

# **ANDROID EVENT ANALYTICS**

Project report submitted in partial fulfilment of the requirement for the  
degree of Bachelor of Technology

in

**Computer Science and Engineering/Information Technology**

By

**AKSHIT GUPTA**

(161304)

Under the Supervision of

Dr. Geetanjali Rathee

to



Department of Computer Science & Engineering and Information  
Technology

**Jaypee University of Information Technology Waknaghat, Solan-  
173234, Himachal Pradesh**

## DECLARATION BY THE SCHOLAR

I hereby declare that this project report titled '**Android Event Analytics**' in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, at **Jaypee University of Information Technology, Wagnaghat, Solan** is an authentic record of my work carried out under supervision of **Dr. Geetanjali Rathee** . I have not submitted this work elsewhere for any other degree or diploma. I am fully responsible for the contents of my B.Tech. Industrial Project.



Akshit Gupta

161304

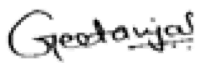
Department of Computer Science

Jaypee University of Information Technology

May, 2020

## SUPERVISOR'S CERTIFICATE

This is to certify that the work reported in BTech. Industrial Training entitled “**Android Event Analytics**” submitted by **Akshit Gupta** at **Jaypee University of Information Technology, Wagnaghat, India** is a bonafide record of his original work carried under my supervision. This work has not been submitted elsewhere for any other degree or diploma.



Dr. Geetanjali Rathee

Assistant Professor (SG)

Department of Computer Science

Jaypee University of Information Technology



Mr. Vihaan Verma

Android Team Lead

Testbook, Navi Mumbai, India

## **TABLE OF CONTENT**

<b>INNER FIRST PAGE</b>	<b>1</b>
<b>DECLARATION BY THE SCHOLAR</b>	<b>2</b>
<b>SUPERVISOR’S CERTIFICATE</b>	<b>3</b>
<b>ACKNOWLEDGEMENT</b>	<b>7</b>
<b>ABSTRACT</b>	<b>8</b>
<b>LIST OF ACRONYMS AND ABBREVIATIONS</b>	<b>9</b>
<b>LIST OF FIGURES</b>	<b>10</b>
<b>CHAPTER 1</b>	
<b>INTRODUCTION</b>	<b>11</b>
1.1. Introduction to Android?	11
1.2. Why to use Android?	12
1.3. Java vs Kotlin for Android	12
<b>CHAPTER 2</b>	
<b>THEORETICAL LEARNINGS AND RESEARCH</b>	<b>18</b>
2.1. Activity In Android	18

2.2. Activity Life Cycle	19
2.3. Fragments In Android	21
2.4. Fragments Life Cycle	22
2.5. Basic Android Architectures	25
2.5.1 MVC	26
2.5.2 MVP	28
2.5.3 MVVM	30

## **CHAPTER-3**

### **TOOLS AND TECHNOLOGIES** **33**

3.1. WEBENGAGE	33
3.2. BITBUCKET	34
3.3. ASANA	35
3.4. JAVA	35
3.5. KOTLIN	36

## **CHAPTER 4**

<b>PROJECT IMPLEMENTATION</b>	<b>37</b>
4.1. Framework Events	37
4.2. Custom Events	38
4.3. Event Triggering	41
<b>CHAPTER 5</b>	
<b>CONCLUSIONS</b>	<b>52</b>
5.1. RISK ANALYSIS	52
5.2. FUTURE SCOPE	53
5.3. CONCLUSION	55
<b>REFERENCES</b>	<b>57</b>

## **ACKNOWLEDGEMENT**

This internship opportunity at Testbook is a great chance for learning and professional development. I would like to express my deepest gratitude and special thanks to the founder and CEO of the company Mr. Ashutosh Kumar who inspite of being extraordinarily busy with his duties, took time out to hear, guide and keep me on the correct path of learning and developing.

I express my deepest thanks to Mr. Ayush Varshney, Managing Partner and Chief Technical Officer for taking part in useful decision & giving necessary advices and guidance and arranged all facilities in the office.

It is my radiant sentiment to place on record my best regards, deepest sense of gratitude to my mentors, Mr. Vihaan Verma, (Android Team Lead), Mr. Abhishek Bharti (Software Developer) for their careful and precious guidance which were extremely valuable for my study both theoretically and practically.

I would like to acknowledge guidance of my institute mentor, Dr. Geetanjali Rathee who constantly guide me during my internship and suggest me to improve on every aspect.

## **ABSTRACT**

Event refers to all the actions performed by users while interacting with your mobile apps, website and campaigns. This enables you to gain in-depth insights into user interactions across your app, website, and channels. You can also leverage this data to segment users, personalize messages and configure campaign targeting. Hence, Event triggering and reviewing helps companies to interact with users indirectly and track their activity.

Based on events triggered, the company gets to know about how exactly is their product performing in the market as compared to various other products in the existing market.

Based on the data analysis of these events and other stats. Companies gets to know where it should improve it's product in order to capture more user interest and interaction.



## **LIST OF ACRONYMS / ABBREVIATIONS**

MVP	Model View Presenter
MVC	Model View Controller
MVVM	Model View ViewModel
CSS	Cascading Style Sheet
WE	WebEngage
IDE	Integrated Development Environment
DOM	Document Object Model
HTTP	HyperText Transfer Protocol
NPE	Null Pointer Exception
JSON	JavaScript Object Notation
MIME	Multi-purpose Internet Mail extension
MVCC	Multi Version Concurrency Control
OOB	Out Of the Box
POC	Proof Of Concept
POJO	Plain Old Java Object
REST	Representational State Transfer
SEO	Search Engine Optimization

UI	User Interface
URL	Uniform resource Locator
VCS	Version Control System

## LIST OF FIGURES

*Fig.1.1 Android*

*Fig.1.2 Why Android*

*Fig.3.2 Activity Life Cycle*

*Fig.3.3 Android Fragments*

*Fig.3.4 Fragment LifeCycle*

*Fig.1.3.1 Kotlin Sample Code*

*Fig.1.3.2 Kotlin code snap (concise)*

*Fig.1.3.2 Kotlin code snap (safe)*

*Fig.1.3.2 Kotlin code snap (interoperable)*

*Fig.3.5.3 FrameWork Events*

*Fig.3.5.3 Custom Events*

# CHAPTER -1

## INTRODUCTION

### 1.1 Introduction to Android?



*Fig.1.1 What is Android?*

It is a free, open-source operating system for tabs, mobiles etc. It is basically based on linux. It was developed by google.

- World's most popular mobile platform is **Android**.

## 1.2 Why to use Android ?



*Fig.1.2 Why Android*

## 1.3 Java vs Kotlin for Android Development

```

KOTLIN

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        ...
        fab.setOnClickListener { view ->
            Snackbar.make(view, "Hello $name", Snackbar.LENGTH_LONG).show()
        }
    }
}

```

Nullable and NonNull types help reduce NullPointerExceptions  
 Use lambdas for concise event handling code  
 Use template expressions in strings to avoid concatenation  
 Semicolons are optional

Fig.1.3.1 Kotlin Sample Code

*Kotlin over-powers Java in development process..WHY??*

We all know that throughout the world, at this very moment, Java is the highly utilized and used languages in the android app development.

But, that does not mean that it's the only best language out of the options that we have.

It is old , it's prone to errors, it's syntax is complicated at times.Kotin is a better option to do app development on other hand.

Engineers have begun to overcome issues related to java 8 but they still are struck up in the previous versions of java... i.e java 6 and java 7. The present scenario is such that we can't expect it to improve soon.

So, this is where Kotlin becomes handy and that's why it has a slight edge over Java. It is easy to adapt to, has a simple syntax and has much more to offer as discussed later.

The JDK engineers are also picking up well with kotlin.

So the reason right ahead is an ideal opportunity for beginning in utilizing the cutting edge, advanced, practical language for your Android improvement ventures.

Nougat version of android additionally was a strong endeavor for help to the highlights of java 8 utilizing, yet a large portion of those are just usable on the off chance that you have min version say 24 which is poor to have. taking into account the amount of moderate our android has become .

### ❖ **Java is prone to errors..**

Perhaps the greatest defect in Java is the manner in which it handles "invalid," prompting the feared Null-Pointer-Exception (NPE)

Today, the NPE is one of the most widely recognized explanations behind accidents in Android applications. Truth be told, it's practically difficult to have an application underway throughout solitary NPE .

What's more, "nullability" is a considerably more serious issue for Android. Invalid is a proficient and basic method of speaking to the nonattendance for worth, the Android-system utilizes this thing to its own system and its own api's. It is however bad fact that Java is making it more challenging for the Enginners to deal with itself.

Another issue is the way in which software engineers frequently overlook the manner in which Java actualizes non-static inward classes and mysterious internal classes, which consistently hold a certain reference to the external class. In this

manner, engineers wind to make the applications defenseless to overflow of memory/space.

## ❖ Kotlin doesn't have any Null-Pointer-Exception

When using kotlin, you need not to stress over this Null-Pointer-Exception on the grounds that invalid security is prepared in kotlin sort framework. It's very ideal getting Null-Pointer-Exception during the time of order as opposed to smashing applications at runtime.

How accomplishes this work by and by? As a matter of course, all factors are non-invalid. On the off chance that you need one variable of "null" type , you use “?” for that. Assume that you are having one string variable also that you are attempting for allocating an incentive :

```
val msg: string msg = “How are you?”
```

That was excellent in working..

What about this now:

```
val msg: string msg = Null
```

In this case, The compiler neglects aggregate, rather gives error , "Invalid can't be an estimation of a invalid sort string." It happened in light of the fact that all factors not-valid

```
val msg: string? msg = Null
```

Presently its ok, however u despite everything should b cautious that you are getting this var. due to the sort framework set up, it will be hard for NPEs to happen in your application at runtime.

## ❖ Kotlin is brief/concise

Kotlin is very concise and due to this the boiler\_plate code is also very less.

```
/*  
  Create a POJO with getters, setters, `equals()`, `hashCode()`, `toString()` and `copy()` in a single line:  
*/  
  
data class Customer(val name: String, val email: String, val company: String)  
  
// Or filter a list using a lambda expression:  
  
val positiveNumbers = list.filter { it > 0 }  
  
// Want a singleton? Create an object:  
  
object ThisIsASingleton {  
    val companyName: String = "JetBrains"  
}
```

*Fig.1.3.2 Kotlin code snap (concise)*

## ❖ Kotlin is Safer than Java



```

/*
  Get rid of those pesky NullPointerExceptions, you know, The Billion Dollar Mistake
*/

var output: String
output = null // Compilation error

// Kotlin protects you from mistakenly operating on nullable types

val name: String? = null // Nullable type
println(name.length()) // Compilation error

// And if you check a type is right, the compiler will auto-cast it for you

fun calculateTotal(obj: Any) {
    if (obj is Invoice)
        obj.calculateTotal()
}

```

*Fig.1.3.3 Kotlin code snap (safe)*

## ❖ It can be used with java

Kotlin is interoperable with java

```

/*
  Use any existing library on the JVM, as there's 100% compatibility, including SAM support.
*/

import io.reactivex.Flowable
import io.reactivex.schedulers.Schedulers

Flowable
    .fromCallable {
        Thread.sleep(1000) // imitate expensive computation
        "Done"
    }
    .subscribeOn(Schedulers.io())
    .observeOn(Schedulers.single())
    .subscribe(::println, Throwable::printStackTrace)

// Target either the JVM or JavaScript. Write code in Kotlin and decide where you want to deploy to

import kotlin.browser.window

fun onLoad() {
    window.document.body!!.innerHTML += "<br/>Hello, Kotlin!"
}

```

*Fig.1.3.4 Kotlin code snap (interoperable)*

## **CHAPTER 2**

### **THEORETICAL LEARNINGS AND RESEARCH**

#### **2.1 Activity in Android**

An action speaks to a solitary screen by taking the help from UI simple window and edge from Java. Action of android may be defined as the subclass ContextThemeWrapper class.

In the event in which the working is done with the help of C, C++ or Java languages by you, then it is more likely than not seen that your program begins from fundamental work. Fundamentally the same as way, Android framework starts the process inside an Activity beginning by an approach on Create callback technique. The grouping for callback strategies which states an action also a

succession for callback techniques which tear apart a movement appeared underneath.

## **2.2 Activities Life Cycle**

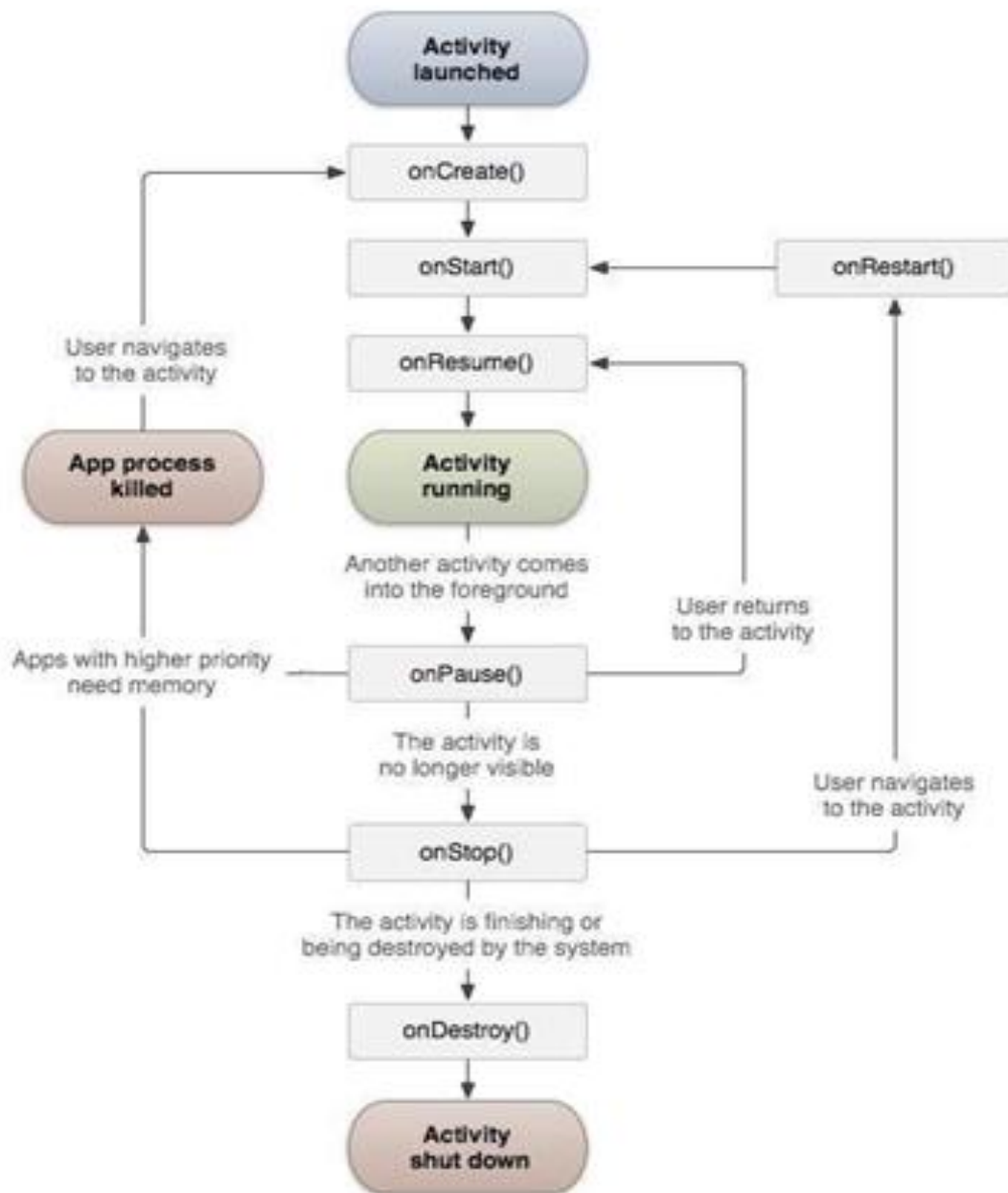


Fig.3.2 Activity Life Cycle

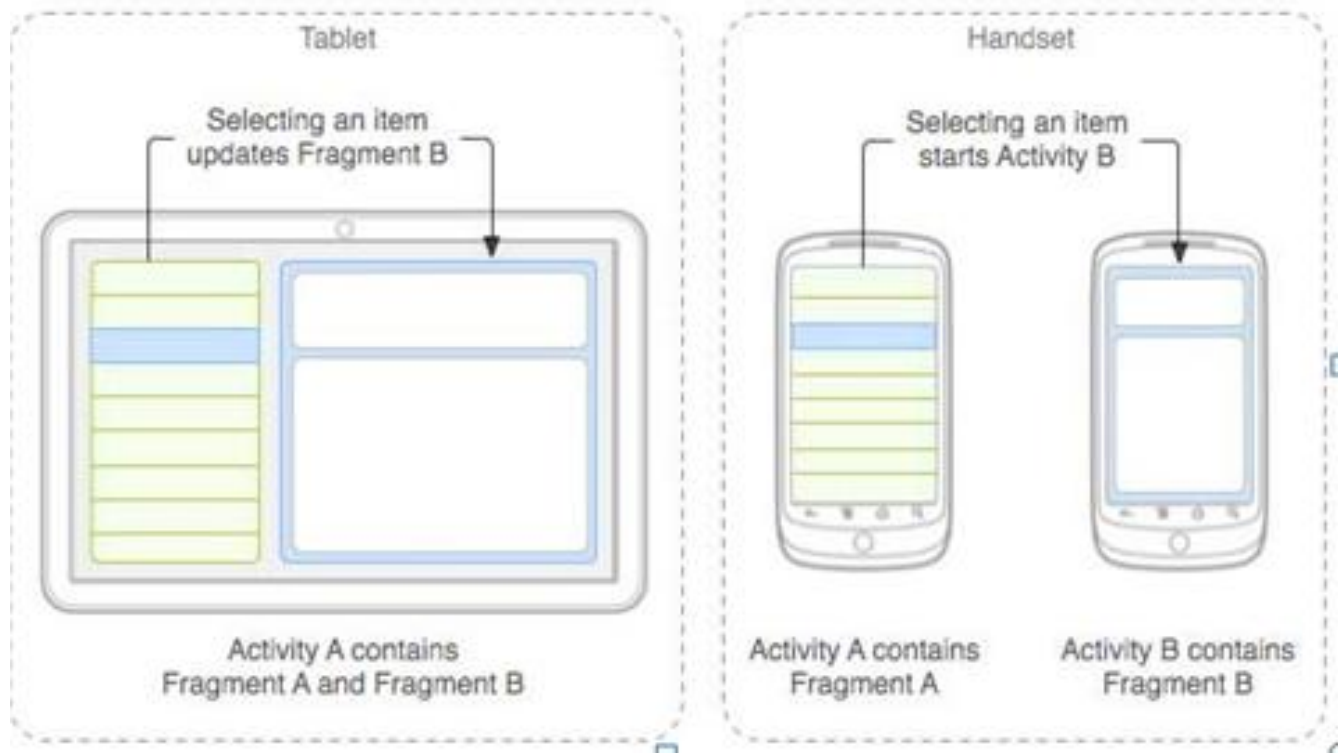
S.No	Callbacks & Descriptions
1	<p><b>on Create()</b></p> <p>It's the very first callback and gets called out as soon as the creation of activity takes place the 1st time.</p>
2	<p><b>on Start()</b></p> <p>This callback gets called as soon as an activity becomes visible for a person.</p>
3	<p><b>on Resume()</b></p> <p>This callback gets called as soon as the person starts interaction in application.</p>
4	<p><b>on Pause()</b></p> <p>A activity which is paused does not gain any input from user, thus can't put codes in action. When called out, currently occupied activity gets holdup also the previous one gets resumed.</p>
5	<p><b>on Stop()</b></p> <p>It is called as soon as an activity, which was visible, disappears.</p>
6	<p><b>on Destroy()</b></p> <p>This one gets calledout as soon as an activity gets destroyed by system.</p>
7	<p><b>on Restart()</b></p> <p>It is the callback which gets calledout as soon as the activity starts again by pausing itself.</p>

## 2.3 Fragments in Android

Fragments in android are "reusable independent parts of a UI". Fragments are associated with Android activity, which is utilized for making dynamic and adaptable interface for client. Android fragment have certain associated life-cycle yet it generally be inserted with an action so the pieces life cycle is legitimately influenced by the parent action's life cycle and sections gets it's own information type. Inside an app of android one could utilize various pieces in a solitary movement to make multiple-pane user interface and furthermore could ready into utilize a solitary section in numerous activity.

The manager of a fragment is answerable to include/evacuate or supplant parts during app's execution phsase from anyplace inside its active .when include a section in an action, it recides inside view group inside action's chain of command and piece characterizes itself for design buy proclaiming the section through the action's format record as a component or through app's coding part by appending it to a current view group.

Shown below an ordinary scenario as to how two different fragments characterized through sections could be consolidated in a particular action for tablet device-type, however isolated when device-type is hand-set.



*Fig.3.3 Android Fragments*

The app could insert 2 pieces inside the activity-A, when the device type is tablet. Notwithstanding, but for device-type – handset, there is no sufficient space to adjust the two pieces, thus the associated-activity An incorporates just its section through rundown of paragraphs, and when the client chooses a paragraph, thus begins activity-B, which consolidates to subsequent part into peruse this paragraph.

## **2.4 Life Cycle of Fragment**

The parts of Android system do have a life cycle of their own, fundamentally the same as an android movement. This segment briefs various phases

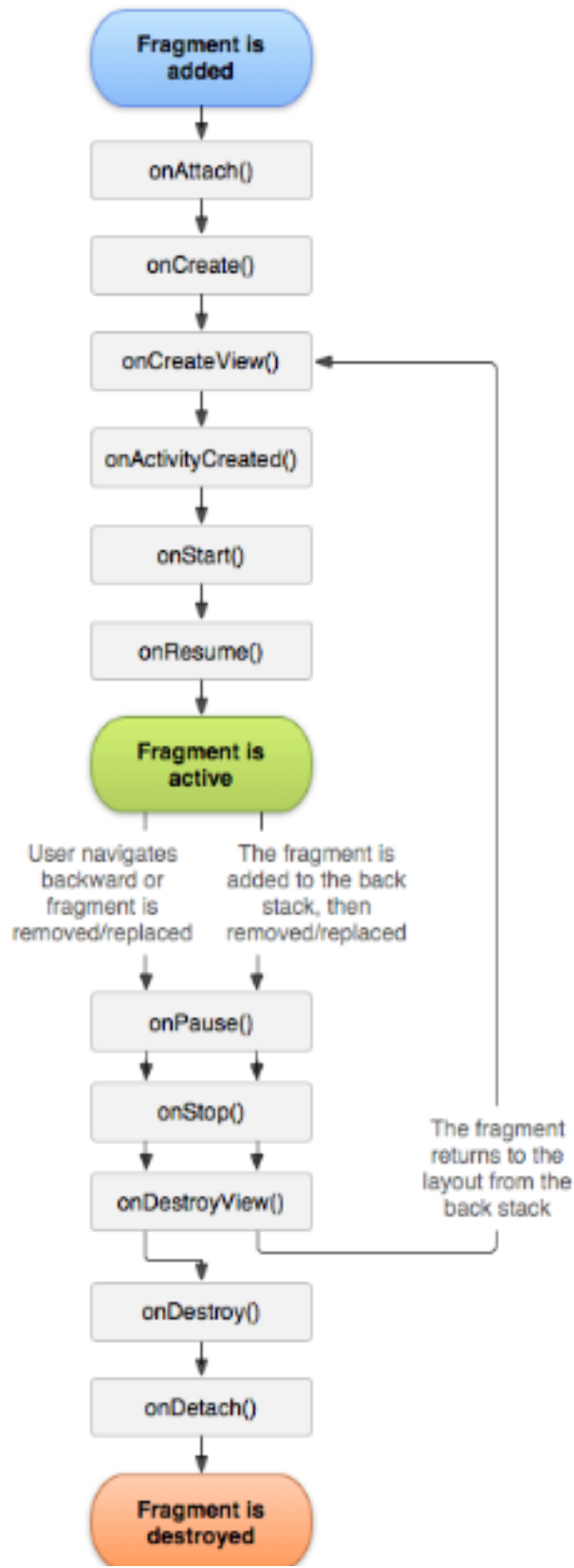


Fig.3.4 Fragment LifeCycle



Let's get to know each of these methods in little detail –

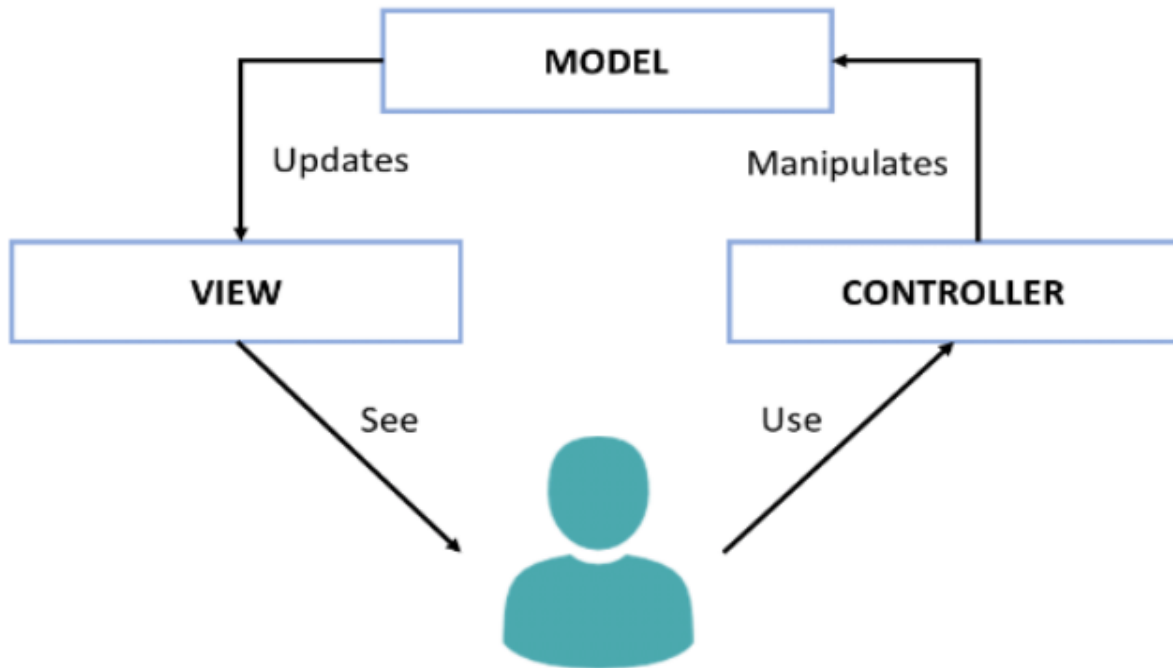
- **onAttach():** This part is often related through an action case. A piece as well as actions isn't completely instantiated. Normally this strategy gives reference for movement that utilizes a section of additional intro work.
- **onCreate():** This technique is called out for making a piece. You ought to introduce a fundamental segment's piece which one may need to hold if the part tends to be delayed or halted, it continues.
- **onCreateView():** This callback is called out if it is the only opportunity of the section from drawing its Ui. For drawing a Ui, one should restore the View segment through the technique which is the base of one's section's format. One may be returning as invalid when the Ui is not provided.
- **onActivityCreated():** This is called out after onCreateView strategy as soon as the host action is taken in consideration. Movement as well as section occasion are made just like view progression for action. Now, view may get to with `findViewById` strategy. Here, one can launch objects that requires a Context object
- **onStart():** This technique is called out after the section is noticeable.
- **onResume():** Here fragment gets dynamic.

- **onPause():** This technique is considered, the main sign referring the client is ready to leave the piece. Generally this one is where one ought to submit the progressions which ought to persevered past and present client meeting.
- **onStop():** Fragment is halted , calling out onStop()
- **onDestroyView():** View of fragment demolishes after calling this certain technique
- **onDestroy():** This one is called to as for last tidy up the part's way yet not destined to call out from Android stage.

## 2.5 Basic Android Architectures (MVC versus MVP versus MVVM)

These are two of the most generally embraced other options, yet engineers are regularly separated regarding which one better fits with Android. There have been various blog entries over the previous year or so emphatically pushing for one over the other, yet regularly these transform into contentions of feeling over target rules. As opposed to quarrel about which approach is better, this article takes a gander at the worth and potential issues with each of the three methodologies so you can settle on an educated choice for yourself.

## 2.5.1 MVC (Model View Controller)



*Fig.3.5.1 MVC*

### **Model**

The model combines the Data plus State plus Business rational for Tic-Tac-Toe app. It is the minds of application in a manner of speaking. It isn't attached with view and controller, along these lines, it can be reused in numerous unique circumstances.

## **View**

The view works as the representation. It has an obligation i.e. to render the User Interface also for conveying the controller as soon as the client collaborates with app. Within the MVC design, views commonly truly "imbecilic" so, having zero information on basic model also, nor comprehension for state to do as soon as the client connects through clicking catch, composing the worth, and rest. Basic realization is: the less one realize, more inexactly they are to model which also implies the more change prone and adjusting they become.

## **Controller**

Controller in technical terms is the Glue which integrates this app. Or the ace controller that knows what ever occurs. The point where the View tell that a client has a catch, controller concludes to collaborate both the model as needs be

## **Assessment**

MVC works superbly of isolating the model and view. Unquestionably the model can be effectively tried on the grounds that it's not attached to anything and the view has not a lot to test at a unit testing level. The Controller has a couple of issues in any case.

## **Controller Concerns**

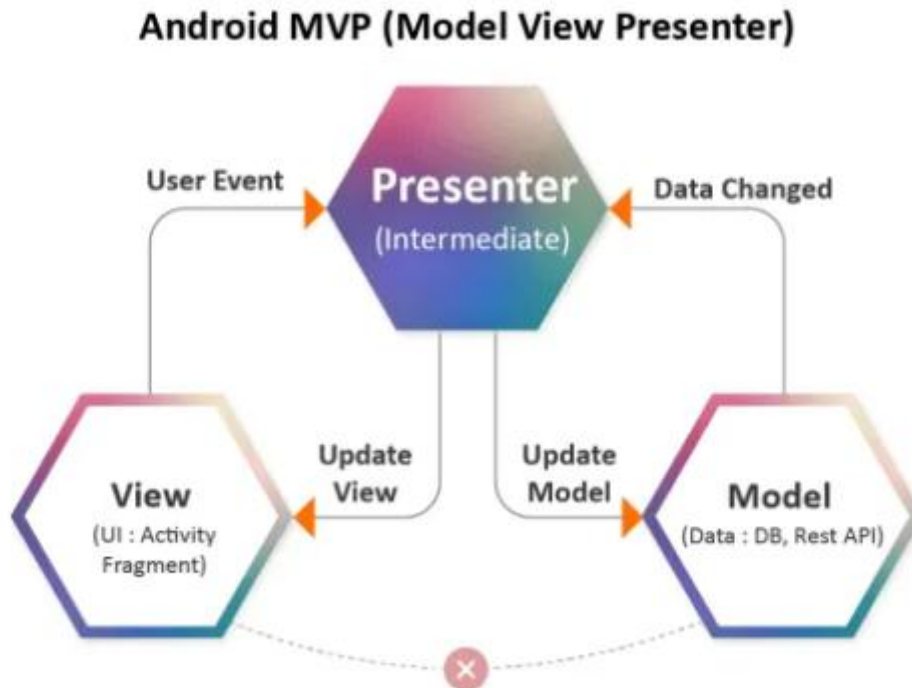
- **Testability** - The controller is tied so firmly to the Android APIs that it is hard to unit test.
- **Modularity and Flexibility** - The controllers are firmly coupled to the perspectives. It should be an expansion of the view. In the event that we change the view, we need to return and change the controller.

- **Maintenance** - Over time, especially in applications with iron deficient models, increasingly more code begins getting moved into the controllers, making them enlarged and weak.

How might we address this? MVP to the salvage!

## 2.5.2 MVP (Model View Presenter)

MVP splits the controller up so the normal view/action coupling can happen without binds it to the remainder of the "controller" obligations. More on this beneath, yet how about we start again with a typical meaning of obligations when contrasted with MVC.



*Fig.3.5.2 MVP*

## **Models**

No change

## **View**

Change which is main over here is Activity currently is viewed as a feature. We fight no further for the regular incline for these to connected at the hip. High progress is for having the Activity execute.

## **Controller**

It is the MVC controller with the exception of that it isn't at all attached to with View. It tends for the worrying of testability just like measured adaptability that we are concerned about in MVC. Indeed, idealists may contend in MVP the moderator ought for not having any concern to do with any Android APIs and also the code.

## **Assessment**

This is a lot of more clean. We can undoubtedly unit test the moderator rationale since it's not attached to any Android explicit perspectives and APIs and that likewise permits us to work with some other view as long as the view actualizes the TicTacToeView interface.

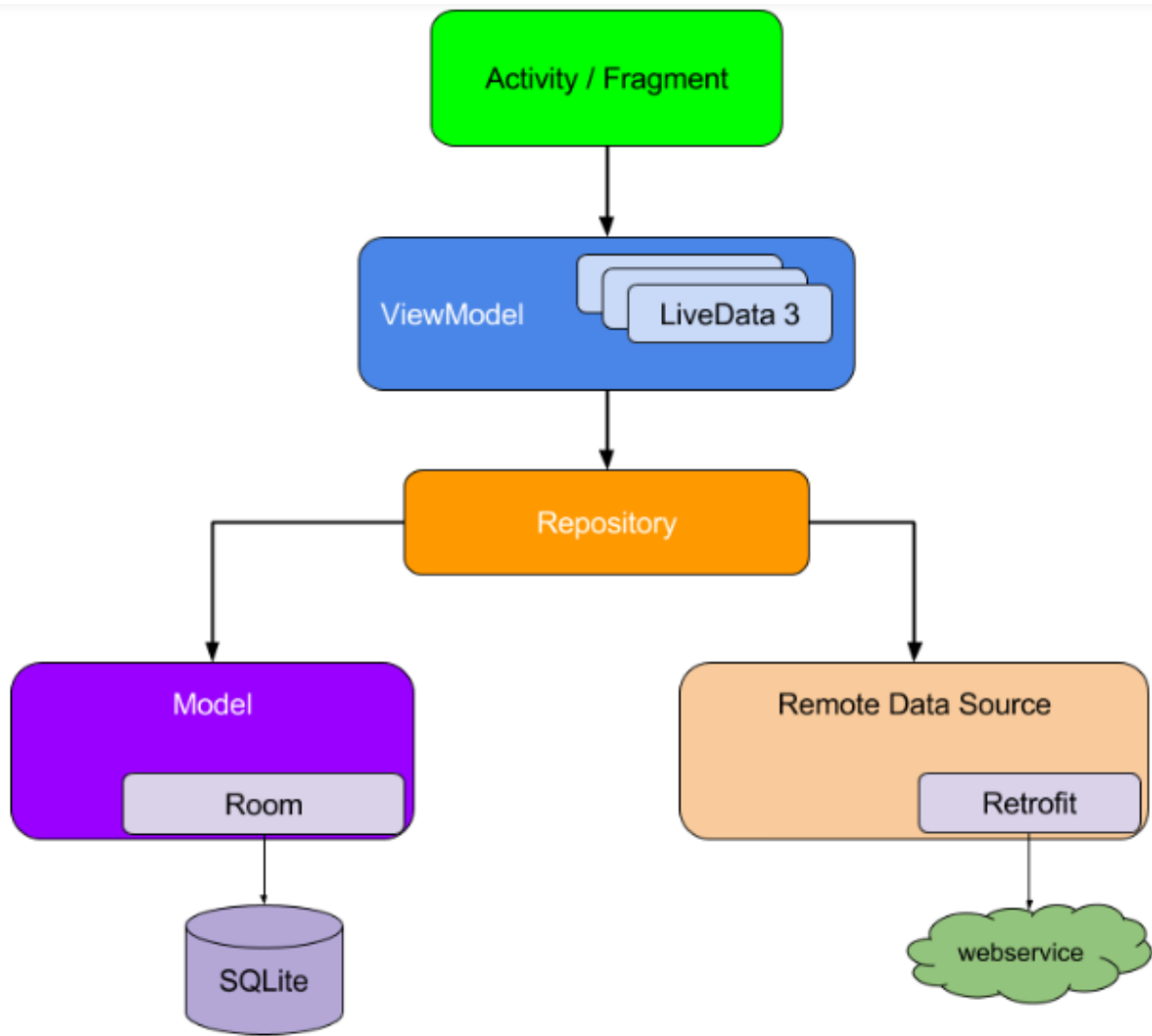
## **Moderator Concerns**

- **Maintenance** - Presenters, much the same as Controllers, are inclined to gathering extra business rationale, sprinkled in, after some time. Sooner or later, designers regularly end up with enormous inconvenient moderators that are hard to break separated.

Obviously, the cautious engineer can assist with forestalling this, by determinedly guarding against this allurements as the application changes after some time. Notwithstanding, MVVM can help address this by doing less to begin.

### **2.5.3 MVVM (Model View ViewModel)**

This with Data Binding have the advantages for simpler test as well as particularity, also additionally lessening measure for paste codes which is needed for writing to associate view plus model.



*Fig.3.5.3 MVVM*

**How about we look at the pieces of MVVM.**

**Model**

Same as MVC/No change



## **View**

The view ties to perceptible factors and activities uncovered by the viewModel in an adaptable manner. More on that in minute.

## **ViewModel**

The ViewModel is answerable for wrapping the model and getting ready noticeable information required by the view. It additionally gives snares to the view to pass occasions to the model. The ViewModel isn't attached to the view be that as it may.

## **Assessment**

Unit testing is much simpler now, since you truly have no reliance on the view. When testing, you just need to confirm that the recognizable factors are set fittingly when the model changes. There is no compelling reason to deride out the view for testing as there was with the MVP design.

## **MVVM Concerns**

- **Maintenance** - Since perspectives can tie to the two factors and articulations, incidental introduction rationale can sneak in after some time, viably adding code to our XML. To maintain a strategic distance from this, consistently get values legitimately from the ViewModel instead of endeavor to register or infer them in the perspectives restricting articulation. Along these lines the calculation can be unit tried fittingly.

## **CHAPTER -3**

### **TOOLS AND TECHNOLOGIES USED**

On 7<sup>th</sup> may, 19, Java was supplanted as favored language used in Android application. Java is as yet upheld, as is C++.

#### **3.1 WebEngage**

WebEngage is a full-stack showcasing mechanization suite that drives development for shopper organizations by empowering them to draw in clients by means of various channels like, Push, In-application, SMS, website location Notifications, Web Push, Email, Facebook-Instagram, and WhatsApp. The stage's logical abilities additionally make it workable for organizations to get a 360° perspective on their clients, item, crusades, and analyze the effect of each channel of commitment.

At WebEngage, we are fixated on helping organizations, similar to yours, fuel economical development by means of information driven commitment over your clients' lifecycle. By helping you influence constant bits of knowledge into your clients and their associations with your application, site, and crusades, our excursion originator empowers you to flawlessly draw in clients at each stage.

This is the reason client related information (all things considered alluded to as User Attributes) and social information (alluded to as Events and Event Attributes) establish the framework for all your client commitment exercises.

## 3.2 BitBucket

Bitbucket is an online version control vault encouraging organization controlled through Atlassian, to source code and for the improvement expands, used :

- 1) Mercurial (from dispatch until 1 July, 20)
- 2) Git (since Oct, 2011)
- 3) Amendment control systems.

It gives the benefit of both, i.e. marketable strategies also the free records. Bit bucket gives these records the endless counts of private stores (that may have maximum 5 customers by virtue for records) for Sep, 2010. It also joins with alternative Atlassian programming such as Hip Chat, Confluence, Bamboo and Jira.

Bitbucket resembles Git Hub, that basically is used by Git. It have large displayed its organizations to capable architects with private selective programming codes, most probably since it was taken in consideration by the Atlassian in the year 2010. On Feb, 2017, Bit bucket pronounced that it's been shown up at six million planners also one million gatherings in the platform. On Apr, 2019 Atlassian made the announcement that Bit bucket showed up at ten million enlisted customers also in excess of twenty eight million repository.

- It had 3 association models i.e.

Cloud, Bit bucket Server, and Data Center.

### **3.3 Asana**

- Asana is the web and versatile application that intended for helping groups sort out, to track, also to deal with the work one have. Forrester, Inc.reports which Asana disentangles based on group work with management.
- Profitability of workers at Facebook.d The item propelled financially in April 2012. In December 2018 the organization was esteemed at \$1.5B.

### **3.4 Java**

After some time some improved variants of Java did discharged. The present variant of Java will be JAVA 1.8 which is otherwise called JAVA 8.

Its characterized through detail , a compiler, center libraries and a runtime JVM. During its runtime ,it permits programming engineers to compose program different dialects than java that despite everything runs on the JVM. its stage is typically connected with the JVM and liberaries based in java.

Java language has following properties:

- Java is independent of platform.
- Java is object-oriented.

- Java is strongly-typed language.
- In Java, we have automatic management of memory.

Syntactically, java is very much same when compared with C++

### **3.5 Kotlin**

Its a broadly useful, easily reliable, free and very composed "even minded" coding lang. at first intended to JVM and to android as to consolidates arranged and useful coding highlights. It's centered around inter-operable, security, lucidity. Adaptations of Kotlin focusing on JS and local codepart for various processing parts are underway also.

Kotlin started at jet-brains, its organization before intellij-idea, in 2011, and is opensource throughout from 2013. Kotlin's group as of now have in excess of 100 full time individuals from jetBrains, also Kotlin venture on gitHub have excess of 200 patrons. jet-brains utilizes Kotlin in huge numbers of its items excluding the leader intellij-idea.

# CHAPTER 4

## PROJECT WORK AND RESEARCH

### Events

Every single social datum focuses are called Events in your dashboard. Furthermore, every Event can additionally be comprehended with regards to its Attributes which incorporates subtleties like time, area, gadget subtleties, value, amount, etc.

This empowers you to pick up top to bottom experiences into client connections over your application, site, and channels. You can likewise use this information to section clients, customize messages and arrange crusade focusing on.

The term Event alludes to all the activities performed by clients while associating with your versatile applications, site and crusades.

For instance, if a client needs to tap on an item to see its subtleties, at that point it is prudent to follow this activity as the Event, Product Viewed as it presents to them a bit nearer to making a buy.

Events are characterized into 2 classes in WebEngage:

**Framework Events:** Pre-characterized by WebEngage, consequently followed for stages post intergation.

**Custom Events:** Defined by you for every stage and followed through the individual SDKs.

We should walk you through this.

## **4.1 Framework Events**

We have pre-characterized a few conventional activities that clients can perform while cooperating with your application, site and battles. These activities are alluded to as System Event and are consequently followed for your foundation once you coordinate them with your WebEngage account.

Here's a rundown of some of the System Events that are naturally followed for every one of your clients post coordination:

Name on Dashboard	Name in Backend	Description
App Installed	app_installed	When the app is installed.
App Upgraded	app_upgraded	When the app is upgraded.
App Crashed	app_crashed	When the app crashes.
App Uninstalled	app_uninstalled	When app is uninstalled.
User Login	user_logged_in	Whenever you call the login function on user login, signup etc.
User Logout	user_logged_out	Whenever you call the logout function.
Session Started	user_session_started	Whenever a new session is started by your user. <i>(Here's what a session means in WebEngage)</i>
Campaign Conversion	goal_accomplish	When a user performs the <i>Conversion Event</i> defined for a campaign/journey.
GCM/APNs Registered	gcm_registered	When a device is registered successfully to receive <i>Push Notifications</i> .
APNs Registration Failed	apns_registration_failed	When an iOS device fails to get registered for receiving <i>Push Notifications</i> .
Push (Mobile) Sent	gcm_notification_response	When a <i>Push Notification</i> is sent to FCM for delivery by WebEngage.

*Fig.4.1 FrameWork Events*

How Does WebEngage Define Session?

For Websites: Session break span = 30 minutes



Meetings time is the measure of time for which clients are 'dynamic' on your site. Here, 'movement' could mean any of the accompanying:

Performing Events like inquiry, peruse, seeing items, classifications, etc.

Invigorating or reloading a similar site page.

Opening another site page.

In the event that a client is distinguished to be 'inert' for 30 minutes, at that point their on-going meeting is finished. Here's 'dormancy' could mean any of the accompanying:

The client is on your site however has not played out any of the activities referenced previously.

The client has explored to another tab.

The client is exchanging between your site and another tab yet has not played out any of previously mentioned activities.

For Mobile Apps (iOS and Android): Session break span = 15 seconds

Meeting time is the measure of time for which the application is open in the gadget's forefront. In the event that the client pushes your application to the foundation and doesn't take it back to the closer view inside 15 seconds, at that point their on-going meeting will end. In the event that they carry it to the closer view following 15 seconds, at that point we'll record it as another meeting.

## **4.2 Custom Events**

Custom Events are conduct information focuses that you would custom be able to characterize and follow for your clients over your applications and site. These empower you to comprehend your clients better and convey logically customized encounters progressively.

Contingent upon your business, these occasions could be in any way similar to:

- Item Page Viewed
- Course Details Viewed
- Subscription Purchased
- Video Played | Video Paused | Video Ended
- Game Started | Game Ended
- Checkout Started | Checkout Completed
- Survey Submitted, etc.

## Event Attributes

Event Attributes are the respective details that are attached to each Event. These attributes convey in what context did the user perform it.

For instance, the properties of a Custom Event, Order Confirmed could be Order Value, Delivery Date, Number of Items, Primary Product Category, Delivery Address, Order ID, Event Time, Device Type, etc.

Event Attributes have been grouped into 2 classifications in WebEngage:

**System Attributes:** Tracked for all System Events and Custom Events as a matter of course.

**Custom Attributes:** Tracked distinctly for the Custom Events to which they're connected.

How about we walk you through this:

## Framework Attributes

These are conventional subtleties that have been predefined by us and are naturally followed for all the System Events and your Custom Events. These information focuses can't be altered by you.

Here's a table showing some of these:

Name on Dashboard	Type	Description
Event Time	DateTime	Time at which the event occurred (in ISO format).
Country	String	Country from where user performed the event.
City	String	City from where user performed the event.
Browser Name	String	Name of the browser on which user performed the event.
OS Name	String	Operating System of the device through which user performed the event.
Device Manufacturer	String	Name of the device manufacturer through which user performed the event.
Device Model	String	Model of the device through which user performed the event.
Carrier	String	Name of the cellular network provider of the device through which user performed the event.
App Version	String	Version of your app on which the user performed the event.
App ID	String	ID of your app on which the user performed the event.
Platform	String	The platform on which user performed the event (Android/ iOS, Website).
Page URL	String	The page URL on which user performed the event. <i>(only for website)</i>
Screen Name	String	The screen name on which user performed the event. <i>(only for apps)</i>
Channel	String	The channel ( <i>Direct, Organic Search, Social</i> ) that resulted in the occurrence of the event.

*Fig.4.2 Custom Events*

## **Custom Attributes**

Custom Attributes are subtleties that can be connected to every Custom Event characterized by you. You can decide to connect a most extreme if 25 custom qualities of a solitary information type to every Custom Event to all the more likely see every client's foundation communications and logically draw in them.

For instance, if an ed-tech stage tracks course buys as the Custom Event, Course-Enrolled then it's Custom Event Attributes would be:

Course Name

Course ID

Course Value

Course Duration

Number of Chapters

Course Category, etc.

## **Tracking Events**

WebEngage begins following a few occasions when you incorporate the SDK. These are called System Events and track some nonexclusive client cooperations with your application and battles. Here's a rundown of the System Events that are naturally followed by us.

You can make Custom Events to follow some other client cooperations that are crucial for your business. Every Custom Event can additionally be characterized by Event Attributes like value, amount, classification, etc. Such granular information empowers you to draw in clients through exceptionally logical and customized crusades through all the channels of commitment.

## Tracking Custom Events

All Events related APIs are a piece of WebEngage Android SDK's Analytics object. Here's the way you can get an example of the WebEngage Analytics object:

Java

```
// import WebEngage 'Analytics'  
import com.webengage.sdk.android.Analytics;  
  
// Get an instance of 'Analytics' object  
Analytics weAnalytics = WebEngage.get().analytics();
```

## Rules

Here are a couple of things to remember:

WebEngage sends all occasions information intermittently in clusters to limit arrange utilization and expand portable battery life for your clients. (Instructions to Set Event Priority)

Custom Event and Custom Event Attribute names are case delicate and must be under 50 characters in length. String property estimations must be under 1000 characters in length.

eventName or eventAttributeName must not begin with we\_. Names beginning with we\_ are held only for inward use at WebEngage. Subsequently, to keep away from information tainting for your record, such information will be disregarded whenever utilized for your Custom Events.

Custom Event Attributes can be of these information types: String, all subclasses of Number, Boolean, Date, List, Map.

You can make a limit of 25 Event Attributes of every datum type for a Custom Event.

In the event that an Event Attribute esteem is List or Map, at that point it can't be utilized to make portions. It must be utilized to customize crusades.

The first datapoint matched up to WebEngage characterizes the information type for that occasion property. In this manner, information types must be predictable with the worth that you need to store against the property. In the event that the information type is changed sometime in the not too distant future, at that point Custom Event Attribute information will quit streaming to your WebEngage dashboard.

After WebEngage has been effectively instated, you can follow a Custom Event in the accompanying way:

Java

```
weAnalytics.track("Product - Page Viewed");
```

## Tracking Event Attributes

You can utilize a Map to determine a Custom Event Attribute and append it to the Custom Event for which you'd prefer to follow it, in the accompanying way:

Java

```
Map<String, Object> addToCartAttributes = new HashMap<>();
addToCartAttributes.put("Product ID", 1337);
addToCartAttributes.put("Price", 39.80);
addToCartAttributes.put("Quantity", 1);
addToCartAttributes.put("Product", "Givenchy Pour Homme Cologne");
addToCartAttributes.put("Category", "Fragrance");
addToCartAttributes.put("Currency", "USD");
addToCartAttributes.put("Discounted", true);

weAnalytics.track("Added to Cart", addToCartAttributes);

Map<String, Object> orderPlacedAttributes = new HashMap<>();
orderPlacedAttributes.put("Amount", 808.48);
orderPlacedAttributes.put("Product 1 SKU Code", "UHUH799");
orderPlacedAttributes.put("Product 1 Name", "Armani Jeans");
orderPlacedAttributes.put("Product 1 Price", 300.49);
orderPlacedAttributes.put("Product 1 Size", "L");
orderPlacedAttributes.put("Product 2 SKU Code", "FBHG746");
orderPlacedAttributes.put("Product 2 Name", "Hugo Boss Jacket");
orderPlacedAttributes.put("Product 2 Price", 507.99);
orderPlacedAttributes.put("Product 2 Size", "L");
String dateStr = "2017-10-06T09:27:37Z";
SimpleDateFormat format = new SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss'Z'");
```



```
try {
    Date date = format.parse(dateStr);
    orderPlacedAttributes.put("Delivery Date", date);
} catch (ParseException e) {
    e.printStackTrace();
}
orderPlacedAttributes.put("Delivery City", "San Francisco");
orderPlacedAttributes.put("Delivery ZIP", "94121");
orderPlacedAttributes.put("Coupon Applied", "BOG017");
```

## Tracking Complex Event Attributes

WebEngage permits you to pass complex occasion characteristics as List and Map information types. You will have the option to utilize this information to customize crusades, as demonstrated as follows. Notwithstanding, you won't have the option to utilize complex characteristics for making sections.

WebEngage Test

Original + Add new variant

✓ Title\* Congrats! Your order has been shipped.

✓ Message\*

Type your push message here

Tip : Click to personalize your message.

Next

← Event

← Order Placed

Products

Delivery Address

Coupons Applied

release

device

country

app\_version

model

Order shipped intimation Edit push notification > Rules > Conversion tracking > Activate

**So, Here is how you pass *Complex Event Attributes* to a *Custom Event* into your account on WebEngage:**

```

Map<String, Object> product1 = new HashMap<>();
product1.put("SKU Code", "UHUH799");
product1.put("Product Name", "Armani Jeans");
product1.put("Price", 300.49);

Map<String, Object> detailsProduct1 = new HashMap<>();
detailsProduct1.put("Size", "L");
product1.put("Details", detailsProduct1);

Map<String, Object> product2 = new HashMap<>();
product2.put("SKU Code", "FBHG746");
product2.put("Product Name", "Hugo Boss Jacket");
product2.put("Price", 507.99);

Map<String, Object> detailsProduct2 = new HashMap<>();
detailsProduct2.put("Size", "L");
product2.put("Details", detailsProduct2);

Map<String, Object> deliveryAddress = new HashMap<>();
deliveryAddress.put("City", "San Francisco");
deliveryAddress.put("ZIP", "94121");

Map<String, Object> orderPlacedAttributes = new HashMap<>();
orderPlacedAttributes.put("Products", Arrays.asList(product1, product2));
orderPlacedAttributes.put("Delivery Address", deliveryAddress);
orderPlacedAttributes.put("Coupons Applied", Arrays.asList("BOG017"));

weAnalytics.track("Order Placed", orderPlacedAttributes);

```

## Event Priority

Events amass after some time, and WebEngage sends occasions in groups intermittently to limit organize use and augment battery life of the gadget. Occasion need permits you to abrogate this conduct of the SDK.

Since high need occasions are accounted for more regularly than bunched occasions, it will cause your application to deplete your client's gadget battery quicker. Thus, we strongly suggest that you client this component sparingly.

setHighReportingPriority works just when revealing technique is set to BUFFER. In the event that announcing procedure is set to FORCE\_SYNC, every one of your occasions will be accounted for at high need of course.

You can set a specific occasion's need to 'high' in the accompanying way:

Java

```
weAnalytics.track("Checkout Started", new Analytics.Options().setHighReportingPriority(true));
```

The WebEngage SDK reports such occasions promptly (gave arrange network is accessible), and doesn't cluster them with occasions whose need isn't set to high.

You can likewise set high need for occasions that have ascribes joined to them, in the accompanying way:

Java

```
Map<String, Object> checkoutStartedAttributes = new HashMap<>();
checkoutStartedAttributes.put("Cart ID", 35651);
checkoutStartedAttributes.put("Cart Size", 3);
checkoutStartedAttributes.put("Cart Value", 445.59);
checkoutStartedAttributes.put("Cart Category", "Women Dresses");

weAnalytics.track("Added to Cart", checkoutStartedAttributes, new Analytics.Options().setHighReportingPriority(true));
```

# CHAPTER 5

## CONCLUSIONS

### 5.1. RISK ANALYSIS

There are many things that needs to be kept in mind in order to code in AEM as it requires special expertise in many fields. There are many risks associated with the implementation.

For AEM itself, in the implemented environment there are various OOB (Out Of the Box) components available for the use of the authors that are yet a little buggy (including minor defects and not covering entire use cases) and not fully ready to go even without some minor tweaks for the real project use. Since, this software (earlier made and acquired by Day CQ) was acquired not very late by Adobe, therefore a lot of Adobe integrated features and solutions are not fully ready-to-use or baked yet. This makes the integrations cumbersome and introduce various defects that are difficult to catch and fix. This AEM tool is not much suitable for small business or start-ups making their websites due to the cost incurred in purchase of the software and high level of developer expertise. Debugger in AEM does not function every time as required because of some untracked errors. Also, the IntelliVault tool which is used to pull or push the code from or to CRXD Lite to IntelliJ/ Eclipse does not work every time as per the desired and the expected behaviour due to unknown reasons.

There are certain functionalities provided by Touch UI which are not fully functional and Javascript needs to be written to make them compatible with the

custom made components. Every new version of AEM covers the small bugs that are incurred in the present version supporting Touch UI.

While working with ClientLibs in AEM (for executing the JavaScript code) there might be cases when JavaScript code breaks on the console due to some properties which might not be supported on the browsers. There might be scenarios where cross-site scripting may creep in and break the code. So, these risks are associated with the JavaScript implementation which needs to be taken care of. The issues can be browser specific, run modes specific, environment specific, server specific etc.

There are many risks associated while working on git as many conflicts arise while taking pull before committing a code. There conflicts might sometime lead to loss of code on a machine when stashing is not done for the current working branch. Hence, working with git can introduce risks on the work that we are trying to send to the remote repository from our local repository.

## **5.2. FUTURE SCOPE**

The present project has been released and the present work is going around modifying the functionalities to make them more generic and flexible. These will be further releases of the project which will be logically divided into sprints and will modify the existing components. The project now will include working on improvement stories to modify the website and to make the components more generic and authorable in order to give more freedom to the authors to display relevant content on the website. For implementing this, we need to modify the dialog structure or the existing components or the look and feel of the components as per the updated designs provided by the client. This may also include changing

the desired behaviour of the existing components by doing some modifications in the backend logic implemented in the specific components. Also, many new pages shall be authored requiring some special components that are template-specific to extend the pages and content of the website.

The work of the developers will include making some new components to fulfil the demands of the client. Also, the work will focus around modifying the existing components and including enhanced functionalities via Javascript coding. There will be use of servlets and ajax calls for the functionality that is specific to some requirement or is triggered based on some conditions. The existing components will be modified as a part of improvements in order to make them flexible to be dragged and dropped anywhere on the page. Many features will be automated as part of the plan and some dynamic features will also be included for the future. There will be integration of Adobe Analytics with the existing framework and automation of task by making different custom workflows that will reduce the coding intervention.

This project shall be extended further in order to work on the login functionality of the present website. It will include customization of the entire login process that will be confidential and handled with security layers. The future of this project shall also include starting of email functionality for the stakeholders dependent on the platform of Office 365. For implementing this functionality we need to write some custom OSGI configurations which shall be implemented in this email workflow that will be send triggered on some specific activities.

The project can include use of AEM Forms to convert the present form functionality developed with the help of HTML and jQuery and JS validations into

AEM Forms. This will make the entire website based on Adobe products (Forms, Analytics etc.) and integrations will be highly efficient and refined.

There are two more projects lined up for this year from different clients which will be based on AEM development and might include dynamic content inclusion which will be handled by the back end logic applied to the custom components.

The scope of learning will extend to learning of various features specific to AEM:

- Dispatcher configurations
- Advanced workflow implementations
- Links and Custom AEM Reports
- Different APIs used in backend logic
- SEO functionalities
- Sonar checking and integration
- Advanced UI and frontend technologies

### **5.3. CONCLUSION**

The internship duration was enriched with learning experience, where a lot of new technologies were learnt theoretically and implemented in the live project. The internship duration was divided into training phase where we learnt the concepts and got the opportunity to research on some domains. The second part of the phase



included opportunity to work on a live project to get practical industrial knowledge and experience. The tasks were reviewed and finally sent for the final project merge. This training was fruitful in both industrial and research perspective as there was findings and hands-on on technologies that are new to the industry and have a very wide scope in the near future. Hence, the training was an amalgamation of learning, experience,

Research and practical knowledge.

## REFERENCES

1. Closser, S. (2013). Adobe Experience Manager Quick-Reference Guide: Android App Development [formerly CQ]. Adobe Press.
2. Hall, R., Pauls, K., McCulloch, S., & Savage, D. (2011). OSGi in action: Creating modular applications in Java. Manning Publications Co.
3. <http://www.testbook.com/>
4. <https://webengage.com/>
5. [https://docs.adobe.com/docs/en/aem/6-2/develop/ref/granite-ui/api/jcr\\_root/libs/granite/ui/index.html](https://docs.adobe.com/docs/en/aem/6-2/develop/ref/granite-ui/api/jcr_root/libs/granite/ui/index.html)
6. <https://docs.adobe.com/docs/en/htl/overview.html>
7. <https://sling.apache.org/documentation/bundles/models.html>
8. <https://docs.adobe.com/docs/en/cq/5-6-1/exploring/architecture-overview.html>
9. Loeliger, J., & McCullough, M. (2012). Version Control with Git: Powerful tools and techniques for collaborative software development. " O'Reilly Media, Inc." .

10. Patil, S. (2006). What is Java content repository. O'Reilly Onjava. com.
11. Phil, I. C. E. (2013). Advancing Platform Technologies in Online Learning. In Conference proceedings of» eLearning and Software for Education (eLSE) (No. 01, pp. 616-621). Universitatea Nationala de Aparare Carol I.
12. Rellermeyer, J. S., Alonso, G., & Roscoe, T. (2007, November). R-OSGi: distributed applications through software modularization. In Proceedings of the ACM/IFIP/USENIX 2007 International Conference on Middleware (pp. 1-20). Springer-Verlag New York, Inc..
13. Schwaber, K., & Beedle, M. (2002). Agile software development with Scrum (Vol. 1). Upper Saddle River: Prentice Hall.
14. Siriwardena, P. (2014). Mastering Apache Maven 3. Packt Publishing Ltd.
15. Siriwardena, P. (2015). Maven Essentials. Packt Publishing Ltd.
16. Somasundaram, R. (2013). Git: Version control for everyone. Packt Publishing Ltd.
17. Sutherland, J., Schwaber, K., Scrum, C. C. O., & Sutherl, C. J. (2007). The scrum papers: Nuts, bolts, and origins of an agile process.
18. Vishnu, K. (2017). Web Content Management Infrastructure Migration. International Journal of Research in Computer Engineering & Electronics, 6(2).
19. XUE, S. J., & CHENG, M. (2009). Research on Java Content Repository and Application in CMS [J]. Computer Technology and Development, 1, 067.

# Plag Check Results

AGR9

by Akshit Guptar9

I

---

**Submission date:** 28-May-2020 06:35PM (UTC+0530)

**Submission ID:** 1333410765

**File name:** AkshitR9.docx(989.24K)

**Word count:** 4527

**Character count:** 23622

## AGR9

### ORIGINALITY REPORT

<b>1</b> %	<b>0</b> %	<b>0</b> %	<b>1</b> %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

### PRIMARY SOURCES

<b>1</b>	Submitted to Postgraduate Schools - <u>Limkokwing University of Creative Technology</u> Student Paper	<1 %
<b>2</b>	Submitted to University of <u>Northumbria at Newcastle</u> Student Paper	<1 %
<b>3</b>	Submitted to Crown Institute of Business and Technology Student Paper	<1 %
<b>4</b>	Submitted to Queen Mary and Westfield College Student Paper	<1 %

Exclude quotes  Off  
Exclude bibliography  On

Exclude matches  On

**PLAGIARISM VERIFICATION REPORT**

Date: .....

Type of Document (Tick):  **PhD Thesis**  **M.Tech Dissertation/ Report**  **B.Tech Project Report**  **Paper**

Name: Akshit Gupta Department: CSE Enrolment No 161304

Contact No. \_\_\_\_\_ E-mail. <[akshitgupta881@gmail.com](mailto:akshitgupta881@gmail.com)> \_\_\_\_\_

Name of the Supervisor: Dr. Geetanjali Rathee

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): ANDROID EVENT ANALYTICS

**UNDERTAKING**

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

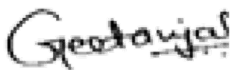
**Complete Thesis/Report Pages Detail:**

- Total No. of Pages =60
- Total No. of Preliminary pages =1
- Total No. of pages accommodate bibliography/references =1

  
(Signature of Student)

**FOR DEPARTMENT USE**

We have checked the thesis/report as per norms and found **Similarity Index** at 1.... (%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

  
(Signature of Guide/Supervisor)

Signature of HOD

**FOR LRC USE**

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	<ul style="list-style-type: none"> <li>• All Preliminary Pages</li> <li>• Bibliography/Images/Quotes</li> <li>• 14 Words String</li> </ul>		Word Counts	
<b>Report Generated on</b>			Character Counts	
		<b>Submission ID</b>	Total Pages Scanned	
			File Size	

Checked by  
Name & Signature

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at [plagcheck\\_juit@gmail.com](mailto:plagcheck_juit@gmail.com)