**WHAT’S HAPPENING – AN ANDROID APPLICATION AS EVENT ORGANISER AND MANAGER**

1AMIT DHANWANI, 2SANKET NAWLE, 3SHRUTI SADALGE, 4ABHIJEET GITE

1Assistant professor, Vidyalankar Institute of Technology, Wadala, Mumbai, Mumbai university

2BE Student, Vidylankar Institute of Technology, Wadala, Mumbai, Mumbai university

3BE Student, Vidylankar Institute of Technology, Wadala, Mumbai, Mumbai university

4BE Student, Vidylankar Institute of Technology, Wadala, Mumbai, Mumbai university

Email id: 1amit.dhanwani@vit.edu.in, [4abhijeetgite1704@gmail.com](mailto:4abhijeetgite1704@gmail.com), [2sanketnawle@gmail.com](mailto:2sanketnawle@gmail.com), [3shrutisadalge@yahoo.com](mailto:3shrutisadalge@yahoo.com),

**Abstract—** There has been an enthralling increase in the usage of smart phones since their invention. These smart phones are going to be the next generation open operating systems. It is very easy for the consumers to search for applications which will suffice their needs. The openness of these new environments leads to new applications and markets and enables greater integration. This paper presents a concept of an application which will help users to be aware of all the important things happening around them. The users will be notified about the events which are significant to them in their day to day life. It will consists of functions like alarm and news (customized). By using this application, the users can connect to other people by sending them requests and can also block people they don’t want to connect with. The users will have the feature invite other people to their events and reply to their invitations as well. Users can also enjoy a feature called area-wise notification to get to know about the events which will be happening around them by tapping their smart phones on NFC tags. This application will also facilitate the users to keep track of all their events using reminders and connect those reminders to google calendar in order to increase the availability and reliability.

**Keywords** – Alarm, News, Invitations, Area-wise notifications, NFC

1. **INTRODUCTION**

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers” [1]. Android is also the world’s most widely used smartphone platform having the highest market share. The platform is made of the open source operating system android, devices that run it and an extensive repository of applications developed for the OS. The open source nature of the OS allows anybody to freely modify and distribute it. The OS has code written in Assembly, Java, C++ and C. The applications are primarily written in Java; however C++ can also be used. Google Play store, the application repository for Android, has more than 700,000 applications. The devices that run Android can have the hardware range from a bare minimum 32MB RAM, 200 MHz processor to those with a state-of-the-art Quad core processor and 2GB of RAM. The primary hardware platform is the ARM architecture. Android’s kernel was based on the Linux 2.6 kernel. The latest version of Android called Jelly Bean has the Linux kernel 3.0.3. Android has added new drivers to the kernel, middleware, libraries and an extensive set of APIs that aid application development on top of the kernel. The relevance of smartphones in today’s world, ubiquity of the Android operating system and intriguing customization of Linux to run on mobile devices make a compelling case to study Android. Through our project we tried to explore the Android software stack and its unique features, primarily the Inter Process Communication (IPC) mechanism – Binder. Binder is a key and complex component that is essential to the functioning of the Android operating system. This report is a documentation of our learning. Understanding the data flow, inner workings of the Binder framework and the components of Android were of particular interest to us. We begin this report with a bit of history to set context, follow that with the Android software stack and then the crucial part about Binder.

**II. LITERATURE REVIEW**

**[1] Android OS Architecture**

Android OS consists of four main layers kernel, libraries, applications framework and applications.

**Kernel Layer**

 The whole Android OS is built on top of the Linux 2.6 Kernel with some further   architectural changes made by Google.  It is this Linux that interacts with the hardware and contains all the essential hardware drivers. Linux was chosen since it has a proven track record in desktop systems and in many cases doesn’t require drivers to be rewritten. Linux provides such things as virtual memory, networking, drivers, and power management.

**Native libraries layer**

The native libraries layer provides Android with the capabilities for its core features. Android is shipped with SGL which acts as the primary 2D graphics renderer. Its counterpart is OpenGL ES which provides 3D graphics support. Android comes packed with SQLite which takes care of most data storage. The WebKit web rendering engine is also shipped with Android and has been tailored to render web pages for smaller screen sizes. Dalvik virtual machine which is a part of this layer. The Dalvik virtual machine is a bytecode interpreter which is highly optimized for executing on the mobile platform. The bytecodes are converted Java binaries that are very quick and efficient to run on smaller processors. The core libraries are written in Java and provide much of the core classes which would normally be available in a Java virtual machine.

**Applications framework Layer**

The applications framework provides all of the major APIs that the applications will use including things like sharing data, accessing the telephony system, and receiving notifications. This layer and the layer above it are written completely in Java.

**Applications layer**

This is the top layer in Android architecture. This layer contains software written by the Android team as well as any third-party software that is installed on the device. Even the most core features such as the phone and the contacts application reside in this layer. Any third party developer can access this layer, as an effect of which any events of core android apps can be handled by third party applications(like phone ringing).

Android provides the services expected in a modern operating system such as virtual memory, multiprogramming, and threads all on a mobile platform. Many of Android’s services are a result of including the Linux kernel. As well Android team has added the telephony stack.

**[2] NFC**

Near Field Communication (NFC) is a set of short-range wireless technologies, typically requiring a distance of 4cm or less to initiate a connection. NFC allows you to share small payloads of data between an NFC tag and an Android-powered device, or between two Android-powered devices.

Tags can range in complexity. Simple tags offer just read and write semantics, sometimes with one-time-programmable areas to make the card read-only. More complex tags offer math operations, and have cryptographic hardware to authenticate access to a sector. The most sophisticated tags contain operating environments, allowing complex interactions with code executing on the tag. The data stored in the tag can also be written in a variety of formats, but many of the Android framework APIs are based around a NFC Forum standard called NDEF (NFC Data Exchange Format).

Android-powered devices with NFC simultaneously support three main modes of operation:

1. **Reader/writer mode**, allowing the NFC device to read and/or write passive NFC tags and stickers.
2. **P2P mode**, allowing the NFC device to exchange data with other NFC peers; this operation mode is used by Android Beam.
3. **Card emulation mode**, allowing the NFC device itself to act as an NFC card. The emulated NFC card can then be accessed by an external NFC reader, such as an NFC point-of-sale terminal. Android-powered devices are usually looking for NFC tags when the screen is unlocked, unless NFC is disabled in the device's Settings menu. When an Android-powered device discovers an NFC tag, the desired behavior is to have the most appropriate activity handle the intent without asking the user what application to use. Because devices scan NFC tags at a very short range, it is likely that making users manually select an activity would force them to move the device away from the tag and break the connection.

To help you with this goal, Android provides a special tag dispatch system that analyzes scanned NFC tags, parses them, and tries to locate applications that are interested in the scanned data. It does this by:

1. Parsing the NFC tag and figuring out the MIME type or a URI that identifies the data payload in the tag.
2. Encapsulating the MIME type or URI and the payload into an intent. These first two steps are described in [How NFC tags are mapped to MIME types and URIs](https://developer.android.com/guide/topics/connectivity/nfc/nfc.html#ndef).
3. Starts an activity based on the intent. This is described in [How NFC Tags are Dispatched to Applications](https://developer.android.com/guide/topics/connectivity/nfc/nfc.html#dispatching).

**[3] NOTIFICATION**

A notification is a message you can display to the user outside of your application's normal UI. When you tell the system to issue a notification, it first appears as an icon in the **notification area**. To see the details of the notification, the user opens the **notification drawer**. Both the notification area and the notification drawer are system-controlled areas that the user can view at any time. Android **Toast** class provides a handy way to show users alerts but problem is that these alerts are not persistent which means alert flashes on the screen for a few seconds and then disappears.

When you need to issue a notification multiple times for the same type of event, you should avoid making a completely new notification. Instead, you should consider updating a previous notification, either by changing some of its values or by adding to it, or both.

For example, Gmail notifies the user that new emails have arrived by increasing its count of unread messages and by adding a summary of each email to the notification. This is called "stacking" the notification; it's described in more detail in the [Notifications](http://developer.android.com/design/patterns/notifications.html) Design guide.

If you wish, you can set the priority of a notification. The priority acts as a hint to the device UI about how the notification should be displayed.

Although they're optional, you should add at least one action to your notification. An action allows users to go directly from the notification to an [Activity](http://developer.android.com/reference/android/app/Activity.html) in your application, where they can look at one or more events or do further work.

A notification can provide multiple actions. You should always define the action that's triggered when the user clicks the notification; usually this action opens an [Activity](http://developer.android.com/reference/android/app/Activity.html) in your application. You can also add buttons to the notification that perform additional actions such as snoozing an alarm or responding immediately to a text message

**[4] TCP Connection**

The Transmission Control Protocol provides a communication service at an intermediate level between an application program and the Internet Protocol. It provides host-to-host connectivity at the Transport Layer of the Internet model. An application does not need to know the particular mechanisms for sending data via a link to another host, such as the required packet fragmentation on the transmission medium. At the transport layer, the protocol handles all handshaking.

At the lower levels of the protocol stack, due to network congestion, traffic load balancing, or other unpredictable network behavior, IP packets may be lost, duplicated, or delivered out of order. TCP detects these problems, requests retransmission of lost data, rearranges out-of-order data, and even helps minimize network congestion to reduce the occurrence of the other problems. If the data still remains undelivered, its source is notified of this failure. Once the TCP receiver has reassembled the sequence of [octets](https://en.wikipedia.org/wiki/Octet_(computing)) originally transmitted, it passes them to the receiving application. Thus, TCP abstracts the application's communication from the underlying networking details.

TCP is a reliable stream delivery service that guarantees that all bytes received will be identical with bytes sent and in the correct order. Since packet transfer over many networks is not reliable, a technique known as positive acknowledgment with retransmission is used to guarantee reliability of packet transfers. This fundamental technique requires the receiver to respond with an acknowledgment message as it receives the data. The sender keeps a record of each packet it sends. The sender also maintains a timer from when the packet was sent, and retransmits a packet if the timer expires before the message has been acknowledged.

**III. FEATURES OF WHAT’S HAPPENING:**

The Proposed Android What’s Happening Application enables users with the following features:

* **Alarm**: The users have the facility to set 3 different alarms. This feature is included in order to support the idea that in order to get the news, the user should first be awake at the proper time.
* **News**: The user can get to know about the latest news in which he/she is interested in.
* **Connect**: The users can connect with other users by sending them requests. The users can also block certain users if they don’t wish to connect with them.
* **Public post**: The users can post an event which they want to conduct and notify the desired people about the event. The users on the receiving end can thereby set a reminder to that event or simply choose to ignore it if they are not interested.
* **Invitation**: This is an extended feature of ‘public post’. The users can invite other users to their events and can also ask them whether they would be attending the event or not. The users on the receiving end can set a reminder and reply with a ‘yes’, ‘may’ be or a ‘no’ option. Thus, the inviting user can get an approximate count of the people attending the event and plan the event accordingly
* **Area-wise notification**: The users have the ability to understand which event is going to occur in the near-by area by tapping their smart phone on an NFC tag which contains information regarding that event.

**IV.** **REQUIREMENTS:**

This project will be implemented using client-server architecture.

Following are some of the requirements:-

**Server Side:-**

**Hardware Requirements:-**

* Device: Smart Phone supporting Android OS
* RAM memory: 512 mb
* Processor: 1 Ghz

**Software requirements:**

* Operating system: Android 4.0 + version (Jelly Bean)
* Connectivity: WI-FI
* Installation file: apk ( Android Application Package File) file

**Client Side:-**

**Hardware Requirements:-**

* Device:- Smart Phone Supporting Android OS
* RAM Memory:- 512 Mb
* Processor:- 1 GHz

**Software Requirements:-**

* Operating System:- Android 4.0 + version

(Jelly Bean)

* Connectivity:- WI-FI
* Installation File:- apk (Android Application

Package File) file.

1. **CONCLUSION:**

Thus the friendly mobile application “What’s Happening” has been proposed which includes features such as “Area wise notification” which lets the user know about any current event taking place in a particular area in the college by simply tapping the android with the NFC tag. It includes a feature called as “public post” where you can connect with other people and get reminders from them regarding the various events, i.e., other people can send you reminders and you can directly add those reminders in your phone. It also includes a feature called as Invitation wherein you can invite your friends to attend a particular event, also you can accept or reject Invitations send by other people. It also includes a basic alarm feature and a customizable news feature. All these features are integrated in the app so that you don’t miss anything important .Thus this app is basically used to get information and updates regarding the important events in our life.

1. **VI. REFERENCES:**

[1] <http://rts.lab.asu.edu/web_438/project_final/CSE_598_Android_Architecture_Binder.pdf>

[2]

<http://www.compiletimeerror.com/2012/12/blog-post.html#.VfgXBxGqpBc>

[3] <https://developer.android.com/guide/topics/connectivity/nfc/index.html>

[4] <https://developer.android.com/guide/topics/connectivity/nfc/nfc.htm>

[5]

<http://trendblog.net/creative-and-useful-ways-to-use-nfc-tags-with-your-smartphone/>

[6] <http://developer.android.com/guide/topics/ui/notifiers/notifications.html>

[7] <http://www.tutorialspoint.com/android/android_notifications.htm>

[8] <https://en.wikipedia.org/wiki/Transmission_Control_Protocol>

[9] <http://searchnetworking.techtarget.com/definition/TCP>