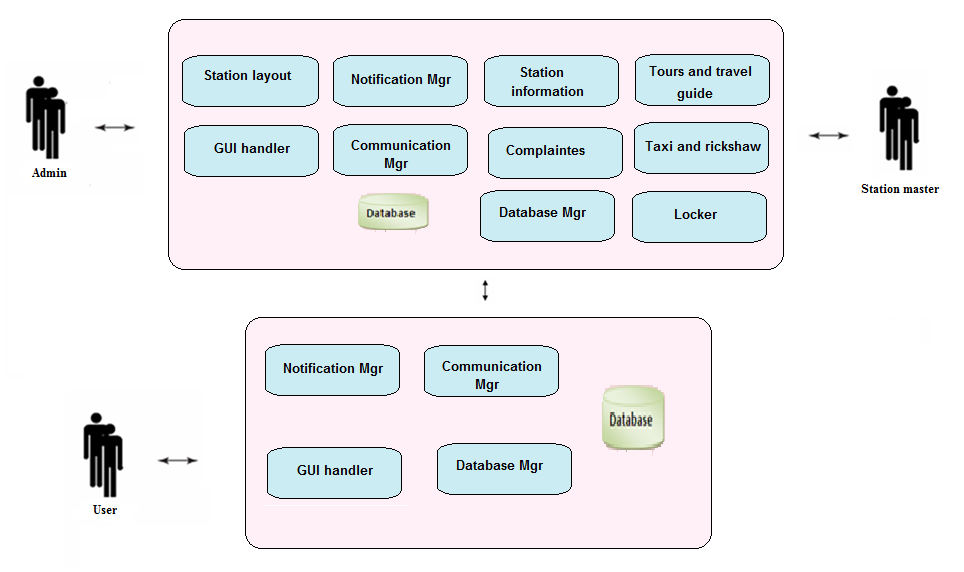
1. **Introduction**

Indian Railways is the principal mode of transport in the country. It is one of the world’s largest rail networks under a single management. The route length is around 63,332 km with more than 8000 stations. As it is the backbone of nation’s transport system, IR owns more than 25,000 wagons, 45,000 different types of coaches and 8000 locomotives. The system carries about 5,000 million passengers generating a traffic output of 340 billion passenger kms.

So many people travel daily in train and came across so many railway station and platforms. There is always problem to locate a book stall, canteen, bathroom, waiting room etc. on unknown station/platform. So we wanted to build an android app which will inform you which will be the next coming station and what is the layout of stations, like how many platforms, where is police station on map/own display.

* 1. **Problem Definition**

In railway guide system user can search all information about train, station, route, time required to reaching to destination. Apart from this through our system passenger access information like nearest tourist places, passenger can view the whole platform wise layout of a particular station. passenger can set a reminder of the particular station, which when arrives, the application will send an alert sound or vibrate. Also The passenger will be able to view the amount of time the train is going to wait at the particular station. Passenger can post complaints about the services in the train. The passenger can receive alerts and alarms for protection against thefts to take. The main aim of the project was to develop a application which would facilitate to access information about train, station layout, nearest tourist places etc through an effective and yet simple GUI for a normal passenger intending to travel in railways. Consequently, the higher number of passenger uses the train to travel from source to destination. so we proposed a system railway guide.



In railway guide system user can search all information about train, station, route, time required to reaching to destination. Apart from this through our system passenger access information like nearest tourist places, passenger can view the whole platform wise layout of a particular station.

* 1. **Literature Survey**
     1. Background of the project

In literature, we study most of railway service systems which have been developed. There are many systems for railway reservation or showing time schedule of railway. But there is no system to give information about a book stall, canteen, bathroom, waiting room etc. on unknown station/platform, nearest tourist places, layout of upcoming station, waiting time for train.

* + 1. Domain/Field of study

Mobile Computing, Web Development

* + 1. Motivation for the project

We wanted to develop a Railway Station Guide Android App which will help user to locate some place/things quickly on unknown station platform.  So many people travel daily in train and came across so many railway station and platforms. There is always problem to locate a book stall, canteen, bathroom, waiting room etc. on unknown station/platform. So we wanted to build an android app which will inform you which will be the next coming station and what is the layout of stations, like how many platforms, where is police station on map.

* + 1. Study of the existing system
       1. Indian Rail Guide App

Indian Rail Guide is a complete travel companion app for the frequent travellers of Indian Railways. With Indian Railway Guide, you can access train ticket status, train timetable, train live running information, train seat availability, train fare, train arrivals/departures at a station and much more from your mobile.

Limitations: Only static information is provided

* + - 1. IndRail Indian Railway App

Using IndRail you can get information about Indian Railway(s) such as current PNR Status, Seat Availability, Fare Enquiry, Train Routes, and information of any train.

Limitations: Only static information is provided

* + - 1. Indian Rail Train, IRCTC Info

Indian Rail Train, IRCTC Info App is developed to easily access information regarding indian rail way reservation.

Limitations: Very Less features.

* + 1. Strength and weakness

Strength:

1.User friendly application

2.Allow multiple user simultaneously

Weakness:

1.Network dependant

This application requires continuous internet, if network gone our application is not working

2.Location dependency

Requires GPS in our mobile, If there is problem in GPS then we get error in finding latitude and longitude

* + 1. Glossary of the domain

Mobile Computing, Web Development

1. **Software Requirement Specification (SRS)**
   1. Introduction
      1. Scope of the development project

**Passenger Module**

The passenger can register to use the app creating id, password. The passenger can login to the system with its own id and password created at the time of registration. The passenger will be receiving the notification of arrival of the next station. The passenger can view the whole platform wise layout of a particular station. The passenger can set a reminder of the particular station, which when arrives, the application will send an alert sound or vibrate. The passenger will be able to view the amount of time the train is going to wait at the particular station. The passenger’s location is sent to the people he/she is going to visit.

**Station Master**

The station master can login to his account of his station. The station master can feed in data of his station. The station master can send important notification to the passenger on the station. If baggage is reported lost the immediately notify the station staff and the other station masters, where ever the train is going to stop.

**Station layout module**

End user can search for next coming station and can see the layout of next station. It check all shops, tea stall, hotel on the coming station, number of platform, on which platform train will arrive etc. Station master add the station layout, information of station.

**Notification**

The passenger can set a reminder of the particular station, which when arrives, the application will send an alert sound or vibrate. Before 30 sec user get notification that train is living in 30 sec from platform. The passenger will be able to view the amount of time the train is going to wait at the particular station. All information of arriving time of train, departure time of train, route add by station master.

**Station information:**

Displays the time table of the trains i.e arrival and departure time of the train. Cost of tickets according date , the train and the source and destination. Whenever the train has been delayed the passengers will be notified of this delay and the new arrival time.

**Complaints:**

Passenger can post complaints about the services in the train. The passenger can receive alerts and alarms for protection against thefts to take precaution. If baggage is reported lost then station master immediately notify the station staff and the other station masters, where ever the train is going to stop.

**Tours and travel guide:**

The passenger can access information about various tourist places at the particular station. Passenger can access all the information of nearest tourist places like how far from station, rickshaw, taxi available or not etc.

**Taxi and rickshaw:**

The passenger can book a prepaid taxi or rickshaw from mobile at any station.

**Locker**

Number of locker, available locker, booked locker all information is add station master. Our user can access all this information and passenger can book a locker.

* + 1. Definitions, Acronyms & Abbreviations

NIL

* 1. General Description
     1. User personas and Characteristics

The user of this product/system will be any citizen who is traveling through train and want information about trains and station.

* + 1. Product perspective

So many people travel daily in train and came across so many railway station and platforms. There is always problem to locate a book stall, canteen, bathroom, waiting room etc. on unknown station/platform. So we wanted to build an android app which will inform you which will be the next coming station and what is the layout of stations, like how many platforms, where is police station on map/own display. This will help user to get to know the next station before reaching the station and help to locate whatever we need on that station. Will would also like to add one more feature like app would show how much time train will stay on this station and remind the user30 sec before. The scope does not involve the out of the station information except for the tourist places. Software has two major component one the server and the second one is the mobile application. The server will required Windows XP/Vista/7 machine with minimum 1GB RAM and 100 GB hard disk. The server machine also required WIFI devices sing which it can create Wireless Ad-hoc network.

* + 1. Overview of Data Requirements

NA

* + 1. Operating Environment

Software has two major component one the server and the second one is the mobile application. The server will required Windows XP/Vista/7 machine with minimum 1GB RAM and 100 GB hard disk. The server machine also required WIFI devices sing which it can create Wireless Ad-hoc network. Mobile application will support Android phones so at least 2 Android devices required getting the output.

* + 1. General Constraints, Assumptions, Dependencies, Guidelines

The user is expected to have Android Mobile phones and should be able to send and receive data when connected to wifi range. First the user has to register to wifi network to use the service. Network gives the functionality to login and registration facility. The registered user uses this network to send and receive messages when connected to wifi network.

* 1. Specific Requirements
     1. External Interface requirements
        1. **User Interfaces**

#### Station master:

1. Registration Page
2. Login Page
3. Profile Page
4. Enter data of its station Page
5. Layout of its station Page
6. Send notification Page
7. Registered compliant Page

#### User:

1. Registration Page
2. Login Page
3. Searching next station Page
4. Station Layout Page
5. Searching tourist place near station Page
6. Message to user according to search Page
7. Message from station master Page
8. Compliant registration Page
   * + 1. **Hardware Interfaces**

Mobile application will get installed on mobile devices. These mobile devices should have WIFI device thorough which it will connect to server.

* + - 1. **Software Interfaces**
* Operating System: Windows XP/Windows Vista/Windows 7.
* Database: MySql 6.0.
* Android 2.2 supported mobile handset
* Tomcat 6
* JDK 1.6
* Eclipse 3.4
  + - 1. **Communication Interfaces**

Here we will be using WIFI network and going to create our own communication protocol. Software will also support BASE64 encryption logic while sending data to server. Server will support HTTP protocol for web based access.

* + 1. Detailed Description of Functional Requirements

Station master:

* System should support Android handset
* System should have internet on mobile
* System should support to GPS on mobile
* System should have support to register new user
* System should have enter in their profile, when they login with their login credentials
* System should support to add and update profile.
* System should support to add new information.
* System should support to add layout of station.
* System should support to send notification message.
* System should allow users to create/update their own profile.
* System should give connectivity to client
* System should have to maintain database.
* System should allow multiple users at a time.
* System should maintain update of user
* System should maintain update of station master

User:

* System should support Android handset
* System should have internet on mobile
* System should support to GPS on mobile
* System should have support to register new user
* System should have enter in their profile, when they login with their login credentials
* System should able to get the layout of station
* System should able to receive notification message
* System should able to receive proper error message when system is fail to connect to internet.
  + 1. Performance requirements

For good performance the server should be tuned to server only server process and most of the RAM should be used for our application. Mobile application should use as much possible RAM. KVM should be tuned on mobile to provide extra address space to application

* + 1. Software Quality Attributes

Quality attributes are the overall factors that affect run-time behavior, system design, and user experience. They represent areas of concern that have the potential for application wide impact across layers and tiers. Some of these attributes are related to the overall system design, while others are specific to run time, design time, or user centric issues. The extent to which the application possesses a desired combination of quality attributes such as usability, performance, reliability, and security indicates the success of the design and the overall quality of the software application.

* 1. Reusability:

Reusability defines the capability for components and subsystems to be suitable for use in other applications and in other scenarios. Reusability minimizes the duplication of components and also the implementation time.

* 1. Availability:

Availability defines the proportion of time that the system is functional and working. It can be measured as a percentage of the total system downtime over a predefined period. Availability will be affected by system errors, infrastructure problems, malicious attacks, and system load.

* 1. Performance:

Performance is an indication of the responsiveness of a system to execute any action within a given time interval. It can be measured in terms of latency or throughput. Latency is the time taken to respond to any event. Throughput is the number of events that take place within a given amount of time.

* 1. Reliability:

Reliability is the ability of a system to remain operational over time. Reliability is measured as the probability that a system will not fail to perform its intended functions over a specified time interval.

* 1. Scalability:

Scalability is ability of a system to either handle increases in load without impact on the performance of the system, or the ability to be readily enlarged.

* 1. Testability:

Testability is a measure of how easy it is to create test criteria for the system and its components, and to execute these tests in order to determine if the criteria are met. Good testability makes it more likely that faults in a system can be isolated in a timely and effective manner.

* 1. Usability:

Usability defines how well the application meets the requirements of the user and consumer by being intuitive, easy to localize and globalize, providing good access for disabled users, and resulting in a good overall user experience.

1. **High Level Design**
   1. Database Schemas
   2. Entity Relationship Diagram
   3. Algorithms:

**Distance calculation algorithm using latitude longitude:**

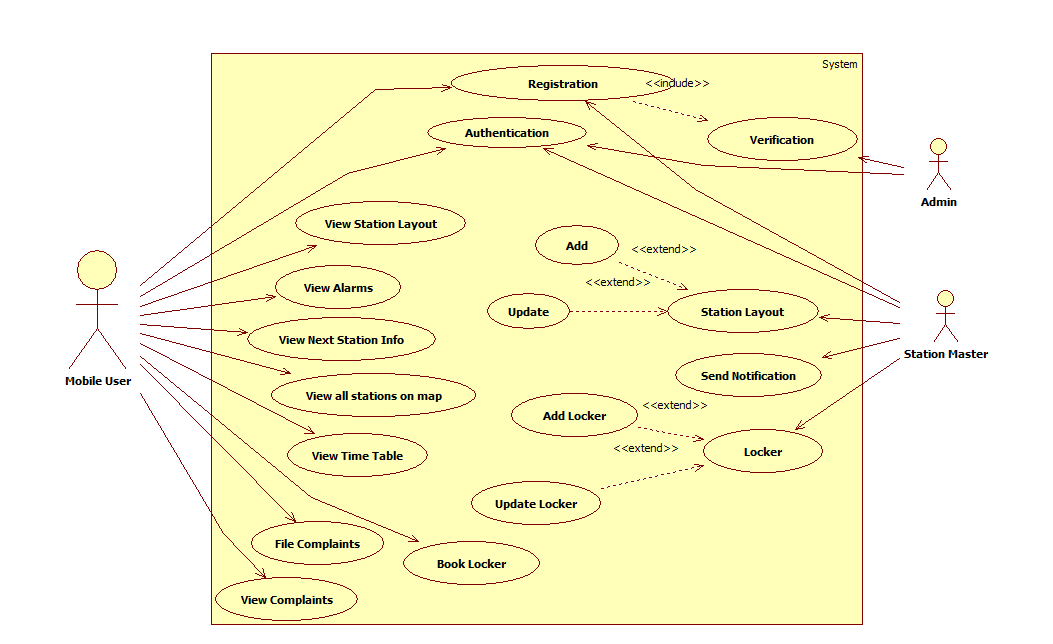
This uses the ‘**haversine**’ formula to calculate the great-circle distance between two points – that is, the shortest distance over the earth’s surface – giving an ‘as-the-crow-flies’ distance between the points (ignoring any hills they fly over, of course!).

|  |  |
| --- | --- |
| *Haversine formula:* | a = sin²(Δφ/2) + cos φ1 ⋅cos φ2 ⋅ sin²(Δλ/2) c = 2 ⋅ atan2( √a, √(1−a) ) d = R ⋅ c |
| *where* | φ*is latitude,*λ*is longitude,*R*is earth’s radius (mean radius = 6,371km); note that angles need to be in radians to pass to trig functions!* |

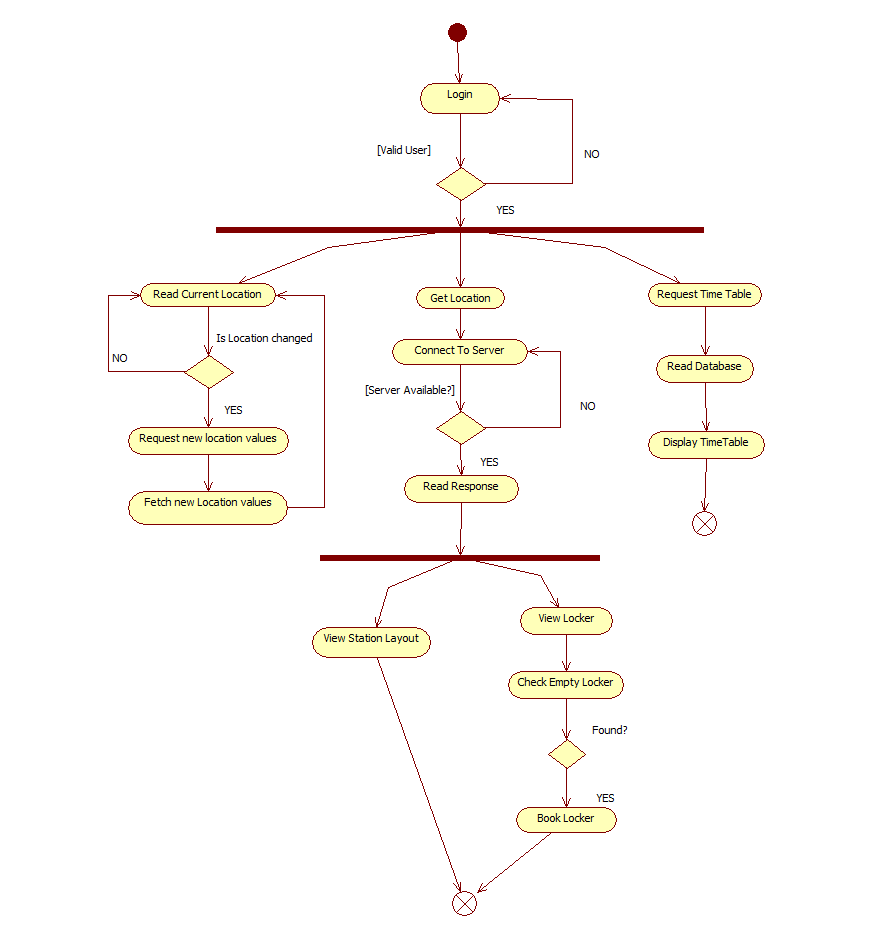
**Base64 encryption/decryption algorithm used while communication between mobile client and server.**

Base64 encoding schemes are commonly used when there is a need to encode binary data that needs to be stored and transferred over media that are designed to deal with textual data. This is to ensure that the data remains intact without modification during transport.

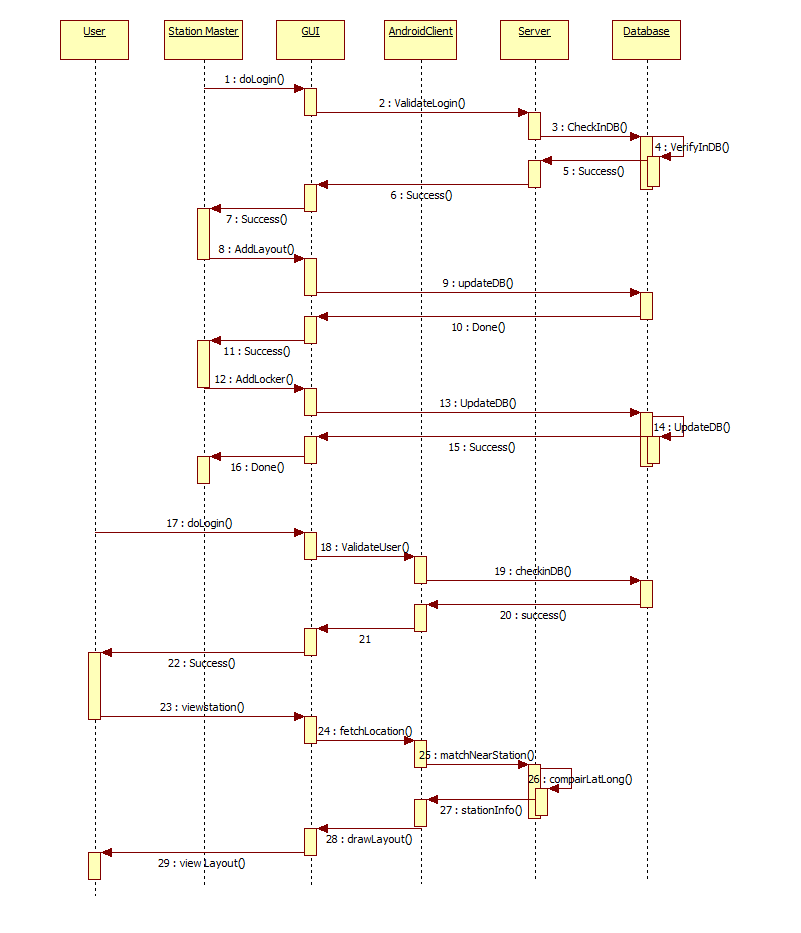
* 1. UML diagrams
     1. Use Case diagram



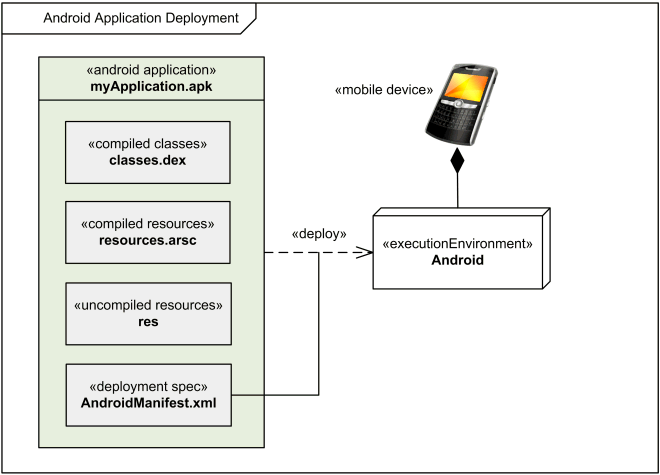
* + 1. Activity diagram

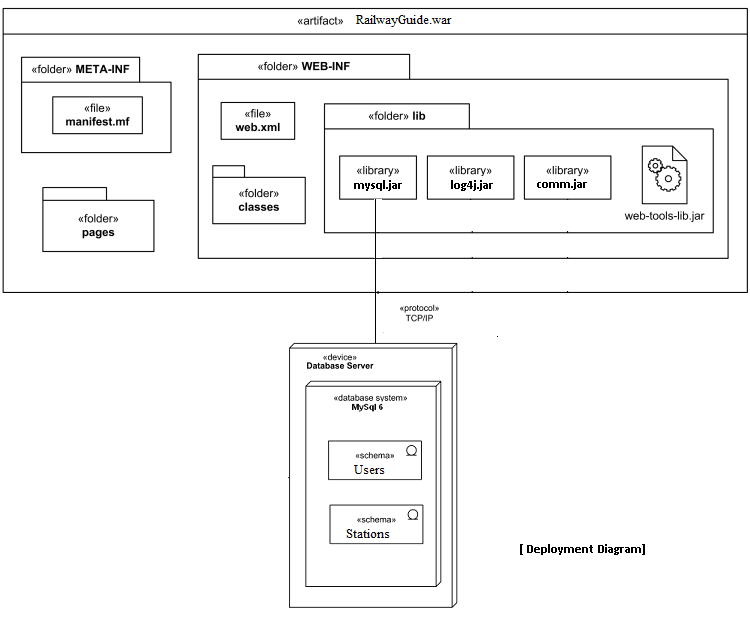


* + 1. Sequence diagram

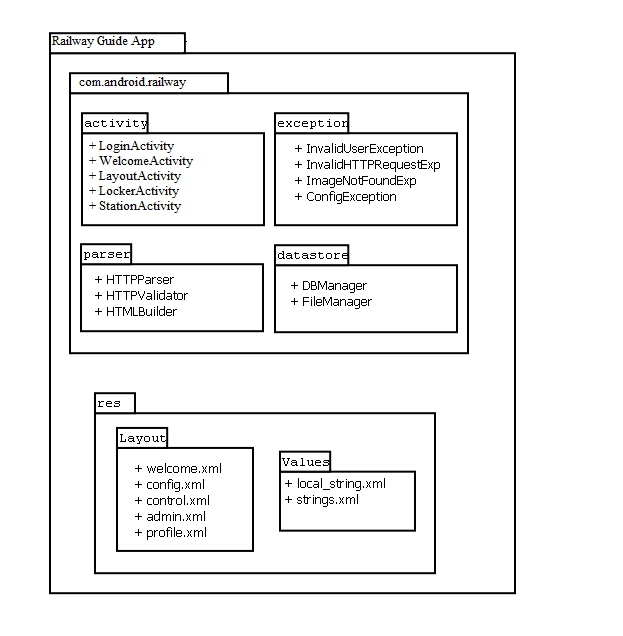


* + 1. Deployment diagram

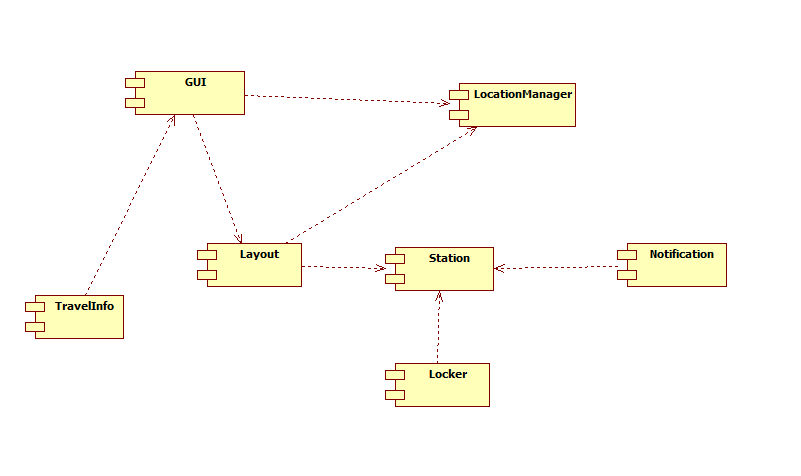




* + 1. Package diagram

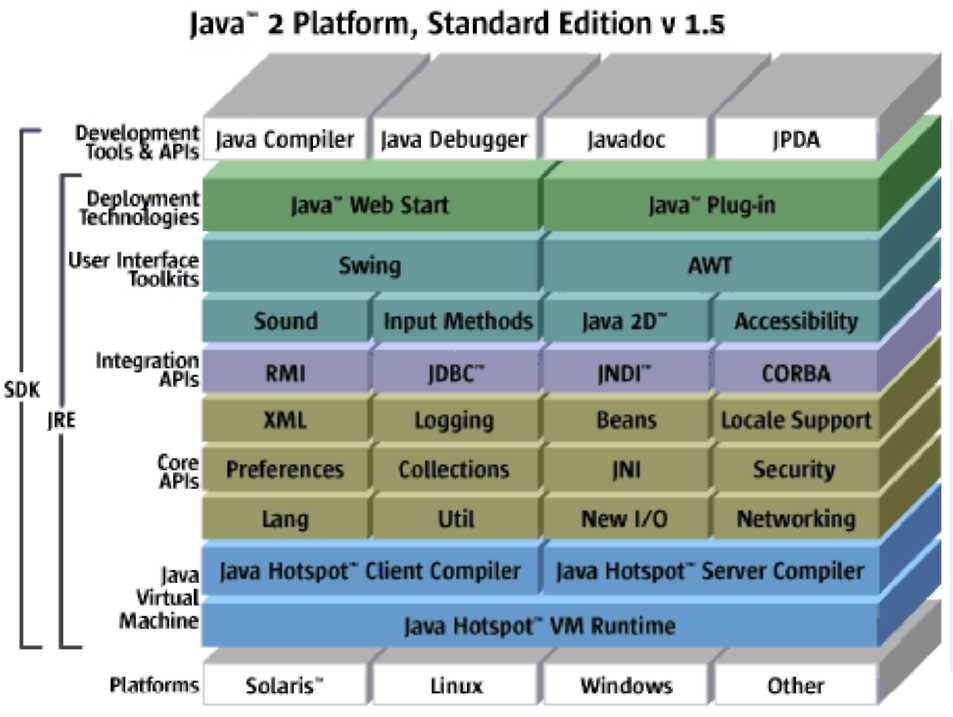


* + 1. Component diagram



1. **Technology**
   1. **Java**

Java was developed at Sun Microsystems. Work on Java initially began with the goal of creating a platform-independent language and OS for consumer electronics. The original intent was to use C++, but as work progressed in this direction, developers identified that creating their own language would serve them better. The effort towards consumer electronics led the Java team, then known as First Person Inc., towards developing h/w and s/w for the delivery of video-on-demand with Time Warner.



Unfortunately (or fortunately for us) Time Warner selected Silicon Graphics as the vendor for video-on-demand project. This set back left the First Person team with an interesting piece of s/w (Java) and no market to place it. Eventually, the natural synergies of the Java language and the www were noticed, and Java found a market. Today Java is both a programming language and an environment for executing programs written in Java Language. Unlike traditional compilers, which convert source code into machine level instructions, the Java compiler translates java source code into instructions that are interpreted by the runtime Java Virtual Machine. So unlike languages like C and C++, on which Java is based, Java is an interpreted language.

# Why Java?

Over the ages people have used tools to help them accomplish tasks, but lately their tools have been getting smarter and interconnected. Microprocessors have appeared inside many commonly used items, and increasingly, they have been connected to networks. As the heart of personal computers and workstations, for example, microprocessors have been routinely connected to networks. They have also appeared inside devices with more specific functionality than the personal computer or the workstation. Televisions, VCRs, audio components, fax machines, scanners, printers, cell phones, personal digital assistants, pagers, and wrist-watches--all have been enhanced with microprocessors; most have been connected to networks. Given the increasing capabilities and decreasing costs of information processing and data networking technologies, the network is rapidly extending its reach.

The emerging infrastructure of smart devices and computers interconnected by networks represents a new environment for software--an environment that presents new challenges and offers new opportunities to software developers. Java is well suited to help software developers meet challenges and seize opportunities presented by the emerging computing environment, because Java was designed for networks. Its suitability for networked environments is inherent in its architecture, which enables secure, robust, platform- independent programs to be delivered across networks and run on a great variety of computers and devices.

* 1. **JSP(Iava server pages)**

## What is JavaServer Pages?

JavaServer Pages (JSP) is a technology for developing web pages that support dynamic content which helps developers insert java code in HTML pages by making use of special JSP tags, most of which start with <% and end with %>.

A JavaServer Pages component is a type of Java servlet that is designed to fulfill the role of a user interface for a Java web application. Web developers write JSPs as text files that combine HTML or XHTML code, XML elements, and embedded JSP actions and commands.

Using JSP, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

JSP tags can be used for a variety of purposes, such as retrieving information from a database or registering user preferences, accessing JavaBeans components, passing control between pages and sharing information between requests, pages etc.

## Why Use JSP?

JavaServer Pages often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But JSP offer several advantages in comparison with the CGI.

* Performance is significantly better because JSP allows embedding Dynamic Elements in HTML Pages itself instead of having a separate CGI files.
* JSP are always compiled before it's processed by the server unlike CGI/Perl which requires the server to load an interpreter and the target script each time the page is requested.
* JavaServer Pages are built on top of the Java Servlets API, so like Servlets, JSP also has access to all the powerful Enterprise Java APIs, including JDBC, JNDI, EJB, JAXP etc.
* JSP pages can be used in combination with servlets that handle the business logic, the model supported by Java servlet template engines.

Finally, JSP is an integral part of J2EE, a complete platform for enterprise class applications. This means that JSP can play a part in the simplest applications to the most complex and demanding.

## Setting up JSP Environment

This step involves downloading an implementation of the Java Software Development Kit (SDK) and setting up PATH environment variable appropriately.

You can downloaded SDK from Oracle's Java site: [Java SE Downloads](http://www.oracle.com/technetwork/java/javase/downloads/index.html).

Once you download your Java implementation, follow the given instructions to install and configure the setup. Finally set PATH and JAVA\_HOME environment variables to refer to the directory that contains java and javac, typically java\_install\_dir/bin and java\_install\_dir respectively.

If you are running Windows and installed the SDK in C:\jdk1.5.0\_20, you would put the following line in your C:\autoexec.bat file.

set PATH=C:\jdk1.5.0\_20\bin;%PATH%

set JAVA\_HOME=C:\jdk1.5.0\_20

Alternatively, on Windows NT/2000/XP, you could also right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, you would update the PATH value and press the OK button.

On Unix (Solaris, Linux, etc.), if the SDK is installed in /usr/local/jdk1.5.0\_20 and you use the C shell, you would put the following into your .cshrc file.

setenv PATH /usr/local/jdk1.5.0\_20/bin:$PATH

setenv JAVA\_HOME /usr/local/jdk1.5.0\_20

Alternatively, if you use an Integrated Development Environment (IDE) like Borland JBuilder, Eclipse, IntelliJ IDEA, or Sun ONE Studio, compile and run a simple program to confirm that the IDE knows where you installed Java.

## Setting up Web Server: Tomcat

A number of Web Servers that support JavaServer Pages and Servlets development are available in the market. Some web servers are freely downloadable and Tomcat is one of them.

Apache Tomcat is an open source software implementation of the JavaServer Pages and Servlet technologies and can act as a standalone server for testing JSP and Servlets and can be integrated with the Apache Web Server. Here are the steps to setup Tomcat on your machine:

* Download latest version of Tomcat from <http://tomcat.apache.org/>.
* Once you downloaded the installation, unpack the binary distribution into a convenient location. For example in C:\apache-tomcat-5.5.29 on windows, or /usr/local/apache-tomcat-5.5.29 on Linux/Unix and create CATALINA\_HOME environment variable pointing to these locations.

Tomcat can be started by executing the following commands on windows machine:

%CATALINA\_HOME%\bin\startup.bat

or

C:\apache-tomcat-5.5.29\bin\startup.bat

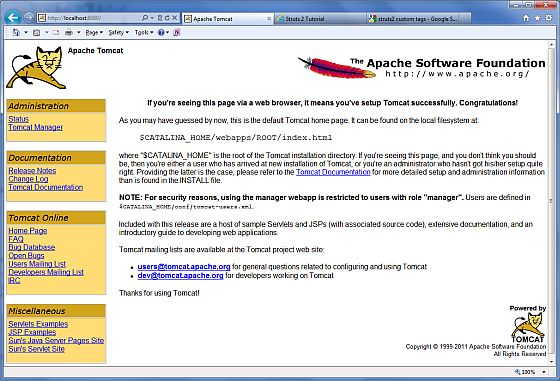
Tomcat can be started by executing the following commands on Unix (Solaris, Linux, etc.) machine:

$CATALINA\_HOME/bin/startup.sh

or

/usr/local/apache-tomcat-5.5.29/bin/startup.sh

After a successful startup, the default web applications included with Tomcat will be available by visiting **http://localhost:8080/**. If everything is fine then it should display following result:



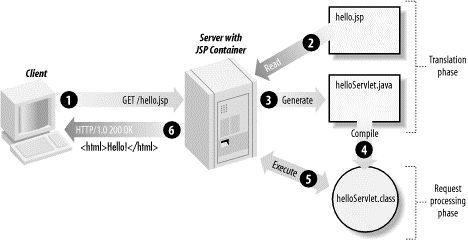
Further information about configuring and running Tomcat can be found in the documentation included here, as well as on the Tomcat web site: http://tomcat.apache.org

## JSP Processing:

The following steps explain how the web server creates the web page using JSP:

* As with a normal page, your browser sends an HTTP request to the web server.
* The web server recognizes that the HTTP request is for a JSP page and forwards it to a JSP engine. This is done by using the URL or JSP page which ends with **.jsp** instead of .html.
* The JSP engine loads the JSP page from disk and converts it into a servlet content. This conversion is very simple in which all template text is converted to println( ) statements and all JSP elements are converted to Java code that implements the corresponding dynamic behavior of the page.
* The JSP engine compiles the servlet into an executable class and forwards the original request to a servlet engine.
* A part of the web server called the servlet engine loads the Servlet class and executes it. During execution, the servlet produces an output in HTML format, which the servlet engine passes to the web server inside an HTTP response.
* The web server forwards the HTTP response to your browser in terms of static HTML content.
* Finally web browser handles the dynamically generated HTML page inside the HTTP response exactly as if it were a static page.

All the above mentioned steps can be shown below in the following diagram:



## The Scriptlet:

A scriptlet can contain any number of JAVA language statements, variable or method declarations, or expressions that are valid in the page scripting language.

Following is the syntax of Scriptlet:

<% code fragment %>

You can write XML equivalent of the above syntax as follows:

<jsp:scriptlet>

code fragment

</jsp:scriptlet>

Any text, HTML tags, or JSP elements you write must be outside the scriptlet. Following is the simple and first example for JSP:

<html>

<head><title>Hello World</title></head>

<body>

Hello World!<br/>

<%

out.println("Your IP address is " + request.getRemoteAddr());

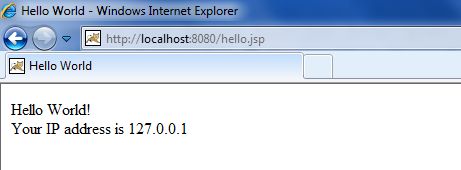
%>

</body>

</html>

**NOTE:** Assuming that Apache Tomcat is installed in C:\apache-tomcat-7.0.2 and your environment is setup as per environment setup tutorial.

Let us keep above code in JSP file hello.jsp and put this file in **C:\apache-tomcat-7.0.2\webapps\ROOT** directory and try to browse it by giving URL http://localhost:8080/hello.jsp. This would generate following result:



## JSP Declarations:

A declaration declares one or more variables or methods that you can use in Java code later in the JSP file. You must declare the variable or method before you use it in the JSP file.

Following is the syntax of JSP Declarations:

<%! declaration; [ declaration; ]+ ... %>

You can write XML equivalent of the above syntax as follows:

<jsp:declaration>

code fragment

</jsp:declaration>

Following is the simple example for JSP Declarations:

<%! int i = 0; %>

<%! int a, b, c; %>

<%! Circle a = new Circle(2.0); %>

## JSP Expression:

A JSP expression element contains a scripting language expression that is evaluated, converted to a String, and inserted where the expression appears in the JSP file.

Because the value of an expression is converted to a String, you can use an expression within a line of text, whether or not it is tagged with HTML, in a JSP file.

The expression element can contain any expression that is valid according to the Java Language Specification but you cannot use a semicolon to end an expression.

Following is the syntax of JSP Expression:

<%= expression %>

You can write XML equivalent of the above syntax as follows:

<jsp:expression>

expression

</jsp:expression>

**4.3Servlet**

## What are Servlets?

Java Servlets are programs that run on a Web or Application server and act as a middle layer between a request coming from a Web browser or other HTTP client and databases or applications on the HTTP server.

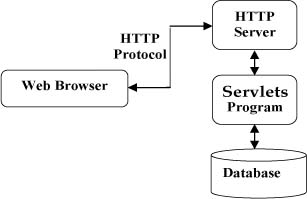
Using Servlets, you can collect input from users through web page forms, present records from a database or another source, and create web pages dynamically.

Java Servlets often serve the same purpose as programs implemented using the Common Gateway Interface (CGI). But Servlets offer several advantages in comparison with the CGI.

* Performance is significantly better.
* Servlets execute within the address space of a Web server. It is not necessary to create a separate process to handle each client request.
* Servlets are platform-independent because they are written in Java.
* Java security manager on the server enforces a set of restrictions to protect the resources on a server machine. So servlets are trusted.
* The full functionality of the Java class libraries is available to a servlet. It can communicate with applets, databases, or other software via the sockets and RMI mechanisms that you have seen already.

## Servlets Architecture:

Following diagram shows the position of Servelts in a Web Application.



## Servlets Packages:

Java Servlets are Java classes run by a web server that has an interpreter that supports the Java Servlet specification.

Servlets can be created using the **javax.servlet** and **javax.servlet.http** packages, which are a standard part of the Java's enterprise edition, an expanded version of the Java class library that supports large-scale development projects.

These classes implement the Java Servlet and JSP specifications. At the time of writing this tutorial, the versions are Java Servlet 2.5 and JSP 2.1.

Java servlets have been created and compiled just like any other Java class. After you install the servlet packages and add them to your computer's Classpath, you can compile servlets with the JDK's Java compiler or any other current compiler.

## Setting up Java Development Kit

This step involves downloading an implementation of the Java Software Development Kit (SDK) and setting up PATH environment variable appropriately.

You can download SDK from Sun's Java servlet site: http://java.sun.com/products/servlet/.

Once you download your Java implementation, follow the given instructions to install and configure the setup. Finally set PATH and JAVA\_HOME environment variables to refer to the directory that contains java and javac, typically java\_install\_dir/bin and java\_install\_dir respectively.

If you are running Windows and installed the SDK in C:\jdk1.5.0\_20, you would put the following line in your C:\autoexec.bat file.

set PATH=C:\jdk1.5.0\_20\bin;%PATH%

set JAVA\_HOME=C:\jdk1.5.0\_20

Alternatively, on Windows NT/2000/XP, you could also right-click on My Computer, select Properties, then Advanced, then Environment Variables. Then, you would update the PATH value and press the OK button.

On Unix (Solaris, Linux, etc.), if the SDK is installed in /usr/local/jdk1.5.0\_20 and you use the C shell, you would put the following into your .cshrc file.

setenv PATH /usr/local/jdk1.5.0\_20/bin:$PATH

setenv JAVA\_HOME /usr/local/jdk1.5.0\_20

Alternatively, if you use an Integrated Development Environment (IDE) like Borland JBuilder, Eclipse, IntelliJ IDEA, or Sun ONE Studio, compile and run a simple program to confirm that the IDE knows where you installed Java.

**4.4Mysql database**

## What is Database?

A database is a separate application that stores a collection of data. Each database has one or more distinct APIs for creating, accessing, managing, searching and replicating the data it holds.

Other kinds of data stores can be used, such as files on the file system or large hash tables in memory but data fetching and writing would not be so fast and easy with those types of systems.

So nowadays, we use relational database management systems (RDBMS) to store and manage huge volume of data. This is called relational database because all the data is stored into different tables and relations are established using primary keys or other keys known as foreign keys.

A **Relational DataBase Management System (RDBMS)** is a software that:

* Enables you to implement a database with tables, columns and indexes.
* Guarantees the Referential Integrity between rows of various tables.
* Updates the indexes automatically.
* Interprets an SQL query and combines information from various tables.

## RDBMS Terminology:

Before we proceed to explain MySQL database system, let's revise few definitions related to database.

* **Database:** A database is a collection of tables, with related data.
* **Table:** A table is a matrix with data. A table in a database looks like a simple spreadsheet.
* **Column:** One column (data element) contains data of one and the same kind, for example the column postcode.
* **Row:** A row (= tuple, entry or record) is a group of related data, for example the data of one subscription.
* **Redundancy:** Storing data twice, redundantly to make the system faster.
* **Primary Key:** A primary key is unique. A key value can not occur twice in one table. With a key, you can find at most one row.
* **Foreign Key:** A foreign key is the linking pin between two tables.
* **Compound Key:** A compound key (composite key) is a key that consists of multiple columns, because one column is not sufficiently unique.
* **Index:** An index in a database resembles an index at the back of a book.
* **Referential Integrity:** Referential Integrity makes sure that a foreign key value always points to an existing row.

## MySQL Database:

MySQL is a fast, easy-to-use RDBMS being used for many small and big businesses. MySQL is developed, marketed, and supported by MySQL AB, which is a Swedish company. MySQL is becoming so popular because of many good reasons:

* MySQL is released under an open-source license. So you have nothing to pay to use it.
* MySQL is a very powerful program in its own right. It handles a large subset of the functionality of the most expensive and powerful database packages.
* MySQL uses a standard form of the well-known SQL data language.
* MySQL works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
* MySQL works very quickly and works well even with large data sets.
* MySQL is very friendly to PHP, the most appreciated language for web development.
* MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase this (if your operating system can handle it) to a theoretical limit of 8 million terabytes (TB).
* MySQL is customizable. The open-source GPL license allows programmers to modify the MySQL software to fit their own specific environments.

**4.5Android**

Android is a recently developed operating system designed for mobile devices. It was developed by Google and uses a Linux based kernel, Java compatible libraries along with the just-in-time compiler for development in the Java programming language. It supports many hardware components. Common hardware consists of cameras, a WiFi communications chip, cellular commutations chip, Bluetooth sender and receiver, and a color touch screen. The Android Application Program Interface (API) contains many functions and classes to control the cellular devices. This functionality is all available in a single device with at least a day worth battery life. For this project H.263 was used in development on the Android device. The initial Android API supports recording in H.263 with Android 3.0 introducing support for H.264. Android ships with a built-in RTP receiver with support for H.263 and H.264 decoding to display video play audio. Android 3.1 introduces RTP encoding support for transmitting audio over a network using the IETF standards. With the RTP encoding integration audio may be transmitted by using the operating system streaming class. Resolutions for the encoders are limited to the recording and playback capabilities of the camera, the processor speed, and the graphics card of the device.

Android is a software stack for mobile devices that includes an operating system, middleware and key applications. The Android SDK provides the tools and APIs necessary to begin developing applications on the Android platform using the Java programming language.

Features

* Application framework enabling reuse and replacement of components
* Dalvik virtual machine optimized for mobile devices
* Integrated browser based on the open source WebKit engine
* Optimized graphics powered by a custom 2D graphics library; 3D graphics based on the OpenGL ES 1.0 specification (hardware acceleration optional)
* SQLite for structured data storage
* Media support for common audio, video, and still image formats (MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF)
* GSM Telephony (hardware dependent)
* Bluetooth, EDGE, 3G, and WiFi (hardware dependent)
* Camera, GPS, compass, and accelerometer (hardware dependent)
* Rich development environment including a device emulator, tools for debugging, memory and performance profiling, and a plugin for the Eclipse IDE

Application Framework

By providing an open development platform, Android offers developers the ability to build extremely rich and innovative applications. Developers are free to take advantage of the device hardware, access location information, run background services, set alarms, add notifications to the status bar, and much, much more.

Developers have full access to the same framework APIs used by the core applications. The application architecture is designed to simplify the reuse of components; any application can publish its capabilities and any other application may then make use of those capabilities (subject to security constraints enforced by the framework). This same mechanism allows components to be replaced by the user.

Underlying all applications is a set of services and systems, including:

A rich and extensible set of Views that can be used to build an application, including lists, grids, text boxes, buttons, and even an embeddable web browser

Content Providers that enable applications to access data from other applications (such as Contacts), or to share their own data

* A Resource Manager, providing access to non-code resources such as localized strings, graphics, and layout files
* A Notification Manager that enables all applications to display custom alerts in the status bar
* An Activity Manager that manages the lifecycle of applications and provides a common navigation backstack
* Libraries
* Android includes a set of C/C++ libraries used by various components of the Android system. These capabilities are exposed to developers through the Android application framework. Some of the core libraries are listed below:
* System C library - a BSD-derived implementation of the standard C system library (libc), tuned for embedded Linux-based devices
* Media Libraries - based on PacketVideo'sOpenCORE; the libraries support playback and recording of many popular audio and video formats, as well as static image files, including MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG
* Surface Manager - manages access to the display subsystem and seamlessly composites 2D and 3D graphic layers from multiple applications
* LibWebCore - a modern web browser engine which powers both the Android browser and an embeddable web view
* SGL - the underlying 2D graphics engine
* 3D libraries - an implementation based on OpenGL ES 1.0 APIs; the libraries use either hardware 3D acceleration (where available) or the included, highly optimized 3D software rasterizer
* FreeType - bitmap and vector font rendering
* SQLite - a powerful and lightweight relational database engine available to all applications

Android Runtime

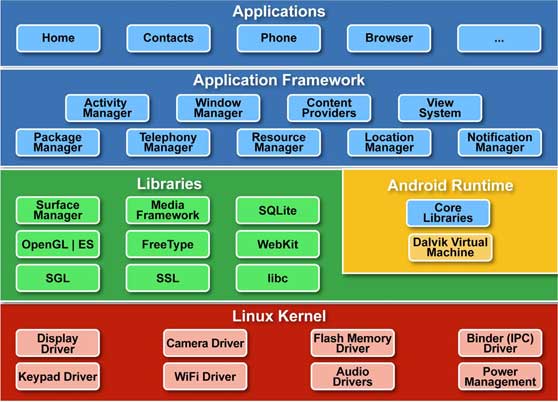
Android includes a set of core libraries that provides most of the functionality available in the core libraries of the Java programming language.Every Android application runs in its own process, with its own instance of the Dalvik virtual machine. Dalvik has been written so that a device can run multiple VMs efficiently. The Dalvik VM executes files in the Dalvik Executable (.dex) format which is optimized for minimal memory footprint. The VM is register-based, and runs classes compiled by a Java language compiler that have been transformed into the .dex format by the included "dx" tool.TheDalvik VM relies on the Linux kernel for underlying functionality such as threading and low-level memory management.

Linux Kernel

Android relies on Linux version 2.6 for core system services such as security, memory management, process management, network stack, and driver model. The kernel also acts as an abstraction layer between the hardware and the rest of the software stack.

Android Architecture

Android operating system is a stack of software components which is roughly divided into five sections and four main layers as shown below in the architecture diagram.



## Linux kernel

At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking and a vast array of device drivers, which take the pain out of interfacing to peripheral hardware.

**4.7SQLite:**

## What is SQLite?

SQLite is an in-process library that implements a self-contained, serverless, zero-configuration, transactional SQL database engine. It is the one database, which is zero-configured, that means like other database you do not need to configure it in your system.

SQLite engine is not a standalone process like other databases, you can link it statically or dynamically as per your requirement with your application. The SQLite accesses its storage files directly.

## Why SQLite?

* SQLite does not require a separate server process or system to operate.(serverless).
* SQLite comes with zero-configuration, which means no setup or administration needed.
* A complete SQLite database is stored in a single cross-platform disk file.
* SQLite is very small and light weight, less than 400KiB fully configured or less than 250KiB with optional features omitted.
* SQLite is self-contained, which means no external dependencies.
* SQLite transactions are fully ACID-compliant, allowing safe access from multiple processes or threads.
* SQLite supports most of the query language features found in the SQL92 (SQL2) standard.
* SQLite is written in ANSI-C and provides simple and easy-to-use API.
* SQLite is available on UNIX (Linux, Mac OS-X, Android, iOS) and Windows (Win32, WinCE, WinRT).

## History:

1. 2000 -- D. Richard Hipp had designed SQLite for the purpose of no administration required for operating a program.
2. 2000 -- In August SQLite 1.0 released with GNU Database Manager.
3. 2011 -- Hipp announced to add UNQl interface to SQLite DB and to develop UNQLite (Document oriented database).

## SQLite Limitations:

There are few unsupported features of SQL92 in SQLite which are shown below:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| RIGHT OUTER JOIN | Only LEFT OUTER JOIN is implemented. |
| FULL OUTER JOIN | Only LEFT OUTER JOIN is implemented. |
| ALTER TABLE | The RENAME TABLE and ADD COLUMN variants of the ALTER TABLE command are supported. The DROP COLUMN, ALTER COLUMN, ADD CONSTRAINT not supported. |
| Trigger support | FOR EACH ROW triggers are supported but not FOR EACH STATEMENT triggers. |
| VIEWs | VIEWs in SQLite are read-only. You may not execute a DELETE, INSERT, or UPDATE statement on a view. |
| GRANT and REVOKE | The only access permissions that can be applied are the normal file access permissions of the underlying operating system. |

## SQLite Commands:

The standard SQLite commands to interact with relational databases are similar as SQL. They are CREATE, SELECT, INSERT, UPDATE, DELETE and DROP. These commands can be classified into groups based on their operational nature:

## DDL - Data Definition Language:

|  |  |
| --- | --- |
| **Command** | **Description** |
| CREATE | Creates a new table, a view of a table, or other object in database |
| ALTER | Modifies an existing database object, such as a table. |
| DROP | Deletes an entire table, a view of a table or other object in the database. |

## DML - Data Manipulation Language:

|  |  |
| --- | --- |
| **Command** | **Description** |
| INSERT | Creates a record |
| UPDATE | Modifies records |
| DELETE | Deletes records |

## DQL - Data Query Language:

|  |  |
| --- | --- |
| **Command** | **Description** |
| SELECT | Retrieves certain records from one or more tables |

**4.7Google Map Integration:**

### Google provides this facility using Google play services library which you have to download externally. After downloading, you have to integrate it with your project.In the end one have to integrate your application with Google via Google console

### GOOGLE MAP - ACTIVITY FILE

Google provides GoogleMap and MapFragmentapi to integrate map in your android application. In order to use GoogleMap , you have to create an object of GoogleMap and get the reference of map from the xml layout file.Its syntax is given below:

GoogleMapgoogleMap;

googleMap=((MapFragment)getFragmentManager().findFragmentById(R.id.map)).getMap()

### GOOGLE MAP - LAYOUT FILE

Now you have to add the map fragment into xml layout file. Its syntax is given below:

<fragment

android:id="@+id/map"

android:name="com.google.android.gms.maps.MapFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

### GOOGLE MAP - ANDROIDMANIFEST FILE

The next thing you need to do is to add some permissions along with the Google Map API key in the AndroidManifest.XML file. Its syntax is given below:

<!--Permissions-->

<uses-permissionandroid:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-permissionandroid:name="android.permission.INTERNET"/>

<uses-permissionandroid:name="com.google.android.providers.gsf.permission.READ\_GSERVICES"/>

<uses-permissionandroid:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<!--Google MAP API key-->

<meta-data

android:name="com.google.android.maps.v2.API\_KEY"

android:value="AIzaSyDKymeBXNeiFWY5jRUejv6zItpmr2MVyQ0"/>

## Customizing Google Map

You can easily customize google map from its default view , and change it according to your demand.

### ADDING MARKER

You can place a maker with some text over it displaying your location on the map. It can be done by via**addMarker()** method. Its syntax is given below:

finalLatLngTutorialsPoint=newLatLng(21,57);

Marker TP =googleMap.addMarker(newMarkerOptions().position(TutorialsPoint).title("TutorialsPoint"));

### CHANING MAP TYPE

You can also change the type of the MAP. There are four different types of map and each give different view of the map. These types are Normal,Hybrid,Satellite and terrain. You can use them as below

googleMap.setMapType(GoogleMap.MAP\_TYPE\_NORMAL);

googleMap.setMapType(GoogleMap.MAP\_TYPE\_HYBRID);

googleMap.setMapType(GoogleMap.MAP\_TYPE\_SATELLITE);

googleMap.setMapType(GoogleMap.MAP\_TYPE\_TERRAIN);

### ENABLE/DISABLE ZOOM

You can also enable or disable the zoom gestures in the map by calling the**setZoomControlsEnabled(boolean)** method. Its syntax is given below:

googleMap.getUiSettings().setZoomGesturesEnabled(true);

Integrating google maps in your application basically consists of these 4 steps.

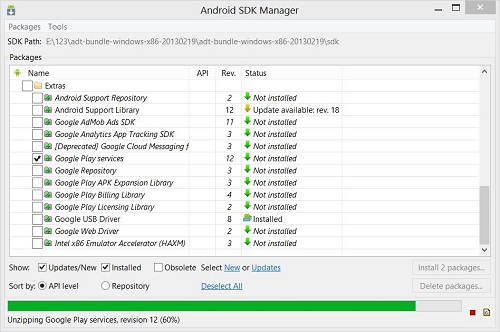
1. Download and configure. Google Play Services SDK
2. Obtain API key from google console
3. Specify Android Manifest settings

## Download and configure. Google Play Services SDK

### INSTALL GOOGLE SERVICES SDK

Open your SDK manager in the eclipse by clicking the Window and then selecting the Android SDK manager.

Navigate to the extras tab and select the Google play services and click on install this package. It would be like this.

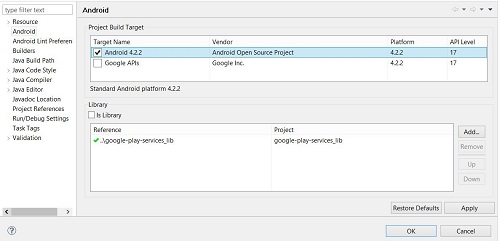


### IMPORT SDK TO ECLIPSE

After you download the SDK , click on file tab and select import option. Select existing android application code and press ok. Browse to your android folder and then sdk folder. In sdk folder expand extras folder. Expand google folder and select google play services.

### CONFIGURE YOUR PROJECT WITH SDK

After you import the SDK , you have to add it into your project. For this , right click on your eclipse project and select properties. Select android from left tab and then select add from right below panel and add the project. It would be like this



## Obtaining the API key

This part is furthur divided into two steps. First you have to get an SHA1 fingerprint key from your pc and then you have to get map API key from google console.

### GETTING CERTIFICATE FROM KEYTOOL

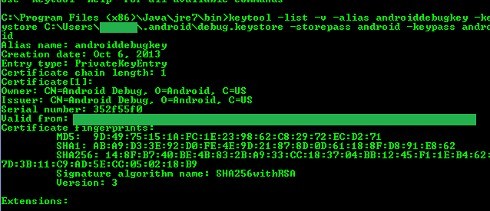
You need to get a certificate key because you have to provide it to google console in order to get your API key for map.

Open your command prompt and move to the path where your java jre has been placed. Now type this command.

keytool-list -v -aliasandroiddebugkey-keystore%%Your path%%-storepass android -keypass android

Replace the percentage part of the command with the path which you will copy from by selecting the window tab and selecting the preferences tab and then selectng the build option under android from left side.

Copy the default debug keystore path and replace it in the cmmand and hit enter. The following result would appear.



Copy the SHA1 key because you need it in the next step.

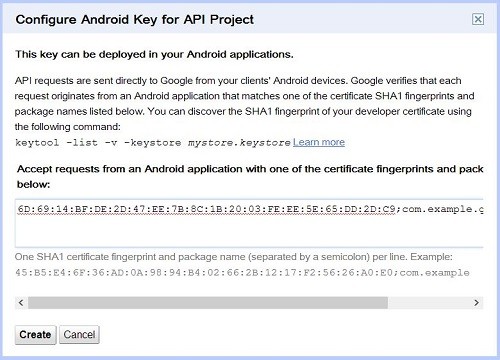
### GETTING KEY FROM GOOGLE CONSOLE

Open [Google Console](https://code.google.com/apis/console/) and sign in by clicking a new project.

Click on services from the left tab and then navigate to the Google Maps Android API v2. You have to turn them on like this



Now again go to the left tab and select API access. And click on create new android key. Now paste the key that you copied and put a semicolon and paste your project name and click create. It would be like this.



Now copy the API key that has been given to your by android , because you have to paste it into your manifest file.

## Specify Android Manifest settings

The final step is to add the API key to your application. Open your manifest file and place this code right before closing the application tag.

<meta-data

android:name="com.google.android.maps.v2.API\_KEY"

android:value="API\_KEY"/>

In the second line replace API\_KEY with your api key and you are done. You need to add some permissions in your manifest too which are given below in the manifest file.

## Adding Google Maps to your application.

Following is the content of the modifed main activity file**src/com.example.googlemaps/MainActivity.java**.

packagecom.example.googlemaps;

importcom.google.android.gms.maps.GoogleMap;

importcom.google.android.gms.maps.MapFragment;

importcom.google.android.gms.maps.model.LatLng;

importcom.google.android.gms.maps.model.Marker;

importcom.google.android.gms.maps.model.MarkerOptions;

importandroid.os.Bundle;

importandroid.app.Activity;

importandroid.widget.Toast;

publicclassMainActivityextendsActivity{

staticfinalLatLngTutorialsPoint=newLatLng(21,57);

privateGoogleMapgoogleMap;

@Override

protectedvoidonCreate(BundlesavedInstanceState){

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

try{

if(googleMap==null){

googleMap=((MapFragment)getFragmentManager().

findFragmentById(R.id.map)).getMap();

}

googleMap.setMapType(GoogleMap.MAP\_TYPE\_HYBRID);

Marker TP =googleMap.addMarker(newMarkerOptions().

position(TutorialsPoint).title("TutorialsPoint"));

}catch(Exception e){

e.printStackTrace();

}

}

}

Following is the modified content of the xml **res/layout/activity\_main.xml**.

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayoutxmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<fragment

android:id="@+id/map"

android:name="com.google.android.gms.maps.MapFragment"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

</RelativeLayout>

Following is the content of **AndroidManifest.xml** file.

<?xml version="1.0" encoding="utf-8"?>

<manifestxmlns:android="http://schemas.android.com/apk/res/android"

package="com.example.googlemaps"

android:versionCode="1"

android:versionName="1.0">

<uses-permissionandroid:name="com.example.googlemaps.permission.MAPS\_RECEIVE"/>

<uses-sdk

android:minSdkVersion="12"

android:targetSdkVersion="17"/>

<permission

android:name="com.example.googlemaps.permission.MAPS\_RECEIVE"

android:protectionLevel="signature"/>

<uses-permissionandroid:name="android.permission.ACCESS\_NETWORK\_STATE"/>

<uses-permissionandroid:name="android.permission.INTERNET"/>

<uses-permissionandroid:name="com.google.android.providers.

gsf.permission.

READ\_GSERVICES"/>

<uses-permissionandroid:name="android.permission.

WRITE\_EXTERNAL\_STORAGE"/>

<uses-permissionandroid:name="android.permission.

ACCESS\_COARSE\_LOCATION"/>

<uses-permissionandroid:name="android.permission.ACCESS\_FINE\_LOCATION"/>

<uses-feature

android:glEsVersion="0x00020000"

android:required="true"/>

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme">

<activity

android:name="com.example.googlemaps.MainActivity"

android:label="@string/app\_name">

<intent-filter>

<actionandroid:name="android.intent.action.MAIN"/>

<categoryandroid:name="android.intent.category.LAUNCHER"/>

</intent-filter>

</activity>

<meta-data

android:name="com.google.android.maps.v2.API\_KEY"

android:value="AIzaSyDKymeBXNeiFWY5jRUejv6zItpmr2MVyQ0"/>

</application>

</manifest>

Let's try to run your GoogleMaps application. I assume you have connected your actual Android Mobile device with your computer. To run the app from Eclipse, open one of your project's activity files and click Run Eclipse Run Icon icon from the toolbar. Before starting your application, Eclipse will display following window to select an option where you want to run your Android application.



Now what you need to do is to tap on the ballon to see the text.

Now you can customize the google map according to your choice with the functions given in the GoogleMap API.

**Appendix A-** Mathematical Model

1. Set Theory
2. Let ‘S’ be the ‘Railway Guide Application’.

S= {…………….}

Set S is divided into 6 modules

S= {S1, S2, S3, S4, S5, S6}

S1= GUI Handler (GH)

S2= Location Manager (LM)

S3= Station Layout Logic (SLL)

S4= Locker Module (LM)

S5= Communication Manager (CM)

S6= Travel Guide (TG)

1. Identify the inputs.

Inputs = {X1, X2, X3, ……..Xn}

X1= Location Information

X2= Locker Information

1. Identify the output as O.

Outputs = {Y1, Y2, Y3, ……..Yn}

Y1= Station Layout

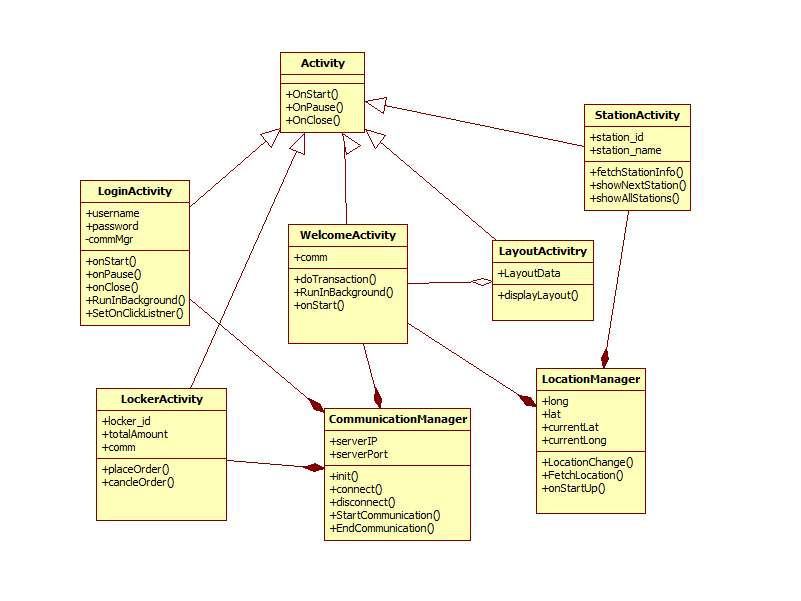
Y2= Locker Room

Y3= Travel Info

**SET THEORY**

|  |  |  |
| --- | --- | --- |
| **Sr No.** | **Description** | **UML Design observations.** |
| **1** | **Problem Description** |  |
|  | Let S be a system which do Railway Guide Application; suchthat S = {S1,S2,S3,S4,S5, S6} where S1 represents GUI Handler (GH); S2 represents Location Manager (LM); S3 represents Station Layout Logic (SLL); S4 represents Locker Module (LM); S5 represents Communication Manager (CM); S6 represents Travel Guide (TG) | S holds list of modules in the system |
| **2** | **Activities** |  |
|  | **2.1 Activity I**  **User Login Process.**  Let S1 be a set of User’s parameters for login.  S1= {userid, password}  Where,  userid – User id of the user  password – password of the user   |  |  | | --- | --- | | *Condition/Parameter* | *Operation/Function* | | If user == valid user | f1:Proceed() | | Else.. | Discard user | | If userid/password of the user is valid then proceed  Else discard the user |
|  | **2.2 Activity II**  **View Railway Station Layout**  Lets S2 be a set of station layout parameters:  S2={userid, destination\_station, layout\_id}  Where,  userid – userid of the user  destination\_station – destination station name  layout\_id – layout of that station   |  |  | | --- | --- | | *Condition/Parameters* | *Operation/Function* | | layout | f1:Search(); | | If (layout is available)  Show Layout  Else  Throw error | f2:CheckLayout() ;  f3:error() | | Search the required layout in database of the destination station  If layout available then show to user, otherwise throw error  Else throw error |
|  | **2.3 Activity III**  **Locker Checking Process**  Let S3 be the set of parameters to validate ticket.  S3:{user\_id, locker\_id, station\_id}  Where,  user\_id – userid of the user  station\_id – station id of the station  locker\_id – Locker id of the locker   |  |  | | --- | --- | | *Condition/Parameter* | *Operation/Function* | | Locker\_id | F1:Validate() | | If(locker present)  Book Locker  Else  Throw Error | F2:CheckLocker()  F3: Invalidate() | | Check if the locker is available on the destination station,  If yes book the locker for the user or else throw error. |
| **3.** | **Venn Diagram**  As described above in entire Process: Railway Guide  Input(Location Info) Output(Layout, Locker, travel guide) |  |

1. Class Diagram



1. OCL(Object Constraint Language)
   * + 1. User Registration is compulsory for accessing application features::

Self.username -> not empty

Self.usermobileno> 10

Self.userpass> 6

Self.user\_address -> not empty

Self.user\_email\_id -> not empty

Self.user\_gender ->M/F

* + - 1. User location information is required for further processing:

Self.lattitude -> not empty

Self.longitude-> not empty

* + - 1. Destination station should be selected:

Self.destination\_station\_name -> not empty

* + - 1. Train should be selected :

Self.train\_no -> valid number

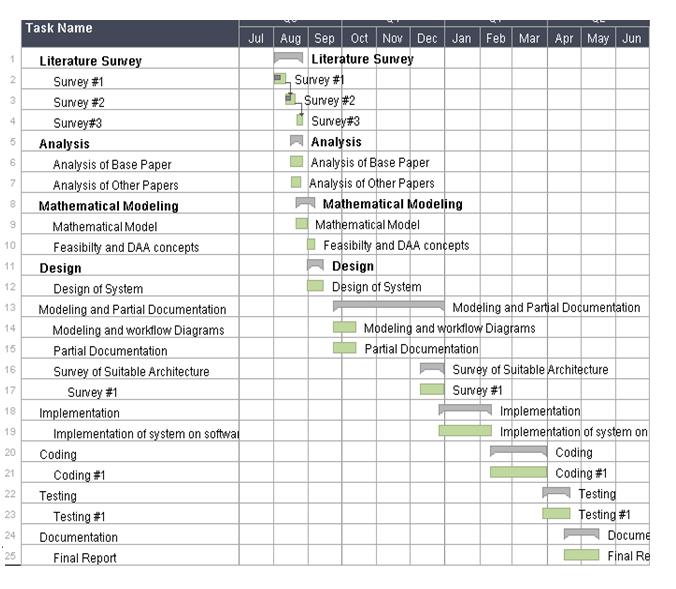
**Appendix B-** Test Planning and Test Case Designing

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case ID** | **Test Case Name** | **User Actions** | **Expected Output** |
| 1 | Apk should get installed properly. | Install the application.  Open the application. | Application should be successful installed. |
| 2 | After clicking on the android icon, application should be started. | Install the application.  First click on the menu  Search the application  Run application as android application. | Application should be successfully started. |
| 3 | System should provide the menu from which user can select the option. | Click on the menu button.  Option menu should get open. | After clicking on the particular button of the menu system should perform the respective operation. |
| 4 | System should provide the Server IP configuration button. | Install the application.  First click on the menu  Search the application  Run application as android application.  Click the configuration button. | After clicking on the button system should provide the page to fill the configuration details. |
| 5 | System should connect to the server using the Ip address. | Install the application.  First click on the menu  Search the application  Run application as android application.  Click the configuration button. | Application should successfully connect to the server. |
| 6 | System should provide the login to the user. | Run application as android application.  Go to login page. | System should provide the username password fields. |
| 7 | System should have the back and ok button. | Run application as android application.  Click the start button.  Create the login.  Click show all parking button. | After clicking on the back button system should display the previous page. |
| 8 | System should ask for destination station name | Run application as android application.  Click the start button.  Do login.  Go to Locker booking page | System should ask for destination station name before showing locker data |

**Appendix C-** Project Plan and Work progress

**Project planning**

We are planning to implement our system in the following manner. In analysis we will gather all the information related to our project and collect all the related papers. Then we are going to focus on mathematical model and finally we will work on design of the system, coding, testing and documentation



**Referances**

[1]Raman, K.S. and Wig, S., ASCI Journal of Management, 2010, vol 39, page 33-54.

[2] Agarwal S., Singhal R., Mittal R., Freight Operations Information System (FOIS) – advanced architecture & its Future. IJLRST 2013, Vol 2 Issue 2.

[3] Agarwal, V.K., Indian Railways in Full steam, www.pib.nic.in

[4] CRIS Projects: passenger r http://cris.org.in/CRIS/Projects/PRS eservation system,

[5] Kumar, A., Gadgil, A., Satapathy, A., Upadhyayula, R., Prabhudesai, S., Indian Railways IT Interface, IIMA Research and Publications, 2001.

[6] Design Document for PRS-CMC-CRIS

[7] CMC Ltd (2013). ‘Reservation system for Indian Railways, <http://www.cmcltd.com/indus> try\_practices/transportation/reservation\_system\_indian\_railways.htm

[8] Agarwal S., Rake management system: operational aspect of freight operating system, IJECS 2013, vol 2, Issue 5.

[9] Railway Audit Report, Chapter 1Computerized Passenger Reservation System of the Indian Railways, Report No 11 of 2007.

[10] http://borjournals.com/Research\_papers/Jun\_2013/1356IT.pdf