

SHIVAJI UNIVERSITY

Application Virtualization

by

Author Name

A thesis submitted in partial fulfillment for the
degree of Doctor of Philosophy

in the

Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering, Yadrav-Ichalkaranji

April 2015

SHIVAJI UNIVERSITY

Abstract

Department of Computer Science and Engineering
Sharad Institute of Technology College of Engineering, Yadrav-Ichalkaranji

Bachelor of Engineering

Cloud Computing is an attractive concept in IT field, since it allows the resources to be provisioned according to the user needs. It provides services on virtual machines whereby the user can share resources, software and other devices on demand. Cloud services are supported both by Proprietary and Open Source Systems. As Proprietary products are very expensive, customers are not allowed to experiment on their product and security is a major issue in it, Open source systems helps in solving out these problems. Cloud Computing motivated many academic and non academic members to develop Open Source Cloud Setup, here the users are allowed to study the source code and experiment it. This paper describes the configuration of a private cloud using Eucalyptus. Eucalyptus an open source system has been used to implement a private cloud using the hardware and software without making any modification to it and provide various types of services to the cloud computing environment.

Acknowledgements

We take this opportunity to express our profound gratitude and deep regards to our guide Mr.O.D.Joshi for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We also take this opportunity to express a deep sense of gratitude to our Principal, Dr. S. A. Khot and our H.O.D., Prof. O.D.Joshi for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are obliged to all faculty and staff members of all departments of our Sharad Institute of Technology College of Engineering, Yadav for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of our assignment.

Lastly, we thank almighty, our parents, brothers, sisters, friends and our colleagues for their constant encouragement without which this assignment would not be possible.

Miss. Shetti Pooja Pravin

Miss. Girmal Sonali Ravaso

Miss. Awati Yogita Ajit

Miss. Buchade Komal Prakash

Miss. Herwade Priyanka Tanaji

Contents

Abstract	i
Acknowledgements	ii
List of Figures	iii
1 INTRODUCTION	1
1.1 Introduction of project	1
1.2 What is Cloud?	2
1.3 Types of Cloud	2
1.3.1 Public Cloud	2
1.3.2 Private Cloud	2
1.3.3 Hybrid Cloud	2
1.4 Cloud Services	3
1.4.1 Android	3
2 LITERATURE REVIEW	4
2.1 A Cloud-Based Interactive Application Service	5
2.2 Virtualization security for cloud computing service	5
2.3 Secure virtualization for cloud computing	5
2.4 Advanced Cloud Protection System	6
3 OBJECTIVE AND SCOPE	7
3.1 Objectives	7
3.2 Scope	7
3.3 Out of Scope	7
4 REQUIREMENT ANALYSIS	8
4.1 Hardware Requirements	8
4.2 Software Requirements	8
4.2.1 Java	8
4.2.2 Eclipse	9
4.2.3 Apache tomcat server	10
5 SYSTEM DESIGN	11
5.1 An Overview of UML	11
5.2 Goals of UML	12

5.3	A Conceptual Model of UML	12
5.4	Diagrams in UML	12
5.5	Use Case Diagram	13
5.5.1	Use-Case Scenarios	14
5.6	Sequence Diagram	15
5.6.1	Contents	15
5.6.2	Definition and Overview	15
5.7	Class Diagram	16
5.7.1	Contents	16
5.7.2	Definition and Common Uses	16
5.8	Deployment Diagram	18
5.8.1	Nodes and Components	18
5.8.2	Deployment Diagram	19
6	CODING	20
6.1	Introduction of Tool and Installation	20
6.1.1	Android	20
6.2	Issues	22
6.2.1	Installing apk	22
6.2.2	WIFI network	22
6.2.3	Firewall	22
6.3	Code for Login Form	22
6.4	Code for Selection Menu	25
7	TESTING	28
7.1	What is Software Testing	28
7.2	Test Methods	29
7.2.1	Black Box Testing	29
7.2.2	White Box Testing	29
7.3	Test Cases and Test Data	30
7.3.1	Test Cases	31
8	DEPLOYMENT	33
8.1	Login Window	33
8.2	Register Screen	33
8.3	Selection Menu	34
8.4	Listening Music	34
8.5	Watching Videos	35
8.6	Code Execution	36
9	CONCLUSION	37
	Bibliography	38

List of Figures

5.1	Use-Case diagram for Application virtualization.	14
5.2	Sequence Diagram for Application Virtualization.	16
5.3	Class Diagram for Application Virtualization.	17
5.4	Deployment Diagram for Application Virtualization.	19
8.1	Login Window.	33
8.2	Registration Form	34
8.3	Menu Managmentl	34
8.4	Listining music	35
8.5	Play music	35
8.6	Watching video	35
8.7	Display for Execute and Get Output	36
8.8	Code writing on android	36

Chapter 1

INTRODUCTION

1.1 Introduction of project

Internet is on the edge of another revolution, where resources are globally networked and can be easily shared. Cloud computing is the main component of this paradigm, that renders the Internet a large repository where resources are available to everyone as services. The architecture for remotely using mobile application on Cloud Server that intends to handle not only user data but also user applications in a cloud server. This approach changes the application lifecycle as follows. Write once, run everywhere. Install once, use everywhere.

Software as a Service (SaaS) represents a novel paradigm and business model expressing the fact that companies do not have to purchase and maintain their own ICT infrastructure, but instead, acquire the services embodied by software from a third party. Here SaaS service is provided for mobile users.[1] This paper proposes Android as a Server Platform that enables many users to use resources on remote cloud servers. This also proposes a multi-tenant architecture of Android on cloud server.

Multi-tenancy, this is defined as a feature where the software running on a server provides services to many users. It is one of the important features for cloud computing. From the viewpoint of both economy and ecology, it is beneficial to share hardware resources among users. Using a mobile OS would be more effective than using a desktop OS because the resource requirements of mobile OSs are smaller.

1.2 What is Cloud?

Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. The storing and accessing of applications and computer data through the web browser rather than installing software on your personal computer or mobile.[2] This approach changes the application lifecycle as follows. "Write once, run everywhere. Install once, use everywhere. The important features for cloud computing is Multi-tenancy, defined as a feature where the software running on a server provides services to many users.

1.3 Types of Cloud

as public Cloud computing is internet-based computing in which large groups of remote servers are networked to allow the centralized data storage, and online access to computer services or resources. Clouds can be classified, private or hybrid.

1.3.1 Public Cloud

In Public cloud the computing infrastructure is hosted by the cloud vendor at the vendor premises. The customer has no visibility and control over where the computing infrastructure is hosted. The computing infrastructure is shared between any organizations.

1.3.2 Private Cloud

The computing infrastructure is dedicated to a particular organization and not shared with other organizations. Some experts consider that private clouds are not real examples of cloud computing. Private clouds are more expensive and more secure when compared to public clouds.

1.3.3 Hybrid Cloud

Organizations may host critical applications on private clouds and applications with relatively less security concerns on the public cloud. The usage of both private and public clouds together is called hybrid cloud.

1.4 Cloud Services

Cloud services means services made available to users on demand via the Internet from a cloud computing provider's servers as opposed to being provided from a company's own on-premises servers. Cloud services are designed to provide easy, scalable access to applications, resources and services, and are fully managed by a cloud services provider.

Cloud service can dynamically scale to meet the needs of its users, and because the service provider supplies the hardware and software necessary for the service, there's no need for a company to provision or deploy its own resources or allocate IT staff to manage the service. Examples of cloud services include online data storage and backup solutions, Web-based e-mail services, hosted office suites and document collaboration services, database processing, managed technical support services and more.

1.4.1 Android

Now a days android is most powerful operating system environment used for mobile. Android is a comprehensive software stack of mobile devices that includes an operating system, middleware and key application. This rich source of software bunch is used in Mobile Technology through its innovation module of The Android Software Development Kit (SDK). Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as Smartphone and tablet computers.[3] Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance: a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first Android-powered phone was sold in October 2008.

Android is Open Source and Google releases the code under the Apache License. This open source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Open source library for application development including SQLite, WebKit, OpenGL, and media manager. Android, Inc. was founded in Palo Alto, California in October 2003 by Andy Rubin (co-founder of Danger), Rich Miner (co-founder of Wildfire Communications, Inc.), Nick Sears (once VP at T-Mobile), and Chris White (headed design and interface development at WebTV) to develop, in Rubin's words "smarter mobile devices that are more aware of its owner's location and preferences".[4]

Chapter 2

LITERATURE REVIEW

In contemporary years Cloud Computing has been assiduously analyzed by the government Japan strategy, the clique of the Ministry of Internal affairs and Communications. In addition to this, different prominent companies are using cloud services, like Windows Azure, Google Apps, and Amazon. People over the globe came up with numerous definition of cloud computing. Different companies have different perspective, for example companies like IBM considers that cloud computing is a cutting edge of IT resources whereas on the other hand Google affirm that cloud computing is service based, paramount, commodious data depository application. So, above all the definitions of cloud computing, the crucial working principle is that it is a parallel and distributed processing model, which convey IT as a service.[5] The core structure of Cloud computing comprises of several computer servers which are named as clouds. All the assests are collected together in order to form a centric data storage and data processing canter. As various resources are gathered diverse configuration tools are needed to stanchion the cloud side of software management. As the client acknowledges the data request to the server, the request is being processed and finally search results return. But, it is exigent to actualize the measurement and monitoring, to accomplish the quality of service and burgeoning of efficient resource allocation and efficiency of resource.[6] Together with an explosive growth of the mobile applications and emerging of cloud computing concept, mobilecloud computing (MCC) has been introduced to be a potential technology for mobile services. MCC integrates thecloud computing into the mobile environment and overcomes obstacles related to the performance (e.g., battery life,storage, and bandwidth), environment (e.g., heterogeneity, scalability, and availability), and security (e.g., reliabilityand privacy) discussed in mobile computing.

2.1 A Cloud-Based Interactive Application Service

Accessing, running and sharing applications and data presents researchers with many challenges. Cloud computing and social networking technologies have the potential to simplify or eliminate many of these challenges. Cloud computing technologies can provide scientists with transparent and on-demand access to applications served over the Internet in a dynamic and scalable manner. Social networking technologies provide a means for easily sharing applications and data.

In this paper they present an on-line/on-demand interactive application service. The service is built on a cloud computing infrastructure that dynamically provisions virtualized application servers based on user demand. An open source social networking platform is leveraged to establish a portal front end that enables applications and results to be easily shared between researchers. Furthermore, the service works with existing/legacy applications without requiring any modifications.

2.2 Virtualization security for cloud computing service

Virtualization is a term that refers to the abstraction of computer resources. The purpose of virtual computing environment is to improve resource utilization by providing a unified integrated operating platform for users and applications based on aggregation of heterogeneous and autonomous resources. More recently, virtualization at all levels (system storage, and network) became important again as a way to improve system security, reliability and availability, reduce costs and provide greater flexibility. In this paper, they address the requirements and solutions for the security of virtualization in cloud computing environment. Moreover, a Virtualization Security framework is presented which contains two parts: virtual system security and virtualization security management.

2.3 Secure virtualization for cloud computing

Cloud computing adoption and diffusion are threatened by unresolved security issues that affect both the cloud provider and the cloud user. In this paper, they show how virtualization can increase the security of cloud computing, by protecting both the integrity of guest virtual machines and the cloud infrastructure components. In particular, they propose a novel architecture, Advanced Cloud Protection System (ACPS), aimed at guaranteeing increased security to cloud resources[7].

2.4 Advanced Cloud Protection System

ACPS can be deployed on several cloud solutions and can effectively monitor the integrity of guest and infrastructure components while remaining fully transparent to virtual machines and to cloud users. ACPS can locally react to security breaches as well as notify a further security management layer of such events. A prototype of our ACPS proposal is fully implemented on two current open source solutions: Eucalyptus and OpenECP. The prototype is tested against effectiveness and performance. In particular: (a) effectiveness is shown testing our prototype against attacks known in the literature; (b) performance evaluation of the ACPS prototype is carried out under different types of workload. Results show that their proposal is resilient against attacks and that the introduced overhead is small when compared to the provided features.

Chapter 3

OBJECTIVE AND SCOPE

3.1 Objectives

Instead of running all the application locally on the machines with all of the associated challenges and costs, applications run centrally and simply deliver screen updates and inputs to clients. **Following are the objectives of the proposed system**

- The ability of the application to share hardware resources.
- The availability of a high degree of configuration of software.
- The architectural approach in which the tenants (or users) make use of a single and database instance.

3.2 Scope

- We are providing the virtual application softwares to the authorized clients such as desktop machines, android mobile and tablets.
- We are providing the memory space to store the user data.

3.3 Out of Scope

- As like application software we are going to provide the operating system virtually using only one CPU.
- We shall provide the more security to the users data.

Chapter 4

REQUIREMENT ANALYSIS

4.1 Hardware Requirements

- Desktop.
- Mobile.
- Aakash Tablet.

4.2 Software Requirements

- OS for server: ubuntu 10.04 enterprise cloud server.
- OS for mobile or tablet clients: Android.
- OS for desktop machines: Any operating system.
- Java Runtime environment 6
- Eclipse
- ADT plug-in
- Apache Tomcat server

4.2.1 Java

a) Java

It is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible.

It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture.

Writing in the Java programming language is the primary way to produce code that will be deployed as byte code in a Java Virtual Machine (JVM); byte code compilers are also available for other languages, including Ada, JavaScript, Python, and Ruby. [?] In addition, several languages have been designed to run natively on the JVM, including Scala, Clojure and Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C.

b) Java platform

It is the name given to the computing platform from Oracle that helps users to run and develop Java applications. The platform does not just enable a user to run and develop a Java application, but also features a wide variety of tools that can help developers work efficiently with the Java programming language.

The platform consists of two essential softwares

- 1) The Java Runtime Environment (JRE), which is needed to run Java applications and applets.
- 2) The Java Development Kit (JDK), which is needed to develop those Java applications and applets.

4.2.2 Eclipse

a) Description

Eclipse is a Java-based open source platform that allows a software developer to create a customized development environment (IDE) from plug-in components built by Eclipse members. The original goal of Eclipse was to create and foster an open source IDE community that would complement the community that surrounds Apache. The Eclipse Platform is written in Java, it supports plug-ins that allows developers to develop and test code written in other languages.[8]

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By

means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

b) Platform

- 1) Plug-in development environment(PDE), provides a number of views and editors that make is easier to build plug-ins for Eclipse.
- 2) The Java development tools (JDT), provides the tool plug-ins for the platform that implement a Java IDE for power-users, that supports the development of any Java application, including Eclipse plug-ins.

4.2.3 Apache tomcat server

a) Description

Apache Tomcat is an open source software implementation of the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed under the Java Community Process.[9] Apache Tomcat also provides by default a HTTP connector on port 8080, i.e., Tomcat can also be used as HTTP server. But the performance of Tomcat is not as good as the performance of a designated web server, like the Apache HTTP server.

b) Platform

- 1) The Java Servlet and JSP Platform, provides a "pure Java" HTTP web server environment for Java code to run in.

Chapter 5

SYSTEM DESIGN

5.1 An Overview of UML

UML is a language for

- Visualizing
- Specifying
- Constructing
- Documenting

UML LANGUAGE

A language provides a vocabulary and the rules for combining words in that vocabulary for the purpose of the communication. A modeling language is a language whose vocabulary and rules focus on conceptual and physical representation of a system. A modeling language such as the UML is thus a standard language for software blueprints. In this context, specifying means building models that are precise, unambiguous, and complete. In particular, the UML addresses the specification of all the important analysis, design and implementation decision that must be made in developing and deploying a software intensive system. The UML is not a visual programming language, but its model can be directly connected to a variety of programming languages. This means that its possible to map from a model in the UML to a programming language such as java, cpp, or visual basic or even to tables in a relational database. Things that are best expressed graphically are done so graphically in the UML, whereas things that best expressed textually are done so in the programming language. A healthy software organization produces all sorts of artifacts in addition to raw executable code. These artifacts include

requirements, architecture, design, source code, project plans, tests, prototypes, releases. The UML also provides for expressing requirements and for tests. Finally, UML provides a language for modeling the activities of project planning and release management.

5.2 Goals of UML

The primary goals in the design of UML were

- Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models. Provide extensibility and specialization mechanisms to extend the core concepts.
- Be independent of particular programming languages and development processes. Provide a formal basis for understanding the modeling language
- Encourage the growth of the OO tools market.
- Support higher-level development concepts such as collaborations, frameworks, patterns and components.
- Integrate best practices

5.3 A Conceptual Model of UML

To understand the UML, you need to form a conceptual model of the language, and this requires learning three major elements: the UMLs basic building blocks, the rules that dictate how those building blocks may be put together, and some mechanisms that apply throughout the UML. Once you have grasped these ideas, you will be able to read UML models and create some basic ones. As you gain more experience in applying the UML, you can build on this conceptual model, using more advanced features of the language.

5.4 Diagrams in UML

A diagram is the graphical presentation of a set of elements, most often rendered as a connected graph of vertices (things) and arcs (relationships). You draw diagrams to visualizing a system from different perspectives, so a diagram is a projection into a system. The same element may appear in all diagrams. In theory, a diagram may contain any combination of things and relationships. The views that comprise the architecture of software intensive system. For this reason, the UML includes following diagrams:

- Use case diagram
- Class diagram
- Sequence diagram
- Deployment diagram

5.5 Use Case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and their relationships. A use case diagram is a just special kind of diagram and shares the same common properties as do all other diagram-a name and graphical contents. A use case diagram is the simplest representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can define the different types of users of a system and the various ways that they interact with the system. They provide the simplified and graphical representation of what the system must actually do. The purpose of the use case diagrams is simply to provide the high level view of the system and convey the requirements.

You can use cases for the following purposes:

- Determine the requirements of the system.
- Describe what the system should do.
- Provide a basis for testing to ensure that the system works as intended.

A use case defines the interactions between external actors and the system under consideration to accomplish a goal. The use cases that a system or component supports appear inside its rectangle. It can be useful to show some use cases outside the rectangle, to clarify the scope of your system. A subsystem in a use case diagram has basically the same type as a component in a component diagram.

Actor: Represents a user, organization, or external system that interacts with your application or system. An actor is a kind of type.

Use Case: Represents the actions performed by one or more actors in the pursuit of a particular goal. A use case is a kind of type.

Association: The system or application that you are working on, or a part of it.

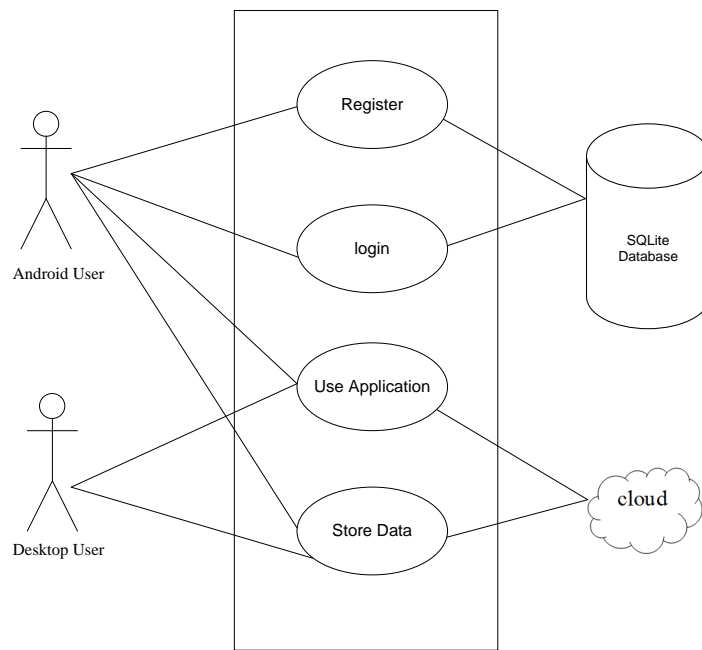


FIGURE 5.1: Use-Case diagram for Application virtualization.

5.5.1 Use-Case Scenarios

,

USE CASE	USE CASE SCENARIO
Register	1) User enters user name in Register form. 2) User enters password in Register form. 3) User re-enter password in Register form. 4) User clicks on Register button.
Login	If user already exist 1) User enters user name in login form. 2) User enters user name in user name text field. 3) User enters user password in user password text field. 4) User click on login button.
Use Application	1) If user-name and password matches user is authenticates. 2) Then user access the cloud services.
Store Data	1) If user-name and password matches user is authenticates 2) Then user access the cloud services. 3) The class file of java program will stored on cloud server.

TABLE 5.1: Use case Scenario for Application Virtualization

5.6 Sequence Diagram

5.6.1 Contents

Sequence diagram commonly contains

- Objects
- Links
- Messages

5.6.2 Definition and Overview

A *sequence* diagram is an interaction diagram that emphasizes the time ordering of messages. A sequence diagram shows a set of objects and the messages sent and received by those objects. The objects are typically named or anonymous instances of classes, but may also represent instances of other things, such as collaborations, components, and nodes. You use sequence diagrams to illustrate the dynamic view of a system. An Actor models a type of role played by an entity that interacts with the subject (e.g., by exchanging signals and data), but which is external to the subject (i.e., in the sense that an instance of an actor is not a part of the instance of its corresponding subject). Actors may represent roles played by human users, external hardware, or other subjects. Note that an actor does not necessarily represent a specific physical entity but merely a particular facet (i.e., "role") of some entity that is relevant to the specification of its associated use cases.

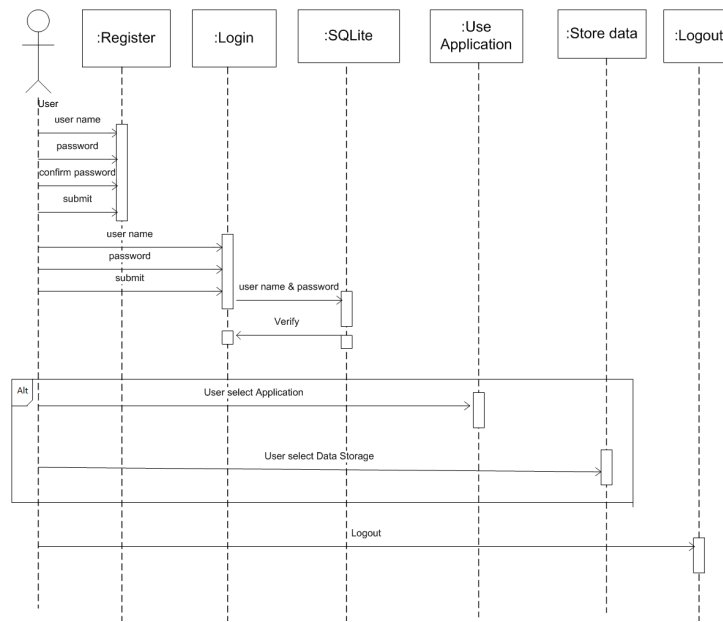


FIGURE 5.2: Sequence Diagram for Application Virtualization.

5.7 Class Diagram

5.7.1 Contents

Class diagram commonly contain the following things:

- Classes
- Interfaces
- Collaborations
- Dependency, generalization, and association relationships.

5.7.2 Definition and Common Uses

A class diagram is a diagram that shows a set of classes, interfaces and their relationships. Graphically, a class diagram is a collection of vertices and arcs. A class diagram will share the same common properties as do all other diagrams. A class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity. Class diagrams are useful in all forms of object-oriented programming (OOP).

The concept is several years old but has been refined as OOP modeling paradigms have evolved. In a class diagram, the classes are arranged in groups that share common characteristics. A class diagram resembles a flowchart in which classes are portrayed as boxes, each box having three rectangles inside. The top rectangle contains the name of the class; the middle rectangle contains the attributes of the class; the lower rectangle contains the methods, also called operations, of the class. Lines, which may have arrows at one or both ends, connect the boxes. These lines define the relationships, also called associations, between the classes.

- Class: A definition of objects that share given structural or behavioral characteristics.
- Attribute: A typed value attached to each instance of a classifier.
- Operation: A method or function that can be performed by instances of a classifier

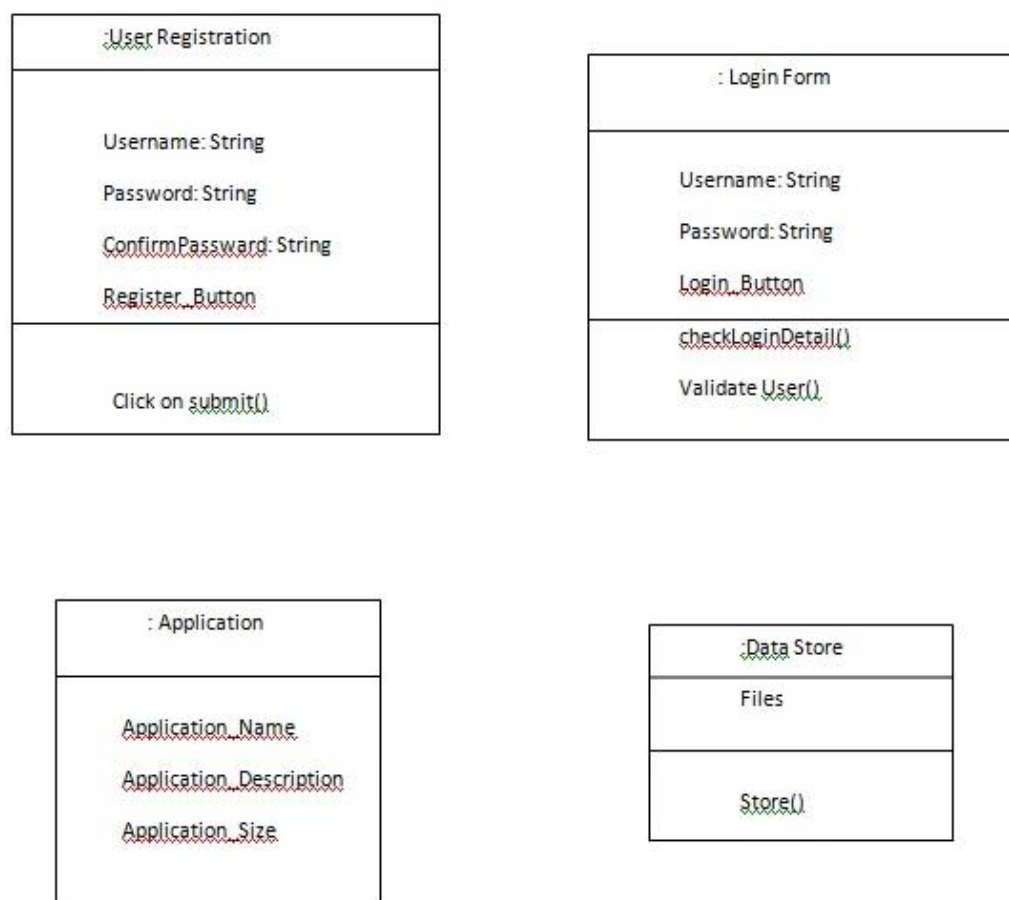


FIGURE 5.3: Class Diagram for Application Virtualization.

5.8 Deployment Diagram

subsectionDefinition A deployment diagram shows the configuration of run time processing nodes and the components that live on them. Deployment diagram address the static deployment view of architecture. They are related to component diagrams in that a node typically encloses one or more components .

5.8.1 Nodes and Components

The UML provides a graphical representation of node. This canonical notation permits you to visualize a node apart from any specific hardware. Using stereotype this notation to represents specific kinds of processors and devices. A *node* is a physical element that exists at run time and represents a computational resource, generally having at least some memory, and often processing capability. Graphically, a node is rendered as a cube. Every node must have a name that distinguishes it from other nodes. A name is a textual string. Components are things that participate in the execution of a system; nodes are things that execute components. Components represent the physical packaging of otherwise logical elements; nodes represent the physical development of components and components that things are executed by nodes. The Deployment Diagram also helps to model the physical aspect of an Object-Oriented software system. It models the run-time configuration in a static view and visualizes the distribution of components in an application. In most cases, it involves modeling the hardware configurations together with the software components that lived on. The UML can often use stereotypes to specify new kinds of nodes that you can use to represent specific kinds of processors and devices. A *Processor* is a node that has processing capability, meaning that it can be executed by component. A *device* is a node that has no processing capability and general, represents something that interfaces to real world.

5.8.2 Deployment Diagram

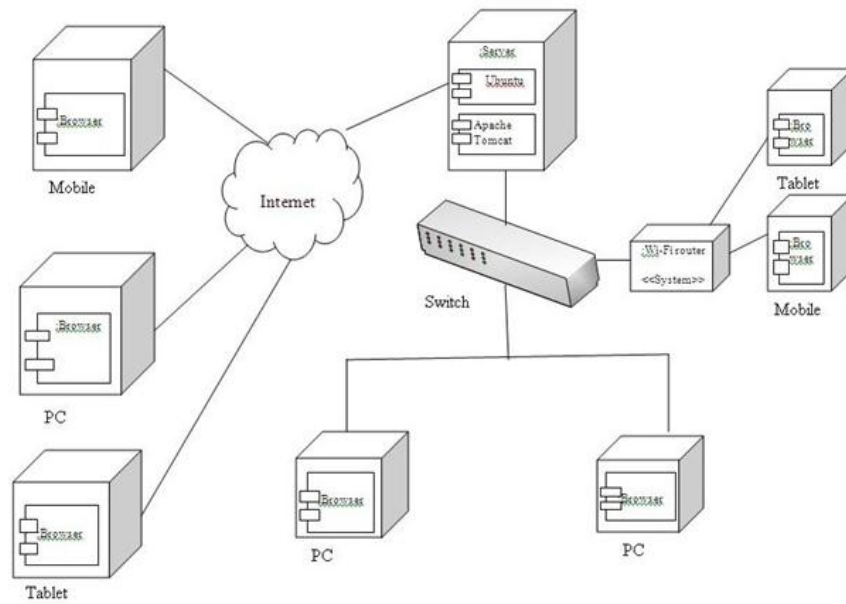


FIGURE 5.4: Deployment Diagram for Application Virtualization.

Chapter 6

CODING

6.1 Introduction of Tool and Installation

6.1.1 Android

There's no other software quite like Android. Google engineered Android, and Google own apps run better on it. And with millions of apps, games, songs, and videos on Google Play, Android is great for fun, and for getting things done.

Android devices come in all kinds of sizes, with all sorts of features, and in all sorts of prices. Each version of Android is named after a dessert, and the most recent version of Android is lollipop. With Android, you're in control of your mobile experience.

The world is contracting with the growth of mobile phone technology. As the number of users is increasing day by day, facilities are also increasing. Starting with simple regular handsets which were used just for making phone calls, mobiles have changed our lives and have become part of it. Now they are not used just for making calls but they have innumerable uses and can be used as a Camera, Music player, Tablet PC, T.V., Web browser etc. . And with the new technologies, new software and operating systems are required.

- What is android

Operating Systems have developed a lot in last 15 years. Starting from black and white phones to recent smart phones or mini computers, mobile OS has come far away. Especially for smart phones, Mobile OS has greatly evolved from Palm OS in 1996 to Windows pocket PC in 2000 then to Blackberry OS and Android.

- ADT Bundle

The Android SDK is a software development kit which provides API libraries and necessary developer tools necessary for building Android applications. Android SDK is officially provided by android developers.

steps for the installation and set-up of Android development environment:

1. Download Eclipse
2. Download JDK and install it, set the environment path.
3. Download ADT plugin inside Eclipse.
4. Set the Preference with Android-SDK path.
5. Download the latest platform-tools and everything.

The ADT Bundle includes everything you need to begin developing apps:

1. Eclipse + ADT plugin
2. Android SDK Tools
3. Android Platform-tools
4. The latest Android platform
5. The latest Android system image for the emulator

Yes there are also possible ways if you want to use existing version of Eclipse or any other IDE.

- **Setting Up the ADT Bundle:**

As you have downloaded ADT bundle, follow below steps to setup it:

1. Unpack the ZIP file named “adt bundle osplatform.zip ” and save it to an appropriate location such as a “Development” directory in your home directory.
2. Open the adt bundle osplatform goto eclipse and next directory and launch eclipse.

6.2 Issues

6.2.1 Installing apk

We face one issue for Tablet because of repeatedly installing the application memory got blocked, because of this problem our application stop unfortunately. the solution is Reset the tablet and then install the application.

6.2.2 WIFI network

we require strong wifi network, because of less range of wifi dis-connectivity will occur and connection may get loss, the client will not going to connect to Server so client will not get the application.

6.2.3 Firewall

the Firewall is next obstacle, if Firewall is off then again connectivity issue arises. then go to control panel type Firewall in search window. go to window Firewall option. on left side there is option "Turn window Firewall on or off". Click on it select the Radio button for "Turn on Firewall".

6.3 Code for Login Form

Here we connect login form with the SQLite database which contains the user name and password. we take input from user, the user name and password when user click on login button it is checked that user is valid or not. if user is valid then Selection Menu will display.

```
package com.example.cloud_computing;
import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;

public class LoginDataBaseAdapter
{
    static final String DATABASE_NAME = "login.db";
```

```
static final int DATABASE_VERSION = 1;
public static final int NAME_COLUMN = 1;
static final String DATABASE_CREATE = "create table
"+"LOGIN"+
"( " +"ID"+" integer primary key autoincrement,"+
    "NAME text,USERNAME text,PASSWORD text); ";
public SQLiteDatabase db;
private final Context context;
private DataBaseHelper dbHelper;
public LoginDataBaseAdapter(Context _context)
{
    context = _context;
    dbHelper = new DataBaseHelper(context,
        DATABASE_NAME, null, DATABASE_VERSION);
}
public LoginDataBaseAdapter open() throws SQLException
{
    db = dbHelper.getWritableDatabase();
    return this;
}
public void close()
{
    db.close();
}
public SQLiteDatabase getDatabaseInstance()
{
    return db;
}

public void insertEntry(String name,String
    userName,String password)
{
    ContentValues newValues = new ContentValues();
    newValues.put("NAME", name);
    newValues.put("USERNAME", userName);
    newValues.put("PASSWORD",password);
    db.insert("LOGIN",null,newValues);
    ///Toast.makeText(context, "Reminder Is
    Successfully Saved", Toast.LENGTH_LONG).show();
}
```

```
    }
    public int deleteEntry(String UserName)
    {

        String where="USERNAME=?";
        int numberOfEntriesDeleted= db.delete("LOGIN",
where, new String[]{UserName}) ;
        "+numberOfEntriesDeleted, Toast.LENGTH_LONG).show();
        return numberOfEntriesDeleted;
    }
    public String getSinlgeEntry(String userName)
    {
        Cursor cursor=db.query("LOGIN", null, " USERNAME=?",
new String[]{userName}, null, null, null);
        if(cursor.getCount()<1) // UserName Not Exist
        {
            cursor.close();
            return "NOT EXIST";
        }
        cursor.moveToFirst();
        String password=
cursor.getString(cursor.getColumnIndex("PASSWORD"));
        cursor.close();
        return password;
    }
    public void updateEntry(String name,String
userName,String password)
    {
        // Define the updated row content.
        ContentValues updatedValues = new ContentValues();
        // Assign values for each row.
        updatedValues.put("NAME", name);
        updatedValues.put("USERNAME", userName);
        updatedValues.put("PASSWORD",password);
        String where="USERNAME = ?";
        db.update("LOGIN",updatedValues, where, new
String[]{userName});
    }
}
```

6.4 Code for Selection Menu

Using this Code we create one Menu Form, on that menu form we provide five options for user i.e Listening music, Watching videos, Study materials, View pictures and Code Execution. by clicking on that options it will connect to cloud Server and Server will execute the perticular Function.

```
package com.example.cloud_computing;

import java.io.BufferedOutputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.net.Socket;
import java.net.UnknownHostException;
import java.util.logging.Logger;
import android.app.Activity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;

public class Home extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.home_page);
    }

    public void Code_Execution_Click(View v)
    {
        Intent i=new Intent(Home.this,Connection.class);
        startActivity(i);
    }
}
```

```
}

    public void Listening_music_click(View v)
    {
        String url = "http://192.168.4.82/Songs/";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }

    public void Watching_videos_click(View v)
    {
        String url = "http://192.168.4.82/Videos/";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }

    public void Study_materials_click(View v)
    {
        String url = "http://192.168.4.82/Study%20Material/";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }

    public void Logout_click(View v)
    {
        Intent i =new Intent(Home.this,MainActivity.class);
        startActivity(i);
    }

    public void View_pictures_click(View v)
    {
        String url = "http://192.168.4.82/Images/";
        Intent i = new Intent(Intent.ACTION_VIEW);
        i.setData(Uri.parse(url));
        startActivity(i);
    }
}
```



```
public void Code_Execution_Click(View v)
{
    String url = "http://192.168.4.82/Code_Execution/";
    Intent i = new Intent(Intent.ACTION_VIEW);
    i.setData(Uri.parse(url));
    startActivity(i);
}
}
```

Chapter 7

TESTING

7.1 What is Software Testing

Software testing is the process of analyzing or operating software for the purpose of finding bugs. Testing can be described as a process used for revealing defects in software, and for establishing that the software has attained a specified degree of quality with respect to selected attribute. The fundamental objective of testing is to find defects, as early as possible and get them fixed.

Software Testing Process

- Test Planning high level plans which list test objectives, test approach, measurement criteria along with test schedule and resources.
- Test Design create test cases, identify test cases for automation(if applicable),prioritize test cases and finalize test iterations.
- Test Implementation Create test scripts using automated testing tools.
- Test Execution Execute the test cases on the test environment and test reports.
- Test analysis Use test and project metrics to calculate key indicators. The data usually will be obtained from your defect tracking system.
- Postmortem reviews Discuss lessons learnt and identify strategies which will prevent such problems in future.

7.2 Test Methods

7.2.1 Black Box Testing

It is also called as functional testing, it is the process of giving the input to the system and checking the output of the system. Without bothering about the system that how the system generates the output. It is also called as Behavior testing.

- Approach to testing where the program is considered as a Black Box.
- Testing based solely on analysis of requirements user specification, user documentation etc.
- The test cases are based on the specifications.
- Black box testing techniques apply to all levels of testing.
- Test planning and design can begin early in the software process.
- Tests are done from a users point of view.

As per black box testing rule we are testing the overview of project. we are testing the our implemented whole project working or not. just checking the giving input and getting output or not. just clicking on any android file it browse or not .

7.2.2 White Box Testing

1) White Box Testing :

White Box Testing method is applicable to the following levels of software testing those applied in your project:

- Unit Testing: For testing paths within a unit.
- Integration Testing: For testing paths between units.
- System Testing: For testing paths between subsystems.

1) Unit Testing : In this testing check the coding of android file those are installed on mobile. also check the php code those are use full for hosting the website. also check the code of interpreter those are interfacing the java compiler.

2) Integration Testing : In Intergration testing we are testing the path of different sub-model in this checking the Android codes whole path those are conneted properly or not means on the clicking on the video the video index is opened properly or not. doing same thing for every unit.

3) System Testing : In System Testing We are testing the whole system. In this mostly We are testing the paths withing subsystems.and also to checking whole system will be connected propely or not we have check as per IP adderss and android aaps linking the whole system working properly not . in that mostly We are testing the cloud is properly acessible on android. In that testing we are checking the those IASS, PASS, SaaS these concept is implemented properly well or not and its working properly or not .

7.3 Test Cases and Test Data

- Test data are inputs that have been devised to test the system.
- Test cases are inputs and output specification plus a statement of the function under test.
- Test data can be generated automatically or real.

Application Virtualization

31

Table 7.1 – continued from previous page

TC ID	OBJECTIVE	PREREQUISITES	STEPS TO BE FOLLOWED	EXPECTED RESULT	ACTUAL RE-SULT	REMARK
4	Register into the application	Login form should open	1.User Should enter the user name. 2.User Should enter the password. 3.User Should enter the confirm password. 4.click on <code>Register</code> button.	User got login id and password.	Register successfully.	pass
5	Use Application	User must login successfully	1.User select the application.	Selected application is open.	Selected application get execute..	pass
6	Store Data	User must click on browse button	1.Display list of files to the user.	Successful browse file..	User get appropriate Output.	pass

TABLE 7.1: Test Cases

Chapter 8

DEPLOYMENT

8.1 Login Window

Login Screen is appeared first when we start the application. The user can login here. If user dont have any login id and password then he / she may be able to create the login id. If the username and password does not matches then it gives the error as Matches are not found.



FIGURE 8.1: Login Window.

8.2 Register Screen

This is simply the registration Form. User can register here. After successful registration user can easily login to the cloud.

The image shows a registration form titled "Cloud Computing WORLD FORUM" with a logo. The form is titled "Registration Form" and contains four input fields: "Name" with placeholder text "Enter name....", "Username" with placeholder text "Enter username....", "Password" with placeholder text "Enter Password.....", and "Confirm Password" with placeholder text "Re-Enter Password.....". Below the fields are two buttons: "Register" and "Cancel". The background of the form has a blue sky with white clouds and the text "deposit photos" repeated several times.

FIGURE 8.2: Registration Form

8.3 Selection Menu

This screen consists of the selection menus which are dependent on logged user. These menus may be listening songs, videos, code execution, and download wallpapers.

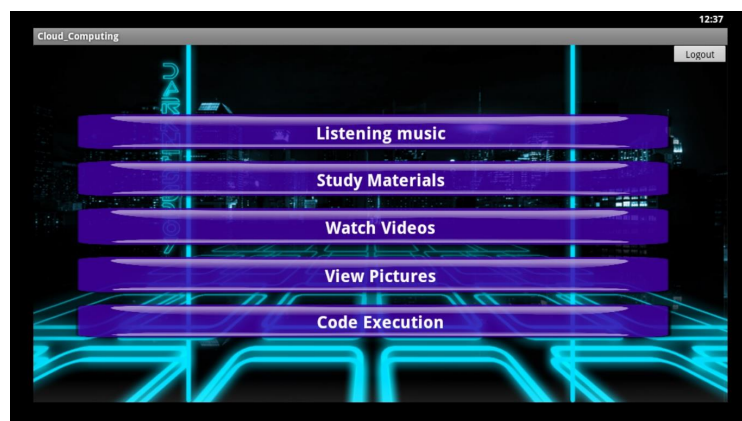


FIGURE 8.3: Menu Managmentl

8.4 Listening Music

This option will play the audio file. This mp3 file is already present on cloud. So within few second users can listen the song. Also users are able to download the mp3 file.



FIGURE 8.4: Listing music

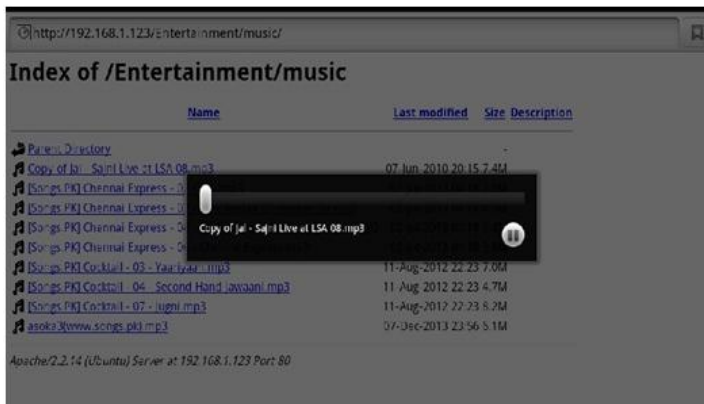


FIGURE 8.5: Play music

8.5 Watching Videos

If user wants to see the videos then selection of this menu will play the video which is stored on the cloud side. These menus have the fast response time as compared to the other website. Moving pictures and audio synchronization is there.



FIGURE 8.6: Watching video

8.6 Code Execution

This option will give the users to browse the program file. These programs file having the .java extension. The users may write the programs here. When user click on browse button then user have to choose the .java file. User have to input the address of the cloud. So when user select the particular file then and click on Execute then file is transferred to the cloud and execution is done at cloud side. When user click on Get Output then it will give the output below area which follows the these two buttons which are Execute and Get Output. **Current Menus On The Tablets**



FIGURE 8.7: Display for Execute and Get Output



FIGURE 8.8: Code writing on android

Chapter 9

CONCLUSION

Cloud computing is an emerging field which provides computation, software, data access, and storage services that do not require end-user knowledge of the physical location and configuration of the system that delivers the services. Cloud Computing is a general term used to describe a new class of network based computing that takes place over the Internet or intranet. The concepts of mobile and cloud computing, working environment and architecture of Cloud Computing, and android application development. Designed and developed a Cloud Computing Application for Java program creation, compilation and execution on the cloud using mobile computing systems like smart phones, iPods and notebooks. An application model can be created based on cloud computing by means of cloud computing mass data storage, high-speed computing capabilities.

In this project we conclude that there is no need to download any software. Because of this, Memory management is achieved, computing time is reduced. So we get fast response as much as possible. In this application platform share the Software as a Service among multiple users on cloud server via network easily.

Bibliography

- [1] Iosr jounaral of computer engineering(iosr-jce) www.iosrjournals.org implementing saas:cloud computing and android application framework for c-programming.
- [2] International journal of scientific and research publication, volume-2, issue-4,april-2012,issn 2250-3153 mobile applications with cloud computing.
- [3] Android-x86 project-run android on your pc(android-x86 - porting android to x86).
<http://www.android-x86.org/>.
- [4] M. satyanarayanan,v.bahl, r. caceres, and n. davies,the case for vm-based cloudlets in mobile computing.:ieee pervasive computing,2009.
- [5] B.g. chun and p. maniatitis. augmented smartphone applications through clone cloud execution.
- [6] A virtual cloud computationgprivoder for mobile devices.san francisco : Mcs'10, 2010.
- [7] Y.royon, s.frenot, and f.l.mouel virtualization of service gateways in multi-provider environments. heidelberg:cbse 2006,2006.
- [8] C.p.bezemer and a.zaidman multi-tenant saas applications: Maintenance dream or nightmare? antwerp, belgium:iwpse-evol'10,2010.
- [9] E.y.chen and m.ito. virtual smartphone over ip.montreal,qc, canada: Ieee wowmom, 2010.