

# PROJECT DOCUMENTATION

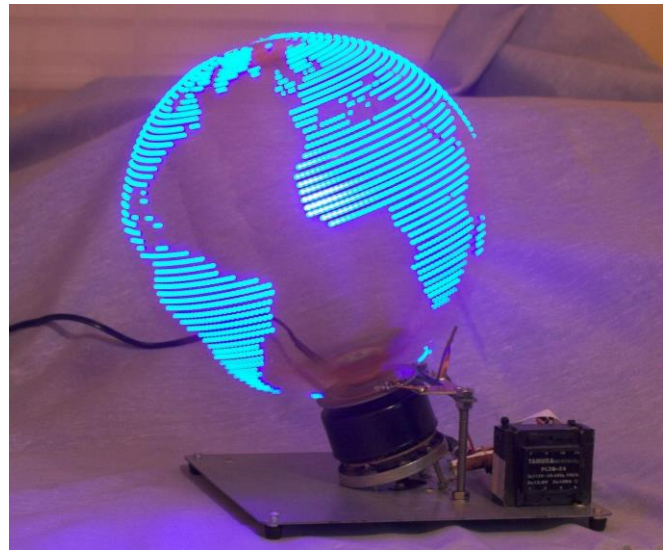
Project Title - Geotagging.

Team Members - Enayat Ullah, Ishaan Kumar, Vishal, Anshul Goyal.

## Motivation

We approached the club with a lot of ideas, some our own, some plagiarised from the internet. But it turned out some were not feasible while the others required a lot of work to be completed within a span of 40 days. Like the one involving 'RFID stickers' where an object which we tend to misplace at times like a key, pen drive are embossed with an RFID sticker on top of it and a mobile app would enable us to track these lost objects. But the fabrication of such stickers was a problem and we were at sea with how to approach it.

Another idea of ours was a 3D LED POV, something like these :



But again, as said by the coordinators, it was too complex for us to complete with the span of our summer stay.

So finally, the coordinators asked us about opinion on doing a summer project on Android Development . At first, we weren't excited about the idea as our team members wanted some hardware exposure and taking this as a summer project would mean coding and working on laptops the whole time. But finally with no alternatives at bay, we decided to go ahead with it and the decision to take up this project was finalized with the coordinators.

## Overview

Our project involved us to get familiar with the Android Development environment and thus develop an application to serve as a personal companion to one.

We basically focussed on the location fetching part and somehow utilize it to develop a reminder app.

So what our app does is fairly simple, you enter a note you want to be reminded of and within that note is the *location* where you want this app to notify you as soon as you get there.

Say for example , you write

“ Return Books in library” and hit save, our app will automatically identify library as a location keyword and tag the co-ordinates in the note.

So, as soon as you enter the 200 m radius of the very location (library) in this case, a notification is popped in your android phone reminding you the same.

We also implemented time notes in our app in case one wants to be reminded at a certain time, pretty much like alarms.

That the functionality part,now for the part, we focussed our efforts to build an intuitive ,clean and simple UI. For the same, we encompassed *card UI* in our ap. These beautiful cards,originally inspired from google now, makes the saved notes look aesthetically pleasing.

Then there's the *drag and drop* feature, a cool way to launch activities in the app by dropping those over a specified region called *Launch pad*.

We also implemented swipeable tabs so as to provide ease in navigating between *Location notes* and *Time notes*.

## Theory

**Android** is a [Linux](#)-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers.Initially developed by Android, Inc., which Google backed financially and later bought in 2005.

Android is [open source](#) and Google releases the code under the Apache License. This open source code and permissive licensing allows the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. Additionally, Android has a large community of developers writing applications ("[apps](#)") that extend the functionality of devices, written primarily in a customized version of the Java programming language.

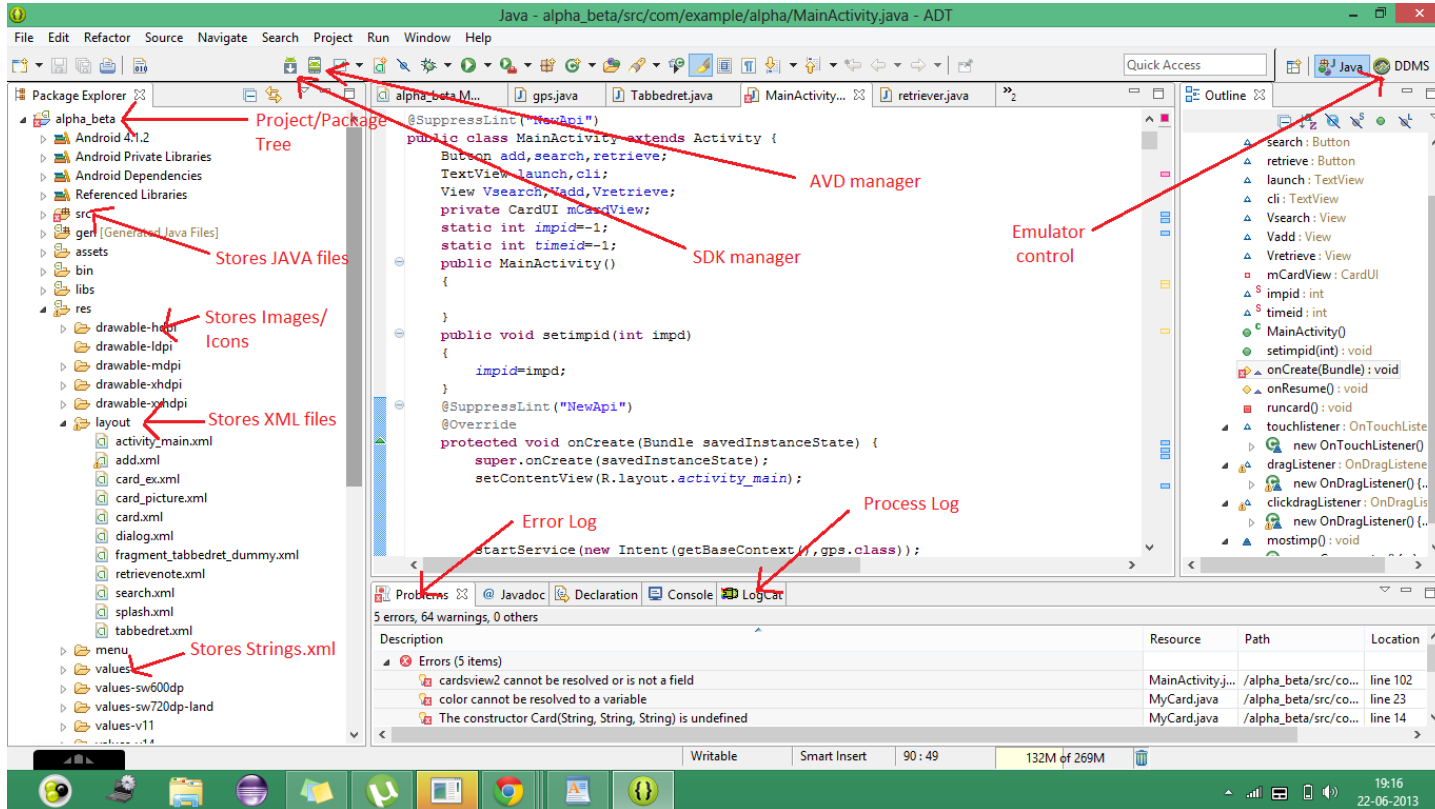
Version	Code Name	Release Date	API Level	Distribution
1.5	<i>Cupcake</i>	April 30,2009	3	0%
1.6	<i>Donut</i>	September	4	0.1%

		15,2009		
2.0-2.1	<i>Eclair</i>	October 26,2009	7	1.5%
2.2	<i>Froyo</i>	May 20, 2010	8	3.2%
2.3.3-2.3.7	<i>Gingerbread</i>	February 9,2010	10	36.4%
2.3-2.3.2	<i>Gingerbread</i>	December 6,2010	9	0.1%
3.2	<i>Honeycomb</i>	May 10, 2011	12	0%
3.2	<i>Honeycomb</i>	July 15,2011	13	0.1%
4.0.x	<i>Ice Cream Sandwich</i>	December 16,2011	15	25.6%
4.1.x	<i>Jelly Bean</i>	July 9,2012	16	29.0%
4.2.x	<i>Jelly Bean</i>	November 13,2012	17	4%

**Android software development** is the process by which new applications are created for the Android operating system. Applications are usually developed in the Java programming language using the Android Software Development Kit.

**ADT** ( Android Development Tools) is the software used to develop android apps. It basically encases Eclipse IDE,which is a multi-language [Integrated development environment](#) (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment.. The latest version comes with ADT plugin preinstalled and bundled to the IDE.

This is how the IDE looks like with the important elements marked.



**Application programming interface (API)** specifies how some software components should interact with each other.

In practice in most of the cases an API is a library that usually includes specification for routines, data structures, object classes, and variables.

An API specification can take many forms, including an International Standard such as POSIX, vendor documentation such as the Microsoft Windows API, the libraries of a programming language, e.g., Standard Template Library in C++ or Java API.

*Google APIs* can be downloaded from Google Code, Google's site for developer tools, APIs and technical resources. The Google Data API] allow programmers to create applications that read and write data from Google services. Currently, these include APIs for [Google Apps](#), [Google Analytics](#), [Blogger](#), [Google Base](#), [Google Book Search](#), [Google Calendar](#), [Google Code](#)

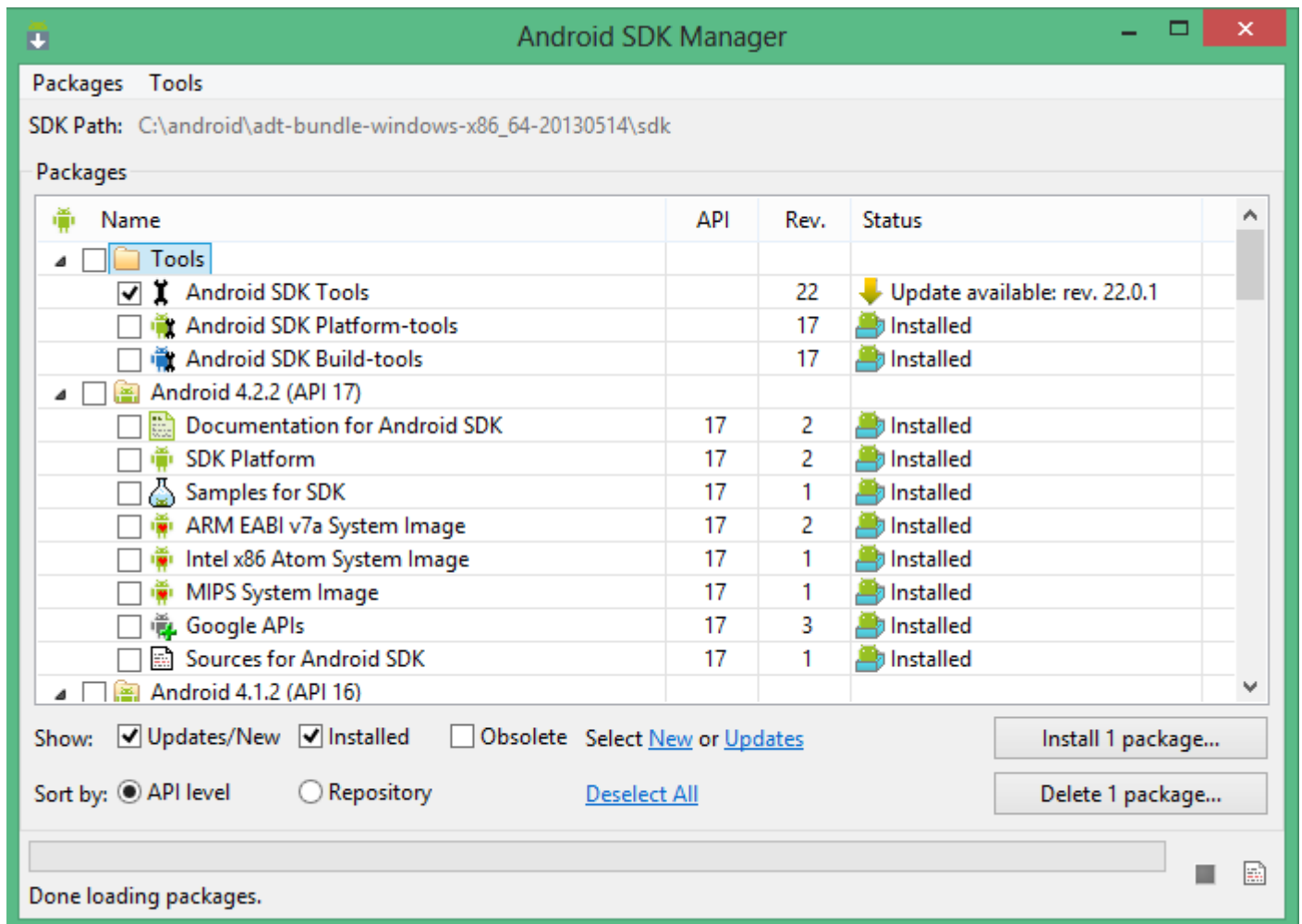
[Search](#), [Google Earth](#), [Google Spreadsheets](#), [Google Notebook](#), and [Picasa Web Albums](#).

**SDK** (Software Development Kit or "devkit") is typically a set of software development tools that allows for the creation of applications for a certain software package, software framework, hardware platform, computer system, video game console, operating system, or similar development platform.

It may be something as simple as an application programming interface (API) in the form of some files to interface to a particular programming language or include sophisticated hardware to communicate with a certain embedded system. Common tools include debugging aids and other utilities often presented in an integrated development environment (IDE).

In the latest version of ADT, the android SDK adds on to the IDE automatically as soon as you unzip and load the IDE.

SDK Manager enables us to download Google APIs and use them in our code.



**Android Virtual Device (AVD)** manager enables us to launch virtual android devices/ emulators in our PC and run the app in the emulator, and at the same time we can track and debug each app activity from the Logcat in our IDE.

### Android Virtual Device Manager

Android Virtual Devices | Device Definitions

List of existing Android Virtual Devices located at C:\Users\sony\.android\avd

AVD Name	Target Name	Platform	API Level	CPU/ABI	
✓ new	Google APIs (Google Inc.)	4.2.2	17	ARM (armeabi-v7a)	 New... Edit... Delete... Repair... Details... Start...  Refresh

✓ A valid Android Virtual Device.    📄 A repairable Android Virtual Device.  
✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.

### Create new Android Virtual Device (AVD)

AVD Name: abc

Device: [Dropdown]

Target: [Dropdown]

CPU/ABI: [Dropdown]

Keyboard: [Dropdown]

Skin: [Dropdown]

Front Camera: [Dropdown]

Back Camera: [Dropdown]

Memory Options: [Dropdown]

Internal Storage: [Dropdown]

SD Card:  
 Size: [Input] MiB [Dropdown]  
 File: [Input] Browse...

Emulation Options:  
 Snapshot     Use Host GPU

Override the existing AVD with the same name

OK    Cancel



## Our Approach

- We approached this project to learn about the android development environment not merely to sneak the code from tutorials and copy-paste to build something.

With this in mind, first we developed a lot of apps which implements different aspects of android ability and which were, totally unrelated to our project.

Among those were a primitive calculator, username-password authentication window, Splash screen music player, a primitive camera app and others

Now, after getting pretty much exposure in android development, we decided to go ahead with our project app. Ignoring the futile attempts to add many features in our app, which unfortunately could not be made possible, the work of our final app can be divided into the following phases:

- Keyword Extraction
- Card UI
- Drag and Drop
- Swipeable Tabs
- *Vani* Search Integration
- Colour Coding

### Keyword Extraction

First we thought that we would let the user enter the tag location explicitly, but then an idea struck to us so as make the app smarter in which it would recognise and isolate the location part from the note.

For the Keyword extraction from our note String, we first tried to use a web hosted API, [Alchemy API](#) whose algorithm works superbly in extracting the keyword from a sentence. We were fairly successful in implementing the same in our app, but the thing which forbade us from doing so was that this API required internet connectivity for keyword extraction, and our app would otherwise seek no net connectivity.

So, we decided to go with an offline Standalone app.

We are now performing keyword extraction from a location database which checks the note String word by word for a location keyword.

## Card UI

We had to display the notes in some simple, yet beautiful manner. Textview as out of the way for being way too simple. We explored the internet for the same.

At last, inspired from Google now cards, we decided to implement those cards in our app. We browsed around the net for tutorials and examples and after a bit of work, we succeeded in including those cards in our app.

## Drag and Drop

Clicking to launch an activity appeared way too simple and orthodox. And thus we thought of using drag and drop, like those in lock screen and other to initiate activities from launch screen.

At first, we decided that a round cavity in the centre would serve as drop region. We implemented the same, and for most part of the time, it was. But in the last days, we gave a thought to making the drop area in a certain region specifically at the bottom of the screen, and designating it *Launch pad*.

## Swipeable Tabs

We added swipeable tabs to be navigate to and from the location and time notes. These are nowadays found in many application supporting the Android 4.x framework. Just a flick of a finger and you move on to the other screen. Implementation of the same seemed easy at first, but proved to be a time-taking and tough task.

## Vani Search Integration

We decided to have a go with a feature to make the user explore and search places within the IITK campus, and what better platform to extract data from , than our IITK wiki Vani.

So when a user enters a search query, we search for the same in Vani and provide the obtained results. The user can now use the page as a full-fledged browser and explore vani content from the browser view of our app.

## Color Coding

At the last moment, we thought of adding a feature so as to prioritize our notes in the notes view. And to show the notes of various priorities , we thought of changing the colors of the note titles.

Red- Urgent

Orange - Approaching

Green - Normal

The user doesn't provided the priority value of notes. We determine the urgency from the time it is set to remind the user. Implementing the same in location notes was unsuccessful due to the unavailability of satellite information in closed areas.

## A Word of Thanks

- TheNewBoston tutorials.
- StackOverflow
- Github
- Vogella Tutorials.

We approached this as project and not a product. It was fun staying over here during summers and doing a project under Electronics Club.

We learned a lot and now could confidently usher into the app development field.