A FILE EXPLORER APP FOR ANDROID DEVICES -

5l XplorR

A MINI PROJECT REPORT

Submitted by

BALAJI .R (211418104034)

DARIN JOSHUA .D (211413104043)

GLADWIN JOSEPH SOLOMON .B (211413104064)

ISAAC .G (211413104091)

JEBASTIN STEPHEN SARJIN JOHN (211418104098)

JOEL NITHISH KUMAR. M (211418104102)

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI

ANNA UNIVERSITY :: CHENNAI 600 025

JUNE 2021

BONAFIDE CERTIFICATE

Certified that this mini project report **"File Explorer App for Android Devices- 5l XplorR"** is the bonafide work of "**BALAJI .R (211418104034), DARIN JOSHUA .D (211418104043), GLADWIN JOSEPH SOLOMON .B (211418104064), ISAAC .G (211418104091), JEBASTIN STEPHEN SARJIN JOHN (211418104098), JOEL NITHISH KUMAR (211418104102)",** who carried out the project work under my supervision.

SIGNATURE
Dr. S. MURUGAVALLI
HEAD OF THE DEPARTMENT
Professor & Head
Department of CSE,
Panimalar Engineering College,
Poonamallee, Chennai - 600 123.

SIGNATURE
C.THYAGARAJAN
SUPERVISOR
Assistant Professor
Department of CSE,
Panimalar Engineering College,
Poonamallee, Chennai - 600 123.

ABSTRACT

These days Device storage management is a major issue for all Smart Phone users as we get limited internal storage which is either covered by multimedia or other apps which occupy most of the available space along with the OS. Android allows every user to access their device's file system using a file manager. They can access Android files and transfer them between a computer and a smartphone or tablet. This is performed by simply plugging in a USB cable in an Android device and firing up your desktop file manager. Android's file manager app reacts as an important part of the software on a device. It provides facilities to browse files, manage storage space, downloads, move files around, and a lot of other activities. However, several Android device manufacturers provide a pre-install Android file manager app on their devices. The pre-install file manager contains fewer features as compared to other third-party file managers. Security is also a major concern for storing any important files or documents in any android device as the data which are stored in any device is not safe as it is not hard to hack a phone or attack a device to destroy/damage/steal any data; In any situation the user loses the files. So, we introduce **5l XplorR** which solves the current problems in a different approach which means you can store any type of file securely on your device without fearing about any loss or theft of data. This is an advanced system which provides the facility to store and retrieve the files without any duplication or if there any attempt to access the data without the knowledge of the user, it alerts the user for further action. In case of large amount of files, the user can take the advantage of manual search or text to speech feature to the system to filter out certain files. Not everyone is too keen on file organization because it can be pretty boring, but it still must be occasionally done. 5l XplorR is open source and focuses on a lighter experience for those who just need to do some light file browsing. It manages to include the most important stuff without feeling bloated. Above everything else, it is free to download.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ADT	-	Android Development Tools
ALG	-	Algorithm
AWT	-	Abstract Window Toolkit
API	-	Application Program Interface
APK	-	Android Application Package
APP	-	Application
CPU	-	Central Processing Unit
DB	-	Database
EMU	-	Emulator
EPL	-	Eclipse Public License
GUI	-	Graphical User Interface
HTML	-	Hyper Text Markup Language
IDE	-	Integrated Development Environment
JDK	-	Java Development Kit
JDT	-	Java Development Tools
MAL	-	Malware
MSG	-	Message
OS	-	Operating System
OSGI	-	Open Services Gateway Initiative
PDA	-	Personal Digital Assistant
PMS	-	Permissions
PDE	-	Plug-in Developer Environment
SDK	-	Software Development Kit
SWT	-	Standard Widget Toolkit

1. INTRODUCTION

This chapter deals with the general introduction, existing system and the proposed system of this project. General gives a broad introduction about the project. Existing system explains the current system and its limitations and the proposed system provides a solution to overcome those limitations.

1.1 GENERAL

The number of smartphone users and mobile applications are growing rapidly. According to a recent report, 45 million people in the U.S. own smartphones and 234 million people subscribe to the mobile phone application stores. Because thousands of application developers construct many kinds of applications for the android platform, users can easily enjoy their individual smartphone phone lifecycle. But there is one major issue in using the apps and other files, we have a storage limitation which forces the users to manage the available space effectively. This 5l XplorR application takes care of the storage issue by helping the users view the files and apps systematically and manage them.

1.2 OBJECTIVE OF THE PROJECT

The application keeps the user files safe and secured. It allows the users to search a file by a search box. Also maintains the files in a categorized and in a structured manner. It avoids the data duplication which prevents re-writing the similar file. In this app we maintain the data in an efficient manner. Users usually create or have many files, folders, this app makes it easy to port files from one place to another place. In this application we can even share files from one device to another device. If this device has its own application that is required to open a particular document, we can open it from the 5l XplorR app. The main purpose of this application is to provide security to the user data while keeping the app user friendly.

1.3 EXISTING SYSTEM

The built-in file manager provided in any android device is minimal and barebones, but it has all the basic features you'll need; unless you need to access network storage locations or gain access to the root file system, which are more advanced features better left to third-party apps.

- The built-in file manager does not provide the required security to the files and sometimes fails to manage the files effectively.
- Not all apps can be opened with the help of the built-in file manager.
- Most built-in file managers have a size limitation in the files or folders that can be moved or copied.
- It is not possible to share the files to another device easily.
- The third party file managers which are available online provide better solutions in managing the files but for that, the user has to grant all access for the app to the device data which many people might feel insecure about.

1.4 PROPOSED SYSTEM

Our proposed system fixes the issues that are highlighted in the existing system and also has some of the features mentioned below.

- It keeps the data secure by providing encryption to the files that the user wants(feels it should be protected).
- Provides a thumbnail view for Applications, images and videos.
- Makes it possible to share files between devices easily while keeping it safe from the wrong hands.
- Helps to manage the files effectively by identifying duplicate files.
- Shows hidden files in the system.
- Can archive or compress files and store them in a separate place for easy access.

2. LITERATURE REVIEW

Android allows every user to access their device's file system using a file manager. They can access Android files and transfer them between a computer and a smartphone or tablet. This is performed by simply plugging in a USB cable in an Android device and firing up your desktop file manager. Android's file manager app reacts as an important part of the software on a device.

It provides facilities to browse files, manage storage space, downloads, move files around, and a lot of other activities. However, several Android device manufacturers provide a pre-install Android file manager app on their devices. The pre-install file manager contains fewer features as compared to other third-party file managers.

3. DESCRIPTION OF THE TOPIC

3.1 HANDHELD DEVICE

The handheld device used for this project is a smartphone which operates on an Android operating system. Android is a mobile operating system developed by Google, based on a modified version of the Linux kernel and other open source software and designed primarily for touch screen mobile devices such as Smartphones and tablets. In addition, Google has further developed Android TV for televisions, Android Auto for cars, and Wear OS for wrist watches, each with a specialized user interface. Variants of Android are also used on game consoles, digital cameras, PCs and other electronics. Initially developed by Android Inc., which Google bought in 2005, Android was unveiled in 2007, with the first commercial Android device launched in September 2008. The operating system has since gone through multiple major releases. The core Android source code is known as Android Open Source Project (AOSP), and is primarily licensed under the Apache License. Android is also associated with a suite of proprietary software developed by Google, including core apps for services such as Gmail and Google Search, as well as the application store and digital distribution platform Google Play, and associated development platform. These apps are licensed by manufacturers of Android devices certified under standards imposed by Google, but AOSP has been used as the basis of competing Android ecosystems, such as Amazon.com's Fire OS, which utilize its own equivalents to these Google Mobile Services. Android has been the best-selling OS worldwide on smart phone’s since 2011 and on tablets since 2013. As of May 2017, it has over two billion monthly active users, the largest installed base of any operating system, and as of 2017, the Google Play store features over 3.5 million apps.

3.2 OPERATING SYSTEM OF THE HANDHELD DEVICE

FEATURES OF OS:

1. INTERFACE:

Android's default user interface is mainly based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects, along with a virtual keyboard. Internal hardware, such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to

additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented. Android devices boot to the homescreen, the primary navigation and information "hub" on Android devices, analogous to the desktop found on personal computers. Android homescreens are typically made up of app icons and widgets; app icons launch the associated app, whereas widgets display live, auto-updating content, such as a weather forecast. Along the top of the screen is a status bar, showing information about the device and its connectivity This status bar can be "pulled" down to reveal a notification screen where apps display important information or updates. Notifications are "short, timely, and relevant information about your app when it's not in use", and when tapped, users are directed to a screen inside the app relating to the notification. An All Apps screen lists all installed applications, with the ability for users to drag an app from the list onto the home screen. A Recent screen lets users switch between recently used apps.

2. APPLICATIONS:

Applications ("apps"), which extend the functionality of devices, are written using the Android software development kit (SDK) and, often, the Java programming language. Java may be combined with C/C++, together with a choice of non-default runtimes that allow better C++ support. The Go programming language is also supported, although with a limited set of application programming interfaces (API). In May 2017, Google announced support for Android app development in the Kotlin programming language. The SDK includes a comprehensive set of development tools, including a debugger, software libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Initially, Google's supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) plugin; in December 2014, Google released Android Studio, based on IntelliJ IDEA, as its primary IDE for Android application development. Android has a growing selection of third-party applications, which can be acquired by users by downloading and installing the application's APK (Android application package) file, or by downloading them using an application store program that allows users to install, update, and remove applications from their devices.

3. MEMORY MANAGEMENT:

Since Android devices are usually battery-powered, Android is designed to manage processes to keep power consumption at a minimum. When an application is not in use the system suspends its operation so that, while available for immediate use rather than closed, it does not use battery power or CPU resources. Android manages the applications stored in memory automatically: when memory is low, the system will begin invisibly and automatically close inactive processes, starting with those that have been inactive for the longest amount of time.

3.3 STUDY OF ANDROID

Android Operating System is developed for smartphones and tablets. It is Open source Software. Android is the most widely used mobile Operating System by the people nowadays. Android Software Stack contains four Layers: application layer, application framework layer, Libraries, Linux kernel. Android was founded by a company, named android Inc. in Palo Alto of California, U.S in 2003 whose founder was Andy Rubin. Early intension of the company was to develop an advanced Os for digital cameras, but then it was realized that market for such devices was not large enough so that diverted their attention to producing a smart phone OS. Android was then acquired by the Google in 2005 and unveiled its distribution in 2007 with formation of Open Handset Alliance led by Google. Android mobile operating system is based on the Linux kernel and is developed by Google. Android OS has its own virtual machine called DVM which is used for executing the android application. One of the reasons for the success of Google's OS is the constant improvement of its many versions, with every new one offering more advanced features, faster access to the internet. Another reason for the Android's popularity is its strong collaboration with mobile devices manufacturers, while it is main global competitor.

3.3.1 Android Execution Environment

Android Runtime (**ART**) is an application runtime environment used by the Android operating system. Replacing Dalvik, the process virtual machine originally used by Android, ART performs the translation of the application's bytecode into native instructions that are later executed by the device's runtime environment.[1]

Android 2.2 "Froyo" brought trace-based just-in-time (JIT) compilation into Dalvik, optimizing the execution of applications by continually profiling applications each time they run and dynamically compiling frequently executed short segments of their bytecode into native machine code. While Dalvik interprets the rest of application's bytecode, native execution of those short bytecode segments, called "traces", provides significant performance improvements.[2][3]

Unlike Dalvik, ART introduces the use of ahead-of-time (AOT) compilation by compiling entire applications into native machine code upon their installation. By eliminating Dalvik's interpretation and trace-based JIT compilation, ART improves the overall execution efficiency and reduces power consumption, which results in improved battery autonomy on mobile devices. At the same time, ART brings faster execution of applications, improved memory allocation and garbage collection (GC) mechanisms, new applications debugging features, and more accurate high-level profiling of applications.

To maintain backward compatibility, ART uses the same input bytecode as Dalvik, supplied through standard .dex files as part of APK files, while the .odex files are replaced with Executable and Linkable Format (ELF) executables. Once an application is compiled by using ART's on-device dex2oat utility, it is run solely from the compiled ELF executable; as a result, ART eliminates various application execution overheads associated with Dalvik's interpretation

and trace-based JIT compilation. As a downside, ART requires additional time for the compilation when an application is installed, and applications take up slightly larger amounts of secondary storage (which is usually flash memory) to store the compiled code.

Android 4.4 "KitKat" brought a technology preview of ART, including it as an alternative runtime environment and keeping Dalvik as the default virtual machine.[6][7] In the subsequent major Android release, Android 5.0 "Lollipop", Dalvik was entirely replaced by ART.

Android 7.0 "Nougat" switched its Java Runtime Environment to OpenJDK, introducing a JIT compiler with code profiling to ART, which lets it constantly improve the performance of Android apps as they run.The JIT compiler complements ART's current Ahead of Time compiler and helps improve runtime performance, and save storage space by only compiling some parts of the apps.

3.4 EMULATOR

The Android Emulator simulates Android devices on your computer so that you can test your application on a variety of devices and Android API levels without needing to have each physical device.

The emulator provides almost all of the capabilities of a real Android device. You can simulate incoming phone calls and text messages, specify the location of the device, simulate different network speeds, simulate rotation and other hardware sensors, access the Google Play Store, and much more.

Testing your app on the emulator is in some ways faster and easier than doing so on a physical device. For example, you can transfer data faster to the emulator than to a device connected over USB. Instead of using more than one device at the same time, which would cause frustration, it could be comfortable to use the Emulator and the necessary evaluations of a particular app or some functionality/ feature can be checked.

The emulator comes with predefined configurations for various Android phone, tablet, Wear OS, and Android TV devices.

You can use the emulator manually through its graphical user interface and programmatically through the command line and the emulator console. For a comparison of the features available through each interface, see Comparison of Android Emulator tools.

Requirements and recommendations

The Android Emulator has additional requirements beyond the basic system requirements for Android Studio, which are described below:

- SDK Tools 26.1.1 or higher
- 64-bit processor
- Windows: CPU with UG (unrestricted guest) support
- HAXM 6.2.1 or later (HAXM 7.2.0 or later recommended)

The use of hardware acceleration has additional requirements on Windows and Linux:

- Intel processor on Windows or Linux: Intel processor with support for Intel VT-x, Intel EM64T (Intel 64), and Execute Disable (XD) Bit functionality
- AMD processor on Linux: AMD processor with support for AMD Virtualization (AMD-V) and Supplemental Streaming SIMD Extensions 3 (SSSE3)
- AMD processor on Windows: Android Studio 3.2 or higher and Windows 10 April 2018 release or higher for Windows Hypervisor Platform (WHPX) functionality

To work with Android 8.1 (API level 27) and higher system images, an attached webcam must have the capability to capture 720p frames.

4. TECHNICAL SECURITY FEATURES

Android applications run in a sandbox, 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, however this may not be possible for pre-installed apps. It is not possible, for example, to turn off the microphone access of the pre-installed camera app without disabling the camera completely. This is valid also in Android versions 7 and 8. Before

installing an application, the Google Play store displays a list of the requirements an app needs to function. After reviewing these permissions, the user can choose to accept or refuse them, installing the application only if they accept.

In Android 6.0 "Marshmallow", the permissions system was changed; apps are no longer automatically granted all of their specified permissions at installation time. An opt-in system is used instead, in which users are prompted to grant or deny individual permissions to an app when they are needed for the first time. Applications remember the grants, which can be revoked by the user at any time. The new permissions model is used only by applications developed for Marshmallow using its software development kit (SDK), and older apps will continue to use the previous all-or-nothing approach. Permissions can still be revoked for those apps, though this might prevent them from working properly, and a warning is displayed to that effect.

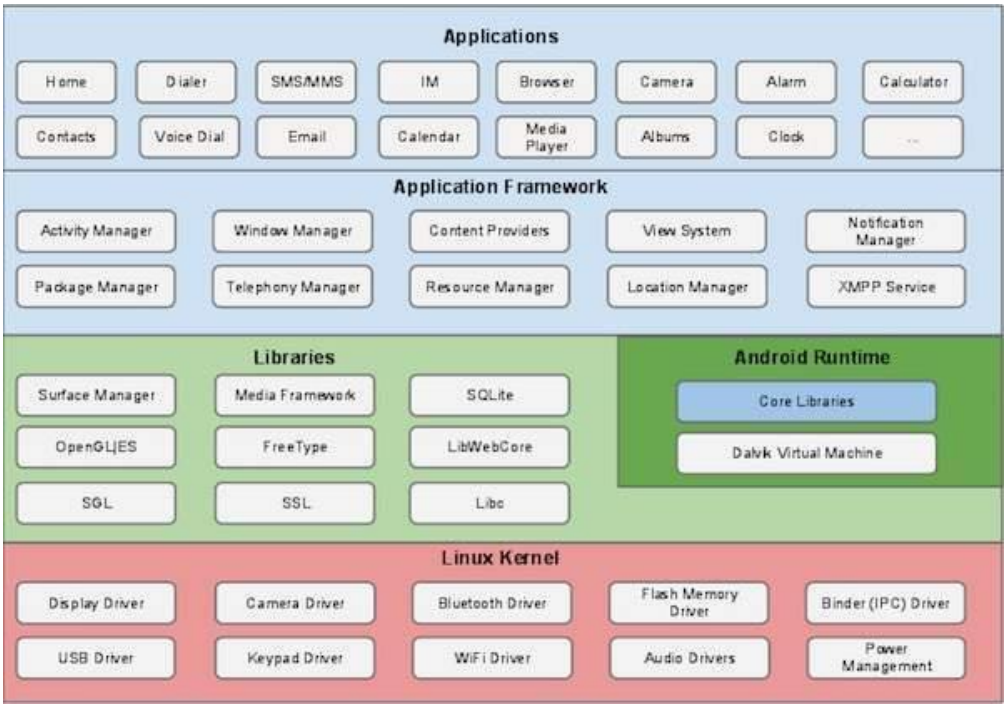
5. SYSTEM DESIGN

Android is developed by Google until the latest changes and updates are ready to be released, at which point the source code is made available to the Android Open Source Project (AOSP), an open source initiative led by Google.The AOSP code can be found without modification on

select devices, mainly the former Nexus and current Android One series of devices. The source code is, in turn, customized by original equipment manufacturers (OEMs) to run on their hardware. Android's source code does not contain the device drivers, often proprietary, that are needed for certain hardware components. As a result, most Android devices, including Google's own, ship with a combination of free and open source and proprietary software, with the software required for accessing Google services falling into the latter category.

5.1 SYSTEM ARCHITECTURE:

The 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 Architecture

Linux kernel

At the bottom of the layers is Linux - Linux 3.6 with approximately 115 patches. This provides a level of abstraction between the device hardware and it contains all the essential hardware drivers 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.

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.

Android Libraries

This category encompasses those Java-based libraries that are specific to Android development. Examples of libraries in this category include the application framework libraries in addition to those that facilitate user interface building, graphics drawing and database access.

Android 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.

Application Framework

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.

The Android framework includes the following key services −

- Activity Manager − Controls all aspects of the application lifecycle and activity stack.
- Content Providers − Allows applications to publish and share data with other applications.
- Resource Manager − Provides access to non-code embedded resources such as strings, color settings and user interface layouts.
- Notifications Manager − Allows applications to display alerts and notifications to the user.
- View System − An extensible set of views used to create application user interfaces.

Applications

You will find all the Android applications 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.

5.2 DESIGN OF FILE EXPLORER

File Management System

Files are used for storing the Information of the user; But Files are organized into the System by using a Specific Manner. Generally for arranging all the Files, directories or Folders are used.

A Folder is also called as the Container of the Files in which many Sub directories and Files are Stored. But there is also an Access Method in which data of the File will be accessed by the user.

Means from which Location Information of the File will be Accessible. There are various types of Storage Architectures which specify from which Locations. File will be Read and from which Locations Data will read and how the data will be Read from the Files.

A file management system has limited capabilities and is designed to manage individual or group files, such as special office documents and records. It may display report details, like owner, creation date, state of completion and similar features useful in an office environment.

A file management system is also known as a file manager.

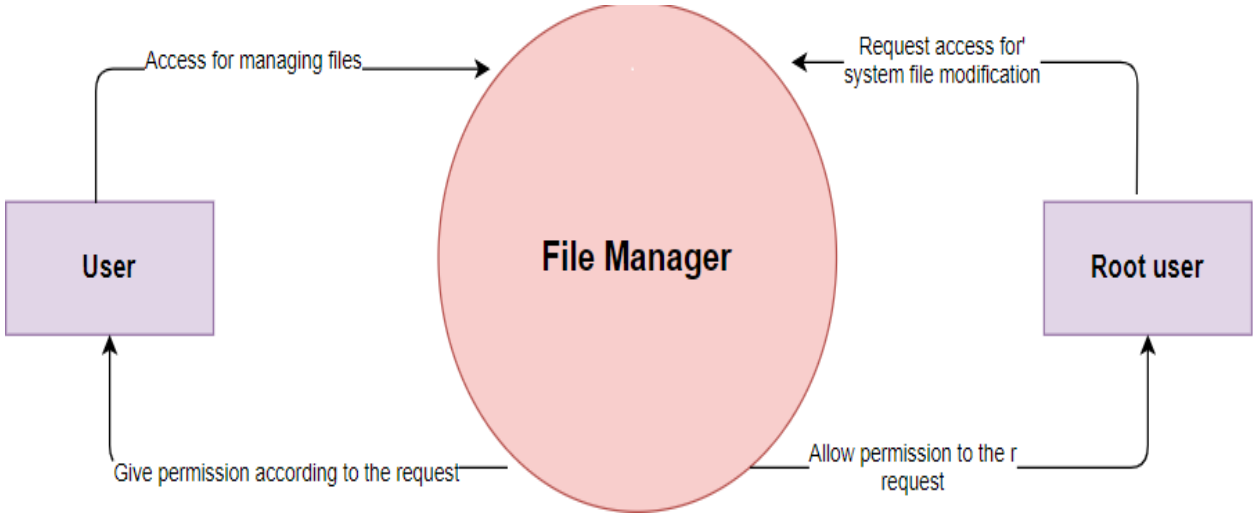
There are three types of Attributes of the Files

1) Read We can only read the data but we can't modify the Data of the File.

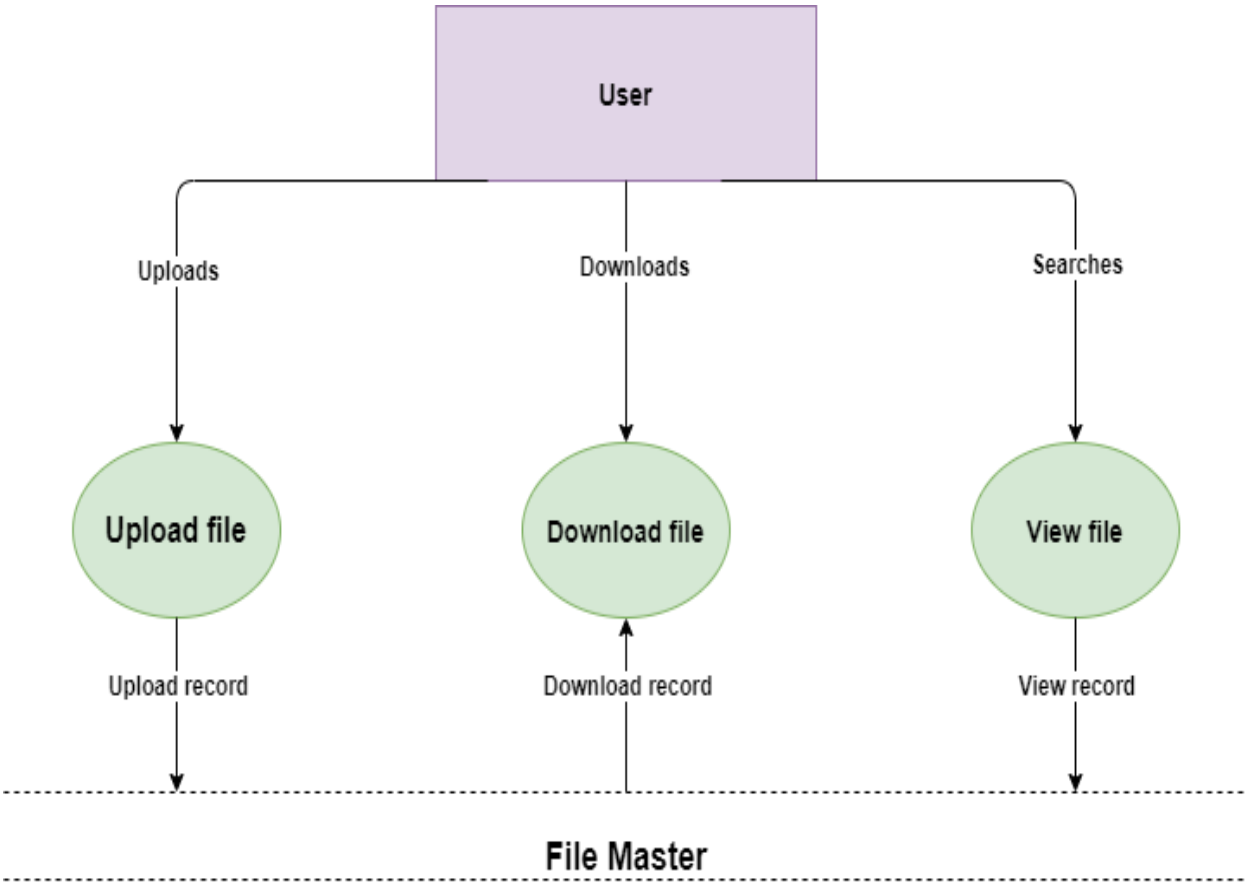
2) Write We can also write the data and also Read the data

3) Hidden File will not be shown to the user. But these will Remain Saved on the user Computer.

Flow of data



Level 1



Level 2

6. DEVELOPING APPLICATION

Design is the first step into the development phase for any engineered product or system. Design is a creative process. A good design is the key to an effective system. An application needs to be built through an efficient method and also it must work properly. It must perfectly suit and play its role for what it has been created. This application "5l Xplorer" is determined to do the tasks of how to organize the files in smartphones, to link required applications to open such files, provide better User Interface to understand the information about the files present. The application is designed with necessary layouts and components.

LOGICAL DESIGN:

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system – its data flows, file content, volumes, Frequencies etc.
- Prepares output specifications – that is, determines the format, content and Frequency of reports.
- Prepares input specifications – format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, Controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

7. CODING

7.1 PROJECT REPOSITORY: (6 Collaborators)

Github link: https://github.com/DarinJoshua-dev/5l-XplorR-Android-App

7.2 AndroidManifest.xml:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.amaze.filemanager">

    <permission android:name="${applicationId}.permission.CONTROL_FTP_SERVER"
        android:protectionLevel="dangerous" />

    <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="com.amaze.cloud.permission.ACCESS_PROVIDER" />
    <uses-permission android:name="android.permission.USE_FINGERPRINT" />
    <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />
```

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE" />
<uses-permission android:name="android.permission.REQUEST_DELETE_PACKAGES" />

<uses-feature
    android:name="android.hardware.touchscreen"
    android:required="false" />

<uses-feature android:name="android.software.leanback"
    android:required="false" />

<application
    android:name=".application.AppConfig"
    android:icon="@mipmap/ic_launcher"
    android:roundIcon="@mipmap/ic_launcher_round"
    tools:replace="android:label"
    android:label="@string/appbar_name"
    android:requestLegacyExternalStorage="true"
    android:banner="@drawable/about_header">

    <activity
        android:label="@string/appbar_name"
        android:launchMode="singleInstance"
        android:name=".ui.activities.MainActivity"
        android:theme="@style/appCompatBlack">

        <intent-filter android:label="@string/appbar_name">
            <action android:name="android.intent.action.MAIN" />
```

```
    <category android:name="android.intent.category.LAUNCHER" />
    <category android:name="android.intent.category.LEANBACK_LAUNCHER" />
</intent-filter>

<intent-filter>
    <action android:name="android.intent.action.GET_CONTENT" />
    <data android:mimeType="*/*" />
    <category android:name="android.intent.category.OPENABLE" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>

<intent-filter>
    <action android:name="android.intent.action.RINGTONE_PICKER" />
    <category android:name="android.intent.category.DEFAULT" />
</intent-filter>

<intent-filter tools:ignore="AppLinkUrlError">
    <action android:name="android.intent.action.VIEW" />
    <data android:mimeType="application/zip" />
    <data android:mimeType="application/rar" />
    <data android:mimeType="application/x-gzip" />
    <data android:mimeType="application/x-rar-compressed"/>
    <data android:mimeType="application/x-bzip2" />
    <data android:mimeType="application/x-xz" />
    <data android:mimeType="application/x-7z-compressed" />
    <!--<category android:name="android.intent.category.OPENABLE" />-->
    <category android:name="android.intent.category.DEFAULT" />
```

```
    </intent-filter>
    <intent-filter>
        <action android:name="android.service.quicksettings.action.QS_TILE_PREFERENCES" />
    </intent-filter>

    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="com.amaze.filemanager" />
    </intent-filter>

    <intent-filter tools:ignore="AppLinkUrlError">
        <action android:name="android.intent.action.VIEW" />
        <data android:mimeType="resource/folder" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="com.amaze.filemanager" />
    </intent-filter>

    <intent-filter android:label="@string/intent_save_as">
        <action android:name="android.intent.action.SEND" />
        <action android:name="android.intent.action.SEND_MULTIPLE" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:mimeType="*/*" />
    </intent-filter>
</activity>
```

```
<activity
   android:uiOptions="splitActionBarWhenNarrow"
   android:label="@string/setting"
   android:name=".ui.activities.PreferencesActivity"
   android:theme="@style/appCompatBlack">
   <intent-filter>
      <action android:name="android.intent.action.APPLICATION_PREFERENCES" />
      <category android:name="android.intent.category.DEFAULT" />
   </intent-filter>

</activity>
<activity
   android:label="@string/textreader"
   android:name=".ui.activities.TextEditorActivity"
   android:theme="@style/appCompatBlack">
   <intent-filter
      tools:ignore="AppLinkUrlError"
      android:label="Amaze Text Editor">
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT" />
      <data android:mimeType="text/*" />
      <data android:mimeType="application/javascript" />
      <data android:mimeType="application/json" />
      <data android:mimeType="application/xml" />
   </intent-filter>

</activity>
```

```
<activity
   android:label="@string/databasereader"
   android:name=".ui.activities.DatabaseViewerActivity"
   android:theme="@style/appCompatBlack"
   android:screenOrientation="locked">
   <intent-filter
      tools:ignore="AppLinkUrlError"
      android:label="Amaze Database Reader">
      <action android:name="android.intent.action.VIEW"/>
      <category android:name="android.intent.category.DEFAULT" />
      <data android:mimeType="db/*" />
   </intent-filter>

</activity>

<activity android:name=".ui.activities.AboutActivity"
   android:theme="@style/aboutBlack"
   android:label="About"
   />

<activity android:name=".crashreport.ErrorActivity"
   android:theme="@style/appCompatBlack"
   android:label="Error"
   />

<service
   android:name = ".asynchronous.services.ExtractService"
```

```
    android:label = "Extracting"
    >
</service>
<service
    android:name = ".asynchronous.services.ZipService"
    android:label = "Compressing"
    >
</service>
<service
    android:name = ".asynchronous.services.CopyService"
    android:label = "Copying"
    >
</service>

<service android:name=".asynchronous.services.EncryptService"
    android:label="@string/crypt_encrypting"
    />

<service android:name=".asynchronous.services.DecryptService"
    android:label="@string/crypt_decrypting"
    />

<service
    android:name=".asynchronous.services.ftp.FtpService"
    android:enabled="true"
    android:exported="true"
    android:permission="${applicationId}.permission.CONTROL_FTP_SERVER"/>
```

```
<service android:name=".asynchronous.services.ftp.FtpTileService"
    android:icon="@drawable/ic_ftp_dark"
    android:label="@string/ftp"
    android:permission="android.permission.BIND_QUICK_SETTINGS_TILE">
    <intent-filter>
        <action
            android:name="android.service.quicksettings.action.QS_TILE" />
    </intent-filter>
</service>

<receiver
    android:name=".asynchronous.services.ftp.FtpReceiver"
    android:exported="true"
    android:permission="${applicationId}.permission.CONTROL_FTP_SERVER">
    <intent-filter>
        <action
android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver.ACTION_START_FTPSERVER" />
        <action
android:name="com.amaze.filemanager.services.ftpservice.FTPReceiver.ACTION_STOP_FTPSERVER" />
    </intent-filter>
</receiver>

<provider
    android:name="androidx.core.content.FileProvider"
    android:authorities="${applicationId}"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
```

```
            android:name="android.support.FILE_PROVIDER_PATHS"
            android:resource="@xml/provider_paths" />
        </provider>
    </application>
</manifest>
```

8. TESTING

Testing refers to testing software so it is also called software testing. Software testing is an investigation conducted to provide stakeholders with information about the quality of the software 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 the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test

· meets the requirements that guided its design and development,

· responds correctly to all kinds of inputs,

· performs its functions within an acceptable time,

· is sufficiently usable,

· can be installed and run in its intended environments, and

· achieves the general result its stakeholders desire.

Unit Testing:

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use.

Parameterized unit tests (PUTs) are tests that take parameters. Unlike traditional unit tests, which are usually closed methods, PUTs take any set of parameters. PUTs have been supported by Testing, JUnit and various .NET test frameworks. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. A unit test provides a strict, written contract that the piece of code must satisfy. As a result, it affords several benefits.

Integration Testing:

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

System Testing:

System testing of software or hardware 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.

As a rule, system testing takes, as its input, all of the "integrated" software components that have passed integration testing and also the software system itself integrated with any applicable hardware system. The purpose of integration testing is to detect any inconsistencies between the software units that are integrated together or between any of the assemblages and the hardware. System testing is a more limited type of testing; it seeks to detect defects both within the "inter-assemblages" and also within the system as a whole.

System testing is performed on the entire system in the context of a Functional Requirement Specification and/or a System Requirement Specification (SRS). System testing tests not only the design, but also the behaviour and even the believed expectations of the customer. It is also intended to test up to and beyond the bounds defined in the software/hardware requirements specification.

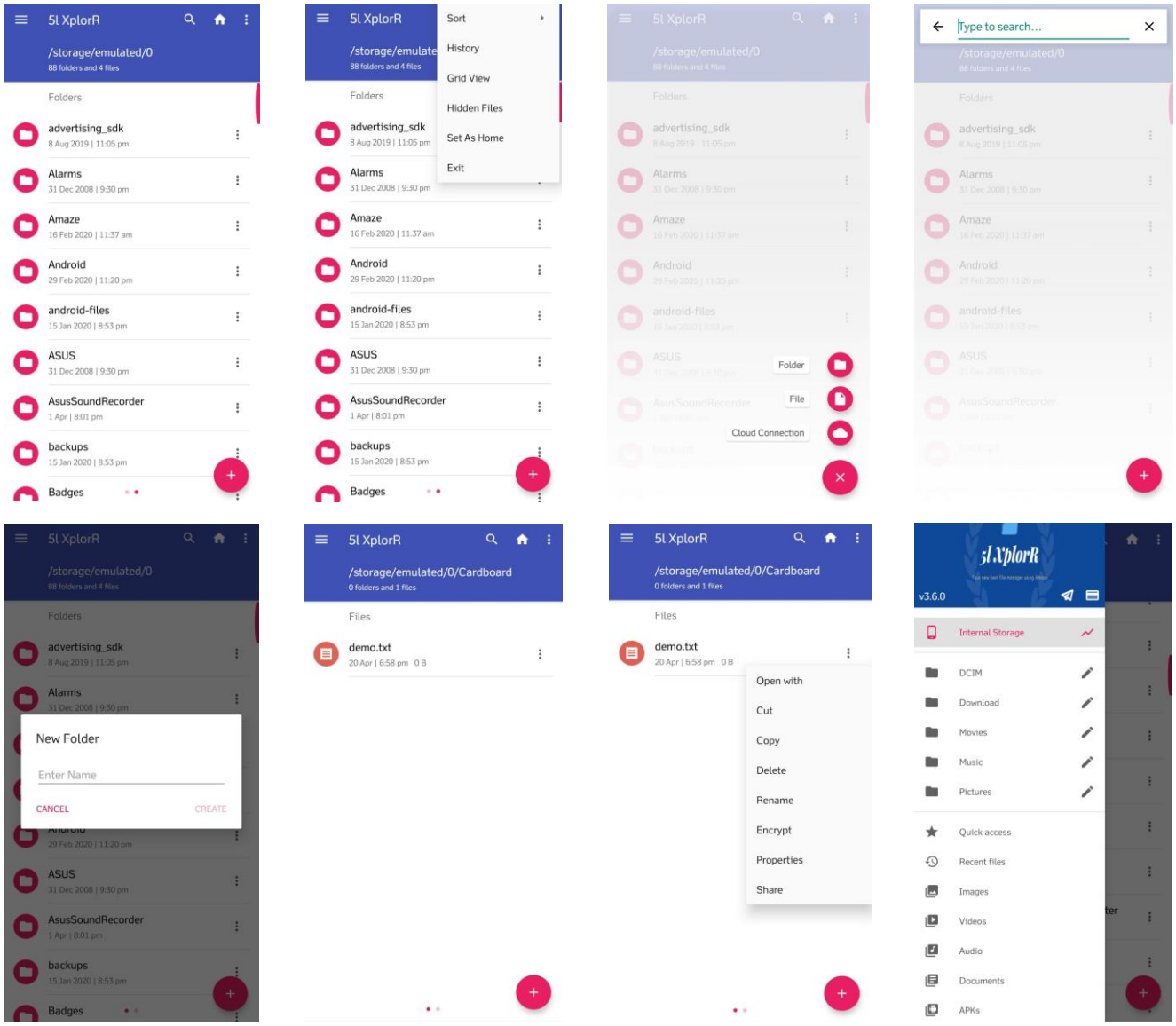
Acceptance Testing:

In engineering and its various sub disciplines, acceptance testing is a test conducted to determine if the requirements of a specification or contract are met. It may involve chemical tests, physical tests, or performance tests.

In systems engineering it may involve black-box testing performed on a system (for example: a piece of software, lots of manufactured mechanical parts, or batches of chemical products) prior to its delivery.

9. RESULT AND ANALYSIS

9.1 Project Screenshots:



Demo & App Icon

9.2 Result Outcome:

The 5l XplorR android application built using java is executed and successfully generated as an apk. The generated apk is installable and can be downloaded in any android device. 5l XplorR is an android file explorer and manager which is fully functional with many different user based features and user interface. Create, Delete, Search, Explore, Manage and Arrange any file, apk, image, etc at ease by using the android application which is well polished and can be used as a regular usage application and manages all data consistent with well formed categories.

9.3 Analysis:

File manager apps are among the most important apps on your device. With them, you can browse your files, find your downloads, manage your storage space, move things around, and a lot more. Not everyone is too keen on file organization because it can be pretty boring, but it still must be occasionally done. 5l XplorR is a new app and it is pretty good. It's open source and focuses on a lighter experience for those who just need to do some light file browsing. It features Material Design, file sharing, a built-in app manager to uninstall apps, root explorer, and more. It manages to include the most important stuff without feeling bloated. Above everything else, it is free to download.

10.CONCLUSION AND FUTURE WORK

This Application is very useful when some urgent tasks are required. This application helps users to do fast work concurrently like share files, copy, paste, cut files from one location to another location. This application is user friendly. This application is very reliable. This application gives very suitable features like we see how much our internal storage is full and see how much our external storage is full, if any memory card is available.

The future scope is to provide security to users and it is very useful for easy managing or sharing files from one place to another securely. Its scope in future is very useful for storing application data and providing a secure file transfer. In future we provide extra security like face recognition and voice recognition system in file manager for improving the security. In the future the android file manager scope is too much. New features like voice recognition can be added, we search files from voice commands so it will be a very easy process for searching the files.