**MAJOR PROJECT REPORT**

**ON**

**“Chat Application”**

A MAJOR PROJECT REPORT SUBMITTED IN PARTIAL FULLFILLMENT FOR THE REQUIREMENT FOR THE AWARD OF THREE YEAR DIPLOMA

**IN**

**COMPUTER ENGINEERING**

UNDER THE GUIDANCE

OF

**Mr.Anoop Mishra**

logo.jpg (PROJECT IN-CHARGE)

**………………. INSTITUTE OF TECHNOLOGY**

**SHAKARPUR DELHI-92**

**SUBMITTED BY:- UNDER SUPERVISION OF: -**

**Chat Application**

A MAJOR PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE AWARD OF THREE YEAR DIPLOMA

IN

COMPUTER ENGINEERING

**…………….. INSITUTE OF TECHNOLOGY**

**SHAKARPUR, DELHI-92**

SESSION: ……………………...

**SUBMITTED TO: SUBMITTED BY:**

**“CHAT APPLICATION”**



**CERTIFICATE**

This is to be certifying that this entitled **“Chat Application”** being submitted by …………..**,**B.T.E Roll Number **…………** in the partial fulfillment of the requirement for the diploma in COMPUTER ENGG in the department of **COMPUTER ENGG,………… Institute of Technology** is a record of bonafide work done by him under my supervision and guidance. It is also certified that the dissertation has not been submitted elsewhere for any other degree.

Under the guidance of:-

**ACKNOWLEDGEMENT**

I take this opportunity to express my gratitude to my project guide **Mr**. **ANOOP MISHRA** for his endeavor encouragement and support throughout this endeavor. His insight and expertise in this field motivated and supported me during the duration of this project. It is my privilege and honor to have worked under his supervision, his invaluable guidance and helpful discussion in every stage if this project really helped me in materializing this project. Without his constructive direction and invaluable advice, this work would not have been completed

I would also like to take this opportunity to present us sincere regards to **Mr. ANOOP MISHRA** (In-charge of major project), **COMPUTER ENGINEERING**, …………….. Institute of Technology Delhi, for this. My gratitude is also extended to all teaching and nonteaching staff for their unwavering encouragement and support in our pursuit for academics. I wish to express my deepest love for my parents & family, whose endless love, understanding, and support during all these years has been the greatest assets in my life.

**SELF DECLARATION OF STUDENT**

This project is submitted as partial fulfillment of the requirement of DIPLOMA IN **COMPUTER ENGINEERING** of …………. **INSTITUTE OF TECHNOLOGY** SHAKARPUR NEW DELHI:-110092 affiliated to BTE DELHI, under the guidance of **Mr. Anoop Mishra**, Head of Computer Engineering Department,

………….INSTITUTE OF TECHNOLOGY, Shakarpur Delhi:-110092.

I hereby declare that present project report on **“CHAT APPLICATION”** is partially original and a bona fine work done by me and wherever the matter has been replicated with or without modification the same has been specially mentioned with the reasons for its usage.

NAME

BTE Roll No.

Computer Engineering

Final Year ()

**ABSTRACT**

People love to chat with their loved ones. Whenever they get time, they love to talk with their near and dear ones. The most convenient way to communicate with each other in the modern age is through various online chat applications.

The problem arises are, is user data Secured? Can my chats be accessed by anyone else also? Is the information being shared secured? Most of the times these questions remain answerless.

While chating with someone the major concern in the mind of the user is that is his data secured? To avoid that problem, we have built this application with proper measures to keep the user data secured, chats of each individual user private. The application also allows user to share important files with people.

The application provides an easy and efficient communication medium to the user where user can chat and share information with anyone in their contacts without any concern about the privacy of the data. It allows user to share information through a secure channel.

Each user can create a unique account secured with a password with the help of which user can communicate with other people in their profile.

**Objective**

This project aims at developing software that allows users to chat and share information with their loved ones. The project is developed, keeping in mind, the concern of every user have i.e. privacy of information that is being shared. To avoid that problem, we have built this application with proper measures to keep the user data secured, chats of each individual user private. Each user can create a unique account secured with a password with the help of which user can communicate with other people in their profile.

People love to chat with their loved ones. Whenever they get time, they love to talk with their near and dear ones. The most convenient way to communicate with each other in the modern age is through various online chat applications.

The problem arises are, is user data Secured? Can the chats be accessed by anyone else? Is the information being shared secured? Most of the times these questions remain answerless.

So this Android App will help to:-

* To provide better service to the users.
* To allow user to chat with the one they want.
* The allow user to share information through a secured Channel.
* Security of data and information being shaed between two users.
* To provide users a platform to chat in a fast and easy way.

**CONTENTS**

**TITLE**

1. **Certificate**
2. **Acknowledgement**
3. **Abstract**
4. **Objective**

**Chapter 1:- Introduction**

* 1. **Problem Statement**
  2. **Proposed solution**
  3. **Deliverables**

**Chapter 2:- Project Description**

**2.1. Feasibility Studies**

**2.2 Introduction**

**2.2. System Specification**

**2.3. Methodology and Tools used**

**2.4. Assumption and dependency**

**2.5. User Characteristics**

**2.6 Architecture**

**Chapter 3:- Functionality**

**3.1. Data Flow Diagram**

**3.2. Use Case Diagram**

**3.3. Flow Chart**

**Chapter 4: - Testing**

**4.1. Testing Methodology**

**4.2. Unit Testing**

**4.3. Integration Testing**

**4.4. System Testing 4.5. Functional Testing**

**4.6. Test Cases**

**Chapter 5:- Conclusion and Reference**

* 1. **Conclusion**
  2. **References/Bibliography**

**Appendix**

1. **Source Code**
2. **Screen Shots**

# 1.1. PROBLEM STATEMENT

People love to chat with their loved ones. Whenever they get time, they love to talk with their near and dear ones. The most convenient way to communicate with each other in the modern age is through various online chat applications.

The problem arises are, is user data Secured? Can the chats be accessed by anyone else? Is the information being shared secured? Most of the times these questions remain answerless.

While chating with someone the major concern in the mind of the user is that is his data secured? To avoid that problem, we have built this application with proper measures to keep the user data secured, chats of each individual user private.

**There are lots of problem also arise :-**

* No internet connection
* Privacy
* Security of Data

# 1.2. PROPOSED SOLUTION

**JUSTIFICATION AND NEED FOR THE SYSTEM**

* To provide better service to the users.
* To allow user to chat with the one they want.
* The allow user to share information through a secured Channel.
* Security of data and information being shaed between two users.
* To provide users a platform to chat in a fast and easy way.

**ADVANTAGES OF THE PROPOSED SYSTEM**

* Communication made easy and efficient.
* Provide users a Platform to chat in a fast and easy way.
* Allows user share information through a secured channel.
* Privacy of data.
* Secure means of Communication.

**1.3 DELIVERABLES**

* Table of contents
* Use Case Diagram
* Data flow diagrams
* Test Plan

**2.1 FEASIBILITY STUDY**

Feasibility study is a preliminary study undertaken before the real work of a project starts to ascertain the likelihood of the project's success. It is an analysis of all possible solutions to a problem and a recommendation on the best solution to use. It involves evaluating how the solution will fit into the corporation.

It is used to determine if the project should get the go-ahead. If the project is to proceed, the feasibility study will produce a project plan and budget estimates for the future stages of development.

* + 1. **Technical Feasibility**

The key customer benefits kept in mind while envisaging this architecture were:

* Higher maintainability, extensibility and configurability
* Improved performance and scalability
* Lower Cost of Ownership
* Better productivity
* Lower business risk
* System Performance
* System Interfaces
* Development Processes
* Risk Assessment
* Failure Immunity
* Customer Support
* Security

**2.1.2 Economical Feasibility**

Cost Benefit Analysis was done in this stage. Following activities were performed during this stage:-

* Each phase of the project was analyzed for the cost involved in it.
* This was calculated based upon resources and infrastructure used.
* Benefits of each phase which were the end products were analyzed and listed.
* Both cost involved and benefits obtained were compared to the details to get the final result.

**2.1.3 Operational Feasibility**

How well the solution will work in the organization and how the end-users and managers feel about the system.

This is important because a workable solution can be thrown away because of end-user or management doesn’t want the system. Therefore usability is another important factor.

**Usability analysis** is often performed with a working prototype of the proposed system. Test of system’s user interfaces and measured in how easy they are to learn and to use and how they support the desired productivity levels of users. Easy to learn, use and user satisfaction are other things which are considered here.

### 2.1.4 Other Feasibility Dimensions

Scheduling Feasibility was one of the dimensions. Measure of how reasonable the project timetable is. Schedule can be mandatory or desirable. It’s better to deliver a properly functioning information system later than to deliver an error-prone.

* Measure of how reasonable the project timetable is.
* Work allocation was finalized.
* Time Value for money was analyze

2.2 INTRODUCTION

There are various chat application but not many are concerned with the security and privacy of user data. This keeps the user in stiff that wether the data he is sharing is being passed through a secured channel. Is the information secured from the reach of hackers?

The proposed system deals with overcoming the problems stated above. The system is an Android application that allows the user to chat with their loved ones keeping in mind the privacy of the user. Proper measures have been taken to keep the user data secured. The platform chosen for this kind of system is Android, reason being Android Operating System has come up on a very large scale and is owned by almost every second person. Also, Android is a user friendly platform, thereby enabling ease of access for all the users.

This application has been developed using IDE (Android Studio) with ADT (Android Development Tools) and Android SDK (Software Development Kit).

There are a number of constraints that need to be satisfied. A few of them may be stated as follows:

1. The phone should not lag each time any member wish to chat with other member.

2. The platform used must be Android only.

3. All the members information and their chat history must be stored must be stored in the database and retrieved whenever asked for.

4. The application must have information about all the members.

5. The application must be user friendly enough for the user to understand it and operate it.

6. The application must not need internet while displaying the pervious chat with any other member.

7. All Android Versions above Kitkat must support the application.

* 1. **SYSTEM SPECIFICATIONS**
     1. **Hardware Requirements**
* **Programmer**
* A computer with Core i3 2.3GHz or higher processor, 4GB RAM.
* Minimum 100 GB Hard Disk Space recommended.
* 1024X768 Pixels Screen Resolution for proper viewing of Screens.
* **User**
* A Smart Phone with full features
* Minimum 100 MB storage space and 512 MB RAM
* 1024X768 Pixels Screen Resolution for proper viewing of Screens
  + 1. **Software Requirements**

**Programmer**

* Operating System : Windows 7 Home Basic
* Scripting language : ANDROID
* IDE & ADT Bundle : ECLIPSE JUNO
* Photoshop Tool : ADOBE PHOTOSHOP CS6

**User**

* Operating System : ANDROID Jelly Bean 4.3

**2.3 METHODOLOGY AND TOOLS USED**

The model that is basically being followed is the WATERFALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once the part is over the Requirement Analysis and Project Planning begins. The design starts after the requirements analysis is complete and the coding begins after the design is complete. Once the coding is completed, the testing is done. In this model the sequence of activities performed in software development project are:

* Requirement Analysis
* Project Planning
* System Design
* Coding
* Unit Testing
* System Integration and Testing

Here the linear ordering of these activities is critical .Output of one phase is the input of another phase. The output of each phase is to be consistent with overall requirement of the system.

Some of the qualities of spiral model are also incorporated like after Interface designing the user was asked to validate the design as per the requirements. Interaction with the user was also done from time to time for identifying further requirements.

WATERFALL Model was being chosen because all the requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

Requirement Analysis &

Specification

Design

Implementation & Unit testing

Integration & System Testing

Operation & Maintenance.

**Fig. 1 - Various Stages of WATERFALL MODEL**

**2.3.1 Requirement Analysis & Specification Phase**

The goal of this phase is to understand the exact requirements of the customer and to document them properly. The requirements describe the “what” of the system, not the “how”. This phase produces a large document, written in a natural language, contains a description of what the system will do without describing how it will be done. The resultant document is known as Software Requirement Specification (SRS).

**2.3.2 Design Phase**

The goal of this phase is to transform the requirements specification into a structure that is suitable for implementation in some programming language.

Here, overall software architecture is defined, and the high level and detailed design work is performed. This work is documented and known as software design description (SDD) document.

**2.3.3 Implementation & Unit testing Phase**

During this phase, design is implemented. If SDD is complete, the implementation or coding phase proceeds smoothly.

During Testing, the major activities are centered on the examination and modification of the code. Initially small modules are tested in isolation from the rest of the software product.

* + 1. **Integration & System testing Phase**

This is a very important phase. Effective testing will contribute to the delivery of higher quality software products, more satisfied users, lower maintenance costs, a and more accurate and reliable results. It is a very expensive activity and consumes one third to one-half of the cost of a typical developments project.

As we know, the purpose of unit testing is to determine that each independent module is correctly implemented. This gives a little chance to determine that the interface between modules is also correct, and for this reason integration testing of the entire system is done whereas software is part of the system. This is essential to build confidence in the developers before software is delivered to the customer or released in the market.

**2.3.5 Operation & Maintenance phase**

Software maintenance is a task that every development group has to face, when the software is delivered to the customer’s site, installed and is operational. Therefore, release of software inaugurates the operation and maintenance phase of the life cycle .The time spent and effort required to keep the software operational after is very significant. Despite the fact that it is very important and challenging task; it is routinely the poorly managed headache that nobody wants to face.

Software maintenance is a very broad activity that includes error correction, enhancement of capabilities and optimization. The purpose of this phase is to preserve the value of the software overtime.

**2.3.6 Technology Used / Tools Selection**

The strength of any project depends upon the technology on which the project is based. Today we are living in a world where technologies related to computer science are evolving every day new technologies are taking an edge over the older ones. Every new technology provides new benefits, but only small part of them remains in the competitive world.

**ANDROID CONCEPT**

**2.3.6.1. Java Concept**

Java is a [programming language](http://en.wikipedia.org/wiki/Programming_language) originally developed by [James Gosling](http://en.wikipedia.org/wiki/James_Gosling) at [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) (now part of [Oracle Corporation](http://en.wikipedia.org/wiki/Oracle_Corporation)) and released in 1995 as a core component of Sun Microsystems' [Java platform](http://en.wikipedia.org/wiki/Java_(software_platform)). The language derives much of its [syntax](http://en.wikipedia.org/wiki/Syntax_(programming_languages)) from [C](http://en.wikipedia.org/wiki/C_(programming_language)) and [C++](http://en.wikipedia.org/wiki/C%2B%2B) but has a simpler [object model](http://en.wikipedia.org/wiki/Object_model) and fewer [low-level](http://en.wikipedia.org/wiki/Low-level_programming_language) facilities. Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [byte code](http://en.wikipedia.org/wiki/Java_bytecode) ([class file](http://en.wikipedia.org/wiki/Class_(file_format))) that can run on any [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) (JVM) regardless of [computer architecture](http://en.wikipedia.org/wiki/Computer_architecture). Java is a general-purpose, concurrent, class-based, object-oriented language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere." Java is currently one of the most popular programming languages in use, particularly for client-server web applications.

The original and [reference implementation](http://en.wikipedia.org/wiki/Reference_implementation_(computing)) Java [compilers](http://en.wikipedia.org/wiki/Compiler), virtual machines, and [class libraries](http://en.wikipedia.org/wiki/Library_(computing)) were developed by Sun from 1995. As of May 2007, in compliance with the specifications of the [Java Community Process](http://en.wikipedia.org/wiki/Java_Community_Process), Sun relicensed most of its Java technologies under the [GNU General Public License](http://en.wikipedia.org/wiki/GNU_General_Public_License). Others have also developed alternative implementations of these Sun technologies, such as the [GNU Compiler for Java](http://en.wikipedia.org/wiki/GNU_Compiler_for_Java) and [GNU Class path](http://en.wikipedia.org/wiki/GNU_Classpath).

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called [Java byte code](http://en.wikipedia.org/wiki/Java_bytecode), instead of directly to platform-specific [machine code](http://en.wikipedia.org/wiki/Machine_code). Java byte code instructions are analogous to machine code, but are intended to be [interpreted](http://en.wikipedia.org/wiki/Interpreter_(computing)) by a [virtual machine](http://en.wikipedia.org/wiki/Virtual_machine) (VM) written specifically for the host hardware. [End-users](http://en.wikipedia.org/wiki/End-user) commonly use a [Java Runtime Environment](http://en.wikipedia.org/wiki/Java_Virtual_Machine) (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java [applets](http://en.wikipedia.org/wiki/Applet).

Standardized libraries provide a generic way to access host-specific features such as graphics, [threading](http://en.wikipedia.org/wiki/Thread_(computer_science)), and [networking](http://en.wikipedia.org/wiki/Computer_network).

A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executable would. Just-in-Time compilers were introduced from an early stage that compile byte codes to machine code during runtime.

[Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems) officially licensed the Java Standard Edition platform for [Linux](http://en.wikipedia.org/wiki/Linux), [Mac OS X](http://en.wikipedia.org/wiki/Mac_OS_X), and [Solaris](http://en.wikipedia.org/wiki/Solaris_(operating_system)). In the past Sun licensed Java to Microsoft but the license expired without renewal. Because Windows does not ship with a Java software platform, a network of third-party vendors and licenseesdevelop them for Windows and other operating system/hardware platforms.

Sun's trademark license for usage of the Java brand insists that all implementations be "compatible". This resulted in a legal dispute with [Microsoft](http://en.wikipedia.org/wiki/Microsoft) after Sun claimed that the Microsoft implementation did not support [RMI](http://en.wikipedia.org/wiki/Java_remote_method_invocation) or [JNI](http://en.wikipedia.org/wiki/Java_Native_Interface) and had added platform-specific features of their own. Sun sued in 1997, and in 2001 won a settlement of US$20 million, as well as a court order enforcing the terms of the license from Sun.As a result, Microsoft no longer ships Java with [Windows](http://en.wikipedia.org/wiki/Microsoft_Windows), and in recent versions of Windows, [Internet Explorer](http://en.wikipedia.org/wiki/Internet_Explorer) cannot support Java applets without a third-party plugin. Sun, and others, have made available free Java run-time systems for those and other versions of Windows.

Platform-independent Java is essential to the [Java EE](http://en.wikipedia.org/wiki/Java_Platform,_Enterprise_Edition) strategy, and an even more rigorous validation is required to certify an implementation. This environment enables portable server-side applications, such as [Web services](http://en.wikipedia.org/wiki/Web_service), [Java Servlets](http://en.wikipedia.org/wiki/Java_Servlet), and [Enterprise JavaBeans](http://en.wikipedia.org/wiki/Enterprise_JavaBean), as well as with [embedded systems](http://en.wikipedia.org/wiki/Embedded_system) based on [OSG](http://en.wikipedia.org/wiki/OSGi)I, using [Embedded Java](http://en.wikipedia.org/wiki/Embedded_Java) environments.

**2.3.6.2. XML Concept**

**Extensible Markup Language** (**XML**) is a [markup language](http://en.wikipedia.org/wiki/Markup_language) that defines a set of rules for encoding documents in a [format](http://en.wikipedia.org/wiki/File_format) which is both [human-readable](http://en.wikipedia.org/wiki/Human-readable_medium) and [machine-readable](http://en.wikipedia.org/wiki/Machine-readable_data). It is defined by the [W3C](http://en.wikipedia.org/wiki/World_Wide_Web_Consortium)'s XML 1.0 Specification and by several other related specifications, all of which are free [open standards](http://en.wikipedia.org/wiki/Open_standard).

The design goals of XML emphasize simplicity, generality and usability across the [Internet](http://en.wikipedia.org/wiki/Internet). It is a textual data format with strong support via [Unicode](http://en.wikipedia.org/wiki/Unicode) for different [human languages](http://en.wikipedia.org/wiki/Language). Although the design of XML focuses on documents, it is widely used for the representation of arbitrary [data structures](http://en.wikipedia.org/wiki/Data_structures) such as those used in [web services](http://en.wikipedia.org/wiki/Web_service).

Several [schema systems](http://en.wikipedia.org/wiki/XML_schema) exist to aid in the definition of XML-based languages, while many [application programming interfaces](http://en.wikipedia.org/wiki/Application_programming_interface) (APIs) have been developed to aid the processing of XML data.

The XML specification defines an XML document as a [well-formed](http://en.wikipedia.org/wiki/Well-formed_element) text – meaning that it satisfies a list of syntax rules provided in the specification. Some key points in the fairly lengthy list include:

* The document contains only properly encoded legal Unicode characters
* None of the special syntax characters such as < and & appear except when performing their markup-delineation roles
* The begin, end, and empty-element tags that delimit the elements are correctly nested, with none missing and none overlapping
* The element tags are case-sensitive; the beginning and end tags must match exactly.
* Tag names cannot contain any of the characters!”#$%&'()\*+,/;<=>?@[\]^`{|}~, nor a space character, and cannot start with -, ., or a numeric digit.
* A single "root" element contains all the other elements.

The definition of an *XML document* excludes texts that contain violations of well-formed rules; they are simply not XML. An XML processor that encounters such a violation is required to report such errors and to cease normal processing. This policy, occasionally referred to as "[draconian](http://en.wikipedia.org/wiki/Draco_(lawgiver)) error handling," stands in notable contrast to the behavior of programs that process [HTML](http://en.wikipedia.org/wiki/HTML), which are designed to produce a reasonable result even in the presence of severe markup errors.[[17]](http://en.wikipedia.org/wiki/XML#cite_note-19) XML's policy in this area has been criticized as a violation of [Pastel’s law](http://en.wikipedia.org/wiki/Postel%27s_law) ("Be conservative in what you send; be liberal in what you accept").

The XML specification defines a [valid XML document](http://en.wikipedia.org/wiki/XML_validation) as a [well-formed XML document](http://en.wikipedia.org/wiki/Well-formed_XML_document) which also conforms to the rules of a [Document Type Definition](http://en.wikipedia.org/wiki/Document_Type_Definition) (DTD).

**2.4 ASSUMPTIONS & DEPENDENCIES**

* User must know internet browsing fundamentals
* User must be having internet access.

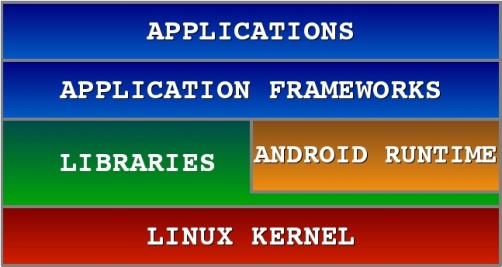
**2.5 USER CHARACTERISTICS**

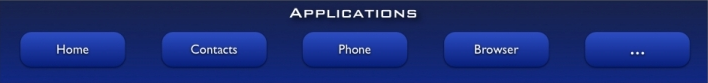
* User must know internet browsing fundamentals
* Registered user must have a valid registration id obtained by registering with the website and must remember his password
* Administrator must remember his secure password

**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.

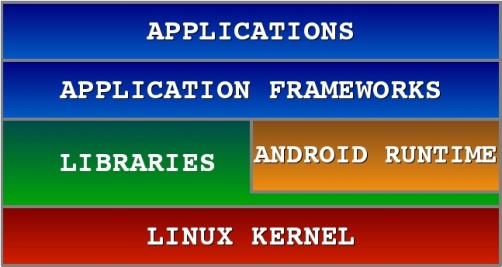


**Android S/W Stack – Application**

****

You will find all the Android application at the top layer. You will write your application to be installed on this layer only. Examples of such applications are Contacts Books, Browser, Games etc.

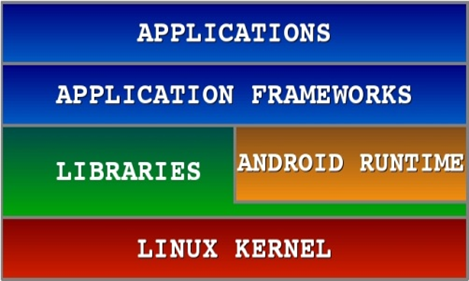
Android S/W Stack – App Framework (Cont)



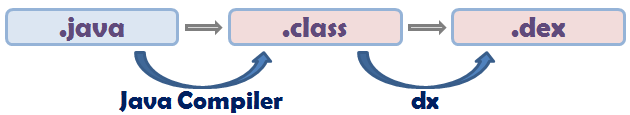


The Application Framework layer provides many higher-level services to applications in the form of Java classes. Application developers are allowed to make use of these services in their applications.

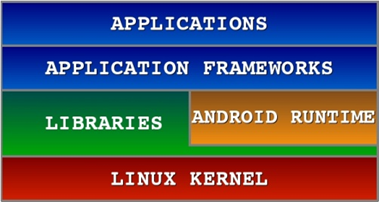
**Android S/W Stack – Runtime**

****

This is the third section of the architecture and available on the second layer from the bottom. This section provides a key component called **Dalvik Virtual Machine** which is a kind of Java Virtual Machine specially designed and optimized for Android. The Dalvik VM makes use of Linux core features like memory management and multi-threading, which is intrinsic in the Java language. The Dalvik VM enables every Android application to run in its own process, with its own instance of the Dalvik virtual machine. The Android runtime also provides a set of core libraries which enable Android application developers to write Android applications using standard Java programming language.

****

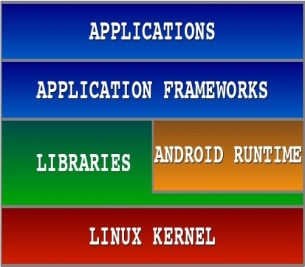
**Android S/W Stack – Libraries**

****

On top of Linux kernel there is a set of libraries including open-source Web browser engine WebKit, well known library libc, SQLite database which is a useful repository for storage and sharing of application data, libraries to play and record audio and video, SSL libraries responsible for Internet security etc.

|  |  |
| --- | --- |
| **Libraries** | **Explanation** |
| SQLite | It is used to access data published by content providers & includes SQLite DB management classes. |
| SSL | It is used to provide internet security. |
| OpenGL | It is used to provide java interface to the OpenGL/ES3D graphics rendering API. |
| Webkit | It is the browser engine used to display internet content |

**Android S/W Stack – Linux Kernel**

****

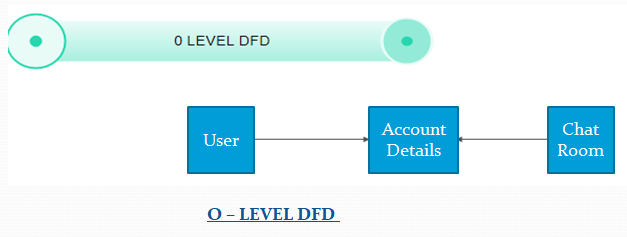
At the bottom of the layers is Linux - Linux 2.6 with approximately 115 patches. This provides basic system functionality like process management, memory management, device management like camera, keypad, display etc. Also, the kernel handles all the things that Linux is really good at such as networking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

**3.1. DATA FLOW DIAGRAM**

**Data flow diagrams** (DFDs) are one of the three essential perspectives of SSADM. The sponsor of a project and the end users will need to be briefed and consulted throughout all stages of a systems evolution.

* The components of a data flow diagram (DFD) are:
* Processes
* Flows
* Stores
* Terminators (sometimes called sources and sinks)

**O – LEVEL DFD of the safe connect**

****

**Fig 3 – DF Diagram**

**3.2. USE CASE DIAGRAM**

**ACTOR**

**Fig 4 – Use Case Diagram**

**Use case description**

**Actor: -** User

**Use case: -** Opens, Account Selection, User Registration, and Chat Room.

3.3 FLOW CHART

START

User Registration

Chat Room

End

FLOW CHART

Chat Appliation

Fig-5 flow chart

**4.1 TESTING METHODOLOGY**

Software testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. Testing is a set of activities that can be planned in advance and conducted systematically. Testing can never completely establish the correctness of arbitrary computer software. In other words, testing is criticism or comparison that is comparing the actual value with an expected one.

Testing is the process of exercising the software item to detect the differences between its behavior and the desired behavior as stipulated by the requirements specifications- 'what is' and 'what should be'. To achieve a very high standard in quality of delivery, a comprehensive and planned testing will be carried out during project execution. The Testing phase follows the coding and unit-testing phase. Testing a program basically consists of providing the program with set of test inputs or the test cases and observing whether the program behaves in the normal and expected manner. The condition under which the programs behaves in an unexpected manner and deviates from its normal course is noted for debugging and correction.

The objective is to design tests that systematically uncover different classes of errors and do so with a minimum amount of time and effort Secondary benefits include

Demonstration of software functions appearing to be working according to specification. Performance requirements appear to have been met Data collection during testing provides a good indication of software reliability and some indication of software quality. Testing cannot show the absence of defects, it can only show that software defects are present.

**4.2 UNIT TESTING**

**Unit testing** is a procedure used to validate that the individual modules or units of source code or functions are working properly. Ideally, each test case is independent from the others; mock objects can be used to assist testing a module or a piece of a module in isolation. Unit testing is typically done by the developers and not by end-users.

The purpose of unit testing is to identify and correct as many internal logic errors as possible. Unit tests would be repeatable and may be conducted at any point in the implementation process in accordance with the approved unit test plan for the module I project The goal for unit testing by developers is to perform selected path testing in which every affected branch is navigated in all possible directions at least once and every affected line of code is executed at least once. The developer would do unit testing at time of coding.

**4.3 INTEGRATION SYSTEM**

**Integration testing** is a procedure used to validate that the modules or units of source code or functions are working properly. Ideally, module is integrated which is independent from the others; mock objects can be used to assist testing a module or a piece of a module in isolation. Integration testing is typically done by the developers and by end-users.

**4.4 SYSTEM TESTING**

System testing is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing falls within the scope of Black box testing, and as such, should require no knowledge of the inner design of the code or logic. It requires testing the system as a whole.

Testing of the system requirements and the resilience of the software. Integration testing constituents checking of the integrated unit technically and functionally achieves its purpose. Pre-requisite to System Testing - Integration testing is completed and specifications for each node.

**4.5 FUNCTIONAL TESTING**

The objective of this test is to ensure that each element of the application meets the functional requirements as outlined in system specifications. Functional testing covers the aspects of the system executing functions it is supposed to execute-including user commands, data manipulation, searches and business processes, user screens, and integrations. Functional testing covers the obvious surface type of functions, as well as the back-end operations (such as security and how upgrades affect the system). Before executing the system test cases in full, a limited functional testing will be performed with a subset of system test case where the system will be run on two (or may be more is to be decided) business days and covering end-to-end two (or may be more is to be decided) event types. This is done to verify if all the components of the system is installed properly and to do a basic functionality testing. This will conclude high-level testing. It will be followed by detailed-level tests, which will aim to test the individual processes and data flows.

**5.1. CONCLSUION**

This application is user friendly. It allow user to chat wth people in their contact with proper privacy of information and security of data and information being shared. It provides efficient and quicker service to all users. It provides easy access to the chat history. It provides flexibility to accommodate future needs.

**5.2. REFERENCES AND BIBILOGRAPHY**

1. For FACEBOOK SHARE - <https://developers.facebook.com/docs/android/share#linkshare>
2. For ImageSswitcher <http://developer.android.com/reference/android/widget/ImageSwitcher.html>
3. For Photoshop tutorials - <http://www.adobephotoshoptutorials.com/free/index.php?cat=13>

**APPENDIX A: SOURCE CODE**

**Splash Activity.java**

**package** com.example.android.chatapp;  
  
**import** android.content.Intent;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
  
**public class** SplashScreen **extends** AppCompatActivity {  
  
 **@Override  
 protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_splash\_screen***);  
  
 **new** Thread()  
 {  
 **@Override  
 public void** run() {  
 **super**.run();  
  
 **try** {  
 Thread.*sleep*(1500);  
 Intent intent=**new** Intent(SplashScreen.**this**,MainActivity.**class**);  
 startActivity(intent);  
  
 } **catch** (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 }.start();  
 }  
}

**MainActivity.java**

**package** com.example.android.chatapp;  
  
**import** android.content.Intent;  
**import** android.support.annotation.**NonNull**;  
**import** android.support.design.widget.FloatingActionButton;  
**import** android.support.design.widget.Snackbar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.text.format.DateFormat;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.view.View;  
**import** android.widget.EditText;  
**import** android.widget.ListView;  
**import** android.widget.RelativeLayout;  
**import** android.widget.TextView;  
  
**import** com.firebase.ui.auth.AuthUI;  
**import** com.firebase.ui.database.FirebaseListAdapter;  
**import** com.google.android.gms.auth.api.Auth;  
**import** com.google.android.gms.tasks.OnCompleteListener;  
**import** com.google.android.gms.tasks.Task;  
**import** com.google.firebase.auth.FirebaseAuth;  
**import** com.google.firebase.database.FirebaseDatabase;  
  
  
**import** java.util.EventListener;  
  
**public class** MainActivity **extends** AppCompatActivity {  
  
 **private static int** *SIGN\_IN\_REQUEST\_CODE*=1;  
  
  
 FloatingActionButton **fab**;  
  
 **private** FirebaseListAdapter<ChatMessage> **adapter**;  
  
 RelativeLayout **relativeLayout**;  
  
 **@Override  
 protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **fab**=findViewById(R.id.***fab***);  
  
 **fab**.setOnClickListener(**new** View.OnClickListener() {  
 **@Override  
 public void** onClick(View view) {  
  
 EditText input=findViewById(R.id.***et\_input***);  
  
 FirebaseDatabase.*getInstance*().getReference().push().setValue(**new** ChatMessage(input.getText().toString(),FirebaseAuth.*getInstance*().getCurrentUser().getEmail()));  
 input.setText(**""**);  
 }  
 });  
  
 **relativeLayout**=findViewById(R.id.***myplace***);  
  
 **if** (FirebaseAuth.*getInstance*().getCurrentUser()==**null**)  
 {  
 startActivityForResult(AuthUI.*getInstance*().createSignInIntentBuilder().build(),*SIGN\_IN\_REQUEST\_CODE*);  
 }  
  
 **else** {  
  
 Snackbar.*make*(**relativeLayout**,**"Welcome "**+FirebaseAuth.*getInstance*().getCurrentUser().getEmail(),Snackbar.***LENGTH\_SHORT***).show();  
  
 }  
  
 *//load content* displayChatMessage();  
 }  
  
 **private void** displayChatMessage() {  
  
 ListView listView=findViewById(R.id.***list\_of\_message***);  
  
 **adapter**=**new** FirebaseListAdapter<ChatMessage>(**this**,ChatMessage.**class**,R.layout.***list\_item***,FirebaseDatabase.*getInstance*().getReference()) {  
  
 **@Override  
 protected void** populateView(View v, ChatMessage model, **int** position) {  
 *// get reference of view* TextView messageUser,messageText,messageTime;  
  
 messageText=v.findViewById(R.id.***message\_text***);  
 messageUser=v.findViewById(R.id.***message\_user***);  
 messageTime=v.findViewById(R.id.***message\_time***);  
  
 messageText.setText(model.getMessageText());  
 messageUser.setText(model.getMessageUser());  
 messageTime.setText(DateFormat.*format*(**"dd-MM-yyyy(HH:mm:ss)"**,model.getMessageTime()));  
  
  
 }  
 };  
  
 listView.setAdapter(**adapter**);  
 }  
  
 **@Override  
 protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
 **super**.onActivityResult(requestCode, resultCode, data);  
  
 **if**(requestCode == *SIGN\_IN\_REQUEST\_CODE*)  
 {  
 **if**(resultCode == ***RESULT\_OK***)  
 {  
 Snackbar.*make*(**relativeLayout**,**"Successfull Signed in Welcome |"**,Snackbar.***LENGTH\_SHORT***).show();  
 displayChatMessage();  
 }  
  
 **else** {  
 Snackbar.*make*(**relativeLayout**,**"Sorry Could not sign in ...."**,Snackbar.***LENGTH\_SHORT***).show();  
  
 finish();  
 }  
 }  
 }  
  
 **@Override  
 public boolean** onCreateOptionsMenu(Menu menu) {  
 **super**.onCreateOptionsMenu(menu);  
  
 getMenuInflater().inflate(R.menu.***main\_menu***,menu);  
  
 **return true**;  
 }  
  
 **@Override  
 public boolean** onOptionsItemSelected(MenuItem item) {  
 **super**.onOptionsItemSelected(item);  
  
 **if**(item.getItemId() == R.id.***menu\_sign\_out***)  
 {  
 AuthUI.*getInstance*().signOut(**this**).addOnCompleteListener(**new** OnCompleteListener<Void>() {  
 **@Override  
 public void** onComplete(**@NonNull** Task<Void> task) {  
  
 Snackbar.*make*(**relativeLayout**,**"Signout Successfully"**,Snackbar.***LENGTH\_SHORT***).show();  
 finish();  
  
 }  
 });  
 }  
  
 **return true**;  
 }  
}

**Chat Message.java**

**package** com.example.android.chatapp;  
**import** java.util.Date;  
**public class** ChatMessage {  
  
 **private** String **messageText**;  
  
 **private** String **messageUser**;  
  
 **private long messageTime**;  
  
 **public** ChatMessage(String messageText, String messageUser) {  
 **this**.**messageText** = messageText;  
 **this**.**messageUser** = messageUser;  
  
  
 **messageTime**=**new** Date().getTime();  
  
 }  
  
 **public** ChatMessage() {  
 }  
  
 **public** String getMessageText() {  
 **return messageText**;  
 }  
  
 **public void** setMessageText(String messageText) {  
 **this**.**messageText** = messageText;  
 }  
  
 **public** String getMessageUser() {  
 **return messageUser**;  
 }  
  
 **public void** setMessageUser(String messageUser) {  
 **this**.**messageUser** = messageUser;  
 }  
  
 **public long** getMessageTime() {  
 **return messageTime**;  
 }  
  
 **public void** setMessageTime(**long** messageTime) {  
 **this**.**messageTime** = messageTime;  
 }  
}

**SplashScreenActivity.XML**

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="@drawable/chatimg"  
 tools:context="com.example.android.chatapp.SplashScreen"**>  
  
</**android.support.constraint.ConstraintLayout**>

**ListItem.XML**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**>  
  
 <**TextView  
 android:textStyle="normal|bold"  
 android:id="@+id/message\_user"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"** />  
  
 <**TextView  
 android:textStyle="normal|bold"  
 android:id="@+id/message\_time"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="@+id/message\_time"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentRight="true"  
 tools:ignore="NotSibling"** />  
  
  
 <**TextView  
 android:layout\_marginTop="5dp"  
 android:textSize="18sp"  
 android:textAppearance="@style/Base.TextAppearance.AppCompat.Body1"  
 android:layout\_below="@+id/message\_user"  
 android:layout\_alignParentStart="true"  
 android:id="@+id/message\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentLeft="true"** />  
  
</**RelativeLayout**>

**MainActivity.XML**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout  
 xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:id="@+id/myplace"  
 tools:context="com.example.android.chatapp.MainActivity"**>  
  
 <**android.support.design.widget.FloatingActionButton  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:clickable="true"  
 android:id="@+id/fab"  
 android:layout\_alignParentEnd="true"  
 android:layout\_alignParentBottom="true"  
 android:tint="@android:color/white"  
 android:src="@drawable/ic\_action\_name"**/>  
  
  
 <**android.support.design.widget.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_toLeftOf="@id/fab"  
 android:layout\_alignParentBottom="true"  
 android:layout\_alignParentStart="true"**>  
 <**EditText  
 android:hint="message...."  
 android:id="@+id/et\_input"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"** />  
 </**android.support.design.widget.TextInputLayout**>  
  
  
 <**ListView  
 android:layout\_marginBottom="16dp"  
 android:divider="@android:color/transparent"  
 android:dividerHeight="16dp"  
 android:layout\_above="@id/fab"  
 android:layout\_alignParentStart="true"  
 android:layout\_alignParentTop="true"  
 android:id="@+id/list\_of\_message"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"**></**ListView**>  
</**RelativeLayout**>

Menu.XML

*<?***xml version="1.0" encoding="utf-8"***?>*<**menu xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"** >  
  
 <**item  
 android:title="Signout"  
 app:showAsAction="never"  
 android:id="@+id/menu\_sign\_out"**></**item**>  
  
</**menu**>

**Manifest**

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.android.chatapp"**>  
  
 <**uses-permission android:name="android.permission.INTERNET"** />  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@drawable/chatimg"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>  
 <**activity android:name=".MainActivity"**>  
  
 </**activity**>  
 <**activity android:name=".SplashScreen"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>  
 </**application**>  
  
</**manifest**>

APPENDIX B: ScreenShot



