**PLAGIO - A PLAGIARISM DETECTOR AS AN ANDROID APPLICATION**

**A PROJECT REPORT**

*submitted by*

**RAMYAA R 21609104055**

**SUPRIYA S 21609104075**

**VAISHALI R 21609104082**

*in partial fulfillment of the award of the degree*

*of*

**Bachelor of Engineering**

in

**Computer Science and Engineering**

****

**SAVEETHAENGINEERINGCOLLEGE , THANDALAM**

**ANNAUNIVERSITY: CHENNAI 600 025**

MAY 2013

**BONAFIDE CERTIFICATE**

Certified that this project report titled “**PLAGIO – A PLAGIARISM DETECTOR AS AN ANDROID APPLICATION***”*,is the bonafide work of **Ms. RAMYAA R, Ms. SUPRIYA S andMs.VAISHALI R** who carried out the project under my supervision, for the partial fulfillment of the requirements for the award of the degree of *Bachelor of Engineering* in *Computer Science and Engineering*.

SIGNATURE SIGNATURE

***Mr.R.Saravanan, M.E.,(Ph.D.),*** ***Mr.G.Nagappan, M.E.,(Ph.D.),***

Associate Professor Associate Professor

HEAD OF THE DEPARTMENT SUPERVISOR

Dept.of Computer Science and Dept.of Computer Science and

Engineering, Engineering,

SaveethaEngineeringCollege, SaveethaEngineeringCollege,

Thandalam, Thandalam,

Chennai 602 105. Chennai 602 105.

Submitted for the Project work and Viva voce Examination held on \_\_\_\_\_\_\_\_\_

INTERNAL EXAMINER EXTERNAL EXAMINER

**ACKNOWLEDGEMENT**

We express our deep sense of gratitude to our honourable and beloved Founder President and Chancellor ***Dr.N.M.Veeraiyan***, beloved President ***Dr.V.Saveetha Rajesh***, beloved Director ***Dr.S.Rajesh***, and other management members for helping us in all the ways to do this project.

We express our whole-hearted gratitude to our Principal ***Dr.R.Venkatasamy,*** for having given us spontaneous and whole-hearted encouragement in completing this project.

We express our sincere thanks of gratitude to our Dean, ***Prof.R.Dheenadayalu***, for having given us whole-hearted valuable suggestions in completing this project.

We express our sincere thanks of gratitude to ***Mr.R.Saravanan,***Associate Professor and Head, Department of Computer Science and Engineering, for having given us valuable suggestions and guidance in completing this project.

We would like to express our sincere thanks and deep sense of gratitude to our internal supervisor ***Mr.G.Nagappan,***Associate Professor, Department of Computer Science and Engineering, Saveetha Engineering College, for his valuable guidance, suggestions and constant encouragement which paved way for the successful completion of the project work.

We express our sincere thanks to our Project Coordinators ***Mr.G.Nagappan and Mr.V.Perumal,*** Associate Professors, Department of Computer Science and Engineering, for their constant encouragement and support throughout our course, especially for the useful suggestions given during the course of the project work.

We are pleased to acknowledge our sincere thanks to all the members of our College ***faculty, staff***, ***and technicians*** for their kind co-operation during the course of the project and to ***our beloved parents, friends and well wishers*** who encouraged us to complete this project successfully.

**ABSTRACT**

The term plagiarism is defined as the practice of taking someone else’s work or ideas and passing them off as one’s own. This project aims at developing an android application to detect plagiarism that exists between documents.

This could help a lecturer to easily find the genuineness of the students in case of assignment submission where each student in the class is expected to come up with his own perspective on the topic put up.

The main goal of this project is to develop the**PLAGIO** app on Android Platform in **AAKASH TABLET** using Eclipse, android SDK and ADT Plug-in to view the text files present in the SD-Card comparing them and identifying the plagiarised documents while displaying the percentage of the plagiarised content. Android SQLite database is used for storing the registration and login details.

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Name** | **Page No.** |
|  |  |  |
| 3.1 | TEST PLAN | 23 |

**LIST OF FIGURES**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | **FIGURE NO.** | **FIGURE NAME** | **PAGE NO.** | | 2.1 | OVERALL SYSTEM ARCHITECTURE | 13 | | 2.2 | ANDROID ARCHITECTURE | 14 | | 2.3 | SQLITE ARCHITECTURE | 15 | | 2.4 | USE CASE DIAGRAM | 16 | | 2.5 | CLASS DIAGRAM | 17 | | 2.6 | ACTIVITY DIAGRAM | 18 | | 2.7 | SEQUENCE DIAGRAM | 19 | | b.1 | LOADING PAGE | 56 | | b.2 | REGISTRATION PAGE | 57 | | b.3 | LOGIN PAGE | 58 | | b.4 | MAIN PAGE | 59 | | b.5 | COMPARISON | 60 | | b.6 | FILE ADDED | 61 | | b.7 | FILE DISCARDED | 62 | |  |  |
|  |  |  |
|  |  |  |
| **LIST OF ABBREVIATIONS**  IDE - Integrated Development Environment  ADT - Android Development Tools  SDK - Software Development Kit  UML - Unified Modeling Language  AVD - Android Virtual Device  DVM - Dalvik Virtual Machine  XML - Extensible Markup Language |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** | **PAGE NO.** |
|  | **ABSTRACT**  **LIST OF TABLES**  **LIST OF FIGURES**  **LIST OF ABBREVIATIONS** | **i**  **ii**  **iii**  **iv** |
| **1**  **CHAPTER NO.** | **INTRODUCTION**   * 1. OVERVIEW OF THE PROJECT   1.1.1 Objective  1.1.2 AAKASH Tablet  1.2 EXISTING SYSTEM  1.3 PROPOSED SYSTEM  1.4 SYSTEM ENVIRONMENT  1.4.1 Hardware Used  1.4.2 Software Used  1.4.3 Java  1.4.4 Android  1.4.5 SQLite  1.5 SYSTEM ANALYSIS  1.5.1 Introduction  1.5.2 System Analysis Activities  1.5.3 Functional Requirements  1.5.4 Non-Functional Requirements  **TITLE**  1.5.5 Feasibility Report | 1  1  1  2  3  3  3  3  3  4  5  6  7  7  8  9  11  **PAGE NO.**  12 |
| **2** | **DESIGN**  2.1 OVERALL SYSTEM ARCHITECTURE  2.2 ANDROID ARCHITECTURE  2.3 SQLITE ARCHITECTURE  2.4 DETAILED DESIGN  2.4.1 UseCase diagram  2.4.2 Class Diagram  2.4.3 Activity Diagram  2.4.4 Sequence Diagram | 13  13  14  15  16  16  17  18  19 |
| **3** | **IMPLEMENTATIONAND TESTING**  3.1 IMPLEMENTATION  3.1.1 Registration  3.1.2 User Login  3.1.3 Upload Files  3.1.4 Comparison and Result  3.2 TESTING  3.2.1 Test Plan | 20  20  20  20  21  21  22  23 |
| **CHAPTER NO.**  **4** | **TITLE**  **CONCLUSION AND FUTURE WORK**  4.1 CONCLUSION  4.2 FUTURE WORK | **PAGE NO.**  24  24  24 |
|  | **APPENDIX A(SAMPLE CODING)**  **APPENDIX B(OUTPUT SCREENSHOTS)**  **REFERENCES** | 25  56  62  63 |

**CHAPTER 1**

**INTRODUCTION**

**1.1 OVERVIEW OF THE PROJECT**

There are many types of plagiarism, such as copy and paste, redrafting or paraphrasing of the text, plagiarism of idea, and plagiarism through translation from one language to another. These types have made plagiarism one of the serious problems while copy and paste being predominant in academic area precisely.

Example, a modern research found that 70% of students confess to a few plagiarism, with about half being guilty of an earnest cheating offence on a written assignment.Additionally, 40% of students confess to using the "cut- paste" method when completing their assignments

Differentiating between the plagiarized documents and non-plagiarized documents in an effective and efficient way is one main issue in plagiarism detection field.Detection of such acts is time consuming and difficult for the lecturerswhen done manually.

Detection methods can be made moreadvanced by implementing it as an android application which will be simple for the lecturers to use.

* + 1. **Objective**

The main objective of this application is to identify the plagiarised documents. It helps the teacher to easily check the papers or assignments submitted by the students, any malpractice such as copying from each other or submitting exact replica from the text books can be automatically detected through PLAGIO. Thus reducing unnecessary manual work.

* + 1. **AAKASH Tablet**

Aakash tablet, Low Cost Access cum Computing Device (LCACD) was launched in India by MHRD for educational use during Oct 2011. In near future, Government of India plans to deploy Millions of these tablets amongst students, to enhance the effectiveness of their learning.

Aakash  is first in a series of [Android](http://en.wikipedia.org/wiki/Android_(operating_system))-based [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer) produced by British company [DataWind](http://en.wikipedia.org/wiki/DataWind). It is manufactured by the India-based company Quad, at a new production centre in [Hyderabad](http://en.wikipedia.org/wiki/Hyderabad,_India), with a planned trial run of 100,000 units.[[4]](http://en.wikipedia.org/wiki/Aakash_(tablet)#cite_note-nyt-4) The tablet was officially launched as the Aakash in New Delhi on 5 October 2011. The Indian [Ministry of Human Resource Development](http://en.wikipedia.org/wiki/Ministry_of_Human_Resource_Development_(India)) announced an upgraded second-generation model called Aakash 2 in April 2012.

The Aakash is a low-cost tablet computer with a 7-inch [touch screen](http://en.wikipedia.org/wiki/Touch_screen), [ARM](http://en.wikipedia.org/wiki/ARM_architecture) [11](http://en.wikipedia.org/wiki/ARM11) processor and 256 MB [RAM](http://en.wikipedia.org/wiki/RAM) running under the Android 2.2 operating system. It has two [universal serial bus](http://en.wikipedia.org/wiki/Universal_serial_bus) (USB) ports and delivers [high definition](http://en.wikipedia.org/wiki/High-definition_video) (HD) quality video. For applications, the Aakash will have access to [Getjar](http://en.wikipedia.org/wiki/Getjar), an independent market, rather than the [Android Market](http://en.wikipedia.org/wiki/Android_Market).

The device was developed as part of the country's aim to link 25,000 colleges and 400 universities in an e-learning program. Originally projected as a "$35 laptop", the device will be sold to the Government of India and distributed to university students – initially at US$50 until further orders are received and projected eventually to achieve the target $35 price. A commercial version of Aakash is currently marketed as UbiSlate 7  at a price of $60.The Aakash 2, codenamed UbiSlate 7Ci, was released on 11 November 2012 and has a configuration that is an improvement over previous versions. The tablet will be sold to MHRD at a cost of Rs.2263 and subsidised to Rs.1130 for students.

* 1. **EXISTING SYSTEM**

The present system followed for assignment evaluation includes a lot of manual work and is also time consuming for the lecturers. It is very difficult for them to check each and every document and analyse it manually. As majority of the students use copy-paste method which is an act of plagiarism, the genuineness cannot be identified. The softwaresavailable for plagiarism detection are mostly online applications and cannot be used efficiently for this purpose.

* 1. **PROPOSED SYSTEM**

The motive of this plagio tool is to allow the user to differentiate between genuine and plagiarized (copied) documents. This can serve as a handy tool for the lecturers to easily evaluate assignments or papers as it will not involve tedious manual work. The lecturers using this application can easily feel the difference from the already existing mechanisms as it is easily available in the Aakash tablet as an offline application. It provides an easy and convenient way to categorize plagiarized documents from the genuine documents.

* 1. **SYSTEM ENVIRONMENT**
     1. **Hardware Used**
* AAKASH Tablet or any equivalent Android Device
* Processor : P4 or Higher
* Memory : 1GB or above
  + 1. **Software Used**
* Front End : Android
* Back End : SQLite
* Open Source Software : Eclipse 3.5 with Android SDK
  + 1. **Java**

Java is a [concurrent](http://en.wikipedia.org/wiki/Concurrent_computing),  [object-oriented](http://en.wikipedia.org/wiki/Object-oriented_programming)programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically [compiled](http://en.wikipedia.org/wiki/Compiler) to [bytecode](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 architectu](http://en.wikipedia.org/wiki/Computer_architecture)re.

Most [Android](http://en.wikipedia.org/wiki/Android_(operating_system)) applications are written in [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), there are many differences between the Java API and the Android API, and Android does not use a [Java Virtual Machine](http://en.wikipedia.org/wiki/Java_Virtual_Machine) but another one called [Dalvik](http://en.wikipedia.org/wiki/Dalvik_(software)).

**Java Features**

* Simple
* Object oriented
* Distributed
* Interpreted
* Secure
* Robust
* Architecture-Neutral
* Portable
* High Performance
* Multi-Threaded
* Dynamic
  + 1. **Android**

Android is a [Linux](http://en.wikipedia.org/wiki/Linux)-based [operating system](http://en.wikipedia.org/wiki/Mobile_operating_system) designed primarily for [touchscreen](http://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smartphones](http://en.wikipedia.org/wiki/Smartphone) and [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer). Android was unveiled in 2007 along with the founding of the[Open Handset Alliance](http://en.wikipedia.org/wiki/Open_Handset_Alliance): a consortium of [hardware](http://en.wikipedia.org/wiki/Computer_hardware), [software](http://en.wikipedia.org/wiki/Software)and [telecommunication](http://en.wikipedia.org/wiki/Telecommunication) companies devoted to advancing [open standards](http://en.wikipedia.org/wiki/Open_standard) for mobile devices.Android is [open source](http://en.wikipedia.org/wiki/Open_source) and Google released the code under the [Apache License](http://en.wikipedia.org/wiki/Apache_License). This open source code and permissive licensing allowed the software to be freely modified and distributed by device manufacturers, wireless carriers and enthusiast developers. These factors have allowed Android to become the world's most widely used smartphone platform,overtaking [Symbian](http://en.wikipedia.org/wiki/Symbian) in the fourth quarter of 2010.

Applications are developed in the [Java](http://en.wikipedia.org/wiki/Java_(programming_language)) language using the [Android software development kit](http://en.wikipedia.org/wiki/Android_SDK) (SDK). The SDK includes a comprehensive set of development tools,including a [debugger](http://en.wikipedia.org/wiki/Debugger), [software libraries](http://en.wikipedia.org/wiki/Software_library), a handset[emulator](http://en.wikipedia.org/wiki/Emulator) based on [QEMU](http://en.wikipedia.org/wiki/QEMU), documentation, sample code, and tutorials. The officially supported [Integrated Development Environment](http://en.wikipedia.org/wiki/Integrated_development_environment)  is Eclipseusing

the Android Development Tools (ADT) plugin. Other development tools are available, including a [Native Development Kit](http://en.wikipedia.org/wiki/Android_NDK) for applications or extensions in C or C++.

Android applications run in a [sandbox](http://en.wikipedia.org/wiki/Sandbox_(computer_security)), an isolated area of the system that does not have access to the rest of the system's resources, unless access permissions are explicitly granted by the user when the application is installed.

**Android Features**

* Handset layouts
* Storage
* Connectivity
* Messaging
* Multiple language support
* Web browser
* Java support
* Media support
* Streaming media support
* Additional hardware support
* Multi-touch
* Bluetooth
* Video calling
* Multitasking
* Accessibility
* Voice based features
* Tethering
* Screen capture
  + 1. **SQLite**

**SQLite**  is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) contained in a small [C](http://en.wikipedia.org/wiki/C_(programming_language)) programming [library](http://en.wikipedia.org/wiki/Library_(computer_science)). In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it. SQLite is [ACID](http://en.wikipedia.org/wiki/ACID)-compliant and implements most of the [SQL](http://en.wikipedia.org/wiki/SQL) standard, using a dynamically and weakly typed SQL [syntax](http://en.wikipedia.org/wiki/Syntax) that does not guarantee the [domain integrity](http://en.wikipedia.org/wiki/Integrity_constraints).

SQLite is a popular choice as [embedded database](http://en.wikipedia.org/wiki/Embedded_database) for local/client storage in [application software](http://en.wikipedia.org/wiki/Application_software) such as [web browsers](http://en.wikipedia.org/wiki/Web_browser). It is arguably the most widely deployed [database engine](http://en.wikipedia.org/wiki/Database_engine), as it is used today by several widespread browsers, [operating systems](http://en.wikipedia.org/wiki/Operating_system), and [embedded systems](http://en.wikipedia.org/wiki/Embedded_system), among others. SQLite has many [bindings](http://en.wikipedia.org/wiki/Language_binding) to programming languages.The [source code](http://en.wikipedia.org/wiki/Source_code) for SQLite is in the [public domain](http://en.wikipedia.org/wiki/Public_domain).

Unlike [client–server](http://en.wikipedia.org/wiki/Client%E2%80%93server) database management systems, the SQLite engine has no standalone [processes](http://en.wikipedia.org/wiki/Process_(computing)) with which the application program communicates. Instead, the SQLite [library](http://en.wikipedia.org/wiki/Library_(computing)) is [linked in](http://en.wikipedia.org/wiki/Linker_(computing)) and thus becomes an integral part of the application program. (In this, SQLite follows the precedent of [Informix SE](http://en.wikipedia.org/wiki/IBM_Informix#Other_Products) of [c. 1984](http://www.iiug.org/faqs/informix-faq/ifaq01.htm.1#1.2)) The library can also be called dynamically. The application program uses SQLite's functionality through simple [function calls](http://en.wikipedia.org/wiki/Subroutine), which reduce [latency](http://en.wikipedia.org/wiki/Latency_(engineering)) in database access: function calls within a single process are more efficient than [inter-process communication](http://en.wikipedia.org/wiki/Inter-process_communication). SQLite stores the entire database(definitions, tables, indices, and the data itself) as a single cross-platform [file](http://en.wikipedia.org/wiki/Computer_file) on a host machine. It implements this simple design by [locking](http://en.wikipedia.org/wiki/Lock_(computer_science)) the entire database file during writing. SQLite read operations can be multitasked, though writes can only be performed sequentially.

Several [computer processes](http://en.wikipedia.org/wiki/Computer_process) or [threads](http://en.wikipedia.org/wiki/Thread_(computer_science)) may access the same database concurrently. Several read accesses can be satisfied in parallel. A write access can only be satisfied if no other accesses are currently being serviced. Otherwise, the write access fails with an[error code](http://en.wikipedia.org/wiki/Error_code) (or can automatically be retried until a configurable timeout expires). This concurrent access situation would change when dealing with temporary tables. This restriction is relaxed in version 3.7 when [WAL](http://en.wikipedia.org/wiki/Write-ahead_logging) is turned on enabling concurrent reads and writes.

* 1. **SYSTEM ANALYSIS**

**1.5.1 Introduction**

The overall objective of the systems analysis phase is to understand the proposed system, ensure that it will support business requirement, and build solid foundation of system development.

Gathering requirements is the main attraction of the Analysis Phase. The process of gathering requirements is usually more than simply asking the users what they need and writing their answers down. Depending on the complexity of the application, the process for gathering requirements has a clearly defined process of its own. This process consists of a group of repeatable processes that utilize certain techniques to capture, document, communicate, and manage requirements. This formal process consists of four basic steps.

* **Elicitation** – I ask questions, you talk, I listen.
* **Validation** – I analyze, I ask follow-up questions.
* **Specification**– I document, I ask follow-up questions.
* **Verification** – We all agree.

**1.5.2 System Analysis Activities**

The systems analysis phase includes three main activities:

1. **Requirements Modeling**

Describing the current system and identification of the requirements for the new system which includes

[**Outputs**](javascript:myWindow('outputs');)**:** Electronic or printed information produced by the system.

[**Inputs**](javascript:myWindow('inputs');)**:** Data that enters the system.

[**Processes**](javascript:myWindow('processes');) **:** The logical rules that are applied to transform the data into information.

[**Performance**](javascript:myWindow('performance');) **:** The system characteristics, speed, volume, capacity, a availability, and reliability.

[**Security**](javascript:myWindow('security');) **:**  Hardware, Software, and Procedural controls that protect the system and data from internal or external threats.

**Scalability :** To ensure that the system can support future growth and expansion.

1. **Data and Process Modeling**

Represent graphically system data and process using various techniques.

1. **Development Strategies**

* Evaluation of alternative solutions
* Preparation of the system requirements document, an overall design for the new system

**1.5.3 Functional Requirements**

* **User Lecturer**
  + The lecturer should be able to access all the text documents from micro SD-Card after login.
  + Students are not given access to the application.
* **Behavioral Requirements**

Behavioral requirements of the system are described using use case view.The actors in the use case diagram are the users. A user first registers with new username, password and email id,then login with the registered username and password. User can view the list of files in the main page , Upload the files from the SD Card. The plagio compares the files and display the plagiarised percentage .If it crosses the minimum threshold value, it discards the file from the list or it will add the file to the list.

A use case diagram is a graph of actors, a set of use cases enclosed by system boundary, a communication association between the actors and the use cases, and a generalization among use cases.



**Fig 1.1:Use case Diagram**

Figure 1.1 represents use case diagram. The actors in the use case diagram are the users. A user first registers with new username, password and email id,then login with the registered username and password. User can view the list of files in the main page , Upload the files from the SD Card. The plagio compares the files and display the plagiarised percentage .If it crosses the minimum threshold value, it discards the file from the list or it will add the file to the list.

**1.5.4 Non-Functional Requirements**

* **Capacity**

PLAGIO shall support the documents submitted by the students with micro SDCard support.

* **Safety**

PLAGIO works in offline mode itself therefore is not prone to network failure. It works flawlessly mostly unaffected by failures except for faults like battery shortage.

* **Accessibility**

Any graphical user interfaces of the system shall use adequate font size to be usable by persons with limited visual capacity.

* **Reusability**

It is the ability of the system to be reused. This project can be reused in various Android Versions.

* **Integrity**

It is protection of the system from unauthorized access. This project provide a high degree of integrity.

**1.5.5Feasibility Report**

**Economic Feasibility**

Economic analysis is most frequently used for evaluation of the effectt-

iveness of the system. More commonly known as cost analysis the procedure is to determine the benefit and saving that are expected from system and compare them with costs, decisions is made todesign and implement the system.

**Technical Feasibility**

Technical feasibility centers on the existing manual system of the test ma-

nagement process and to what extent it can support the system.

According to feasibility analysis procedure the technical feasibility of the

system is analyzed and the technical requirements such as softwarefacilities, procedure, inputs are identified. It is also one of the importantphases of the system development activities.

**CHAPTER 2**

**DESIGN**

**2.1 OVERALL SYSTEM ARCHITECTURE**

Yes

Registration and Login

Tablet

Open Application

If micro sd card available

No

SQLite Database

Tablet

Insert sd card

User/Lecturer

Select a file

Compare files

Browse a file from sd card

Touch Sliding

Drawer

Analyse the plagiarised percent

Discard the file

If below the threshold value

Choose a file

No

File added to the folder

Yes

End

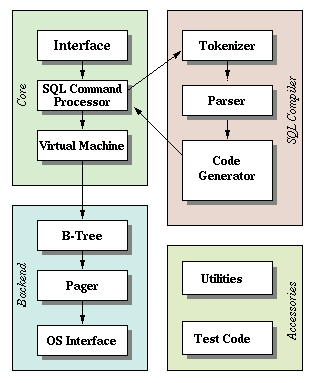
**2.2 ANDROID ARCHITECTURE**

****

**Figure 2.2: Android Architecture**

Android is a [Linux](http://en.wikipedia.org/wiki/Linux)-based [operating system](http://en.wikipedia.org/wiki/Mobile_operating_system) designed primarily for [touch screen](http://en.wikipedia.org/wiki/Touchscreen) mobile devices such as [smart phones](http://en.wikipedia.org/wiki/Smartphone) and [tablet computers](http://en.wikipedia.org/wiki/Tablet_computer). Android was unveiled in 2007 along with the founding of the [Open Handset Alliance](http://en.wikipedia.org/wiki/Open_Handset_Alliance): a consortium of [hardware](http://en.wikipedia.org/wiki/Computer_hardware), [software](http://en.wikipedia.org/wiki/Software)and [telecommunication](http://en.wikipedia.org/wiki/Telecommunication) companies devoted to advancing [open standards](http://en.wikipedia.org/wiki/Open_standard) for mobile devices.

**2.3 SQLITE ARCHITECTURE**

****

**Figure 2.3: SQLite Architecture**

**SQLite**  is a [relational database management system](http://en.wikipedia.org/wiki/Relational_database_management_system) contained in a small [C](http://en.wikipedia.org/wiki/C_(programming_language)) programming [library](http://en.wikipedia.org/wiki/Library_(computer_science)). In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it.

SQLite is a popular choice as [embedded database](http://en.wikipedia.org/wiki/Embedded_database) for local/client storage in [application software](http://en.wikipedia.org/wiki/Application_software) such as [web browsers](http://en.wikipedia.org/wiki/Web_browser). It is arguably the most widely deployed [database engine](http://en.wikipedia.org/wiki/Database_engine), as it is used today by several widespread browsers, [operating systems](http://en.wikipedia.org/wiki/Operating_system), and [embedded systems](http://en.wikipedia.org/wiki/Embedded_system), among others.

SQLite has many [bindings](http://en.wikipedia.org/wiki/Language_binding) to programming languages. The [source code](http://en.wikipedia.org/wiki/Source_code) for SQLite is in the [public domain](http://en.wikipedia.org/wiki/Public_domain).

**2.4 DETAILED DESIGN**

**2.4.1Use case Diagram**

A use case diagram is a graph of actors, a set of use cases enclosed by a system boundary, a communication association between the actors and the use cases, and a generalization among use cases.

****

**Fig 2.4:UseCase Diagram**

Figure 2.4 represents use case diagram. The actors in the use case diagram are the users. A user first registers with new username, password and email id,then login with the registered username and password. User can view the list of files in the main page , Upload the files from the SD Card. The plagio compares the files and display the plagiarised percentage .If it crosses the minimum threshold value, it discards the file from the list or it will add the file to the list.

**2.4.2 Class Diagram**

A class diagram in the Unified Modeling Language(UML) is a type of static structure diagram that describes the structure of a system by showing the systems classes, their attributes, and the relationship between the classes. The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed.

****

**Fig 2.5:Class Diagram**

Figure 2.5 represents the class diagram. It shows the classes login, database, user account, file array adapter, files list. In login class username and password acts as attributes and validate is used as method. In fileslist class item, path, listadapter, contentview acts as attributes and browse,compare considered as methods.

**2.4.3Activity Diagram**

An activity diagram is a variation or special case of a state machine, in which the states are the activities representing the performance of operations and the transitions are triggered by the completion of operation.



**Fig 2.6: Activity Diagram**

Figure 2.6,represents the activity diagram.The activities mentioned are registration, login, uploading files, comparing the files, viewing the plagiarised percentage by the user and finally adding the file or discarding the file.

**2.4.4 Sequence Diagram**

A sequence diagram in Unified Modelling Language(UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence chart. Sequence diagrams are sometimes called Event-trance diagrams, event scenarios and timing diagram.

**Fig 2.7: Sequence Diagram**

Figure 2.7,shows the sequence diagram. User, plagio, and database are the sequences used in this diagram. Database stores the registration and login details. Plagio compares the files, displays the plagiarised percentage and adds the file based on the condition. User uploads the file and view the results.

**CHAPTER 3**

**IMPLEMENTATION AND TESTING**

**3.1 IMPLEMENTATION**

**3.1.1 Registration**

This module contains a registration form where new users can enter details like Name , Password and Email id. Only a registered user can use this application.

**Algorithm:**

**Step1:** Start

**Step2:** Click on the Register button.

**Step3:** Enter the Username and the Password twice for confirmation.

**Step4:** Enter valid Email address and click Save button.

**Step5:** Stop.

**3.1.2 User Login**

This module contains a login form where registered users can login with their respective UserName and Password. Authentication is carried out with the help of the SQLite.

**Algorithm:**

**Step1:** Start.

**Step2:** Enter valid Username and Password.

**Step3:** Click Login button to view the homepage.

**Step4:** Stop.

**3.1.3 Upload files**

In this module, the user is allowed to upload the files which are to be compared by clicking the BROWSE button. Any file can be selected from the SD Card of the Device used.

**Algorithm:**

**Step1:** Start

**Step2:** Click on the Browse button.

**Step3:** Select the desired file which is to be compared.

**Step4:** The file gets uploaded.

**Step5:** Stop.

**3.1.4 Comparison and Result**

This module displays the contents of the files which are to be compared. It also displays a fileslist which will include the files present in the SD-Card from which the user selects a file which is to be compared with the uploaded file. The similarity percentage is displayed by clicking the COMPARE button.

If the percentage crosses a minimum threshold value, it is inferred that the file is plagiarised and thus discarded. Else it is added to the fileslist as “a genuine original file”.

**Algorithm:**

**Step1:** Start.

**Step2:** After selecting the file click the Compare button.

**Step3:** The plagiarised percentage is displayed as a prompt message.

**Step4:** The file is added or discarded depending upon the threshold value.

**Step5:** Stop

**3.2 TESTING**

**Software testing** is an investigation conducted to provide stakeholders with information about the quality of the product or service under test.Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include, but are not limited to, the process of executing a program or application with the intent of finding[software bugs](http://en.wikipedia.org/wiki/Software_bug) (errors or other defects).

Software testing, depending on the testing method employed, can be implemented at any time in the development process. Traditionally most of the test effort occurs after the requirements have been defined and the coding process has been completed, but in the [Agile](http://en.wikipedia.org/wiki/Agile_software_development) approaches most of the test effort is on-going. As such, the methodology of the test is governed by the chosen software development methodology.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as[Agile](http://en.wikipedia.org/wiki/Agile_software_development), often employ [test-driven development](http://en.wikipedia.org/wiki/Test-driven_development) and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers.

In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

Software testing can be stated as the process of validating and verifying that a computer program/application/product:

* Meets the requirements that guided its design and development,
* Works as expected,
* Can be implemented with the same characteristics.

**3.2.1Test Plan**

The project is tested to verify its correctness and efficiency. The test plan includes following test cases:

|  |  |  |
| --- | --- | --- |
| **Test Case** | **Expected Output** | **Actual Output** |
| E-mail id verification in registration form | Should accept all numeric, special characters and symbols | Accepts numerical characters but displays a red warning symbol for certain special characters.(eg :\_,$) |
| Login Validation | Only registered users should have access to Plagio. | If any unregistered details are encountered , a prompt message is popped indicating invalid username and password. |
| Display of percentage | After the file is added the similarity percentage should be stored and displayed. | The percentage is displayed only as a prompt while the files are being compared and is not stored. |
| SD Card | The entire process takes place only if the device has a SD Card. | If there is no SD Card then Plagio stops with an appropriate prompt. |

**Table 3.1: Test Plan**

**CHAPTER 4**

**CONCLUSION AND FUTURE WORK**

**4.1 CONCLUSION**

Thus this project provides a handy tool for the lecturers to ease their evaluation methods and saves a lot of manual work while being accurate and fast. Our techniques for detection are more efficient (in terms of delay and memory consumption) and do not rely on any special hardware.

**4.2 FUTURE WORK**

Plagio can be further enhanced in future by extending the comparison of the files by automatically collecting contents related to the concerned topic from the internet. So that even the contents copied from the web can also be identified and detected. However, it will not remain an offline application and will require an internet connection.

**APPENDIX A**

**SAMPLE CODING**

**MAIN.XML**

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

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

android:background="@drawable/splashscreen" android:layout\_width="wrap\_content" android:layout\_height="match\_parent">

<ProgressBar android:id="@+id/myprogressbar01"

android:progress="50" android:secondaryProgress="0"

android:layout\_centerHorizontal="true"

android:layout\_alignParentBottom="true"

style="?android:attr/progressBarStyleHorizontal" android:layout\_width="fill\_parent"

android:layout\_marginLeft="10dp" android:layout\_marginRight="10dp"

android:layout\_marginBottom="30dp" android:layout\_height="10dp"

android:progressDrawable="@drawable/myprogressbar"

></ProgressBar>

<TextView android:id="@+id/textView1" android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:textSize="15dip"

android:gravity="center"

android:textColor="#FFFF00"

android:maxHeight="80dp"

android:text="LOADING.."

android:layout\_marginBottom="80dp"

android:layout\_alignBottom="@layout/main" android:layout\_centerInParent="true">

</TextView>

</RelativeLayout>

**SPLASHSCREEN.JAVA**

package com.example.plagio;

import com.example.plagiarism.R;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.os.Handler;

import android.widget.ProgressBar;

publicclass Splashscreen extends Activity {

private ProgressBar progressBar;

private Handler handler;

protectedint progressStatus;

/\*\* Called when the activity is first created. \*/

@Override

publicvoid onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

progressBar = (ProgressBar) findViewById(R.id.myprogressbar01);

handler = new Handler();

new Thread(new Runnable() {

publicvoid run() {

while (progressStatus <= 10) {

// Update the progress bar

handler.post(new Runnable() {

publicvoid run() {

progressBar.setProgress(progressStatus \* 10);

}

});

try {

Thread.sleep(500);

}catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

progressStatus++;

}

startActivity(new Intent(getApplicationContext(), login.class));

finish();

}

}).start();

}

}

**REGISTRATION.XML**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<RelativeLayout

android:id="@+id/rl3"

android:layout\_width="fill\_parent"

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

<RelativeLayout

android:id="@+id/rl4"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/relativeLayout3"

android:layout\_marginTop="5dp"

android:background="#ffffff">

<TextView

android:id="@+id/pas\_tv1"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:background="@drawable/textview"

android:gravity="center"

android:text="Password"

android:textColor="#000000">

</TextView>

<EditText

android:id="@+id/pas\_et2"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/pas\_tv1"

android:password="true">

</EditText>

</RelativeLayout>

<RelativeLayout

android:id="@+id/relativeLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp">

<EditText

android:id="@+id/et1"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/tv1">

</EditText>

<TextView

android:id="@+id/tv1"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:background="@drawable/textview"

android:gravity="center"

android:text="UserName"

android:textColor="#000000">

</TextView>

</RelativeLayout>

<RelativeLayout

android:id="@+id/rl5"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/rl4">

<TextView

android:id="@+id/cp\_tv1"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_toLeftOf="@+id/cp\_tv1"

android:background="@drawable/textview"

android:gravity="center"

android:text="Confirm Password"

android:textColor="#000000">

</TextView>

<EditText

android:id="@+id/cp\_et1"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/cp\_tv1"

android:password="true">

</EditText>

<RelativeLayout

android:id="@+id/rl6"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/rl5"

android:layout\_marginTop="50dp">

<TextView

android:id="@+id/mail\_tv1"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:background="@drawable/textview"

android:gravity="center"

android:text="E-Mail Id"

android:textColor="#000000">

</TextView>

<EditText

android:id="@+id/email\_et1"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/mail\_tv1"

android:text="">

</EditText>

<RelativeLayout

android:id="@+id/rl7"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignLeft="@+id/email\_et1"

android:layout\_below="@+id/email\_et1">

<Button

android:id="@+id/save\_bt1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20dp"

android:text="Save"

android:textColor="#000000">

</Button>

<Button

android:id="@+id/hme\_bt2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:layout\_alignTop="@+id/save\_bt1"

android:layout\_marginRight="26dp"

android:text="Home" />

</RelativeLayout>

</RelativeLayout>

</RelativeLayout>

</RelativeLayout>

</ScrollView>

**REGISTRATION.JAVA**

package com.example.plagio;

import java.util.ArrayList;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import com.example.plagiarism.R;

import android.os.Bundle;

import android.app.Activity;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

publicclass registration extends Activity {

publicstatic String regname, regpass, regemail, rname, rpass, cnfrm,

email;

//public static ArrayList<String> username = new ArrayList<String>();

//public static ArrayList<String> password = new ArrayList<String>();

EditText username, pass, conpass, mai;

Button save, hme;

privateboolean isFound = false;

publicvoid onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.registration);

username = (EditText) findViewById(R.id.et1);

pass = (EditText) findViewById(R.id.pas\_et2);

conpass = (EditText) findViewById(R.id.cp\_et1);

mai = (EditText) findViewById(R.id.email\_et1);

save = (Button) findViewById(R.id.save\_bt1);

save.setOnClickListener(new OnClickListener() {

publicvoid onClick(View v) {

SharedValue.db = null;

SharedValue.TableName = "reg";

try {

SharedValue.db = getApplicationContext()

.openOrCreateDatabase("RegisterDatabase",

MODE\_PRIVATE, null);

SharedValue.db.execSQL(" DROP TABLE IF EXISTS TableName ");

String un = username.getText().toString();

String pa = pass.getText().toString();

String paa = conpass.getText().toString();

String cp = conpass.getText().toString();

String cpp = conpass.getText().toString();

String ma = mai.getText().toString();

Pattern ptn = Pattern

.compile("[a-zA-Z.\_%+-]\*.[a-zA-Z]\*[0-9]\*@[a-zA-Z]\*.[a-zA-Z]\*");

Matcher m = ptn.matcher(ma);

// Matcher m=p.matcher(args[0]);

boolean bln = m.matches();

if (TextUtils.isEmpty(un) || un.length() < 0

&&TextUtils.isEmpty(pa) || pa.length() < 0

&&TextUtils.isEmpty(cp) || cp.length() < 0

&&TextUtils.isEmpty(ma) || ma.length() < 0 || (bln==false)

) {

username.setError("Required Fields");

pass.setError("Required Fields");

conpass.setError("Required Fields");

mai.setError("Required Fields or invalid email id");

}

elseif(pa.equals(paa)){

SharedValue.db

.execSQL(" CREATE TABLE IF NOT EXISTS "

+ SharedValue.TableName

+ "(Field1 VARCHAR PRIMARY KEY,Field2 VARCHAR not null,Field3 VARCHAR NOT NULL,Field4 VARCHAR NOT NULL);");

String encpa = SimpleCrypto.encrypt("Password", pa);

String encconpa = SimpleCrypto.encrypt("Password", pa);

SharedValue.db.execSQL(" INSERT INTO "

+ SharedValue.TableName

+ "(Field1,Field2,Field3,Field4)" + " VALUES('"

+ username.getText().toString() + "','" + encpa

+ "','" + conpass.getText().toString() + "','"

+ mai.getText().toString() + "');");

Log.i("TAG", "pass encry" + encpa);

Log.i("TAG", "con pass encry" + encconpa);

String decrpa = SimpleCrypto.decrypt("Password", encpa);

Log.i("TAG", "pass decryption" + decrpa);

String decrconpa = SimpleCrypto.decrypt("Password",

encconpa);

Log.i("TAG", "confirm pass decryption" + decrpa);

String TAG = null;

Toast.makeText(getApplicationContext(),

"Registered Successfully", Toast.LENGTH\_SHORT)

.show();

Intent my = newIntent(getApplicationContext(),

login.class);

startActivity(my);

}

else{

Toast.makeText(getApplicationContext(), "password not matched", 0).show();

}

} catch (Exception e) {

Toast

.makeText(getApplicationContext(),

"Already same Username Exist !",

Toast.LENGTH\_SHORT).show();

e.printStackTrace();

} finally {

if (SharedValue.db != null)

SharedValue.db.close();

}

}

privatevoid showToast(String data) {

// TODO Auto-generated method stub

Toast.makeText(getApplicationContext(), "SAVED",

Toast.LENGTH\_SHORT).show();

}

});

hme = (Button) findViewById(R.id.hme\_bt2);

hme.setOnClickListener(new OnClickListener() {

publicvoid onClick(View v) {

Intent my = newIntent(getApplicationContext(),

login.class);

startActivity(my);

}

});

}

}

**LOGIN.XML**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<RelativeLayout

android:id="@+id/relativeLayout2"

android:layout\_width="match\_parent"

android:layout\_height="120dp"

android:background="@drawable/plagiarism">

<TextView

android:id="@+id/login\_tit"

android:layout\_width="wrap\_content"

android:layout\_height="60dp"

android:text=""

android:textColor="#356aab">

</TextView>

</RelativeLayout>

<ScrollView

android:id="@+id/scrollView1"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#ffffff">

<RelativeLayout

android:id="@+id/rl3"

android:layout\_width="wrap\_content"

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

<RelativeLayout

android:id="@+id/rl4"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:layout\_below="@+id/relativeLayout3"

android:layout\_marginTop="10dp"

android:background="#ffffff">

<TextView

android:id="@+id/pas\_tv1"

android:layout\_width="80dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:background="@drawable/textview"

android:gravity="center"

android:text="Password"

android:textColor="#000000">

</TextView>

<EditText

android:id="@+id/pas\_et2"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/pas\_tv1"

android:password="true">

</EditText>

<Button

android:id="@+id/reg\_bt2"

android:layout\_width="60dp"

android:layout\_height="50dp"

android:layout\_alignBaseline="@+id/pas\_et2"

android:layout\_alignBottom="@+id/pas\_et2"

android:layout\_alignParentRight="true"

android:layout\_marginRight="28dp"

android:background="@drawable/regbtn"

android:textColor="#ffffff" />

</RelativeLayout>

<RelativeLayout

android:id="@+id/relativeLayout3"

android:layout\_width="wrap\_content"

android:layout\_height="fill\_parent"

android:layout\_marginTop="30dp">

<EditText

android:id="@+id/et1"

android:layout\_width="100dp"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:layout\_marginLeft="10dp"

android:layout\_marginTop="50dp"

android:layout\_toRightOf="@+id/tv1">

</EditText>

<TextView

android:id="@+id/tv1"

android:layout\_width="80dp"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginTop="30dp"

android:background="@drawable/textview"

android:gravity="center"

android:text="UserName"

android:textColor="#000000">

</TextView>

<Button

android:id="@+id/button1"

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_alignParentRight="true"

android:layout\_alignTop="@+id/tv1"

android:layout\_marginRight="19dp"

android:background="@drawable/logbtn"

android:textColor="#ffffff" />

</RelativeLayout>

</RelativeLayout>

</ScrollView>

</LinearLayout>

**LOGIN.JAVA**

package com.example.plagio;

import com.example.plagiarism.R;

import android.app.Activity;

import android.content.Intent;

import android.content.SharedPreferences;

import android.database.Cursor;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

publicclass login extends Activity {

String u, p;

publicstaticfinal String PREF\_FILE\_NAME = "PrefFile";

publicstaticfinal String KEY\_PRIVATE = "KEY\_PRIVATE";

@Override

publicvoid onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.login);

final EditText txtUserName = (EditText) findViewById(R.id.et1);

final EditText txtPassword = (EditText) findViewById(R.id.pas\_et2);

Button btnLogin = (Button) findViewById(R.id.button1);

btnLogin.setOnClickListener(new OnClickListener() {

privateint storedPreference;

int i=0;

publicvoid onClick(View v) {

String username = txtUserName.getText().toString();

String password = txtPassword.getText().toString();

try {

u = txtUserName.getText().toString();

p = txtPassword.getText().toString();

if (TextUtils.isEmpty(u) || u.length() < 0

&& (TextUtils.isEmpty(p)) || p.length() < 0) {

txtUserName.setError("Required fields");

txtPassword.setError("Required fields");

} else {

SharedValue.db = getApplicationContext()

.openOrCreateDatabase("RegisterDatabase",

MODE\_PRIVATE, null);

Cursor c = SharedValue.db.rawQuery(

"SELECT \* FROM reg;",null);

int row = c.getCount();

int Column1 = c.getColumnIndex("Field1");

int Column2 = c.getColumnIndex("Field2");

c.moveToFirst();

if (c != null) {

// Loop through all Results

do {

// String Name =

// c.getString(c.getColumnIndex("Field1"));

String Name = c.getString(Column1);

String PassWord = c.getString(Column2);

String decrpa = SimpleCrypto.decrypt(

"Password", PassWord);

if ((username.equals(Name))

&& (password.equals(decrpa))) {

i++;

Toast.makeText(login.this,"LogIn Successfully",Toast.LENGTH\_SHORT).show();

Intent my = newIntent(

getApplicationContext(),

FilesList.class);

startActivity(my);

SharedPreferences preferences = getSharedPreferences(

PREF\_FILE\_NAME, MODE\_PRIVATE);

SharedPreferences.Editor editor = preferences.edit();

editor.putString(KEY\_PRIVATE, txtUserName.getText().toString());

editor.commit();

String val="value";

preferences.getString(KEY\_PRIVATE, val);

if(val.equals(""))

{

Toast.makeText(getApplicationContext(), "invalid username and password", 0).show();

}

else

{

Intent ma=newIntent(getApplicationContext(),FilesList.class);

startActivity(ma);

}

}else{

}

} while (c.moveToNext());

}

if(i==0){

Toast.makeText(getApplicationContext(), "invalid username and password", 0).show();

}

//Toast.makeText(login.this, "Inavalid username and password", Toast.LENGTH\_LONG).show();

}

} catch (Exception e) {

//Toast.makeText(getApplicationContext(), "Invalid username and password", 0).show();

//Log.e("error", "error occured");

}finally{

i=0;

}

}

privatevoid showToast(String data) {

// TODO Auto-generated method stub

//Toast.makeText(getApplicationContext(), "SAVED",Toast.LENGTH\_SHORT).show();

}

});

Button reg = (Button) findViewById(R.id.reg\_bt2);

reg.setOnClickListener(new OnClickListener() {

publicvoid onClick(View v) {

// TODO Auto-generated method stub

Intent my = newIntent(getApplicationContext(),registration.class);

startActivity(my);

}

});

}

@Override

publicvoid onBackPressed() {

super.onBackPressed();

Intent setIntent = newIntent(Intent.ACTION\_MAIN);

setIntent.addCategory(Intent.CATEGORY\_HOME);

startActivity(setIntent);

}

}

**SECONDACTIVITY.XML**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content">

<LinearLayout

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical">

<ImageView

android:id="@+id/imageView1"

android:layout\_width="351dp"

android:layout\_height="wrap\_content"

android:src="@drawable/plagiarism" />

<TextView

android:id="@+id/path"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Browse to upload files"

android:textStyle="bold" />

<EditText

android:id="@+id/editText2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="45dp"

android:ems="20">

<requestFocus />

</EditText>

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Browse" />

<Button

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Compare" />

<EditText

android:id="@+id/editText1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="60dp"

android:ems="10" />

<ListView

android:id="@android:id/list"

android:layout\_width="fill\_parent"

android:layout\_height="201dp"

android:layout\_marginTop="35dp">

</ListView>

</LinearLayout></ScrollView>

**FILESLIST.JAVA**

package com.example.plagiarism;

import java.io.BufferedReader;

import java.io.BufferedWriter;

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.IOException;

import java.util.ArrayList;

import java.util.List;

import java.util.Random;

import org.apache.commons.io.FileUtils;

import com.example.plagiarism.R;

import android.app.AlertDialog;

import android.app.ListActivity;

import android.content.Intent;

import android.os.Bundle;

import android.os.Environment;

import android.util.Log;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ListView;

import android.widget.TextView;

import android.widget.Toast;

public class FilesList extends ListActivity

{

File file = null;

private List<String> item = null;

private List<String> path = null;

// private ListView filelist;

// private ArrayAdapter<String> listAdapter ;

private TextView myPath;

private String root;

private EditText contentview, entertext;

String entercontent;

Button compare;

StringBuilder text = null;

String strline = null;

int value = 0;

String[] enteredtextwords = null;

String content = null;

int i=0;

int k=0;

@Override

protected void onCreate(Bundle savedInstanceState)

{

// TODO Auto-generated method stub

super.onCreate(savedInstanceState);

setContentView(R.layout.secondactivity);

contentview = (EditText) findViewById(R.id.editText1);

entertext = (EditText) findViewById(R.id.editText2);

compare = (Button) findViewById(R.id.button1);

Button browse = (Button) findViewById(R.id.button2);

try

{

String textpath = getIntent().getExtras().getString("Title");

entertext.setText(textpath);

}

catch (Exception e)

{

e.printStackTrace();

}

myPath = (TextView) findViewById(R.id.path);

root = Environment.getExternalStorageDirectory().getPath()

+ "/plagiarism";

getDir(root);

browse.setOnClickListener(new OnClickListener()

{

@Override

public void onClick(View v)

{

// TODO Auto-generated method stub

Intent intent = new Intent(FilesList.this, FileChooser.class);

intent.putExtra("Title", "");

startActivity(intent);

}

});

compare.setOnClickListener(new OnClickListener()

{

@Override

public void onClick(View v)

{

// TODO Auto-generated method stub

String strLine = null;

// entered text in the text box

entercontent = entertext.getText().toString();

Log.i("ramya", entercontent);

// file reading from the sd card

try

{

FileReader fReader = new FileReader(file);

BufferedReader bReader = new BufferedReader(fReader);

int a = 0;

int b = 0;

while ((strLine = bReader.readLine()) != null)

{

Log.i("ramya", "filecontentcheck" + strLine);

enteredtextwords = entercontent.split(" ");

// String str= new String(strLine);

for (int i = 0; i < enteredtextwords.length; i++)

{

if (strLine.contains(enteredtextwords[i]))

{

a++;

}

else if (!strLine.contains(enteredtextwords[i]))

{

b++;

Log.i("ramya",

"count ......... inside if bblock try of the unmatword..."

+ b);

}

}

int divident = a + b;

value = (int) (((double) a / (double) divident) \* 100);

Toast.makeText(getApplicationContext(),

"matched word" + value + " %",

Toast.LENGTH\_SHORT).show();

Log.i("ramya", "count try of the matched word..." + a);

Log.i("ramya", "count try of the unmat word..." + b);

}

if (value <= 55) {

int j=0;

Random random = new Random();

int r = random.nextInt();

j=1;

if(j==1){

ContentChecking(entercontent);

}

System.out.println("random values:............" + r);

Log.i("ramya", "inside file creation......."+ entertext.getText().toString());

}

else

{

Toast.makeText(getApplicationContext(),

"it contains large copy right contents",

Toast.LENGTH\_LONG).show();

}

}

catch (Exception e)

{

e.printStackTrace();

}

}

public void ContentChecking(String entercontent) throws IOException

{

i=1;

if(i==1){

File fileDirectory = new File(Environment

.getExternalStorageDirectory().getPath()

+ "/plagiarism");

String fileOutput = null;

StringBuffer st = new StringBuffer();

File[] dirFiles = fileDirectory.listFiles();

if (dirFiles.length != 0)

{

// loops through the array of files, outputing the name to

// console

/\* for (int ii = 0; ii <= dirFiles.length; ii++)

{\*/

//File list = dirFiles[ii];

Log.i("sathya", "inside for looppp..........");

for (File file : dirFiles) {

content = FileUtils.readFileToString(file);

if(entercontent.contains(content)){

Toast.makeText(getApplicationContext(), "already this file is added in sdcard", Toast.LENGTH\_SHORT).show();

k=2;

Log.i("sathya", "if..........loop:"+content);

}else{

k++;

Log.i("sathya", "else............loop:"+content);

if(k==1){

contentAdd();

Log.i("sathya", "content added........:");

}

}

}

}

}

}

});

}

public void contentAdd(){

Log.i("sathya", "inside of content add method");

try {

Random random = new Random();

int r = random.nextInt();

FileWriter fs = new FileWriter(root + "/plagiarism"

+ r + ".txt");

BufferedWriter out = new BufferedWriter(fs);

out.append(entertext.getText().toString());

out.close();

Toast.makeText(getApplicationContext(),

"file added successfully",

Toast.LENGTH\_LONG).show();

} catch (Exception e) {

e.printStackTrace();

}

}

private void getDir(String dirPath) {

// TODO Auto-generated method stub

{

myPath.setText("Location: " + dirPath);

item = new ArrayList<String>();

path = new ArrayList<String>();

File f = new File(dirPath);

File[] files = f.listFiles();

if (!dirPath.equals(root)) {

item.add(root);

path.add(root);

item.add("../");

path.add(f.getParent());

}

for (int i = 0; i < files.length; i++) {

File file = files[i];

if (!file.isHidden() && file.canRead()) {

path.add(file.getPath());

if (file.isDirectory()) {

item.add(file.getName() + "/");

} else {

item.add(file.getName());

}

}

ArrayAdapter<String> fileList = new ArrayAdapter<String>(this,

R.layout.row, item);

setListAdapter(fileList);

Log.i("ramya", dirPath);

}

}

}

protected void onListItemClick(ListView l, View v, int position, long id) {

// TODO Auto-generated method stub

file = new File(path.get(position));

String strLine = "";

text = new StringBuilder();

if (file.isDirectory()) {

if (file.canRead()) {

getDir(path.get(position));

} else {

new AlertDialog.Builder(this)

.setIcon(R.drawable.ic\_launcher)

.setTitle(

"[" + file.getName()

+ "] folder can't be read!")

.setPositiveButton("OK", null).show();

}

} else {

/\*

\* new AlertDialog.Builder(this) .setIcon(R.drawable.ic\_launcher)

\* .setTitle("[" + file.getName() + "]") .setPositiveButton("OK",

\* null).show();

\*/

try {

FileReader fReader = new FileReader(file);

BufferedReader bReader = new BufferedReader(fReader);

/\*\* Reading the contents of the file , line by line \*/

while ((strLine = bReader.readLine()) != null) {

text.append(strLine + "\n");

}

} catch (Exception e) {

e.printStackTrace();

}

}

contentview.setText(text);

Log.i("ramya", "content of file:" + text);

}

}

**APPENDIX B**

**OUTPUT SCREENSHOTS**

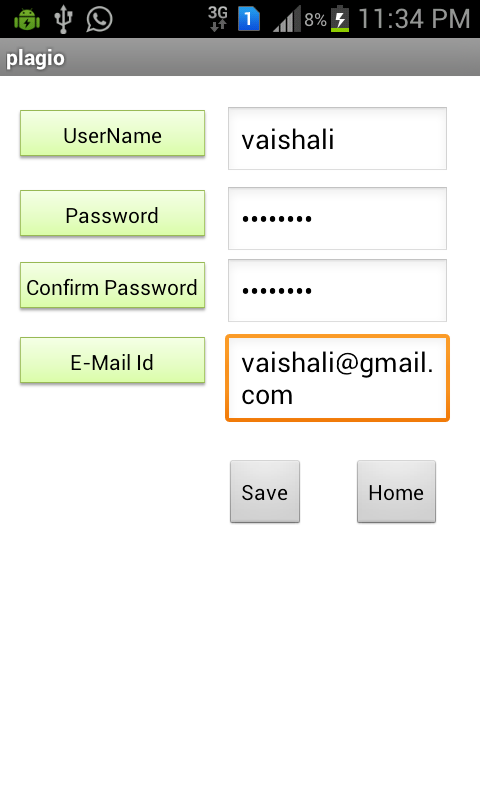
**LOADING PAGE**

****

**Figure b.1: Loading Page**

The above screenshot represents the Loading Page. When Plagio is clicked, the loading page appears for about 10seconds and the application starts.

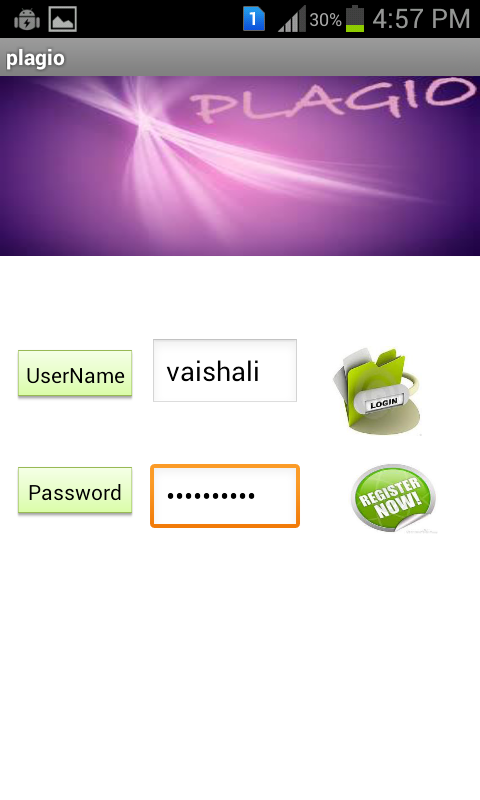
**REGISTRATION PAGE**

****

**Figure b.2: Registration Page**

The above screenshot represents the Registration Page. New users can enter details like Name , Password and Email id. Only a registered user can use this application.

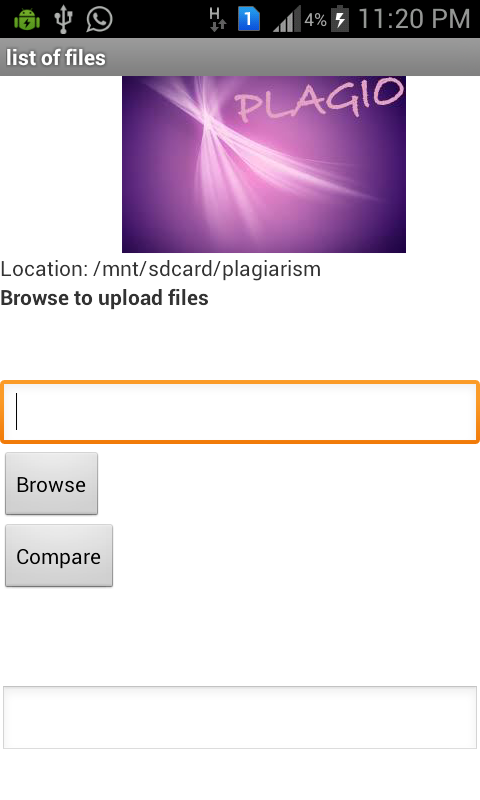
**LOGIN PAGE**

****

**Figure b.3: Login Page**

The above screenshot represents the Login Page.Users can login with their respective UserName and Password. Authentication is carried out with the help of the SQLite.

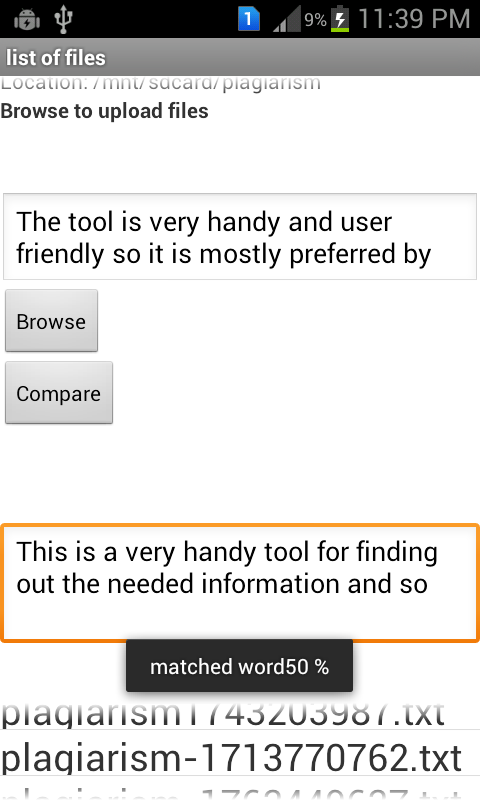
**MAIN PAGE**

****

**Figure b.4: Main Page**

The above screenshot represents the Main Page. This Page contains BROWSE button with the help of which the desired file can be selected and COMPARE button for comparison of files.

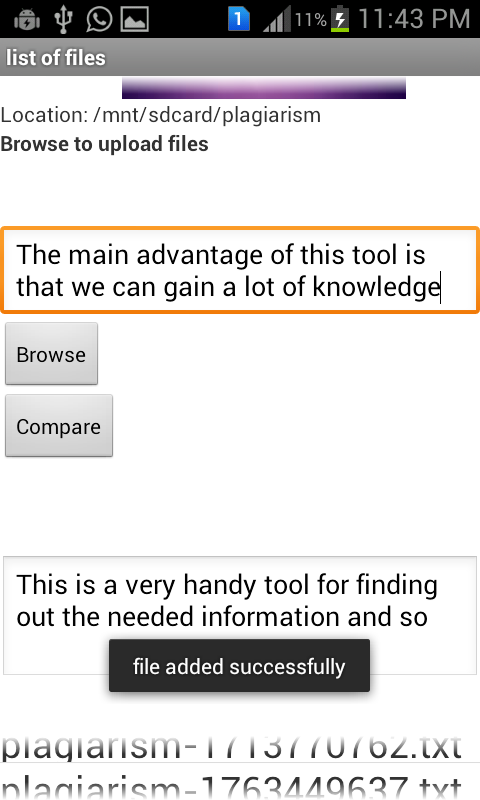
**COMPARISON**

****

**Figure b.5: Comparison**

The above screenshot represents the activity of Compare button where the files are compared and plagiarized percentage is displayed.

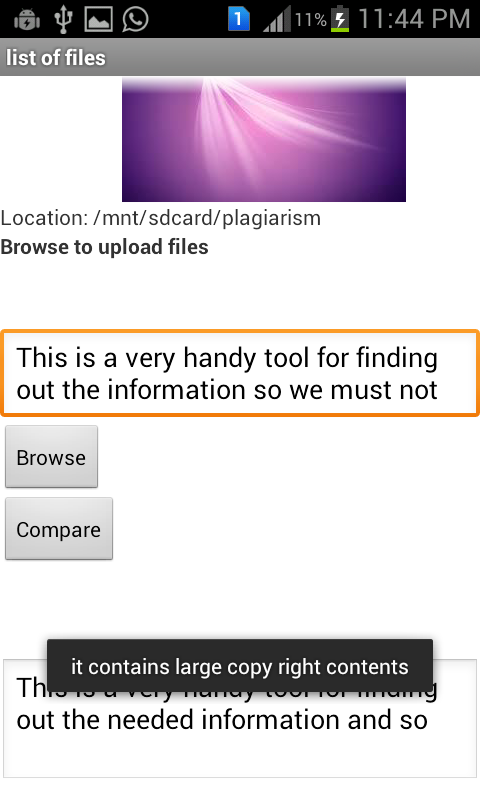
**FILE ADDED**

****

**Figure b.6: File Added**

The above screenshot displays a prompt “file added successfully” when the genuine file is added to the list.

**FILE DISCARDED**

****

**Figure b.7: File Discarded**

The above screenshot displays a prompt “it contains large copy right contents” when the particular file is plagiarized with the files in the list

**REFERENCES**

1. Understanding Plagiarism Linguistic Patterns, Textual Features and

Detection Methods-Salha M. Alzahrani, Naomie Salim, and Ajith Abraham*, Senior Member, IEEE.*

1. <http://www.android.com>
2. <http://developer.android.com>
3. Aakash user manual, Aakash Lab, IIT Bombay.
4. Using design patterns in UML –

**link**-<http://www.developer.com/design/article.php/3309461/Using-Design-Patterns-in-UML.htm>- by [Mandar Chitnis, Pravin Tiwari, & Lakshmi Ananthamurthy](http://www.developer.com/author/Mandar-Chitnis-Pravin-Tiwari-Lakshmi-Ananthamurthy-73080.htm) (posted on Feb 6 2004).