

# **ADCAPSULE :PRACTICAL CONFINEMENT OF ADVERTISEMENTS IN ANDROID APPLICATIONS**

Submitted in partial fulfillment of the requirements for the award of Bachelor  
of Science Degree in Computer Science

By

**K. Nandhini (REG. NO. 38290051)**

**M.SABA TABUSSUM (REG.NO.38290074)**



**DEPARTMENT OF COMPUTER SCIENCE**

**SCHOOL OF COMPUTING**

**SATHYABAMA**

**INSTITUTE OF SCIENCE AND TECHNOLOGY**

**JEPPIAAR NAGAR, RAJIV GANDHI SALAI,**

**CHENNAI – 600119, TAMILNADU**

**MARCH 2021**



**SATHYABAMA**  
INSTITUTE OF SCIENCE AND TECHNOLOGY  
(DEEMED TO BE UNIVERSITY)  
Accredited with Grade "A" by NAAC  
(Established under Section 3 of UGC Act, 1956)  
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI- 600119  
[www.sathyabama.ac.in](http://www.sathyabama.ac.in)



---

## DEPARTMENT OF COMPUTER SCIENCE

### BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **M. SABA TABUSSUM (REG. NO. 38290074)** and **K. NANDHINI (REG. NO. 38290051)** who carried out the project entitled "**ADCAPSULE :PRACTICAL CONFINEMENT OF ADVERTISEMENTS IN ANDROID APPLICATIONS**" under our supervision from November 2020 to March 2021

Internal Guide

**Dr. J. ALBERT MAYAN M.E., Ph.D.,**

**Head of the Department**

**Submitted for Viva voce Examination held on \_\_\_\_\_**

**Internal Examiner**

**External Examiner**

## **DECLARATION**

I, **M. SABA TABUSSUM** hereby declare that the Project Report entitled **ADCAPSULE PRACTICAL CONFINEMENT OF ADVERTISEMENTS IN ANDROID APPLICATIONS** done by me under the guidance of **Ms.S.L.JanyShabui** is submitted in partial fulfillment of the requirements for the award of Bachelor of Science degree in Computer Science.

**DATE:**

**PLACE:CHENNAI**

**SIGNATURE OF THE CANDIDATE**

## ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to Board of Management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M. E., Ph.D., Dean**, School of Computing, **Dr. S. Vigneshwari M. E., Ph. D., and Dr. L. Lakshmanan, M. E., Ph. D., Heads of the Department** of Computer Science for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Ms.S.L.JanyShabu** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **DEPARTMENT OF COMPUTER SCIENCE** who were helpful in many ways for the completion of the project.

# ABSTRACT

Nowadays, app developers tend to integrate advertisement libraries (or ad libraries) into their apps to get revenue from ad networks. However, researches have shown that both ad libraries and ad contents could raise serious security and privacy concerns. In this paper, we propose AdCapsule, a user-level solution to practically confine advertisements, including ad libraries and ad contents. Our solution does not need to change the Android framework, nor requires the root privilege, thus can be readily deployed. Specifically, we propose the permission sandbox, which isolates the permissions used by ad libraries from the host app, and the file sandbox, which separates the file operations of advertisements. The ad library and ad content cannot read or write any file outside this sandbox. We have implemented a prototype of AdCapsule. Our evaluation results indicate that AdCapsule can successfully enforce security policies to block attempts of accessing private information or manipulating files of the host app, and the performance overhead introduced by AdCapsule is low.

## TABLE OF CONTENTS

| <b>CHAPTER<br/>No</b> | <b>TITLE</b>                        | <b>PAGE</b> |
|-----------------------|-------------------------------------|-------------|
|                       | LIST OF FIGURES                     | 8           |
|                       | LIST OF ABBREVIATIONS               | <b>9</b>    |
| <b>1</b>              | <b>INTRODUCTION</b>                 | 10          |
| 1.1                   | About the Project                   | 10          |
| <b>2</b>              | <b>SYSTEM ANALYSIS</b>              | 11          |
| 2.1                   | Existing system                     | 11          |
| 2.2                   | Proposed system                     | 12          |
| <b>3</b>              | <b>REQUIREMENTS SPECIFICATION</b>   | <b>13</b>   |
| 3.1                   | Introduction                        | 13          |
| 3.2                   | Hardware and Software specification | 14          |
| 3.3                   | Technologies Used                   | 15          |
| 3.3.1                 | Java                                | 15          |

|          |                                       |           |
|----------|---------------------------------------|-----------|
| 3.3.3.1  | Introduction to Java                  | 15        |
| 3.3.1.2  | Working of Java                       | 17        |
| <b>4</b> | <b>Project Purpose and Scope</b>      | <b>28</b> |
| 4.1      | Purpose                               | 29        |
| 4.2      | Project Scope                         | 29        |
| 4.3      | Product Perspective                   | 30        |
| 4.4      | System Features                       | 31        |
| 4.5      | Design and Implementation Constraints | 31        |
| 4.6      | Other Nonfunctional Requirements      | 32        |
| <b>5</b> | <b>SYSTEM DESIGN</b>                  | <b>43</b> |
| 5.1      | Architecture Diagram                  | 43        |
| <b>6</b> | <b>SYSTEM DESIGN – DETAILED</b>       | <b>43</b> |
| 6.1      | Modules                               | 45        |
| 6.2      | Module explanation                    | 45        |
| <b>7</b> | <b>CODING AND TESTING</b>             | <b>46</b> |
| 7.1      | Coding                                | 46        |
| 7.2      | Coding standards                      | 48        |
| 7.3      | Test procedure                        | 49        |

|     |                      |           |
|-----|----------------------|-----------|
| 7.4 | Test data and output | 50        |
|     | <b>SOURCE CODE</b>   | <b>54</b> |
|     | <b>SNAP SHOTS</b>    | <b>67</b> |
|     | <b>REFERENCES</b>    | <b>73</b> |

## LIST OF FIGURES

5.1 System Design

5.2 Sequence Diagram

5.3 Use Case Diagram

5.4 Activity Diagram

5.5 Collaboration Diagram:

5.6 Data Flow Diagram

5.7 Class Diagram

## **LIST OF ABBREVIATIONS**

JDK - Java Development Kit

DEX – Dalvik Executables

TCP –Transmission Control Protocol

IP – Internet Protocol

HTTP – Hyper Text Transfer Protocol

ADT – Android Development Tool

## **CHAPTER 1**

### **INTRODUCTION**

#### **Aim:**

The main aim is to have control of the in app advertisements in order to restrict the unauthorized access of personal information.

#### **Synopsis:**

Due to the proliferation of mobile phones in recent years there are many advantages and some disadvantages in smart-phones usage. Many applications like Ecommerce, Social media and games get access of user's permission to use some privileges. Normally while installing any application we used to grant permission to the particular application to access all the privileges. So when the applications which we install in our smart-phones have some tie-up with some advertisement company. So that for the benefits of the app developer those ads will run in our app. Here in this case the advertisements can behave maliciously some times. This malicious behavior extracts personal information like user's contact and files by using the permission access policy of particular app. This data theft may happen without the knowledge of user; the malicious ad will upload the collected information to the desired server. We propose a third party Server to validate the advertisement before the ad get loaded into the app.

## **CHAPTER 2**

### **SYSTEM ANALYSIS**

#### **2.1 EXISTING SYSTEM**

Though we have shown security concerns in previous sections reported by other researchers, we want to further understand the state of the art of ad libraries and the privacy related behaviors. Due to the special nature of the Android ecosystem in China, e.g., highly customized Android systems without Google services, we also collected 29 ad libraries which are popular in China.

Moreover, to check whether ad libraries perform more privacy-concerned operations than other libraries, we collected 119 popular development tool libraries on App-Brain for comparison.

Normally while installing any application we used to grant permission to the particular application to access all the privileges. So when the applications which we install in our smartphones have some tie-up with any advertisement company.

So for the benefits of the app developer those ads will run in our app. Here in this case the advertisements can behave maliciously some times. This malicious behavior extracts personal information like user's contact and files by using the permission access policy of particular app.

## **2.2 PROPOSED SYSTEM**

We propose a third party Server to monitor the advertisement library, Here the User whenever install any new application the third party server will start checking the incoming adds.

So before loading a new advertisement into the application our system will analyze the triggering components inside the ads and its behavior. If there is any chance of accessing the user details or files or contacts, thus an ad can get unauthorized access permissions from the hosted app.

Our system is capable of detecting that kind of ads and restricts their access, then pop up the required access permission list to the user. Only when the user accepts that permission it will allow the ad to run inside the hosted app. Otherwise our system will block the malicious ads.

## **CHAPTER 3**

### **REQUIREMENT SPECIFICATIONS**

#### **3.1 INTRODUCTION**

There is no doubt that smart phones have become an essential part of our life. One potential reason of such popularity is that there are millions of applications (or apps) available to users. For instance, 2; 600; 000 apps are available in Google Play Store. Among these apps, more than 90% are free ones. Users download and use these apps without paying app developers.

To compensate for their work, app developers have incentives to integrate one or more advertisement (or ad) libraries in their apps (or host apps), and get paid. To facilitate the integration, ad service providers offer SDKs or libraries to app developers. App developers integrate these libraries with minor efforts, sometimes adding only small amount lines of code. After that, ad libraries communicate with the ad network, retrieve and render ad contents.

In this work, we propose AdCapsule, a user-level solution to confine ads in Android apps. Our solution does not need to change the Android framework, and it does not require the root privilege, thus can be readily deployed. Specifically, it leverages two sandboxes, i.e., the permission sandbox and the file sandbox for the purpose. The permission sandbox isolates permissions used by ad libraries from the ones used by the host app. In this case, ad libraries are running inside an isolated sandbox with their own permissions and contexts, thus cannot abuse permissions of the host app to perform aggressive operations. The file sandbox separates file-related operations of ad libraries and ad contents. The entire files read and write operations are confined inside a sandbox. They cannot touch any file outside this sandbox. By doing so, our system ensures that ad libraries and ad contents cannot directly retrieve or indirectly infer any private information by accessing files of the host app.

## **3.2 HARDWARE AND SOFTWARE SPECIFICATION**

### **3.2.1 HARDWARE REQUIREMENTS**

- RAM : 4GB & Above
- Processor : Pentium IV & Above
- Hard disk : 250GB and Above

### **3.2.2 SOFTWARE REQUIREMENTS**

- Windows 7 operating system and above
- JDK 1.7
- Tomcat7
- My-SQL 5
- Android Studio 3.1

### **3.2.3 Android Phone Requirements:**

- OS : Jelly-Bean & Above
- No of Devices : 1

## **3.3 TECHNOLOGIES USED**

- Java
- JSP Servlet.
- Android

### **3.3.1 JAVA**

Java is an object-oriented programming language developed initially by James Gosling and colleagues at Sun Microsystems. The language, initially called Oak (named after the oak trees outside Gosling's office), was intended to replace C++, although the feature set better resembles that of Objective C.

#### **3.3.1.1 INTRODUCTION TO JAVA**

Java has been around since 1991, developed by a small team of Sun Microsystems developers in a project originally called the Green project. The intent of the project was to develop a platform-independent software technology that would be used in the consumer electronics industry. The language that the team created was originally called Oak.

The first implementation of Oak was in a PDA-type device called Star Seven (\*7) that consisted of the Oak language, an operating system called GreenOS, a user interface, and hardware. The name \*7 was derived from the telephone sequence that was used in the team's office and that was dialed in order to answer any ringing telephone from any other phone in the office.

Around the time the First Person project was floundering in consumer electronics, a new craze was gaining momentum in America; the craze was called "Web surfing." The World Wide Web, a name applied to the Internet's millions of linked HTML documents was suddenly becoming popular for use by the masses. The reason for this was the introduction of a graphical Web browser called Mosaic, developed by ncSA. The browser simplified Web browsing by combining text and graphics into a single interface to eliminate the need for users to learn many confusing UNIX and DOS commands. Navigating around the Web was much easier using Mosaic.

It has only been since 1994 that Oak technology has been applied to the Web. In 1994, two Sun developers created the first version of Hot Java, and then called Web Runner, which is a graphical browser for the Web that exists today. The browser was coded entirely in the Oak language, by this time called Java. Soon after, the Java compiler was rewritten in the Java language from its original C code, thus proving that Java could be used effectively as an application language. Sun introduced Java in May 1995 at the Sun World 95 convention.

Web surfing has become an enormously popular practice among millions of computer users. Until Java, however, the content of information on the Internet has been a bland series of HTML documents. Web users are hungry for applications that are interactive, that users can

execute no matter what hardware or software platform they are using, and that travel across heterogeneous networks and do not spread viruses to their computers. Java can create such applications.

### 3.3.1.2 WORKING OF JAVA

For those who are new to object-oriented programming, the concept of a class will be new to you. Simplistically, a class is the definition for a segment of code that can contain both data (called attributes) and functions (called methods).

When the interpreter executes a class, it looks for a particular method by the name of **main**, which will sound familiar to C programmers. The main method is passed as a parameter an array of strings (similar to the `argv[]` of C), and is declared as a static method.

To output text from the program, we execute the **println** method of **System.out**, which is java's output stream. UNIX users will appreciate the theory behind such a stream, as it is actually standard output. For those who are instead used to the Wintel platform, it will write the string passed to it to the user's program.

Java consists of two things :

- Programming language
- Platform

### 3.3.1.3 THE JAVA PROGRAMMING LANGUAGE

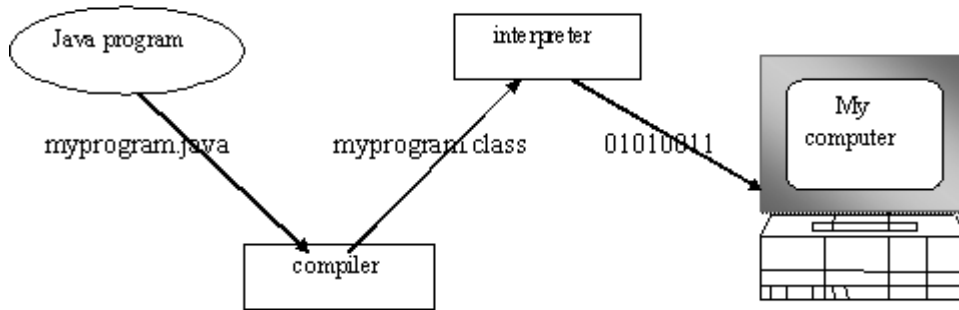
Java is a high-level programming language that is all of the following:

- Simple
- Object-oriented

- Distributed
- Interpreted
- Robust
- Secure
- Architecture-neutral
- Portable
- High-performance
- Multithreaded
- Dynamic

The code can bring about changes whenever felt necessary. Some of the standards needed to achieve the above-mentioned objectives are as follows:

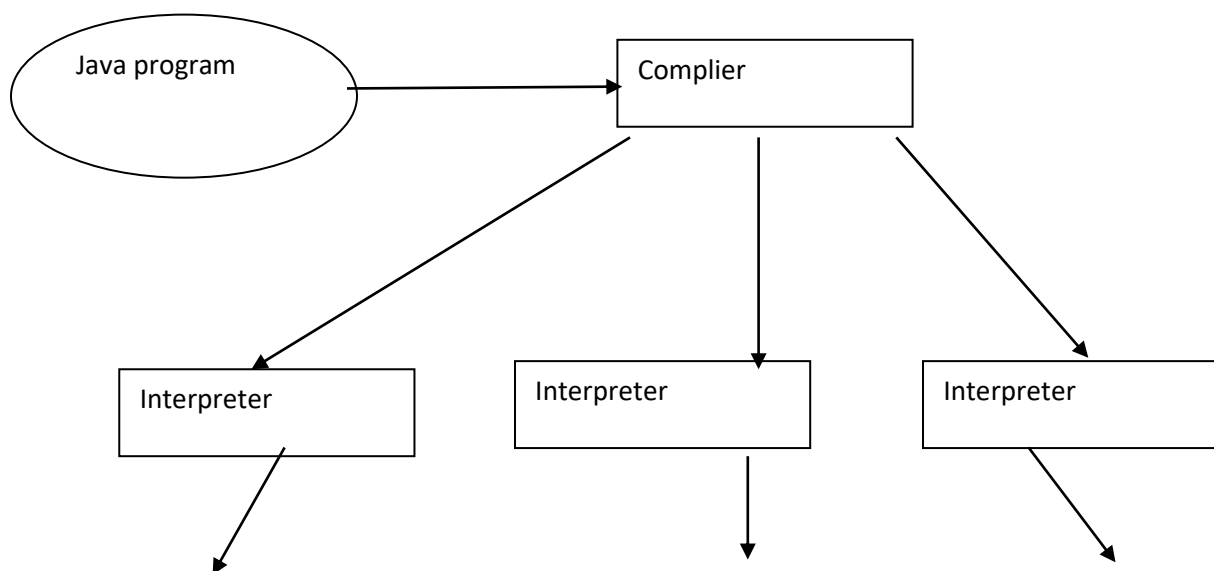
Java is unusual in that each Java program is both compiled and interpreted. With a compiler, you translate a Java program into an intermediate language called **Java byte codes** – the platform independent codes interpreted by the Java interpreter. With an interpreter, each Java byte code instruction is parsed and run on the computer. Compilation happens just once; interpretation occurs each time the program is executed. This figure illustrates how it works:



**Fig.3.1**

You can think of Java byte codes as the machine code instructions for the **Java Virtual Machine (JVM)**. Every Java interpreter, whether it's a Java development tool or a Web browser that can run Java applets, is an implementation of JVM. That JVM can also be implemented in hardware. Java byte codes help make “write once, run anywhere” possible.

You can compile your Java program into byte codes on any platform that has a Java compiler. The byte codes can then be run on any implementation of the JVM. For example, that same Java program can be run on Windows NT, Solaris and Macintosh





**PC-Compatible**

**Sun Ultra Solaris**

**Power macintosh**

**Windows NT**

**System 8**

### 3.3.1.4 THE JAVA PLATFORM

A platform is the hardware or software environment in which a program runs. The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other, hardware-based platforms. Most other platforms are described as a combination of hardware and operating system.

The Java platform has two components :

- The Java Virtual Machine (JVM)
- The Java Application Programming Interface (Java API)

You've already been introduced to the JVM. It's the base for the Java platform and is ported onto various hardware-based platforms.

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets. The Java API is grouped into libraries (**packages**) of related components. The following figure depicts a Java program, such as

an application or applet, that's running on the Java platform. As the figure shows, the Java API and Virtual Machine insulates the Java program from hardware dependencies.

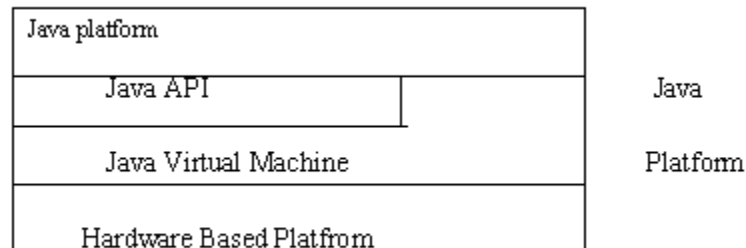


Fig.3.3

As a platform-independent environment, Java can be a bit slower than native code. However, smart compilers, weel-tuned interpreters, and just-in-time byte compilers can bring Java's performance close to that of native code without threatening portability.

### 3.3.2 APACHE TOMCAT SERVER

Apache Tomcat (formerly under the Apache Jakarta Project; Tomcat is now a top level project) is a web container developed at the Apache Software Foundation. Tomcat implements the servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, providing an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management but can also be configured by editing configuration files that are normally XML-formatted. Because Tomcat includes its own HTTP server internally, it is also considered a standalone web server.

## **Environment**

Tomcat is a web server that supports servlets and JSPs. Tomcat comes with the Jasper compiler that compiles JSPs into servlets.

The Tomcat servlet engine is often used in combination with an Apache web server or other web servers. Tomcat can also function as an independent web server. Earlier in its development, the perception existed that standalone Tomcat was only suitable for development environments and other environments with minimal requirements for speed and transaction handling. However, that perception no longer exists; Tomcat is increasingly used as a standalone web server in high-traffic, high-availability environments.

Since its developers wrote Tomcat in Java, it runs on any operating system that has a JVM.

## **Product features**

Tomcat 3.x (initial release)

- implements the Servlet 2.2 and JSP 1.1 specifications
- servlet reloading
- basic HTTP functionality Tomcat 4.x
- implements the Servlet 2.3 and JSP 1.2 specifications
- servlet container redesigned as Catalina
- JSP engine redesigned as Jasper
- Coyote connector
- Java Management Extensions (JMX), JSP and Struts-based administration
- Tomcat 5.x

- implements the Servlet 2.4 and JSP 2.0 specifications
- reduced garbage collection, improved performance and scalability
- native Windows and Unix wrappers for platform integration
- faster JSP parsing

## **History**

Tomcat started off as a servlet specification implementation by James Duncan Davidson, a software architect at Sun. He later helped make the project open source and played a key role in its donation by Sun to the Apache Software Foundation.

Davidson had initially hoped that the project would become open-sourced and, since most open-source projects had O'Reilly books associated with them featuring an animal on the cover, he wanted to name the project after an animal. He came up with Tomcat since he reasoned the animal represented something that could take care of and fend for itself. His wish to see an animal cover eventually came true when O'Reilly published their Tomcat book with a tomcat on the cover.

### **3.3.3 Android Introduction:**

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The operating system has developed a lot in last 15 years starting from black and white phones to recent smart phones or mini computers. One of the most widely used mobile OS these days is android. The android is software that was founded in Palo Alto of California in 2003.

The android is a powerful operating system and it supports large number of applications in Smart phones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform. The android is an open source operating system means that it's free and any one can use it. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 and latest updated version is jelly bean.

The android is a operating system and is a stack of software components which is divided into five sections and four main layers that is



### 3.3.3.1 Linux kernel:

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management, device management like camera, keypad, display etc the kernel handles all the things. The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hard ware devices on the system.

### **3.3.3.2 Libraries:**

The on top of a Linux kernel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

### **3.3.3.3 Android Runtime:**

The android runtime provides a key component called Dalvik Virtual Machine which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run its own process. The Dalvik VM executes the files in the .dex format.

### **3.3.3.4 Application frame work:**

The application frame work layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

### **3.3.3.5 Applications:**

You will find all the android applications at the top layer and you will write your application and install on this layer. Examples of such applications are contacts, books, browsers, services etc. Each application performs a different role in the overall applications.

### **Advantages:**

- Android is Linux based open source operating system , it can be developed by any one
- Easy access to the android apps
- You can replace the battery and mass storage, disk drive and UDB option
- Its supports all Google services
- The operating system is able to inform you of a new SMS and Emails or latest updates.
- It supports Multitasking
- Android phone can also function as a router to share internet
- Its free to customize
- Can install a modified ROM
- Its supports 2D and 3D graphics

## CHAPTER 4

### 4.1 Project Purpose and Scope

#### 4.1 Purpose

The purpose of this project is to compensate for their work, app developers have incentives to integrate one or more advertisement (or ad) libraries in their apps (or host apps), and get paid. To facilitate the integration, ad service providers offer SDKs or libraries to app developers. App developers integrate these libraries with minor efforts, sometimes adding only small amount lines of code. After that, ad libraries communicate with the ad network, retrieve and render ad contents.

Android takes advantage of the underlying Linux operating system to isolate different apps and their resources by making each app run in its own Virtual Machine with a unique UID. Unless explicitly specified, one app cannot read other app's files by default. To communicate with system services and perform dangerous operations, an app should require corresponding permissions. And these permissions should be granted by users when the app is being installed. For instance, if the app wants to send SMS messages in the background, then the app should have the `SEND_SMS` permission.

Ad libraries and the host app are running inside a same virtual machine and sharing the same privileges. Thus, ad libraries could probe and abuse permissions of the host app, which causes privacy concerns to users, and further ruins the reputation of the host app. However, in order to get revenue from the ad network, app developers have no choice but trusting and integrating ad libraries. We need a practical solution to confine these libraries.

## 4.2 Project Scope

Our system is mainly developed for app developers. In particular, app developers integrate our library together with ad libraries into their projects, and add a few lines of code to activate AdCapsule when the app is started. After that, developers can define the security policies according to ad library's package name list and put the policy file into the raw apk file.

Finally, developers can sign the apk file and upload it to the app store. Developers may set the policy of none to some operations. In this case, users should take care the prompt windows when using the app. These windows will tell users that the ad library is performing some operations that need to take further actions.

Since our system has the same privilege with the ad library, and they are running inside the same process, malicious ad libraries could bypass our system leveraging the similar hooking mechanisms we used. Further, ad libraries can invoke the services or remove our system by native code. Our system took this into consideration and proposed several mechanisms for self-protection.

## 4.3 Product Perspective

In this work, in-app ad libraries and ad contents are untrusted. Ad libraries are untrusted either because they are aggressive or malicious, or they are not malicious but vulnerable. For better ad targeting, ad libraries have the motivation to aggressively track users, using the retrieved identifier, location data or the list of installed apps. While for ad contents, they are

untrusted because they are possibly malicious. Note that ad contents fetched from the ad network are usually provided by advertisers, not the ad providers.

Advertisers could leverage malicious ad contents to infer user's sensitive information by accessing user's location or local files using JavaScript. App developers are trusted and they want to get revenue from ad providers by integrating ad libraries into their apps. At the same time, they have the incentive to maintain the reputation of their apps and regulate these libraries. Similar to other works, we trust the underlying Android framework and the operating system.

## **4.4 System Features**

In summary, this paper makes the following contributions:

We propose a practical solution called AdCapsule to confine advertisements, including both ad libraries and ad contents, without the need to change the Android framework or existing ad libraries.

We propose two user-level sandboxes, i.e., the permissions sandbox, to isolate the privacy-related operations of ad libraries, and the file sandbox to confine the file operations of ad libraries and ad contents.

We have implemented and evaluated a prototype of AdCapsule. The evaluation results demonstrate the effectiveness and compatibility of AdCapsule. And it incurs low performance overhead.

## **4.5 Design and Implementation Constraints**

### **4.5.1 Constraints in Analysis**

- ◆ Constraints as Informal Text

- ◆ Constraints as Operational Restrictions
- ◆ Constraints Integrated in Existing Model Concepts
- ◆ Constraints as a Separate Concept
- ◆ Constraints Implied by the Model Structure

#### **4.5.2 Constraints in Design**

- ◆ Determination of the Involved Classes
- ◆ Determination of the Involved Objects
- ◆ Determination of the Involved Actions
- ◆ Determination of the Require Clauses
- ◆ Global actions and Constraint Realization

#### **4.5.3 Constraints in Implementation**

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

### **4.6 Other Nonfunctional Requirements**

#### **4.6.1 Performance Requirements**

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;
- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following.

Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

#### **4.6.2 Safety Requirements**

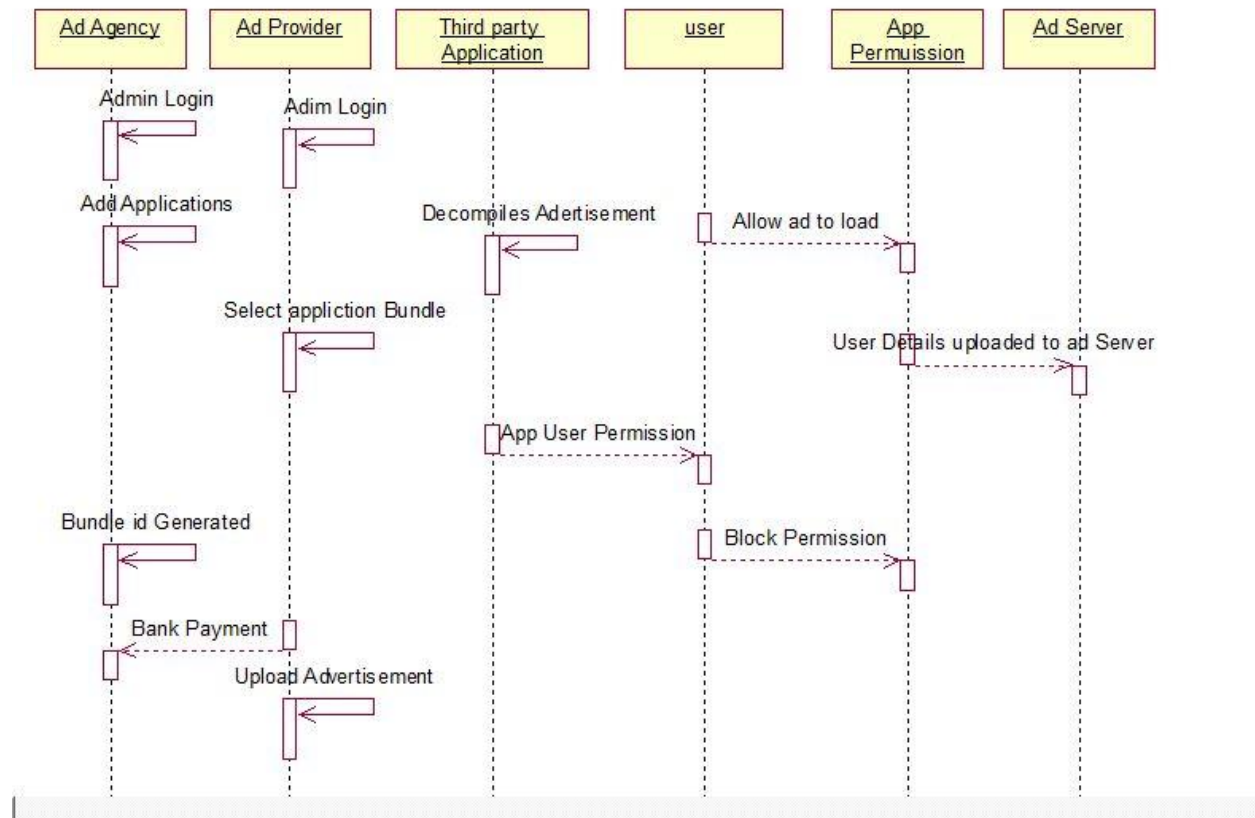
1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

## **CHAPTER 5**

### **Architecture Diagram**



A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.



## 5.3 Use Case Diagram:

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

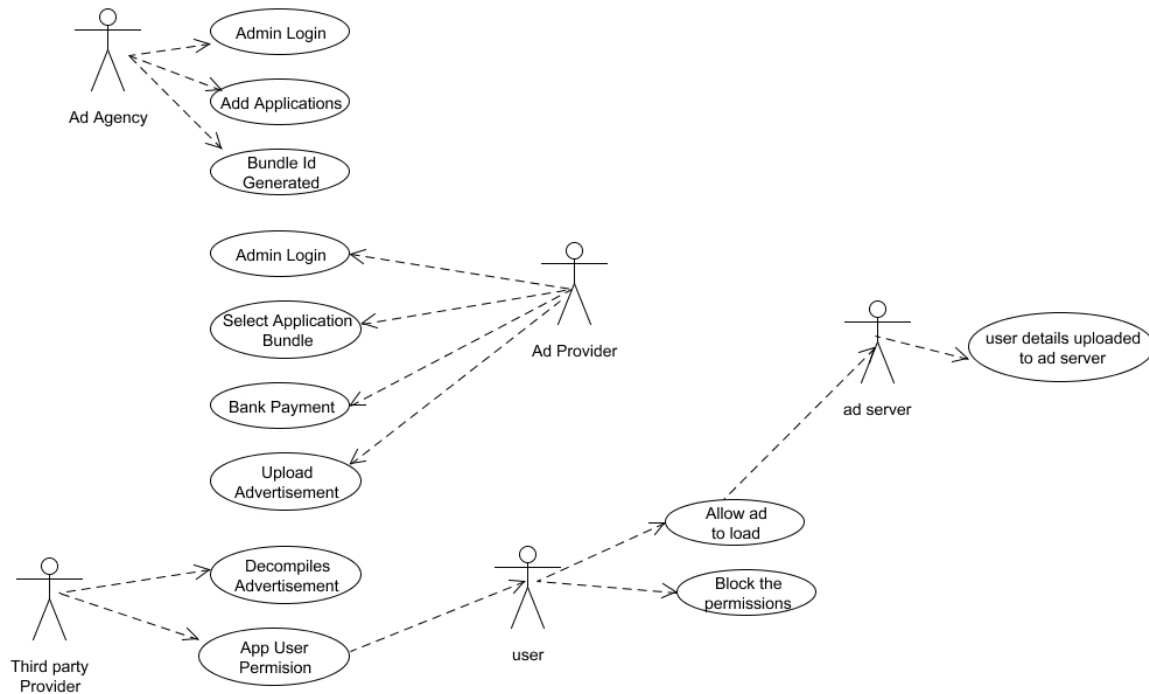
### 5.3.1. USECASE DIAGRAM

A Use case Diagram is used to present a graphical overview of the functionality provided by a system in terms of actors, their goals and any dependencies between those use cases.

Use case diagram consists of two parts:

**Use case:** A use case describes a sequence of actions that provided something of measurable value to an actor and is drawn as a horizontal ellipse.

**Actor:** An actor is a person, organization or external system that plays a role in one or more interaction with the system.

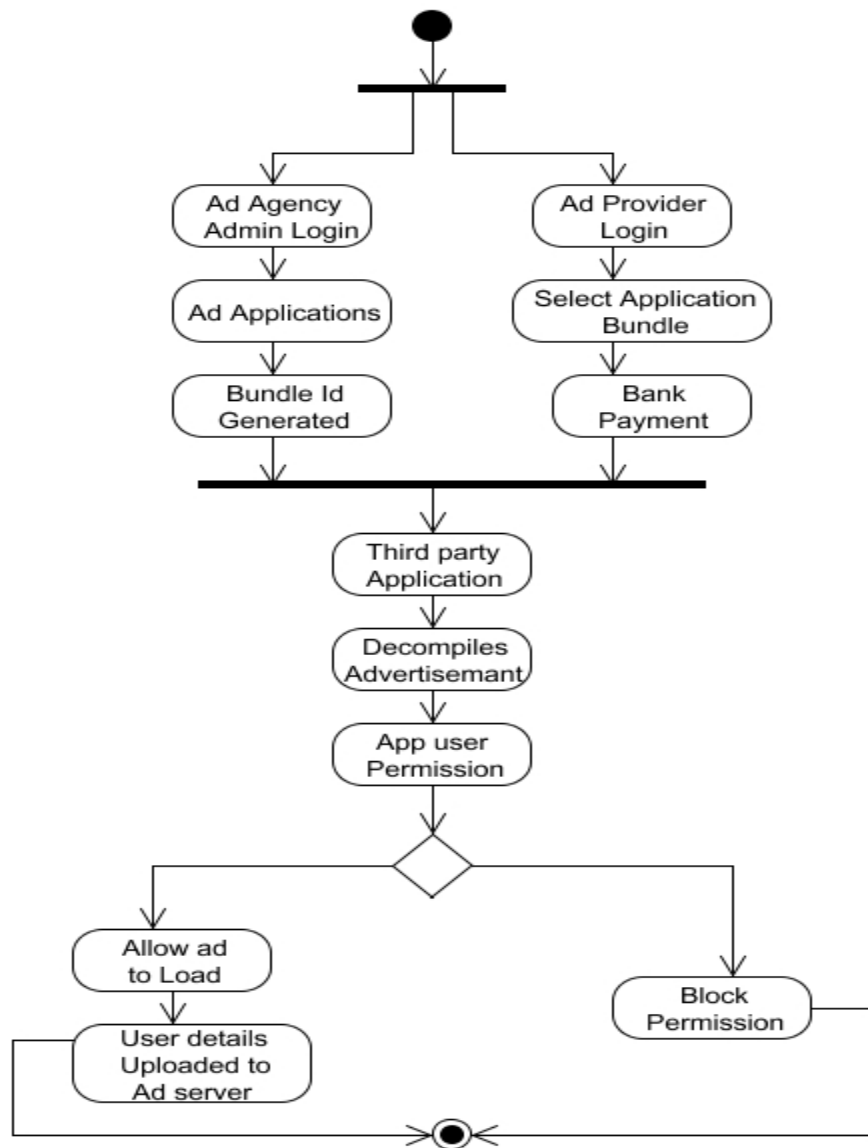


## 5.4 Activity Diagram:

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

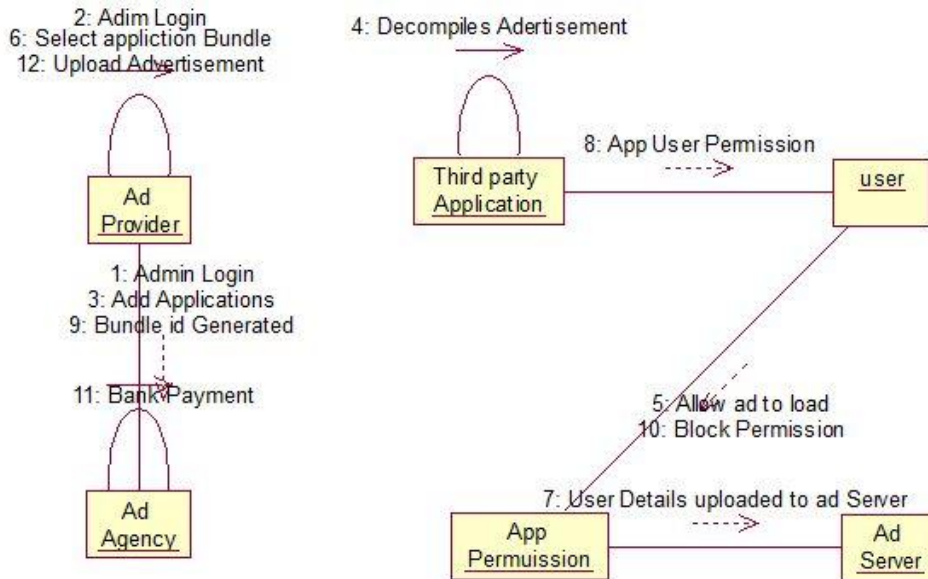
- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.



## 5.5 Collaboration Diagram:

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist.

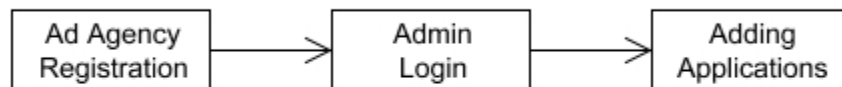
The collaboration diagram illustrates messages being sent between classes and objects.



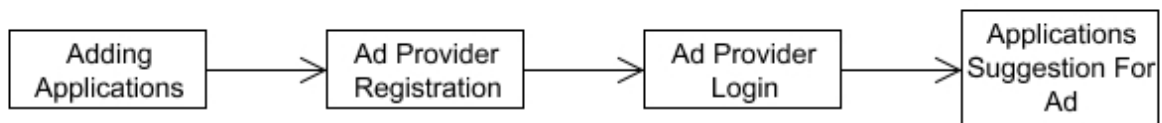
## 5.6 DATA FLOW DIAGRAM:

A Data Flow Diagram (DFD) is a graphical representation of the “flow” of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

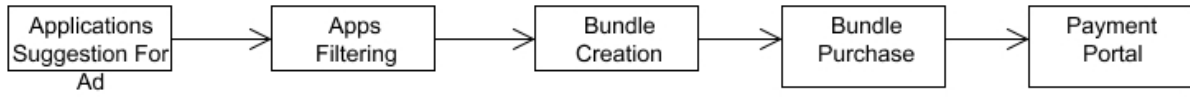
### Level 0:



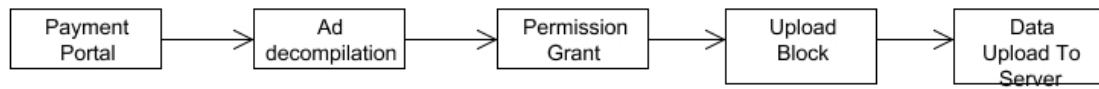
### Level 1:



## Level 2:L

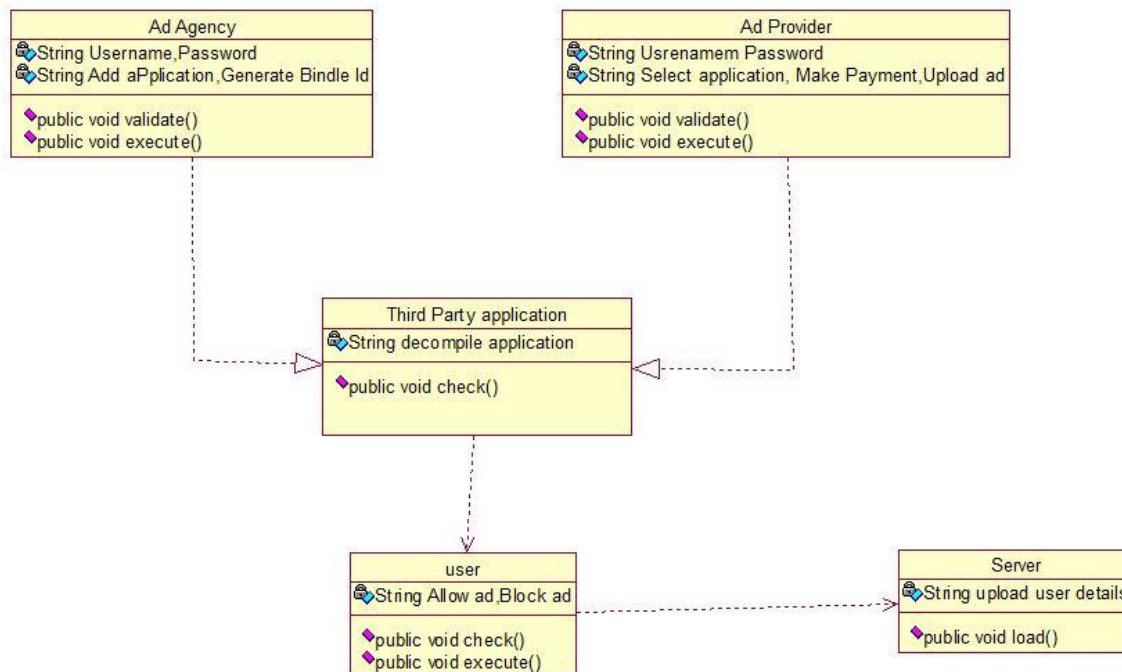


## Level 3:



## 5.7 Class Diagram

A Class diagram in the Unified Modelling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.



## **CHAPTER 6**

### **SYSTEM DESIGN**

In this work, we have designed a user-level solution to confine advertisements, including ad libraries and ad contents. Our system could be readily deployed since it does not need to change Android framework, nor needs the root privilege. It leverages two sandboxes to confine privacy-concerned operations and file operations of advertisements. We have implemented a prototype of our system and the evaluation results demonstrate the effectiveness and compatibility of our system, and the performance overhead is low.

We have implemented a prototype of AdCapsule. The permission sandbox and file sandbox are implemented in Java programming language. The security policies are implemented in XML format.

#### **6.1 MODULES**

- **Adding Applications in Ad Agency**
- **Ad Provider Purchasing APP Bundle**
- **Hosting Ad in APP Bundle**
- **Third-Party Service Provider**

## **6.2 MODULE EXPLANATION:**

### **6.2.1 Adding Applications in Ad Agency:**

In-order to use our application the users need to do user registration. After Ad Agency user Authentication the users or application developers will add their applications to the Ad-Agency portal. While adding an application they has to give basic information about the app like number of downloads, number of daily user logins, ratings, and App Id is mandatory. After that when an Ad-Agency admin sign-in into the portal the requested applications for advertisements will be shown. Here the admin has the privilege to set amount for the requested applications.

### **6.2.2 Ad Provider Purchasing APP Bundle:**

After the Ad Provider User registers into Ad Agency, The Provider as well as Agency has to generate new bank account using our Bank application. After accounts created successfully they can do money transfer for any desired accounts. Then the Ad Provider has to sign-in the Ad-Agency page and look for apps satisfying their criteria. They can select the apps in a bundle by applying filters based on their type of advertisement content, ratings, cost , number of downloads and number of daily users. After filtration an app bundle will be formed with Agency's applied offers. So now the Ad Provider has to purchase the app bundle.

### **6.2.3 Hosting Ad in APP Bundle:**

After bundle purchase the Provider has to upload the suitable ad content which is of types transport, education or health. After uploading the ad content into Agency, it will be automatically loaded into the end applications whose app id is there in the selected bundle. For each category of application we will be showing a related category Android Application. For

example we run a health related app. Once an advertisement gets hosted into that application the ad content will start initiating its malicious behavior. If the application uses permission like sms or contacts the ad-content will make use of it and send all user's sms or contacts to their ad-Server automatically without user's knowledge in background.

#### **6.2.4 Third-Party Service Provider:**

We implement a Third Party Service Provider in Ad-Agency side to verify the malicious contents of incoming advertisements .So whenever a new adcomponent is loaded into the empty spaces allocated for specific ads. The ads get loaded into certain locations inside the host apps. We need to do a preprocessing technique in our proposed Third party Service Provider. So when an advertisement is loaded first into the application the Service provider will check or process all the running components using decompilation and find out the possible permission access that the ads going to get from the hosted application. Then the Third party application will send a request to the application user in order to grant permission for authorized access. A popup will be shown to the app user and get the authorized access. Hereby we can restrict the ad components from getting access to user's sensitive data. Thus we can handle the ads from misbehaviors. This is known as confinement of advertisements.

## **CHAPTER 7**

### **CODING AND TESTING**

#### **7.1 CODING**

Once the design aspect of the system is finalized the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

#### **7.2 CODING STANDARDS**

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

### 7.2.1 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

#### **Class names**

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

#### **Member Function and Data Member name**

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

### 7.2.2 VALUE CONVENTIONS

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

### **7.2.3 SCRIPT WRITING AND COMMENTING STANDARD**

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

### **7.2.4 MESSAGE BOX FORMAT**

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.
- ! – Information to the user.

## **7.3 TEST PROCEDURE**

### **SYSTEM TESTING**

Testing is performed to identify errors. It is used for quality assurance. Testing is an integral part of the entire development and maintenance process. The goal of the testing during phase is to verify that the specification has been accurately and completely incorporated into the design, as well as to ensure the correctness of the design itself. For example the design must not have any logic faults in the design is detected before coding commences, otherwise the cost of fixing the

faults will be considerably higher as reflected. Detection of design faults can be achieved by means of inspection as well as walkthrough.

Testing is one of the important steps in the software development phase. Testing checks for the errors, as a whole of the project testing involves the following test cases:

- Static analysis is used to investigate the structural properties of the Source code.
- Dynamic testing is used to investigate the behavior of the source code by executing the program on the test data.

## **7.4 TEST DATA AND OUTPUT**

### **7.4.1 UNIT TESTING**

Unit testing is conducted to verify the functional performance of each modular component of the software. Unit testing focuses on the smallest unit of the software design (i.e.), the module. The white-box testing techniques were heavily employed for unit testing.

### **7.4.2 FUNCTIONAL TESTS**

Functional test cases involved exercising the code with nominal input values for which the expected results are known, as well as boundary values and special values, such as logically related inputs, files of identical elements, and empty files.

Three types of tests in Functional test:

- Performance Test
- Stress Test
- Structure Test

### **7.4.3 PERFORMANCE TEST**

It determines the amount of execution time spent in various parts of the unit, program throughput, and response time and device utilization by the program unit.

#### **7.4.4 STRESS TEST**

Stress Test is those test designed to intentionally break the unit. A Great deal can be learned about the strength and limitations of a program by examining the manner in which a programmer in which a program unit breaks.

#### **7.4.5 STRUCTURED TEST**

Structure Tests are concerned with exercising the internal logic of a program and traversing particular execution paths. The way in which White-Box test strategy was employed to ensure that the test cases could Guarantee that all independent paths within a module have been have been exercised at least once.

- Exercise all logical decisions on their true or false sides.
- Execute all loops at their boundaries and within their operational bounds.
- Exercise internal data structures to assure their validity.
- Checking attributes for their correctness.
- Handling end of file condition, I/O errors, buffer problems and textual errors in output information

#### **7.4.6 INTEGRATION TESTING**

Integration testing is a systematic technique for construction the program structure while at the same time conducting tests to uncover errors associated with interfacing. i.e., integration testing is the complete testing of the set of modules which makes up the product. The objective is to take untested modules and build a program structure tester should identify critical modules. Critical modules should be tested as early as possible. One approach is to wait until all the units have passed testing, and then combine them and then tested. This approach is evolved from unstructured testing of small programs. Another strategy is to construct the product in increments of tested units. A small set of modules are integrated together and tested, to which another module is added and tested in combination. And so on. The advantages of this approach are that, interface dispenses can be easily found and corrected.

The major error that was faced during the project is linking error. When all the modules are combined the link is not set properly with all support files. Then we checked out for interconnection and the links. Errors are localized to the new module and its intercommunications. The product development can be staged, and modules integrated in as they complete unit testing. Testing is completed when the last module is integrated and tested.

## **7.5 TESTING TECHNIQUES / TESTING STRATERGIES**

### **7.5.1 TESTING**

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet –undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang

together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing.

The software testing process commences once the program is created and the documentation and related data structures are designed. Software testing is essential for correcting errors. Otherwise the program or the project is not said to be complete. Software testing is the critical element of software quality assurance and represents the ultimate the review of specification design and coding. Testing is the process of executing the program with the intent of finding the error. A good test case design is one that as a probability of finding an yet undiscovered error. A successful test is one that uncovers an yet undiscovered error. Any engineering product can be tested in one of the two ways:

#### **7.5.1.1 WHITE BOX TESTING**

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Cyclometric complexity
- Deriving test cases

- Graph matrices Control

### **7.5.1.2 BLACK BOX TESTING**

In this testing by knowing the internal operation of a product, test can be conducted to ensure that “all gears mesh”, that is the internal operation performs according to specification and all internal components have been adequately exercised. It fundamentally focuses on the functional requirements of the software.

The steps involved in black box test case design are:

- Graph based testing methods
- Equivalence partitioning
- Boundary value analysis
- Comparison testing

### **7.5.2 SOFTWARE TESTING STRATEGIES:**

A software testing strategy provides a road map for the software developer. Testing is a set activity that can be planned in advance and conducted systematically. For this reason a template for software testing a set of steps into which we can place specific test case design methods should be strategy should have the following characteristics:

- Testing begins at the module level and works “outward” toward the integration of the entire computer based system.
- Different testing techniques are appropriate at different points in time.
- The developer of the software and an independent test group conducts testing.

- Testing and Debugging are different activities but debugging must be accommodated in any testing strategy.

#### **7.5.2.1 INTEGRATION TESTING:**

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with. Individual modules, which are highly prone to interface errors, should not be assumed to work instantly when we put them together. The problem of course, is “putting them together”- interfacing. There may be the chances of data lost across on another’s sub functions, when combined may not produce the desired major function; individually acceptable impression may be magnified to unacceptable levels; global data structures can present problems.

#### **7.5.2.2 PROGRAM TESTING:**

The logical and syntax errors have been pointed out by program testing. A syntax error is an error in a program statement that in violates one or more rules of the language in which it is written. An improperly defined field dimension or omitted keywords are common syntax error. These errors are shown through error messages generated by the computer. A logic error on the other hand deals with the incorrect data fields, out-off-range items and invalid combinations. Since the compiler s will not deduct logical error, the programmer must examine the output. Condition testing exercises the logical conditions contained in a module. The possible types of elements in a condition include a Boolean operator, Boolean variable, a pair of Boolean parentheses A relational operator or on arithmetic expression. Condition testing method focuses

on testing each condition in the program the purpose of condition test is to deduct not only errors in the condition of a program but also other a errors in the program.

#### **7.5.2.3 SECURITY TESTING:**

Security testing attempts to verify the protection mechanisms built in to a system well, in fact, protect it from improper penetration. The system security must be tested for invulnerability from frontal attack must also be tested for invulnerability from rear attack. During security, the tester places the role of individual who desires to penetrate system.

#### **7.5.2.4 VALIDATION TESTING**

At the culmination of integration testing, software is completely assembled as a package. Interfacing errors have been uncovered and corrected and a final series of software test-validation testing begins. Validation testing can be defined in many ways, but a simple definition is that validation succeeds when the software functions in manner that is reasonably expected by the customer. Software validation is achieved through a series of black box tests that demonstrate conformity with requirement. After validation test has been conducted, one of two conditions exists.

- \* The function or performance characteristics confirm to specifications and are accepted.
- \* A validation from specification is uncovered and a deficiency created.

Deviation or errors discovered at this step in this project is corrected prior to completion of the project with the help of the user by negotiating to establish a method for resolving deficiencies. Thus the proposed system under consideration has been tested by using validation

testing and found to be working satisfactorily. Though there were deficiencies in the system they were not catastrophic

#### **7.5.2.5 USER ACCEPTANCE TESTING**

User acceptance of the system is a key factor for the success of any system. The system under consideration is tested for user acceptance by constantly keeping in touch with prospective system and user at the time of developing and making changes whenever required. This is done in regarding to the following points.

- Input screen design.
- Output screen design.

### **Source Code**

#### **Applicationdetails.java**

```
package logics;

import java.io.IOException;

import java.io.PrintWriter;

import javax.servlet.RequestDispatcher;

import javax.servlet.ServletException;

import javax.servlet.http.HttpServletRequest;

import javax.servlet.http.HttpServletResponse;

import com.http.servlet.HttpServlet;
```

```

public class ApplicationDetails extends HttpServlet {

    public void doPost(HttpServletRequest request, HttpServletResponse response)

        throws ServletException, IOException {

        response.setContentType("text/html");

        PrintWriter out = response.getWriter();

        String appid=request.getParameter("1");

        String appname=request.getParameter("2");

        String dv=request.getParameter("3");

        String ratings=request.getParameter("4");

        String down=request.getParameter("5");

        String logins=request.getParameter("6");

        String purpose=request.getParameter("7");

        ApplicationPojoap=new ApplicationPojo();

        ap.setAppId(appid);

        ap.setAppName(appname);

        ap.setDailyVisitors(dv);

        ap.setRatings(ratings);

        ap.setNumOfDownloads(down);

        ap.setDailyUserLogins(logins);

        ap.setAppPurpose(purpose);

        boolean result=AppDao.idCheck(appid);

```

```

if(result!=true)

{

int status=AppDao.saveApplicationDetails(ap);

if(status>0)

{

    request.setAttribute("msg", "Application details are added!");

    RequestDispatcher rd=request.getRequestDispatcher("menu.jsp");

    rd.forward(request, response);

}else

{

    request.setAttribute("msg", "Application details are not added!");

    RequestDispatcher rd=request.getRequestDispatcher("menu.jsp");

    rd.forward(request, response);

}

}

else

{

    request.setAttribute("msg", "Application id already exists!");

    RequestDispatcher rd=request.getRequestDispatcher("menu.jsp");

    rd.forward(request, response);

}

```

```
        out.close();

    }

}
```

## **AppDao.java**

```
package logics;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import logics.AppDao;

public class AppDao

{

    public static Connection getConnection(){

        Connection con=null;

        try{

            Class.forName("com.mysql.jdbc.Driver");

            con=DriverManager.getConnection("jdbc:mysql://localhost/adcapsule", "root", "root");
```

```

        //System.out.println("-----CLOUD USER CREATED-----");

    }catch(Exception e){System.out.println(e);}

    return con;

}

    public static int saveApplicationDetails(ApplicationPojo ap)

    {

        int status=0;

        try

        {

            Connection con=AppDao.getConnection();

            String query="insert into

application(appid,appname,daily_visitors,ratings,no_of_downloads,user_logins,app_purpose,price) values (?,?,?,?,?,?,?,?)";

            PreparedStatement ps=con.prepareStatement(query);

            ps.setString(1, ap.getAppId());

            ps.setString(2, ap.getAppName());

            ps.setString(3, ap.getDailyVisitors());

            ps.setString(4, ap.getRatings());

            ps.setString(5, ap.getNumOfDownloads());

            ps.setString(6, ap.getDailyUserLogins());

            ps.setString(7, ap.getAppPurpose());

```

```

        ps.setString(8, null);

        status=ps.executeUpdate();

        con.close();

    }

    catch (Exception e) {

        e.printStackTrace();

    }

    return status;

}

```

```

public static List<ApplicationPojo>getApplicationData()

{

    ArrayList<ApplicationPojo> list=new ArrayList<ApplicationPojo>();

    try

    {

        Connection con=AppDao.getConnection();

        String query="select * from application";

        PreparedStatementps=con.prepareStatement(query);

        ResultSetsrs=ps.executeQuery();

        while(rs.next())

        {

```

```

        ApplicationPojo aps=new ApplicationPojo();

        aps.setAppId(rs.getString("appid"));

        aps.setAppName(rs.getString("appname"));

        aps.setDailyVisitors(rs.getString("daily_visitors"));

        aps.setRatings(rs.getString("ratings"));

        aps.setNumOfDownloads(rs.getString("no_of_downloads"));

        aps.setDailyUserLogins(rs.getString("user_logins"));

        aps.setAppPurpose(rs.getString("app_purpose"));

        aps.setPrice(rs.getString("price"));

        list.add(aps);

    }

}

catch (Exception e) {

    e.printStackTrace();

}

return list;

}

public static boolean idCheck(String appid)

{

    boolean result=false;

```

```

try
{
    Connection con=AppDao.getConnection();

    String query="select * from application where appid=?";

    PreparedStatementst=con.prepareStatement(query);

    st.setString(1, appid);

    ResultSetsrs=st.executeQuery();

    if(rs.next())
    {
        result=true;
    }
}

catch (Exception e) {

    e.printStackTrace();

}

return result;

}

```

```

public static intsaveAgency(AgencyPojo app)

{

    int status=0;

```

```

        try
        {
            Connection con=AppDao.getConnection();

            String query="insert into agency(name,email,password,mobile,status)
values (?,?,,?,?)";

            PreparedStatementps=con.prepareStatement(query);

            ps.setString(1, app.getName());

            ps.setString(2, app.getEmail());

            ps.setString(3, app.getPassword());

            ps.setString(4, app.getMobile());

            ps.setString(5, "off");

            status=ps.executeUpdate();

            con.close();

        }

        catch (Exception e) {

            e.printStackTrace();

        }

        return status;

    }

```

```

public static boolean agencyCheck(String email)
{
    boolean result=false;

    try
    {
        Connection con=AppDao.getConnection();

        String query="select * from agency where email=?";

        PreparedStatement st=con.prepareStatement(query);

        st.setString(1, email);

        ResultSet rs=st.executeQuery();

        if(rs.next())
        {
            result=true;
        }
    }

    catch (Exception e) {
        e.printStackTrace();
    }

    return result;
}

```

```

public static String agencyLogin(String email, String password)
{
    String result="no";

    try
    {
        Connection con=AppDao.getConnection();

        String query="select * from agency where email=? and password=?";

        PreparedStatementst=con.prepareStatement(query);

        st.setString(1, email);

        st.setString(2, password);

        ResultSetsrs=st.executeQuery();

        if(rs.next())
        {
            result=rs.getString("name");
        }
    }

    catch (Exception e) {

        e.printStackTrace();

    }

    return result;
}

```

```

public static int saveAgencyPrice(String appid, String price)
{
    int status=0;

    try
    {
        Connection con=AppDao.getConnection();

        String query="update application set price=? where appid=?";

        PreparedStatement ps=con.prepareStatement(query);

        ps.setString(1, price);

        ps.setString(2, appid);

        status=ps.executeUpdate();

        con.close();

    }

    catch (Exception e) {

        e.printStackTrace();

    }

    return status;

}

public static int onOrOff(String status, String name)
{

```

```

int result=0;

try
{
    Connection con=AppDao.getConnection();

    String query="update agency set status=? where name=?";

    PreparedStatementps=con.prepareStatement(query);

    ps.setString(1, status);

    ps.setString(2, name);

    result=ps.executeUpdate();

    con.close();

}

catch (Exception e) {

    e.printStackTrace();

}

return result;

}

public static String statusCheck()

{

    String res="";

    try {

        Connection con=AppDao.getConnection();

```

```
String query="select * from agency";

PreparedStatementps=con.prepareStatement(query);

ResultSetrs=ps.executeQuery();

if (rs.next()) {

    res=rs.getString("status");

}

else {

    res="error";

}

} catch (Exception e) {

    e.printStackTrace();

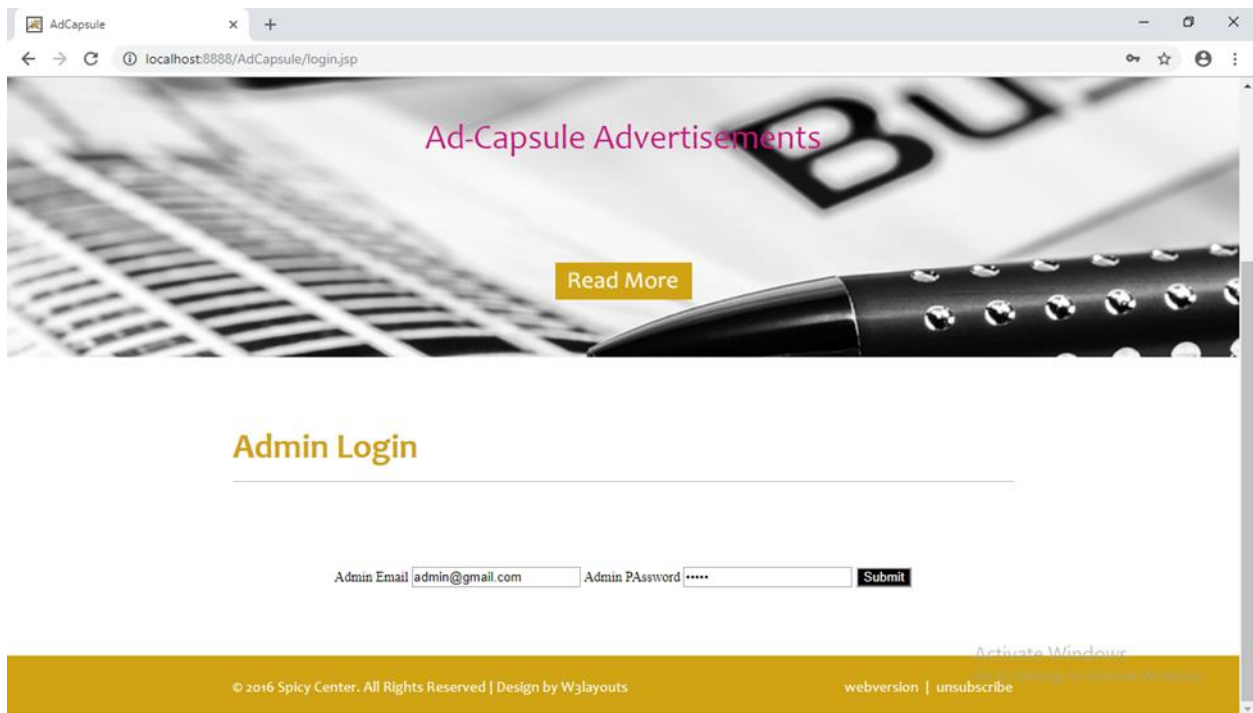
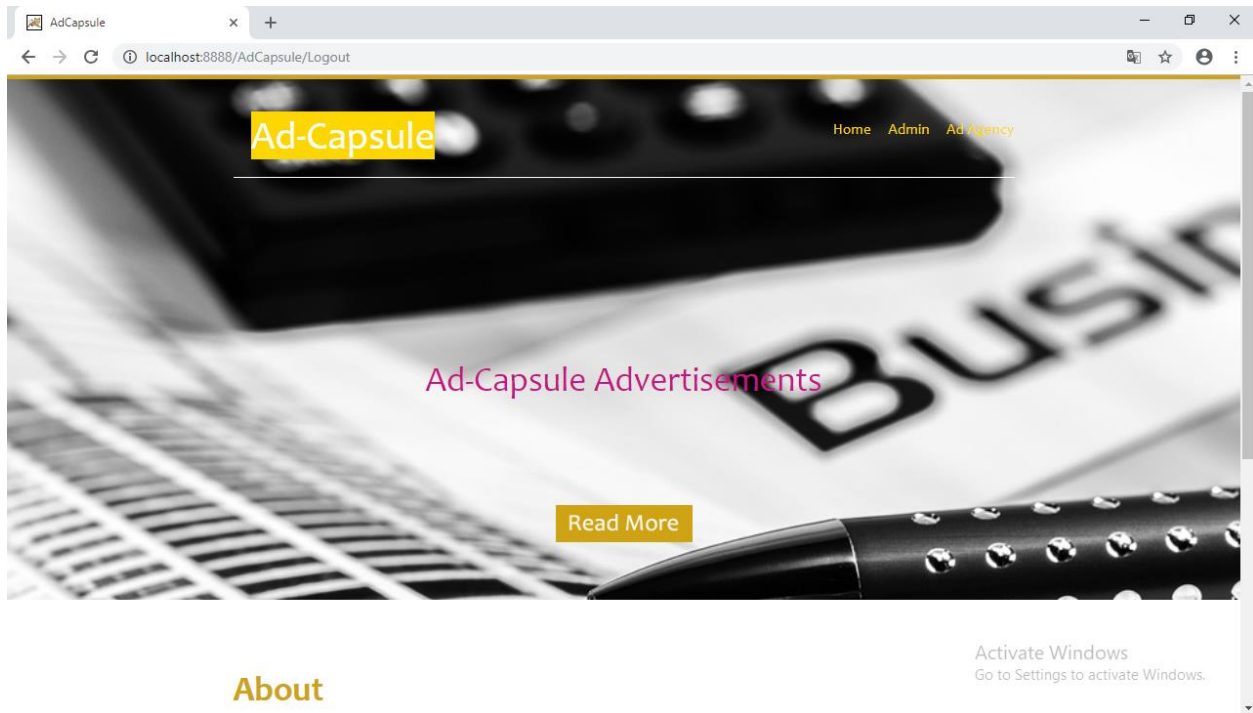
}

return res;

}

}
```

**Screenshots:**



Spicy Center a Email Template | X

localhost:8888/AdCapsule/Admin#

## Add Applications

Application Id

Application Name

Daily Visitors

Ratings

Number of Downloads

Daily User Logins

Application Purpose / Description

-----Choose Here-----

Submit

Logout

Activate Windows  
Go to Settings to activate Windows.

AdCapsule

localhost:8888/AdCapsule/agency.jsp

## Ad-Capsule Advertisements

Read More

## Ad-Agency Login

Email

a@gmail.com

Psssword

Submit

Not yet registered? [Click here...](#)

Activate Windows  
Go to Settings to activate Windows.

AdCapsule

localhost:9999/AdCapsule/areg.jsp

Read More

## Ad-Agency Registration

Name

Email

Pssword

Mobile

Submit

Already registered? [Login here](#)

Activate Windows  
Go to Settings to activate Windows.

maran - Lenovo K33a42 - V... X /manager X Spicy Center a Email Temp... X New Clinic a Medical Cate... X New Tab X

localhost:8888/AdCapsule/AddPrice

## De-Compilation Mode

ON OFF

## Applications List

|                                                                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                                                                 |                                                                                                                                                                                                                                                                                                                |                                                                                                                                                                                                                                                                                              |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <p>Application Id : App-6638</p> <p>Application Name : ZoooCars</p> <p>Daily Visitors : 45000</p> <p>Ratings : 4</p> <p>No. of Downloads : 50000</p> <p>No. of User Logins : 30000</p> <p>App Description : transportation</p> <p>Price : 6000 /-</p> <p>Submit</p> <p>ZoooCars</p> <p>f t v g</p> | <p>Application Id : App-1074</p> <p>Application Name : SpicyHut</p> <p>Daily Visitors : 14000</p> <p>Ratings : 3</p> <p>No. of Downloads : 40000</p> <p>No. of User Logins : 50000</p> <p>App Description : health care</p> <p>Price : 7000 /-</p> <p>Submit</p> <p>SpicyHut</p> <p>f t v g</p> | <p>Application Id : App-3465</p> <p>Application Name : Application123</p> <p>Daily Visitors : 12000</p> <p>Ratings : 4</p> <p>No. of Downloads : 60000</p> <p>No. of User Logins : 15001</p> <p>App Description : transportation</p> <p>Price : 8000 /-</p> <p>Submit</p> <p>Application123</p> <p>f t v g</p> | <p>Application Id : App-7306</p> <p>Application Name : Bizmi</p> <p>Daily Visitors : 25000</p> <p>Ratings : 3</p> <p>No. of Downloads : 10000</p> <p>No. of User Logins : 16000</p> <p>App Description : transportation</p> <p>Price : 9000 /-</p> <p>Submit</p> <p>Bizmi</p> <p>f t v g</p> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

Activate Windows  
Go to Settings to activate Windows.

New Clinic a Medical Category B: x +

localhost:8888/AdProvider/Login

Welcome a ...!!

## ANDROID APPLICATIONS

Category

Health Care

Ratings: 3 to 4

3

4

Number of Downloads: 10000 to 7000000

10000

5000000

Daily User Logins: 6000 to 5000000

10000

5000000

**SUBMIT**

Activate Windows

Go to Settings to activate Windows.

Bank Server x Spicy Center a Email Template | x New Clinic a Medical Category B: x +

localhost:8888/AdProvider/bundle.jsp?appname=SpicyHut.HealthCare.&appid=App-1074,App-1396.&appprice=7000,10000,

Welcome a ...!!

## APPLICATIONS TEMPLATE



### HEALTH CARE

**SPICYHUT,HEALTHCARE**

**APP-1074,APP-1396**

**Total Price**

Rs. ~~47000~~/-

You have 10% discount...

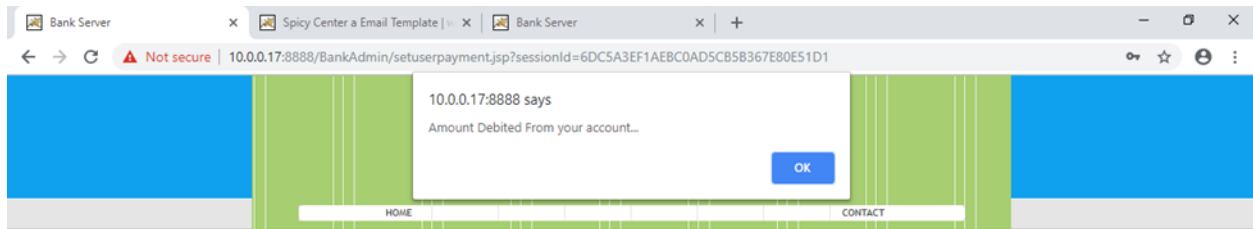
Final Price for Applications is..

Rs. **15300**/-

**Pay Now**

Activate Windows

Go to Settings to activate Windows.



Payment

Account No:

29335501

Atm Pin:

\*\*\*\*

OPass:

\*\*\*\*\*

Submit

©Google zone. All right reserved.

Activate Windows  
Go to Settings to activate Windows.

Bank Server | Spicy Center a Email Template | New Clinic a Medical Category B

Not secure | 10.0.0.17:8888/AdProvider/view.jsp

Welcome a ...!!

APPLICATIONS DATA

| User    | Name    | Type | Number       |
|---------|---------|------|--------------|
| dani-53 | Gopi    | sms  | 9840193127   |
| dani-53 | Gopi    | sms  | 9840193127   |
| dani-53 | RMK-csc | sms  | 994-044-5659 |
| dani-53 | RMK-csc | sms  | 996-577-3901 |
| dani-53 | RMK-csc | sms  | 801-576-1063 |
| dani-53 | RMK-csc | sms  | 917-114-9900 |
| dani-53 | RMK-csc | sms  | 956-668-1656 |
| dani-53 | RMK-csc | sms  | 050-710-4645 |

Activate Windows  
Go to Settings to activate Windows.

Welcome a ...!!

### APPLICATIONS DATA

| S.No | Category    | Bundle   | Application Id | Number of Requests |
|------|-------------|----------|----------------|--------------------|
| 1    | HEALTH CARE | App-1396 | App-1396       | 3                  |

© 2018 New Clinic. All rights reserved | Design by W3layouts

Activate Windows  
Go to Settings to activate Windows.

## REFERENCES

1. “Number of Android applications.” <https://www.appbrain.com/stats/number-of-android-apps>.
2. “Free vs. paid Android apps.” <https://www.appbrain.com/stats/free-and-paid-android-applications>.
3. “Security Alert: New Stealthy Android Spyware –Plankton – Found in Official Android Market.” <https://www.csc.ncsu.edu/faculty/jiang/Plankton/>, 2011.
4. “PontiFlex Ad Library - Remote JavaScript CommandExecution.” <https://labs.mwrinfosecurity.com/blog/pontiflex-ad-library-remote-javascript-Commandexecution/>,2013.

5. “Setting the Record Straight on MopluSDK and the Wormhole Vulnerability.”<http://blog.trendmicro.com/trendlabs-securityintelligence/setting-the-record-straight-on-moplusdk-and-the-wormhole-vulnerability/>, 2015.
6. S. Demetriou, W. Merrill, W. Yang, A. Zhang, and C. A. Gunter, “Free for All! Assessing User Data Exposure to Advertising Libraries on Android,” in Proceedings of the 23rd Annual Symposium on Network and Distributed System Security, NDSS, 2016.
7. M. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi, “Unsafe Exposure Analysis of Mobile In-App Advertisements,” in Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM Wi sec, 2012.
8. S. Son, D. Kim, and V. Shmatikov, “What Mobile Ads Know about Mobile Users,” in Proceedings of the 23<sup>rd</sup> Annual Symposium on Network and Distributed System Security, NDSS, 2016.
9. N. Daswani and M. Stoppelman, “The anatomy of clickbot. a,” in Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, pp. 11–11, USENIX Association, 2007.
10. A. Dewald, T. Holz, and F. C. Freiling, “Adsandbox: Sandboxing javascript to fight malicious websites,” in Proceedings of the 2010 ACM Symposium on Applied Computing, pp. 1859–1864, ACM, 2010.
11. A. Zarras, A. Kapravelos, G. Stringhini, T. Holz, C. Kruegel, and G. Vigna, “The dark alleys of Madison avenue: Understanding malicious advertisements,” in Proceedings of the 2014 Conference on Internet Measurement Conference, pp. 373–380, ACM, 2014.

12. X. Dong, M. Tran, Z. Liang, and X. Jiang, "Adsentry:comprehensive and flexible confinement of javascriptbasedadvertisements," in Proceedings of the 27th AnnualComputer Security Applications Conference, pp. 297–306,ACM, 2011.
13. T. Book and D. S. Wallach, "An empirical study ofmobile ad targeting," CoRR, vol. abs/1502.06577, 2015.
14. S. Han, J. Jung, and D. Wetherall, "A study of thirdpartytracking by mobile apps in the wild."
15. "How to inject JavaScript into an Android web viewfor a more dynamic UX." <http://www.techrepublic.com/article/pro-tip-inject-javascript-into-an-androidweb-view-for-a-more-dynamic-ux/>, 2014.
16. S. Poeplau, Y. Fratanio, A. Bianchi, C. Kruegel, andG. Vigna, "Execute this! analyzing unsafe and maliciousdynamic code loading in android applications.,"in NDSS, vol. 14, pp. 23–26, 2014.
17. J. Seo, D. Kim, D. Cho, T. Kimy, and I. Shinz, "Flexdroid:Enforcing in-app privilege separation in android,"in Proceedings of the 23rd Annual Symposium onNetwork and Distributed System Security, 2016.
18. P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, "Ad-Droid: Privilege Separation for Applications and Advertisersin Android," in Proceedings of the 7th ACMSymposium on Information, Computer and CommunicationsSecurity, ACM ASIACCS, 2012.
19. S. Shekhar, M. Dietz, and D. S.Wallach, "AdSplit: SeparatingSmartphone Advertising from Applications," inPresented as part of the 21st USENIX Security Symposium,USENIX Security, 2012.

20. X. Zhang, A. Ahlawat, and W. Du, “AFrame: Isolating Advertisements from Mobile Applications in Android,” in Proceedings of the 29th Annual Computer Security Applications Conference, ACM ACSAC, 2013.
21. P. Mutchler, Y. Safaei, A. Doupe, and J. Mitchell, “Target Fragmentation in Android Apps,” in IEEE Security and Privacy Workshops, pp. 204–213, 2016.
22. B. Liu, B. Liu, H. Jin, and R. Govindan, “Efficient Privilege De-escalation for Ad Libraries in Mobile Apps,” in Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services, ACM MobiSys, 2015.
23. “Android GOT Hook.” <https://shunix.com/androidgot-hook/>, 2016.
24. M. Backes, S. Gerling, C. Hammer, M. Maffei, and P. von Styp-Rekowsky, “AppGuard: Enforcing User Requirements on Android Apps,” in International Conference on Tools and Algorithms for the Construction and Analysis of Systems, 2013.