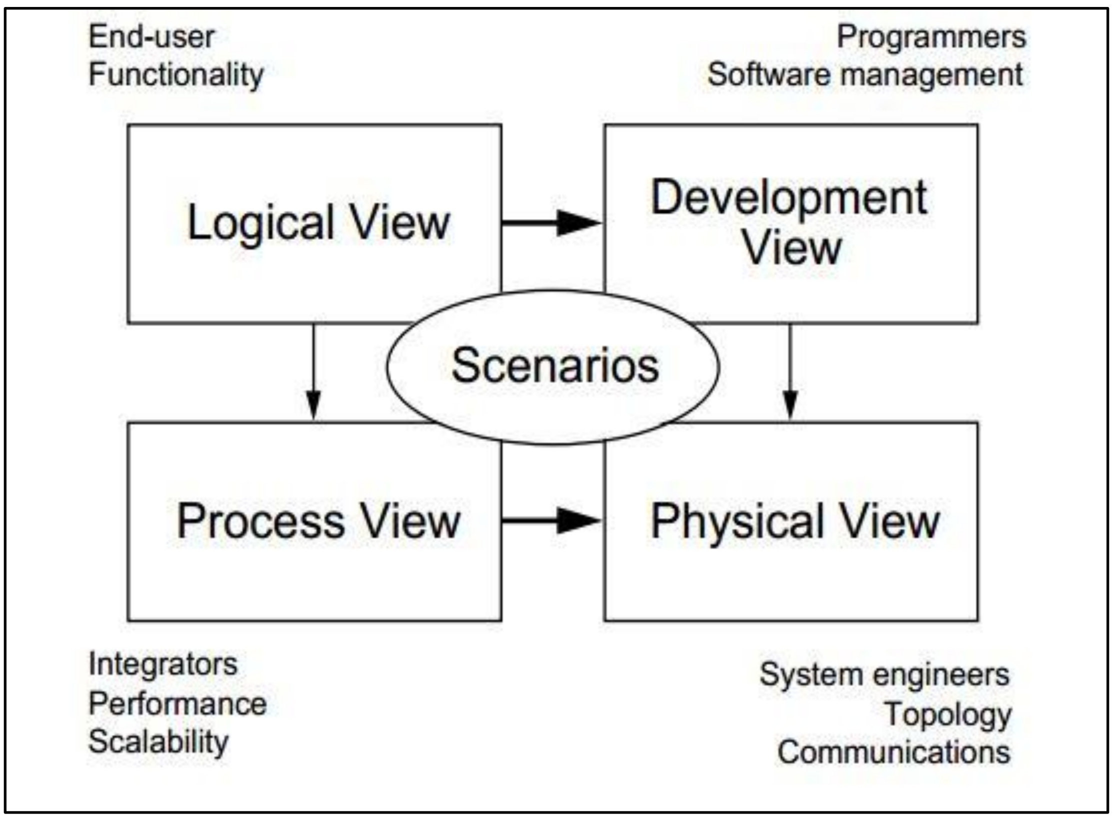


Figure 6.1: Kruchten "4+1" View Model (Source: [70])

6.2 Use Cases

Selective use cases of the capacity building resource sharing system are shown in Figure 6.1. The figure incorporated most important use cases of the prototype implementation. A quick description of each use case is provided in the following subsections.

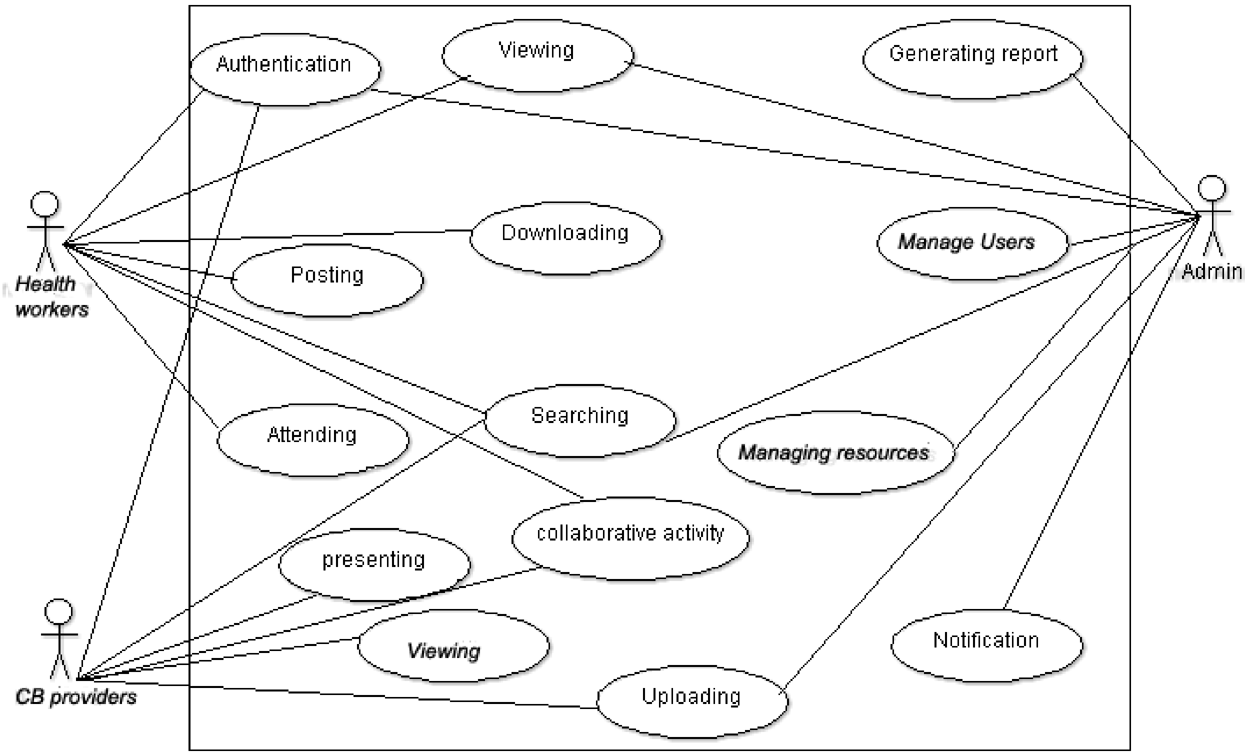


Figure 6.2: Use Cases

6.2.1 Authentication

Use Case Description: Before the capacity building materials, tools, approaches and their associated services provided to the end users from the OpenShift online cloud service provider, these need to be authenticated by the provider. Upon requesting the capacity building system services, the system asks the users to register. After registering to the system, the user will have User Name and password which is used to access the requested services.

Pre Condition: Open the healthcare capacity building resource sharing system web interface by writing the URL on the browser.

Post Condition: User added to the system and has user name and password.

6.2.2 Posting and retrieving comments, experiences and other activities

Use Case Description: The user comment, experience and other activities submitted to the site should be able to post for other peers.

Pre Condition: Login in to the system

Post Condition: The comments and other posted resources are pushed as they created.

6.2.3 Viewing and downloading resources

Use Case Description: The CBRS should support the viewing and downloading of provisioned resources to utilize the features of the system.

Pre Condition: The available resources provided by the cloud service provider must authenticate users

Post Condition: After successful verification, viewing and downloading of the corresponding capacity building cloud resources is done.

6.2.4 Reading the timely provided Articles

Use Case Description: This use case facilitates the users to directly access the online articles provided by any concerned users inside or outside of the country.

Pre Condition: User login

Post Condition: Articles delivered to users.

6.2.5 Searching the available materials, tools and approaches

Use Case Description: This use case used to find the resources without wasting more time by visiting the entire list of resources to select their required one.

Pre Condition: User inputs the keyword (name or title) to find the available resources

Post Condition: The searched resources will be displayed for the users.

6.2.6 Collaborative Activities

Use Case Description: The healthcare capacity building system provides collaboration among the healthcare practitioners throughout the country and with outside world.

Pre Condition: The user creates collaboration group

Post Condition: There will be collaboration among the healthcare practitioners throughout the country and with outside world.

6.2.7 Writing/Adding Articles online

Use Case Description: This use case adds articles and other important notes by differentiating the article category.

Pre Condition: The author prepares the content.

Post Condition: The articles are available to users.

6.2.8 Uploading resources/materials, tools and approaches

Use Case Description: The capacity building providers prepares the required resources to share and categorizes these resources and tools for ease of accessibility. During this process the capacity building providers share the resources using the system and these resources should be provisioned to the remote servers.

Pre Condition: Login to the system

Post Condition: The resources are available online.

6.3 Logical view

The logical view of the CBRS system is explained by using a sequence diagram, two diagrams showing the capacity building resource sharing or uploading and accessing features of the CBRS system. The first diagram, figure 6.3 shows how the capacity building providers and other concerned bodies uploads and publishes resources, tools, approaches and other materials. The second diagram, figure 6.4 shows how the users access the capacity building resources from the CBRS system. The communication between the user’s web browser and the CBRS system is done via HTTP, due to the ability of HTTP to pass information in fastest way [71]. Also HTTP is best because almost all modern browsers understand this protocol and it fulfills accessibility requirement [72].

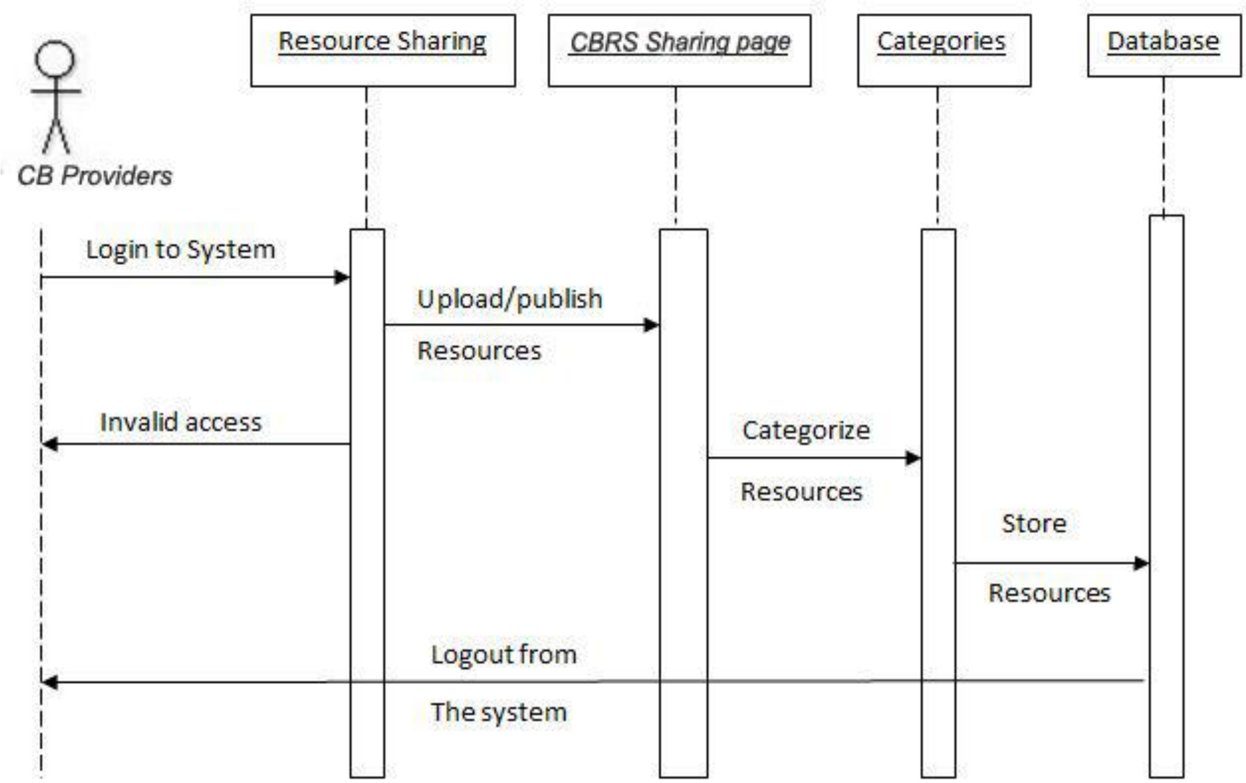


Figure 6.3: Sequence diagram-capacity building providers uploading resources

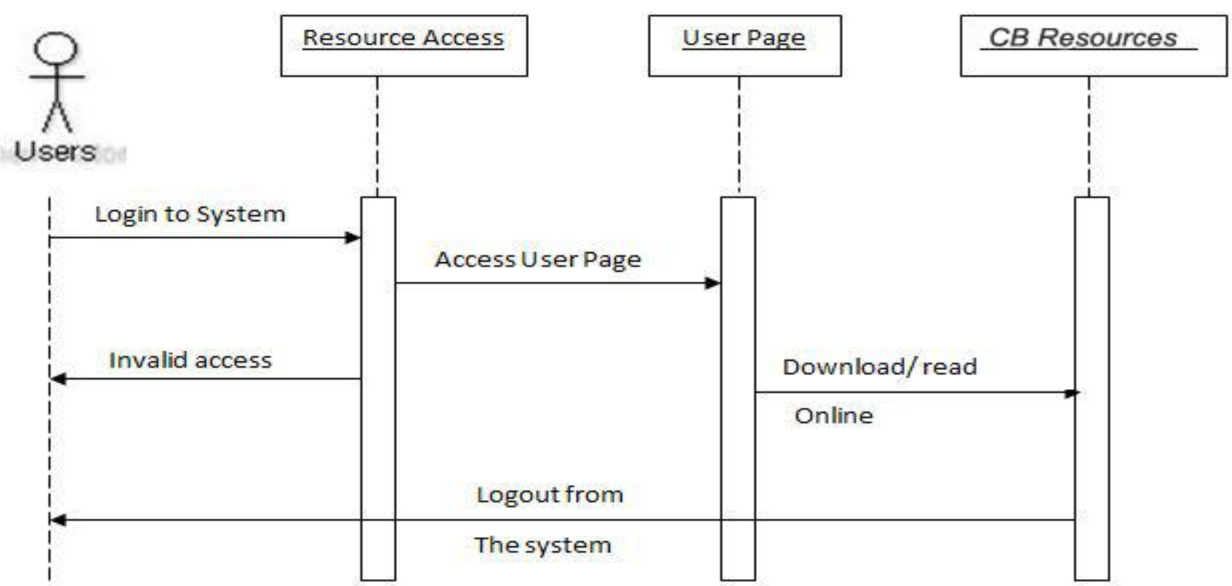


Figure 6.4: Sequence diagram-users resource access

6.4 Process view

The process view is illustrated through activity diagrams corresponding to registering, logging and accessing the CBRS system. Figure 6.5 shows the registration and login process to be part of the CBRS system users and have access privilege to the system. The activity begins by requesting the registration and login through the Registration and Login features of the system. During this process, the registration form is available to users and if they fulfill correctly, the users registered to the system. But if the user did not fulfill the form correctly, the error message is displayed for them and they fulfill the registration form again until they fulfill the form correctly and the process will be finished. In the login activity diagram, the user enters the User Name and Password after requesting the login and login will be validated. If the validation is ok, the process will end otherwise the invalid login notification will be displayed and they request login again until they make valid login.

Activity diagram for the overall CCBRS system services are shown in figure 6.6. The user selects the services or features at the top of the healthcare CBRS system interface and they read the details of the selected service. If the user is not interested with the selected service, he/she will select another service. However, if the user wants the selected service, he/she starts to access the service, finish their task or activities associated with the selected feature and the process will end. On the other hand, if the user has comments related to the accessed service or has data to share for others, they prepare their comment or data, they select the browse menu, they provide their comment or data and they finish the process by publishing or uploading their comment or data to others.

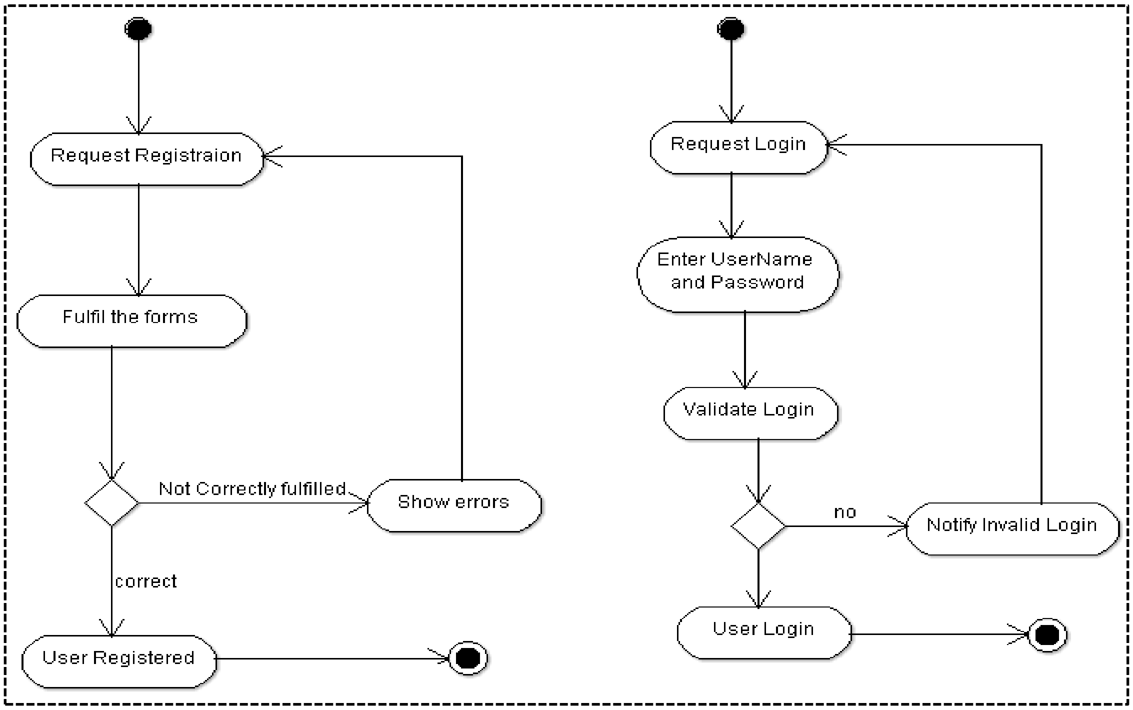


Figure 6.5: Activity diagram for registration and login

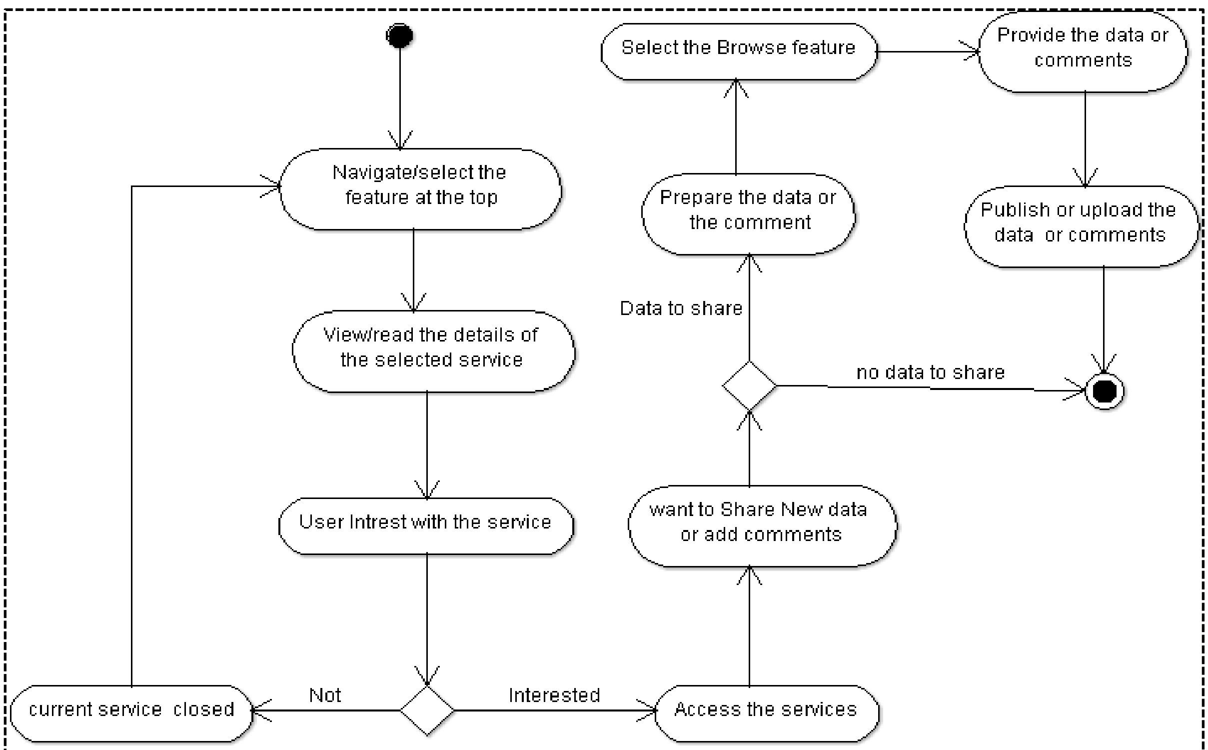


Figure 6.6: Activity diagram to access the overall CBRS system services

6.5 Data model

Data model primarily supports the functional requirements and they define the services that the system should provide to the users. The data model used for the CBRS system is shown in figure 6.7. In order to avoid unnecessary complexities, we are not providing details of the entities inside each entity. Inter entity relationships are logical relations maintained in form of aggregation and composition of services corresponding to the entity whereas the relationship between entities is maintained at services level, i.e. relation between entities inside entity is maintained at database level in form of referential integrity constrains. The CBRS system has from one to several users. And the CBRS system has one to several repositories / databases. The repository has zero to several capacity building resource stored in the database.

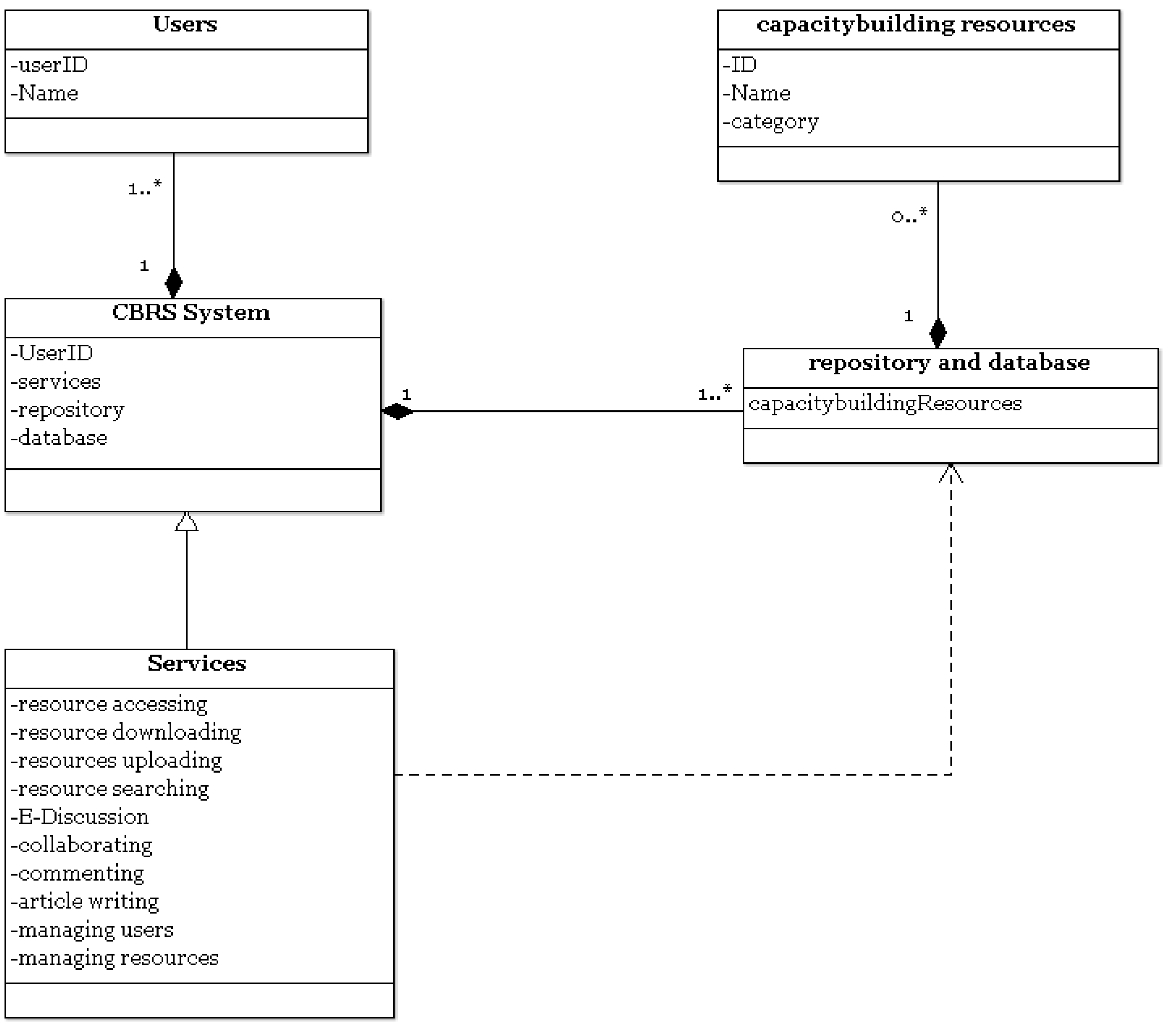


Figure 6.7: Data Model

6.6 Implementation Overview

The CBRS cloud portal is designed to be easy to navigate, simple to look at, easy to access the available resources and experiences, user friendly and easy for healthcare capacity building providers and other voluntaries who share their medical data and knowledge for others. The technologies that were used for development of the prototype are:

- JS, JQ, HTML and CSS for designing the web interface of the web application
- PHP language for backend side of the prototype
- OpenShift online platform for creating and testing the CBRS application
- Web and embedded cartridges to support languages and databases on the OpenShift online
- RHC client to manage the code
- Git repository to clone and deploy locally developed prototype on OpenShift online

The prototype is designed for both desktop and mobile devices. Figure 6.8 shows the developed application in OpenShift online web console with the associated gears and cartridges. A screenshot or web interface of the homepage for desktop or laptop view and mobile view is displayed in Figure 6.9 and 6.10 respectively.

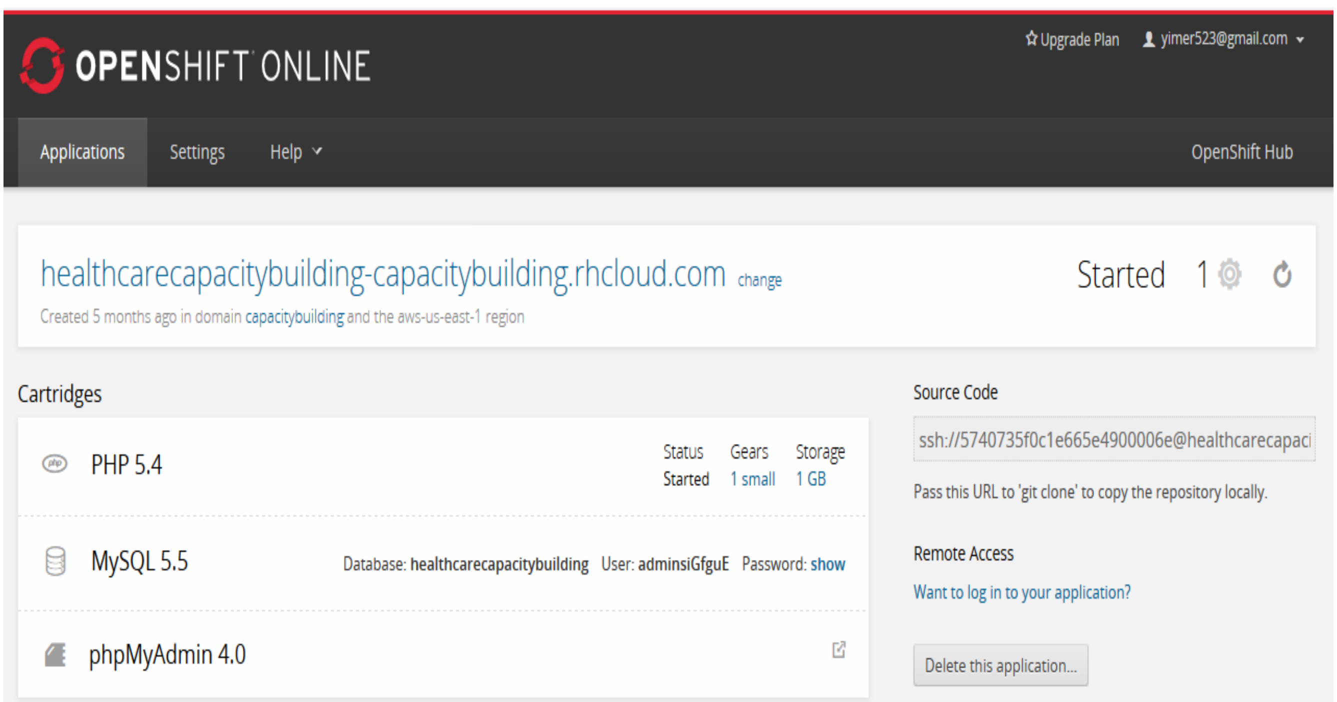
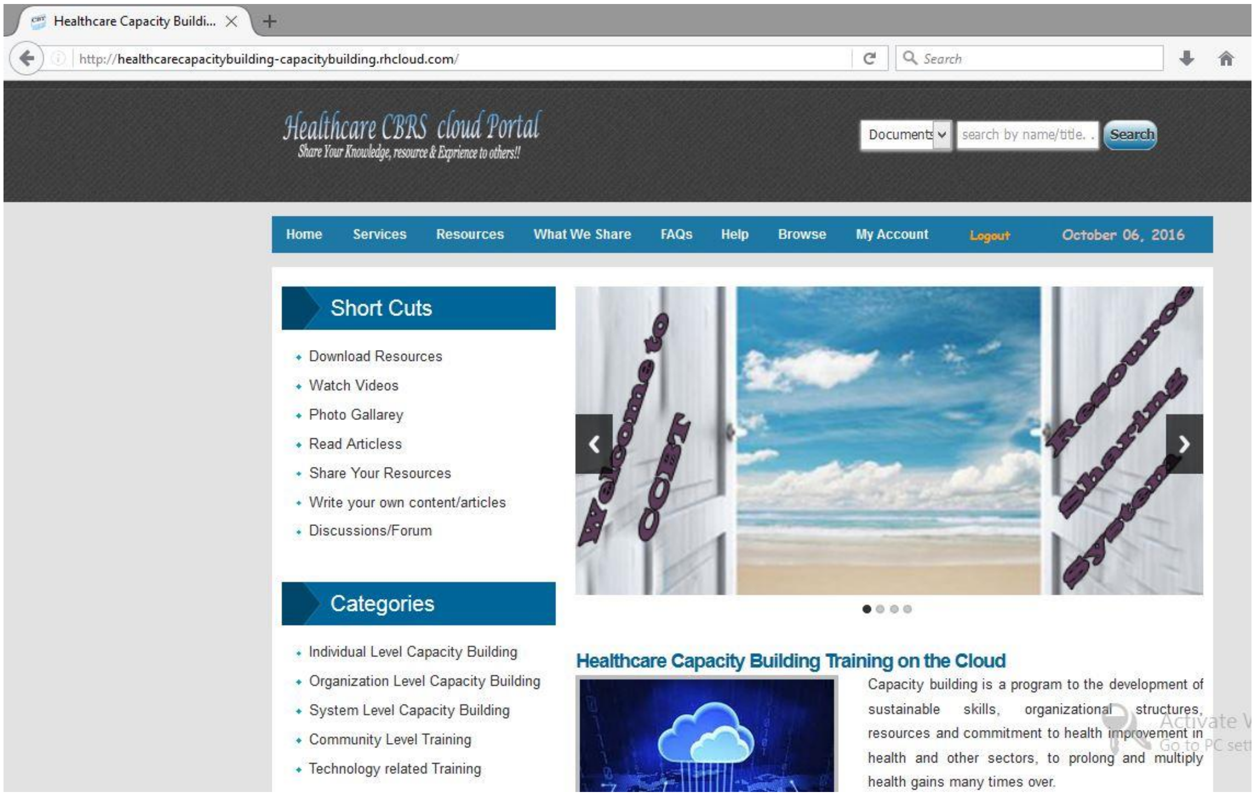


Figure 6.8: The developed healthcare capacity building application on the cloud



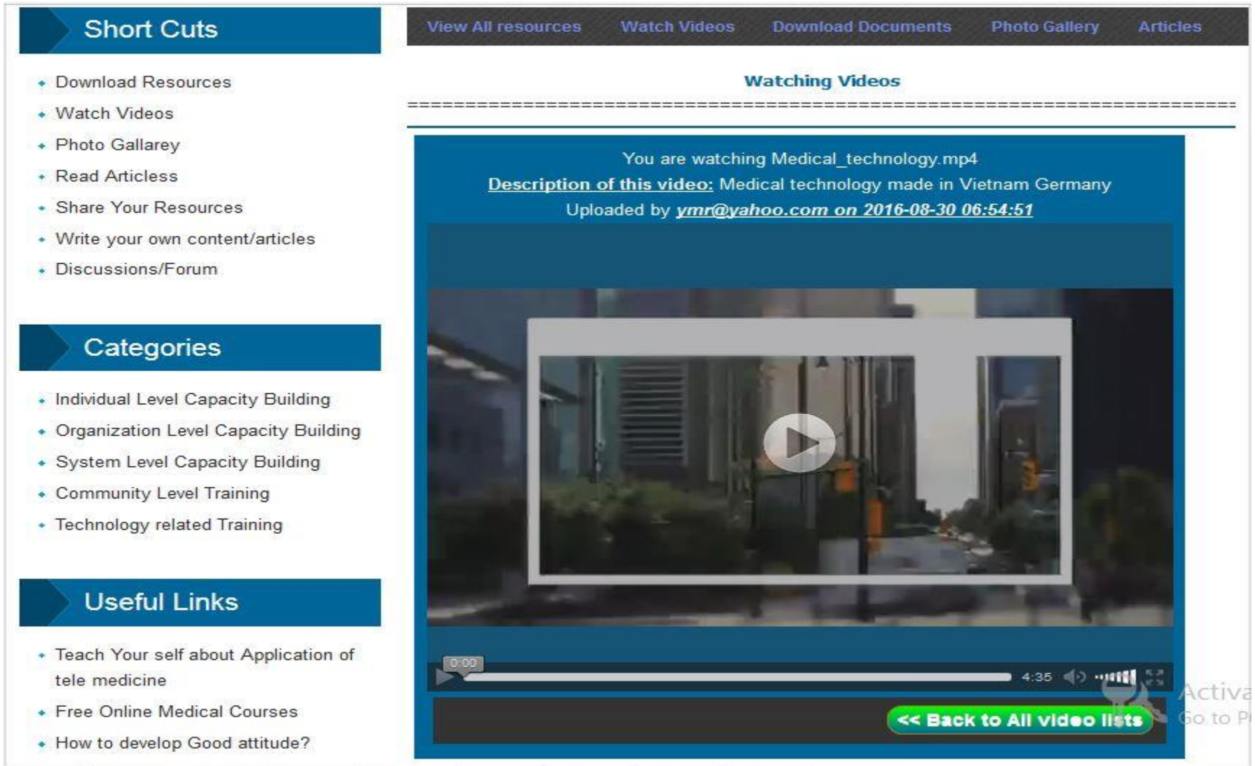


Figure 6.9: CBRS cloud portal desktop interface front page and watching video looking respectively

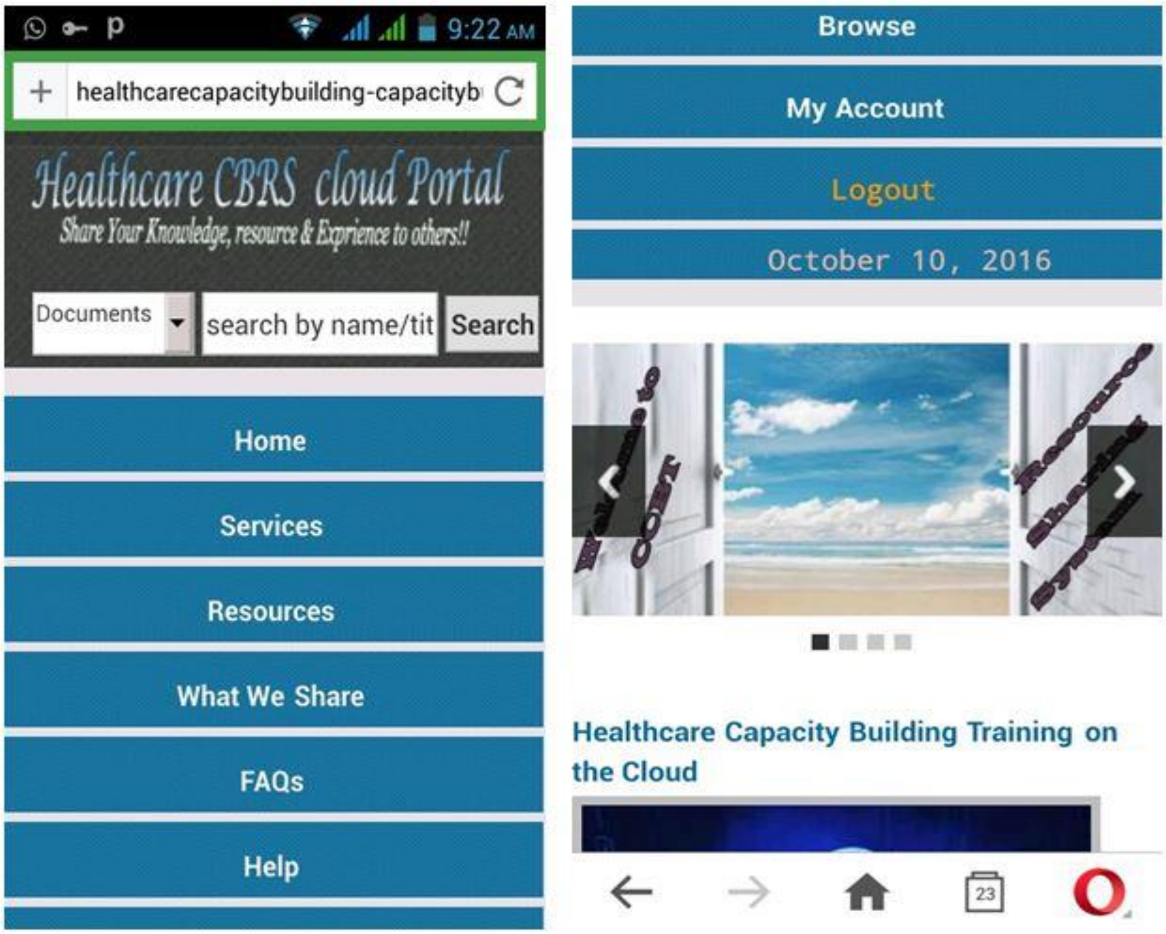


Figure 6.10: CBRS cloud portal mobile interface

6.7 Testing

To check the workability of the developed cloud portal, the CBRS system test was performed through unit testing, integration and usability testing.

6.7.1 Unit testing

The CBRS resource and knowledge sharing center has been tested in perspective of resource sharing and gaining access to these resources. Both resource sharing and resource accessing were tested in OpenShift online Redhat's platform after it has been implemented using different web and cloud technologies. All the concerned bodies including the capacity building providers, medical practitioners, HDI's ICT professionals and others provide their resources and knowledge to this common alliance central cloud place and the administrator manages it in the cloud. Each CBRS cloud interface is tested independently based on the following test case

Table 6.1: Sample unit test case

TC ID	TC Objective	TC description	Expected output
1	Registering users by identifying their level	Click on registration from the front page ; provide information	User information sent to database
2	Get access to the resource and knowledge sharing	Click on login from the front page and provide information	Users login to the system
3	Storing data on OpenShift online	Point on Browse menu and click on add/upload resources link	Store the resources to the OpenShift online database
4	Watching videos and Downloading available resources	Point to resources menu; click on watch videos and select documents or articles	Watching videos and files stored in user's local machine

6.7.2 Integration testing

This system was integration of different components and tested based on unit interface, completeness of interface and by creating integrated test condition for all interface. The following integration test conditions were used

- Resource and comment storing and retrieving test, file processing test and search test were checked. Each interface are validated based on providing input and using output
- CBRS cloud portal were checked on OpenShift online
- Resource were sent from the web portal and stored in the OpenShift online server and transactions were managed accordingly.
- Integration was done for resource accessing, sharing and management

Table 6.2: Sample Integration Test case

TC ID	TC objective	TC description	Expected output
1	Check the link between resources and services	Click on resources link and then service link	To be directed to services from resources
2	Check the link between login and Browse	Enter username and password and click on login	To be directed to Browse feature

6.7.3 Usability testing

CBRS cloud portal should be deployed for healthcare practitioners, capacity building providers, ICT professionals working in the HDIs, researchers, medical students, any respective organization and the whole community knowledge users. Usability test was done on the perspectives of above users. This test was performed on how all users use CBRS cloud portal and attractiveness of the interface to use. They really liked the designed user interface and other features of the developed prototype. Also they suggest other issues that are to be included in the system. Table 7 shows how the HDIs and other knowledge seekers interact with CBRS cloud portal and rate their feeling about this portal.

Table 6.3: usability test case

No_	Activities	Interface rate	Technical issue
1	Register and login to CBRS system	Very good	Easy to use
2	Accessing resources and tools	Good	Very easy
3	Writing and reading articles online	Good	Easy
4	Uploading/downloading resources	Very good	Very easy

Technical system design was checked by HDI's ICT professionals, colleagues and experts from software engineering and related fields and they provide their comment and suggestions about the developed CBRS system. They recommend that, the system should be more users friendly and easy to use to make it more effective. Hence, the feedback was positive and those comment and recommendation have been left to respective healthcare capacity building providers and professionals to play their roles then developing full implementation will be easy for software engineers. After all, this architecture to share capacity building resources and to create healthcare knowledge sharing network was found to be effective and important when it is fully implemented with those comments given by all stakeholders.

6.8 Finding and discussion

Current practice of providing capacity building program is through manual method where the materials, approaches and tools are used by the capacity building providers in each HDI. After that these resources and best practices are not published or electronically available for other organizations and for the healthcare practitioners. The requirement of having many trainers to train all the healthcare practitioners through the country needs high cost and more time. Furthermore, the healthcare practitioners did not access the resources as they need and could not collaborate to share experience and best practices with their peers. The proposed cloud based platform architecture and the implemented prototype creates common alliance to share medical resources, best practices and experiences which is not only cost effective and interactive but also easy for collaboration, sharing and accessing the resources to develop knowledge, skill and capability in their professions. In addition, the healthcare practitioners can conduct online discussion by raising specific issues and forwards their challenges and solutions.

Compared with the manual method, it is resource full, interactive and supported by current technologies and tools but it does not mean it replaces the whole manual resource usage rather it is used as a bridge by supporting and increasing availability and collaboration with other HDIs, voluntaries and fundraising bodies. Therefore, the proposed cloud based platform architecture increases availability and sharing of capacity building resources, tools, approaches, best practices and experiences among the HDIs and healthcare practitioners. This reduces cost of providing and distributing capacity building resources reduces health inequality, lack of well-trained man power and lack of technologies. It increases ability of the practitioners by using the system as knowledge gaining network and self-learning environment.

The proposed architecture and the implemented prototype seems to have capability of sharing and accessing resources of all types such as documents (word document, PDF files, text files, excel files), pictures, audios, videos and other types of medical information. So it adds more flexibility in accessing resources that can be used to develop practitioner's ability and skill. Also the prototype is customizable and managed by the cloud provider which reduces management overhead, the interactive user interface allows to easily registering and login to the system to access all available resources and to share their experiences and data to others, therefore increases collaboration of healthcare practitioners. This capacity building resource sharing system can be integrated into the current healthcare system as one additional feature of tele medicine. As the users of CBRS system with the implemented prototype shows, it can be accessed using the existing ICT infrastructures and 3G and above mobile technologies.

The tools selected to design the prototype is based on open source technologies, which allow future development with less effort, which can be affordable and manageable by the economy of the country. The prototype has been developed from the open source cloud tool (OpenShift online) selected after evaluating several available tools.

In OpenShift online, the **applications** are typical web applications that will run on OpenShift. At this time, OpenShift is focused on hosting web applications. When a new application is created, a URL (http://healthcarecapacitybuilding-capacitybuilding.rhcloud.com/) with name of the application and the name of the domain is registered in DNS. A copy of the application code is checked out locally into a folder with the same name as the application. With each new application that is created with the client tools, a remote Git repository is populated with the selected cartridge, which is then cloned to the current directory on the local machine. There are four components that make up an OpenShift online application: domain (capacitybuilding), application name (healthcarecapacitybuilding), Alias and Git-repository. Most of OpenShift online application life cycle has four processes: code, build, deploy and manage. The **code** involves the development of application code with the desired language and tools. The build process builds applications, OpenShift online supports various build mechanism for this purpose. The application deployment process involves making any required changes to the application code, committing those changes to the local repository, and then updating the remote repository, deploying to OpenShift online and managing (*see section 4.3 of chapter four*).

Chapter Seven: Conclusions and Future work

7.1 Conclusion

In this thesis, we have presented the community cloud platform architecture and prototype implementation of capacity building program named CCBRSA for sharing and accessing of capacity building resources in the healthcare area. Hence, this study introduces the concept of implementation of cloud computing in the healthcare sectors to share and access capacity building resources to make the health workers more capable in their profession and technology adaption. CCBRSA were designed based on CCRA reference architecture and Kruchten's 4+1 architectural view model as a baseline. This CCBRS architecture focused on sharing and accessing of capacity building resources to increase availability with low cost and to create healthcare learning and collaborating network. It facilitates equal delivery of healthcare services and reduces health inequality by building professional capability and problem solving skill. High level conceptual architecture and different views of the architecture are shown. Redhat's OpenShift online cloud computing platform as a service technology was used to develop the prototype as well as to share and access capacity building resources. So the CBRS cloud portal prototype was developed to test usability of the architecture. Then healthcare practitioners and other users can easily share and access the capacity building resources through the portal using their computer and mobile phones to develop ability, skill and experience. Consequently, the CBRS users are capable not only to handle any challenges in their profession but also to deliver quality healthcare services for the community and to create awareness, understanding and culture of using medical and other technologies.

7.2 Future Work

CCBRSA was developed for healthcare delivery institutions to share and access capacity building resources among healthcare practitioners in the country which includes different components and some of them are implemented. However, the architecture presented and prototype implemented in this thesis need to be further enhanced and evaluated for more comprehensive and noteworthy capacity building system. In future, extending the architecture to prepare and deliver live capacity building training using virtual world with the help of cloud computing is needed. The training can be delivered from one capacity building provider to all HDIs to create worldwide healthcare learning environment and to implement more robust, interactive and effective system. In some cases, because of economic problem and other issues, only limited healthcare practitioners can participate in capacity building training but the majority of the workers have not the chance of attending this training in their work place. So including this feature in the future is recommended. In current architecture and prototype, the users can write, post, read and comment articles online but using such features to support different local and international language translation is not included. Hence extending the architecture to support translation of languages to expand accessibility of the system by different speakers for sharing and accessing resources, experiences and best practices is indispensable. Also even if the integration of this CBRS system with the current health system were showed in the development view of the architecture, this feature is not implemented in the prototype. Thus, it needs to be implemented in the future to perform all in one service together with the delivery of notification about the newly added resources and notes for the member of the system using their mobile phone. Lastly, in some cases the user of the system may share unwanted or out of the systems purpose files or videos. Such files may cause security threats on user’s resources, busy servers by processing these unwanted large size files and distrust by users. In future, this must be handled by specifying intermediated actions to process files before it is stored to the OpenShift server and transmitted to users.

References

1. Ahmed E. Youssef M. A framework for Secure Healthcare Systems Based on Big Data Analytics in Mobile Cloud Computing Environments. International Journal of Ambient Systems and Applications (IJASA). June 2014
2. Yao Q et al. Cloud-based hospital information system as a service for grassroots healthcare institutions. Springer J Med Syst (2014, Jul 12) 38(9):104.doi:10.1007/s10916-014-0104-3.
3. NSW HEALTH (2001). A Framework for Building Capacity to Improve Health.
4. N. Karthikeyan& R. Sukanesh. Cloud Based Emergency Health Care Information Service in India. Springer 3 August 2012
5. P. Mell, and T. Grace, The National Institute of Standard and Technology (NIST), Definition of Cloud Computing, Communications ACM 2010, 53(6):50.
6. World Health Organization. Working Together for Health: The World HealthReport 2006. World Health Organization (2006).
7. IBM Research Collaborations with Leading Taiwanese Institutions to Deliver Wellness-Centric Healthcare via Cloud Computing, IBM Press Release, 2010.
8. Samuel, O.W., Omisore, M.O., Ojokoh, B.A. Atajeromavwo, E.J. Enhanced Cloud based Framework for Healthcare Delivery Organizations in Developing Countries. International Journal of Computer Applications July 2013
9. Adu et al. Cloud Computing Framework for E-Health in Ghana: Adoption Issues and Strategies: Case Study of Ghana Health Service. International Journal of Computer Applications (0975 – 8887) Volume 118 – No.17, May 2015.
10. Lisanne Brown, A. L. (2001, March). Measuring Capacity Building. Carolina Population Center, Chapel Hill, North Carolina.
11. Donald Kaniaru, X. K., Mkandla, S., & Kavagi, L. (2002, December). Capacity Building for Sustainable Development:An overview of UNEP environmental capacity development initiatives. (D. Simpson., Ed.) UNEP Divisions, Regional Offices.
12. Brief, T. (2010, April). Challenges encountered in Capacity Building: A Review of Literature and Selected Tools.
13. (EuroHealthNet), C. C. (2010, April). DETERMINE Working Document # 5: Capacity Building for Health Equiy: a report of capacity building actions to address health inequalities and the socio-economic determinants of health. EuroHealthNet, Czech Republic.

14. IFPS Technical Assistance Project (ITAP). 2012. Capacity Building of Institutions in the Health Sector: Review of Experiences in Uttar Pradesh, Uttarakhand and Jharkhand.
15. Parisi, J. Capacity building in nonprofit organisations in the development aid sector: explanatory research of capacity building in Indonesia in 2008 and an investigation into the diffusion of capacity building techniques between sectors, DBA thesis, Southern Cross University, Lismore, NSW 2009
16. UNDP .Capacity Development: A UNDP Primer Capacity is Development. geneva: UNDP Capacity development group. January 2010.
17. Josephine et al. DHIS2: The Tool to Improve Health Data Demand and Use in Kenya. Journal of Health Informatics in Developing Countries 9 march 2014
18. VicHealth. (August 2012).Capacity building for healthcare promotion: information sheet retrieved from: www.vichealth.vic.gov.au/Publications/Vichealth-General-Publications/capacity_Building_FactSheet.pdf/
19. Sameh El-Saharty, S. K. (August 2009). Ethiopia: Improving Health Service Delivery. Health, Nutrition and Population (HNP) Discussion Paper. World Bank. Washingiton DC: World Bank 1818 H street, NW.
20. Destefani Neto, M. (2015). A Brief history of cloud computing. Retrieved May 10, 2016, from Thoughts On Cloud Website: http://www.thoughtsoncloud.com/2014/03/a-brief-history-of-cloud-computing/
21. Infoway. (2010, July 6). Cloud Computing Is The Latest Buzzword. Retrieved May 3, 2016, from Infoway.us: http://www.infoway.us/article/viewarticle.php?title=Cloud-Computing-Is-The-Latest-Buzzword&view=A-1278418848FWLDg
22. Maguire, J. (2008, December 16). Cloud Computing: The Ever Expanding Buzzword. Retrieved May 3, 2016, from Datamation.com: http://www.datamation.com/entdev/article.php/3791456/Cloud-Computing-The-Ever-Expanding-Buzzword.htm
23. Vogel, P. S. (2011, November 9). Cloud Computing - New Buzzword, Old Legal Issues. Retrieved May 3, 2016, from Ecommerce Times: http://www.ecommercetimes.com/story/73714.html

24. Barry M. Leiner, V. G. (2015, October 15). Brief history of Internet. Retrieved May 3, 2016, from internetsociety.org: http://www.internetsociety.org/internet/what-internet/history-internet/brief-history-internet
25. Marcus Y. (2015). Implementing Cloud Design Patterns for AWS: Create highly efficient design patterns for scalability, redundancy, and high availability in the AWS Cloud. Birmingham. Packt Publishing Ltd.
26. George R. (2009). Cloud application architecture: building application and infrastructure in the cloud. United States of America. O'Reilly Media, Inc.
27. Mahesh S .Darak. (2014). Cloud Computing & its Applications in various sectors. Asian Journal of Management Sciences 02 (03 (Special Issue)); 2014; 07-11.
28. Lorraine M. Kieran C; 2013. FACTORS AFFECTING THE ADOPTION OF CLOUD COMPUTING: AN EXPLORATORY STUDY", Proceedings of 21st European Conference on Information Systems (ECIS), 2013.
29. Techopedia. (2016). Cloud Enabler. Retrieved May 9, 2016, from techopedia website: https://www.techopedia.com/definition/26525/cloud-enabler47
30. Aldrich, S. (2016, March 10). Data Backup & Recovery Blog: Four Enablers of Cloud Computing. Retrieved May 9, 2016, from evs corporation.com: http://www.evscorporation.com/cloud-blog/four-enablers-of-cloud-computing
31. Yogesh K. Dwivedi Navonil Mustafee, (2010),"It's unwritten in the Cloud: the technology enablers for realising the promise of Cloud Computing", Journal of Enterprise Information Management, Vol. 23 Iss 6 pp. 673 – 679.
32. Golden, B. (2009, April 29). Choosing application architecture for the cloud. Retrieved May 12, 2016, from TechTarget website: http://searchcloudcomputing.techtarget.com/news/1355058/Choosing-an-application-architecture-for-the-cloud
33. Lydford, S. (2015, Jun 19). Cloud Design: Cloud Application Design Considerations. Retrieved May 14, 2016, from codurance website: http://codurance.com/2015/06/19/cloud-application-design-considerations/
34. Sanders, J. (2015, July 22). Mini-glossary: Cloud computing terms you should know. Retrieved May 12, 2016, from Techrepublic website: http://www.techrepublic.com/blog/the-enterprise-cloud/mini-glossary-cloud-computing-terms-you-should-know/

35. Tutorials, C. (2016). Technologies related to cloud computing. Retrieved May 12, 2016, from Cloud tutorials website: http://thecloudtutorial.com/related.html
36. He C, Fan X, Li Y. Toward ubiquitous healthcare services with a novel efficient cloud platform. IEEE Trans Biomed Eng. 2013;60(1):230–4.
37. Siddiqui Z, Abdullah AH, Khan MK, Alghamdi AS. Smart environment as a service: three factor cloud based user authentication for telecare medical information system. J Med Syst. 2014;38 (1):9997.
38. Van Gorp P, Comuzzi M. Lifelong personal health data and application software via virtual machines in the cloud. IEEE J Biomed Health Inform. 2014;18(1):36–45.
39. Takeuchi H, Mayuzumi Y, Kodama N, Sato K. Personal healthcare system using cloud computing. Stud Health Technol Inform. 2013;192:936.
40. Ratnam KA, Dominic PD, Ramayah T. A structural equation modeling approach for the adoption of cloud computing to enhance the Malaysian healthcare sector. J Med Syst. 2014; 38 (8):82.
41. Hsieh JC, Hsu MW. A cloud computing based 12-lead ECG telemedicine service. BMC Med Inform Decis Mak. 2012; 12:77.
42. Hiden H, Woodman S, Watson P, Cala J. Developing cloud applications using the e-Science Central platform. Philos Trans A Math Phys Eng Sci.2013; 371 (1983):20120085.
43. Endris Mohammed, “Participatory Indigenous Knowledge Management Architecture for Traditional Medicine”, Unpublished Master’s Thesis, Adama Science and Technology University, February 2016.
44. C. R. Kothari, Research methodology: methods and techniques: New Age International, 2008.
45. NJ, Hoboken, Introduction to qualitative research methods, 3rd ed. US: John Wiley & Sons Inc, 1998.
46. Uma D and Pansiri, Jaloni ogulu, "Mixed methods: A research design for management doctoral dissertations," Management research review, vol. 34, no. 6, pp. 687--701, 2011.
47. Daniel Fallman, "Why Research-oriented Design Isn’t Design oriented research," Knowledge, Technology & Policy, vol. 20, no. 3, pp. 193-200, January 2007.
48. Hevner A. etal "design Science In Information Systems Research," MIS Quarterly, vol. vol. 28, no. no. 1, pp. pp. 75-105, 2004.

49. Mattsson, M. et al (n.d.). Software Architecture Evaluation Methods for Performance, Maintainability, Testability, and Portability.
50. Paul Clements, Rick Kazman Len Bass, Software architecture in practice, second edition ed. Boston, USA: Addison Wesley, 2003.
51. Bell, D. (2005). Software Engineering for students A Programming Approach fourth edition. Great Britain: Henry Ling Ltd.
52. Wikipedia. (n.d.). Software prototyping. Retrieved July 17, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Software_prototyping.
53. Patton, M. Q. (2001). Qualitative evaluation and research methods (3rd ed.). Newbury Park, CA: Sage Publications.
54. Sullivan, D. (2014, January 31). PaaS Providers List: Comparison And Guide. Retrieved July 19, 2016, from toms it pro: http://www.tomsitpro.com/articles/paas-providers,1-1517.html
55. J. Creswell, Educational research: Planning, conducting, and evaluating quantitative and qualitative research, 3rd ed. Upper Saddle River, NJ: Pearson Education., 2008.
56. Wikipedia. (n.d.). OpenShift. Retrieved July 19, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/OpenShift
57. Wikipedia. (n.d.). Microsoft Azure. Retrieved July 20, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Microsoft_Azure
58. Wikipedia. (n.d.). Cloud Foundry. Retrieved July 21, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Cloud_Foundry
59. Wikipedia. (n.d.). Heroku. Retrieved July 22, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Heroku
60. Wikipedia. (n.d.). Google App Engine. Retrieved July 24, 2016, from wikipedia.org: https://en.wikipedia.org/wiki/Google_App_Engine
61. RedHat. (2014). OpenShift Online 1 User Guide: Managing Applications in the Cloud with OpenShift Online. Retrieved Jun 11, 2016, from Openshift.com: http://www.openshift.com/.../OpenShift_Online-2.0-User_Guide-en-Us.pdf
62. OpenShift. (2012). Benefits document OpenShift platform features. Retrieved June 17, 2016, from RedHat website: http://www.redhat.com/13633302.pdf
63. RedHat. (2014). OpenShift Online 1 User Guide: Managing Applications in the Cloud with OpenShift Online. Retrieved Jun 11, 2016, from Openshift.com:

http://www.openshift.com/.../OpenShift_Online-2.0-Client_Tools_Installation_Guide-en-US.pdf

64. IBM Cloud Computing Reference Architecture (CCRA) 4.0 Overview, 2014
65. NIST cloud reference architecture
66. Fehling, C., Leymann, F., Mietzner, R., Schupeck, W.: A Collection of Patterns for Cloud Types, Cloud Service Models, and Cloud-Based Application Architectures in Institute Architecture of Application Systems (IAAS) Report, Daimler A G (2011).
67. Keeling, M. (2014, October 22). Reflections on Software Engineering: Dealing with Constraints in Software Architecture Design. Retrieved May 27, 2016, from neverletdown.net: http://www.neverletdown.net/2014/10/dealing-with-constraints-in-software-architecture.html
68. Ali, M. S. (n.d.). Cloud Security Mechanisms. Retrieved July 12, 2016, from slide Share: http://www.slideshare.net/SajjadAliMOHAMMED/cloud-security-mechanisms
69. Ali, M. S. (n.d.). Cloud Management Mechanisms. Retrieved July 12, 2016, from SlideShare: http://www.slideshare.net/SajjadAliMOHAMMED/cloud-management-mechanisms.
70. P. B. Kruchten, "The 4+1 View Model of architecture," Software, IEEE, vol. 12, pp. 42-50, 1995.
71. Wiki. (n.d.). Feature: HTTPS (HTTP Secure or HTTP over SSL/TLS). Retrieved July 26, 2016, from squid-cache wiki: http://wiki.squid-cache.org/Features/HTTPS
72. w3.org. (n.d.). HTTP - Hypertext Transfer Protocol. Retrieved July 12, 2016, from w3.org: https://www.w3.org/Protocols/
73. Redhat. (n.d.). OpenShift framework architecture. Retrieved June 12, 2016, from openshift.com: https://www.openshift.com/sites/default/files/rhc-openshift-enterprise-paasdatasheet.pdf

Appendix

Appendix I: Sample open ended interview questioners

We select the interviewee and create a channel by making local greeting

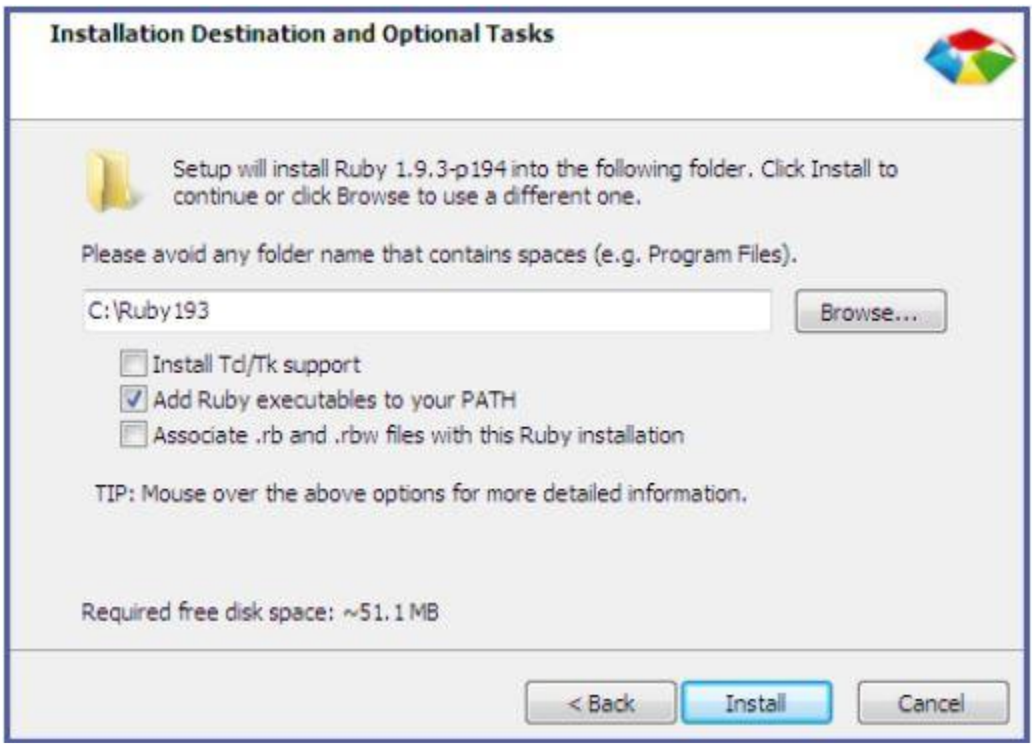
- What do you mean by capacity building program in the healthcare area? Could you explain some of its advantages?
- How capacity building is delivered previously? Have you clear approach or architecture that you follow during the process?
- Is the current approach to share and access capacity building resources is effective and efficient related to the following terms?
    - Its cost to provide
    - Availability of the capacity building resources
    - Usage of the current technology
    - Way of delivery
    - Creation of collaboration
    - Creation of healthcare network and healthcare learning environment
- Where you got the resources? Have you prepared by yourself or from else one? Is it the latest one that matches the current technology and problem solving methods?
- How do you share resources and collaborate with other professional in your area?
- What are the challenges/problems in the current approach/architecture to share and access capacity building resources for the health workers (especially for fresh graduate medical practitioners) in order to make them capable to deliver quality healthcare services for all society?
- What do you think is the promising solution to deliver capacity building resources for the health workers consecutively with reduced cost and man power at high availability?
- If sharing of capacity building resources supported by different devices ubiquitously from one place, would you prefer and use these resources to increase your knowledge, experience and technology adaption in your profession?

Appendix II: Working with OpenShift RHC client tools and GIT

OpenShift client tools also called RHC is a command line tool used to create and manage the applications. To install OpenShift RHC client tool on windows, we have to install three items - Rubyinstaller, Git and OpenShift RHC.

Rubyinstaller

RubyInstaller is a self-contained window-based installer that includes the Ruby language and execution environment. We have downloaded the latest RubyInstaller package for widows from *rubyinstaller.org* and install by following and accepting the default instructions. But it is mandatory to select the **Add Ruby executables to your PATH** check box in order to run Ruby from the command line.



Git Version Control system

Git is a software configuration management (SCM) that keeps track of changes to software source code. With git, we can manage the OpenShift application source repository. We have installed the latest version of Git for Windows from *http://msysgit.github.com/* by following the instructions with all defaults and we have verified correct installation by running the following command from command line: **C :> git –version**, then out puts **git version 1.9.3**.

RHC Client tool

After **RubyInstaller** and **Git** have been successfully installed, we have installed the client tool by running ***$ gem install rhc*** command using window command prompt. After installation is complete we have to configure OpenShift RHC to work with the OpenShift account. RHC setup goes through the following steps:

- Run **C:>rhc setup** command to launch the setup wizard and Login to **openshift.redhat.com**: yimer523@gmail.com and provide Password to the rhc client tool. For User ID, we have to enter the e-mail ID we used to create the OpenShift Account.
- RHC creates and stores a token on disc, which allows us to carry out application management and access the OpenShift shell without using a password. OpenShift provides access to the terminal of the virtual host (gear) on which our application is running.

Having this in mind, git uses three commands to add, commit and deploy locally developed applications on OpenShift online server. The commands are:

- **git add .** ➔tells Git that a file or set of files will become part of a large check in
- **git commit –am "commit message"** ➔complete the check in and inform what change is made
- **git push** ➔deploy the change on OpenShift online server

First we have created domain and application using the RHC client tool as follow:

```
C:\Windows\system32>rhc create-domain CapacityBuilding
DL is deprecated, please use Fiddle
Creating domain 'CapacityBuilding' ... done
You may now create an application using the 'rhc create-app' command
```

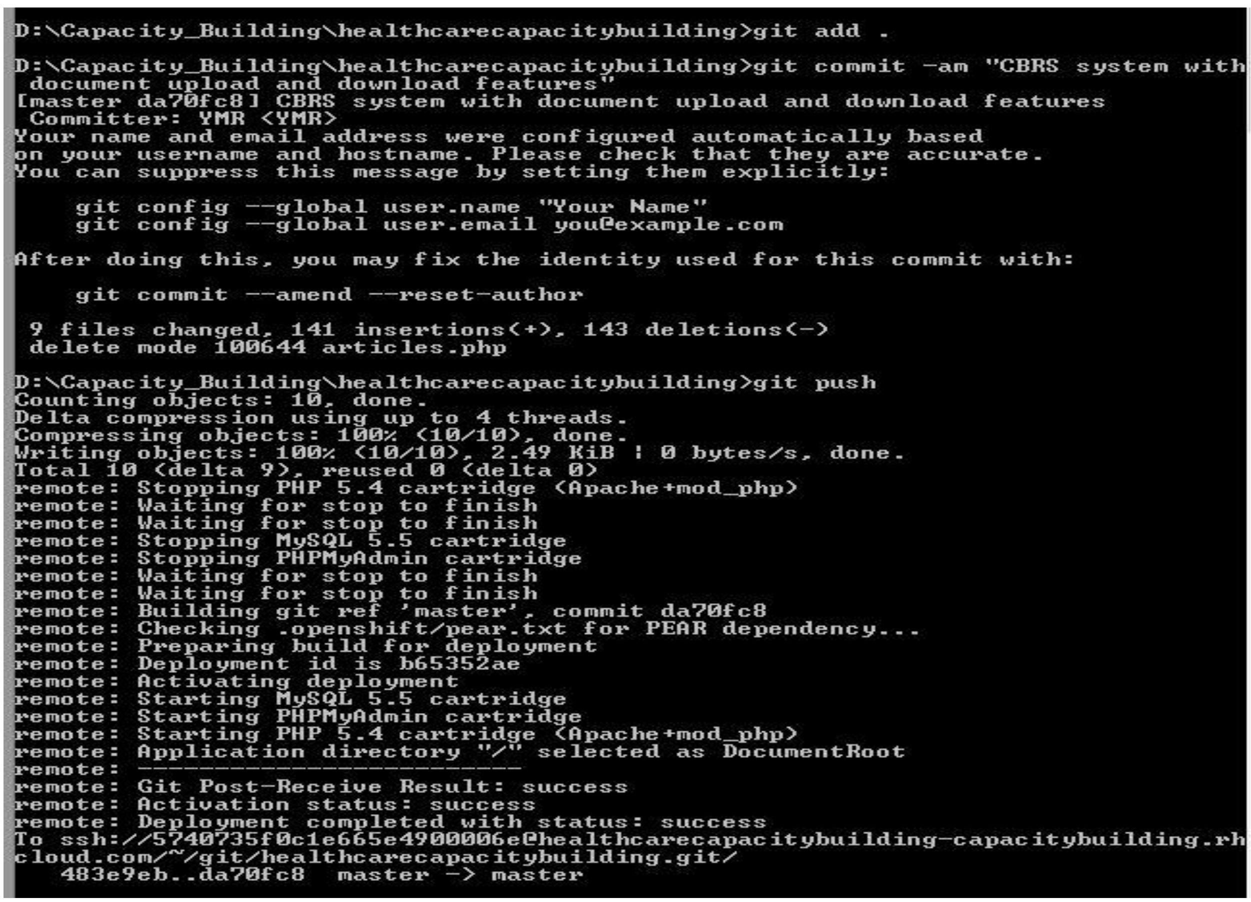
```
C:\Windows\system32>rhc create-app HealthcareCapacityBuilding php-5.4
DL is deprecated, please use Fiddle
Application Options
-------------------
Domain:     capacitybuilding
Cartridges: php-5.4
Gear Size:  default
Scaling:    no

Creating application 'HealthcareCapacityBuilding' ... done


Waiting for your DNS name to be available ... done
```

Figure: Creating domain and application using rhc client tool

Making changes to the Application

As a developer on OpenShift, You make code changes on your local machine, check those changes locally and then push those changes to OpenShift. So the greatest use of Git is that it does not require a continuous online presence in order to run. Simply check in (in Git terminology, 'commit') and revert changes locally before deciding to upload those changes to OpenShift. Hence every OpenShift Application you create has its own Git repository that only you can access. If you create your application from the command line as shown in the above figure, **rhc** will automatically download a copy of that directory (Git calls this 'cloning') to your local system. If you create an application from the web console, you will need to tell Git to clone the repository by running the following code:

$ git clone <git_url> <directory to create>



```
D:\Capacity_Building\healthcarecapacitybuilding>git add .

D:\Capacity_Building\healthcarecapacitybuilding>git commit -am "CBRS system with
 document upload and download features"
[master da70fc8] CBRS system with document upload and download features
 Committer: YMR <YMR>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 9 files changed, 141 insertions(+), 143 deletions(-)
 delete mode 100644 articles.php

D:\Capacity_Building\healthcarecapacitybuilding>git push
Counting objects: 10, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 2.49 KiB | 0 bytes/s, done.
Total 10 (delta 9), reused 0 (delta 0)
remote: Stopping PHP 5.4 cartridge (Apache+mod_php)
remote: Waiting for stop to finish
remote: Waiting for stop to finish
remote: Stopping MySQL 5.5 cartridge
remote: Stopping PHPMyAdmin cartridge
remote: Waiting for stop to finish
remote: Waiting for stop to finish
remote: Building git ref 'master', commit da70fc8
remote: Checking .openshift/pear.txt for PEAR dependency...
remote: Preparing build for deployment
remote: Deployment id is b65352ae
remote: Activating deployment
remote: Starting MySQL 5.5 cartridge
remote: Starting PHPMyAdmin cartridge
remote: Starting PHP 5.4 cartridge (Apache+mod_php)
remote: Application directory "/" selected as DocumentRoot
remote: -------------------------
remote: Git Post-Receive Result: success
remote: Activation status: success
remote: Deployment completed with status: success
To ssh://5740735f0c1e665e4900006e@healthcarecapacitybuilding-capacitybuilding.rh
cloud.com/~/git/healthcarecapacitybuilding.git/
   483e9eb..da70fc8  master -> master
```

Figure: Modifying application and deploying changes to OpenShift online server using Git.

Appendix III: Sample code

The prototype of the proposed architecture was developed and the sample php code includes:

Php code for database connection with OpenShift online platform

```
<?php //start of php code
            $dbhost=getenv("OPENSHIFT_MYSQL_DB_HOST");
            $dbport=getenv("OPENSHIFT_MYSQL_DB_PORT");
            $dbuser=getenv("OPENSHIFT_MYSQL_DB_USERNAME");
            $dbpwd=getenv("OPENSHIFT_MYSQL_DB_PASSWORD");
    $connection=mysql_connect($dbhost, $dbuser, $dbpwd);
     if (!$connection)
            {//checking the OpenShift database host, port, user and password
         die("Database Connection Failed" . mysql_error());
          }
   $databaseconnection=mysql_select_db('healthcarecapacitybuilding');
   if (!$databaseconnection)
   { //checking the database connection and exit if not connected
       die("Database Selection Failed" . mysql_error());
    }
//end of php code
?>
```

Php code for sharing video files

```
<?php
            <div id='box'>
            <form method="post" enctype="multipart/form-data">
            <?php
                        //getting the video from the
                                    if(isset($_FILES['file'])){
                                                $name = $_FILES['file']['name'];
                                                $extension = explode('.', $name);
                                                $extension = end($extension);
                                                $type = $_FILES['file']['type'];
                                                $sizef = $_FILES['file']['size'] /1024/1024;
                                                $size =number_format($sizef, 2);
                                                $random_name = rand();
```

```
                                        $tmp = $_FILES['file']['tmp_name'];
                                        $description=$_POST['description'];
                                        $uploaded_by=$_SESSION['username'];
                                        $uploaded_date=date("Y-m-d H:i:s");

if ((strtolower($type) != "video/mp4") && (strtolower($type) != "video/mpeg") && (strtolower($type) != "video/mpeg1") && (strtolower($type) != "video/mpeg4") && (strtolower($type) != "video/avi") && (strtolower($type) != "video/flv") && (strtolower($type) != "video/wmv") && (strtolower($type) != "video/mov") && (strtolower($type) != "video/ogg") && (strtolower($type) != "video/3GP"))
{
$message= "<font color=yellow face=\"Comic Sans MS,Lucida Console\"><u>Video Format is not supported</u> !</font>";
 }
else{
move_uploaded_file($tmp, 'uploaded/videos/'.$random_name.'.'.$extension);
mysqli_query($connection, "INSERT INTO videos VALUES('', '$name' , '$description', '$size', '$random_name.$extension', '$uploaded_by', '$uploaded_date')");
$message=" <font color=#36ee66 face=\"Comic Sans MS,Lucida Console\"><u>Video </u></font> "." <font color=#ffffff face=\"Comic Sans MS,Lucida Console\"><u> ".$name."</u></font>"." <font color=#36ee66 face=\"Comic Sans MS,Lucida Console\"><u>has been successfully uploaded</u> !</font>";
        }
?>
 <center>  <a href="view_all_videos.php"><img src="./images/list.jpg"\></a> <br /><br />  </center>
<?php echo "<font color=#ffffff face=\"Comic Sans MS,Lucida Console\"> Upload another video . . . . </font><br/>"; }?>
         Select Video : <br/>
        <input name="UPLOAD_MAX_FILESIZE" value="20971520"  type="hidden" required/>
        <input type="file" name="file" id="file" required/>
         <br/>Add description: <br/>
         <input type="text" name="description" id="description" size="31"required/> <br/>
        <input type="submit" value="Share Video" />
      </form>
</div>
```

Php code to watch video files

```
<div id='box'>
   <?php
         require('include/db.php');
         $video= $_GET['video'];
         $query = mysqli_query($connection, "SELECT * FROM videos where url='$video'");
                        while($row = mysqli_fetch_array($query)){
                              $name=$row['video_name'];
                              $description=$row['description'];
                              $uploaded_by=$row['uploaded_by'];
                              $date=$row['uploaded_date'];
echo "<font color='white'><center> You are watching "." ".$name." <br /><b><u>Description of
this video:</u></b> ".$description."  ". "<br />Uploaded by <u><b><i>".$uploaded_by." on
".$date." </i></b></u><br /></center> </font>";}
?>
<video id="my_video" class="yy" controls data-setup="{}" preload="auto" width="100%"
height="450" poster="ymr">
   <source src="uploaded/videos/<?php echo $video; ?>" type='video/mp4'>
</video>
                  <script type="text/javascript">
                        function myScript() {
                              console.log;
                                    }
                  </script>
  <div id='back'>
       <a href="view_all_videos.php" class="backto"><< Back to All video lists</a>
</div>
</div>
```