TABLE OF CONTENT

Chapter 1

INTRODUCTION

This project "FILE MANAGER ANDROID APPLICATION" provides us a simple interface to work with file systems. The most common operations used are create, open, edit, view, print, play, rename, move, copy, delete, attributes, properties, search/find, and permissions. Files are typically displayed in a hierarchy. Some file managers contain features inspired by web browsers, including forward and back navigational buttons. Some file managers provide network connectivity. In Windows the program that does this is called Windows Explorer.

While transferring files, a file manager may show the source and destination direcotires, transfer progress in percentage and/or size, progress bar, name of the file currently being transferred, remaining and/or total number of files, numerical transfer rate, and graphical transfer rate. The ability to pause the file transfer allows temporarily granting other software full sequential read access while allowing to resume later without having to restart the file transfer.

Chapter 2

SYSTEM REQUIREMENTS

2.1 Hardware Requirements

Mobile

- Operating System : Android 5.1 or greater
- Memory : 2 GB minimum, 4 GB recommended
- Screen resolution : 1280*1024 or larger
- Internet Connection : Not required

2.2 Software Requirements

- Client : Android OS
- Language : XML and Java

Chapter 3

PROBLEM ANALYSIS

Development of a file manager application to enable user to save , copy , delete ,view, files from a cell phone.

Table 3.1

COLUMN NAME	DATATYPE & SIZE	CONSTRAINTS	DESCRIPTION
		-	
		-	
		-	

Chapter 4

IMPLEMENTATION

CODE

Java code

```
package com.example.filemanager;


import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;


import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.os.Bundle;

import android.os.Environment;

import android.view.View;

import android.widget.Toast;


import com.google.android.material.button.MaterialButton;


public class MainActivity extends AppCompatActivity {


    @Override

    protected void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    MaterialButton storageBtn = findViewById(R.id.storage_btn);

    storageBtn.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(checkPermission()){
                //permission allowed
                Intent intent = new Intent(MainActivity.this, FileListActivity.class);
                String path = Environment.getExternalStorageDirectory().getPath();
                intent.putExtra("path",path);
                startActivity(intent);
            }else{
                //permission not allowed
                requestPermission();

            }
        }
    });

}

private boolean checkPermission(){
    int result =
```

```
ContextCompat.checkSelfPermission(MainActivity.this,Manifest.permission.WRITE_EXTE
RNAL_STORAGE);
      if(result == PackageManager.PERMISSION_GRANTED){
         return true;
      }else
         return false;
   }

   private void requestPermission(){

if(ActivityCompat.shouldShowRequestPermissionRationale(MainActivity.this,Manifest.permi
ssion.WRITE_EXTERNAL_STORAGE)){
         Toast.makeText(MainActivity.this,"Storage permission is requires,please allow from
settings",Toast.LENGTH_SHORT).show();
      }else
      ActivityCompat.requestPermissions(MainActivity.this,new String[]
{Manifest.permission.WRITE_EXTERNAL_STORAGE},111);
   }
}
……
package com.example.filemanager;

import android.content.Context;
import android.content.Intent;
import android.net.Uri;
import android.view.LayoutInflater;
import android.view.MenuItem;
```

```
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.PopupMenu;
import android.widget.TextView;
import android.widget.Toast;

import java.io.File;

import androidx.recyclerview.widget.RecyclerView;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.ViewHolder>{

    Context context;
    File[] filesAndFolders;

    public MyAdapter(Context context, File[] filesAndFolders){
        this.context = context;
        this.filesAndFolders = filesAndFolders;
    }


    @Override
    public ViewHolder onCreateViewHolder(ViewGroup parent, int viewType) {

        View view = LayoutInflater.from(context).inflate(R.layout.recycler_item,parent,false);
```

```
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(MyAdapter.ViewHolder holder, int position) {

    File selectedFile = filesAndFolders[position];
    holder.textView.setText(selectedFile.getName());

    if(selectedFile.isDirectory()){
        holder.imageView.setImageResource(R.drawable.ic_baseline_folder_24);
    }else{
        holder.imageView.setImageResource(R.drawable.ic_baseline_insert_drive_file_24);
    }

    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if(selectedFile.isDirectory()){
                Intent intent = new Intent(context, FileListActivity.class);
                String path = selectedFile.getAbsolutePath();
                intent.putExtra("path",path);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                context.startActivity(intent);
            }else{
                //open thte file
```

```
            try {
                Intent intent = new Intent();
                intent.setAction(android.content.Intent.ACTION_VIEW);
                String type = "image/*";
                intent.setDataAndType(Uri.parse(selectedFile.getAbsolutePath()), type);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                context.startActivity(intent);
            }catch (Exception e){
                Toast.makeText(context.getApplicationContext(),"Cannot open the
file",Toast.LENGTH_SHORT).show();
            }
        }
    }
});

holder.itemView.setOnLongClickListener(new View.OnLongClickListener() {
    @Override
    public boolean onLongClick(View v) {

        PopupMenu popupMenu = new PopupMenu(context,v);
        popupMenu.getMenu().add("DELETE");
        popupMenu.getMenu().add("MOVE");
        popupMenu.getMenu().add("RENAME");

        popupMenu.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            @Override
```

```
            public boolean onMenuItemClick(MenuItem item) {
                if(item.getTitle().equals("DELETE")){
                    boolean deleted = selectedFile.delete();
                    if(deleted){
                        Toast.makeText(context.getApplicationContext(),"DELETED
",Toast.LENGTH_SHORT).show();
                        v.setVisibility(View.GONE);
                    }
                }
                if(item.getTitle().equals("MOVE")){
                    Toast.makeText(context.getApplicationContext(),"MOVED
",Toast.LENGTH_SHORT).show();


                }
                if(item.getTitle().equals("RENAME")){
                    Toast.makeText(context.getApplicationContext(),"RENAME
",Toast.LENGTH_SHORT).show();


                }
                return true;
            }
        });


        popupMenu.show();
        return true;
    }
});
```

```
    }

    @Override
    public int getItemCount() {
        return filesAndFolders.length;
    }


    public class ViewHolder extends RecyclerView.ViewHolder{

        TextView textView;
        ImageView imageView;

        public ViewHolder(View itemView) {
            super(itemView);
            textView = itemView.findViewById(R.id.file_name_text_view);
            imageView = itemView.findViewById(R.id.icon_view);
        }
    }
}
```

……

```
package com.example.filemanager;

import androidx.appcompat.app.AppCompatActivity;
```

```
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.os.Bundle;
import android.view.View;
import android.widget.TextView;

import java.io.File;

public class FileListActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_file_list);

        RecyclerView recyclerView = findViewById(R.id.recycler_view);
        TextView noFilesText = findViewById(R.id.nofiles_textview);

        String path = getIntent().getStringExtra("path");
        File root = new File(path);
        File[] filesAndFolders = root.listFiles();

        if(filesAndFolders==null || filesAndFolders.length ==0){
            noFilesText.setVisibility(View.VISIBLE);
            return;
```

```
        }
        noFilesText.setVisibility(View.INVISIBLE);
        recyclerView.setLayoutManager(new LinearLayoutManager(this));
        recyclerView.setAdapter(new MyAdapter(getApplicationContext(),filesAndFolders));

    }
}
```

XML CODE:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".FileListActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerInParent="true"
        android:id="@+id/nofiles_textview"
        android:visibility="invisible"
        android:text="NO FILES FOUND"/>

    <androidx.recyclerview.widget.RecyclerView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/recycler_view"/>

</RelativeLayout>
…….
<?xml version="1.0" encoding="utf-8"?>
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    xmlns:tools="http://schemas.android.com/tools"
    android:orientation="horizontal"
    android:padding="10dp"
    android:layout_height="wrap_content">



    <ImageView
        android:layout_width="36dp"
        android:layout_height="36dp"
        tools:src="@drawable/ic_baseline_folder_24"
        android:id="@+id/icon_view"/>


    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/file_name_text_view"
        android:layout_toEndOf="@id/icon_view"
        android:padding="10dp"
        android:textColor="@color/black"
        tools:text="Documents"/>


   <LinearLayout
       android:layout_width="match_parent"
       android:layout_height="1dp"
```

```
    android:layout_marginTop="2dp"
    android:background="@color/cardview_shadow_start_color"
    android:layout_below="@id/file_name_text_view"/>

</RelativeLayout>
```

……..

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:background="#1DE9B6"
  tools:context=".MainActivity">

  <com.google.android.material.button.MaterialButton
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/storage_btn"
    android:text="STORAGE"
    android:padding="8dp"
    android:layout_margin="16dp"
    android:layout_centerInParent="true"/>
</RelativeLayout>
```

Chapter 5

LIST OF FIGURES

SCREENSHOTS

FIG : 5.1

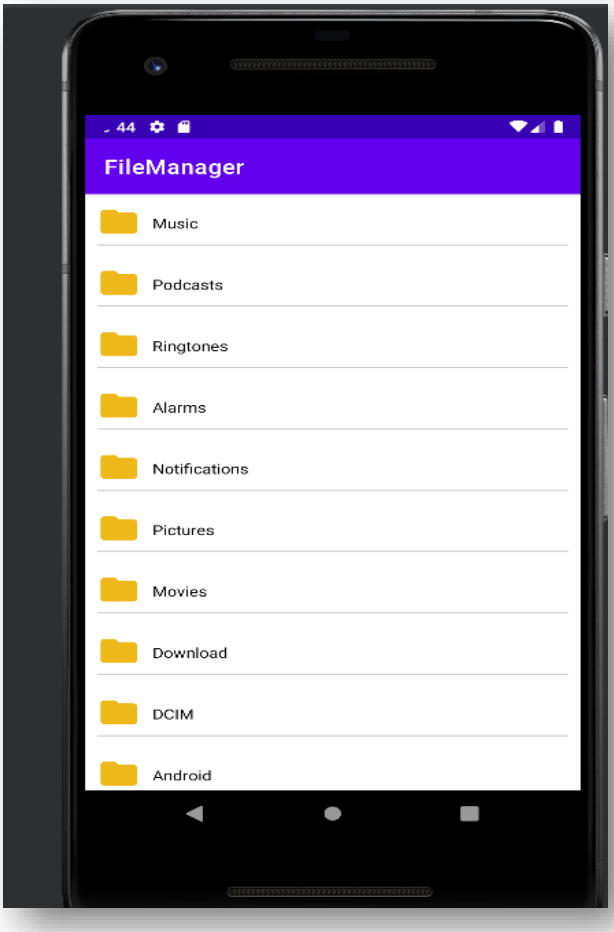
44
FileManager
Music
Podcasts
Ringtones
Alarms
Notifications
Pictures
Movies
Download
DCIM
Android

FIG :5.2

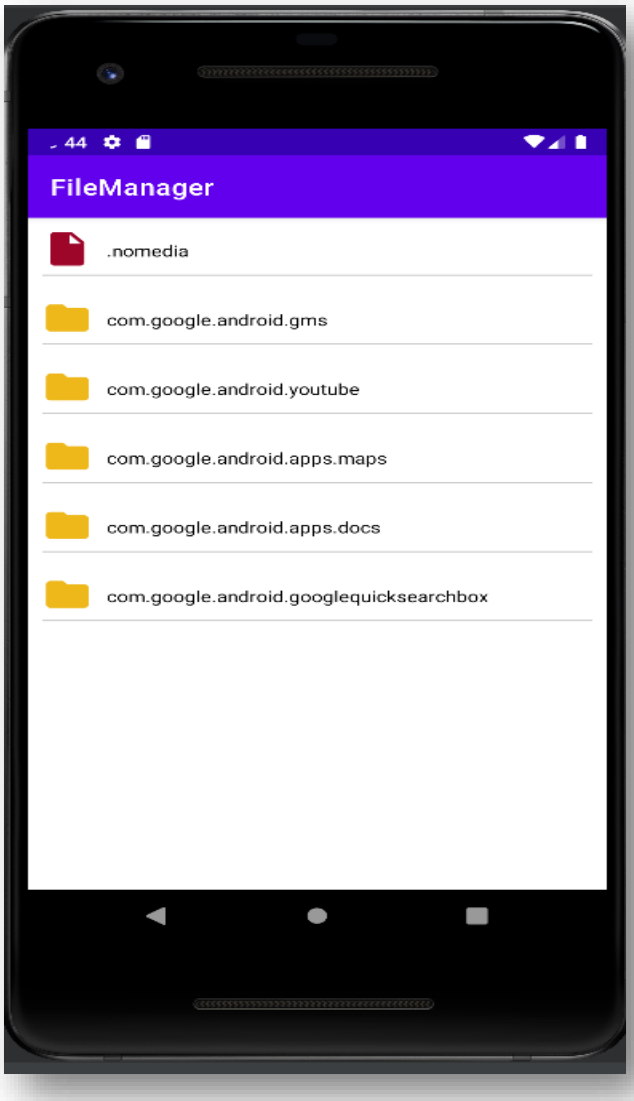
44
FileManager
.nomedia
com.google.android.gms
com.google.android.youtube
com.google.android.apps.maps
com.google.android.apps.docs
com.google.android.googlequicksearchbox

FIG:5.3

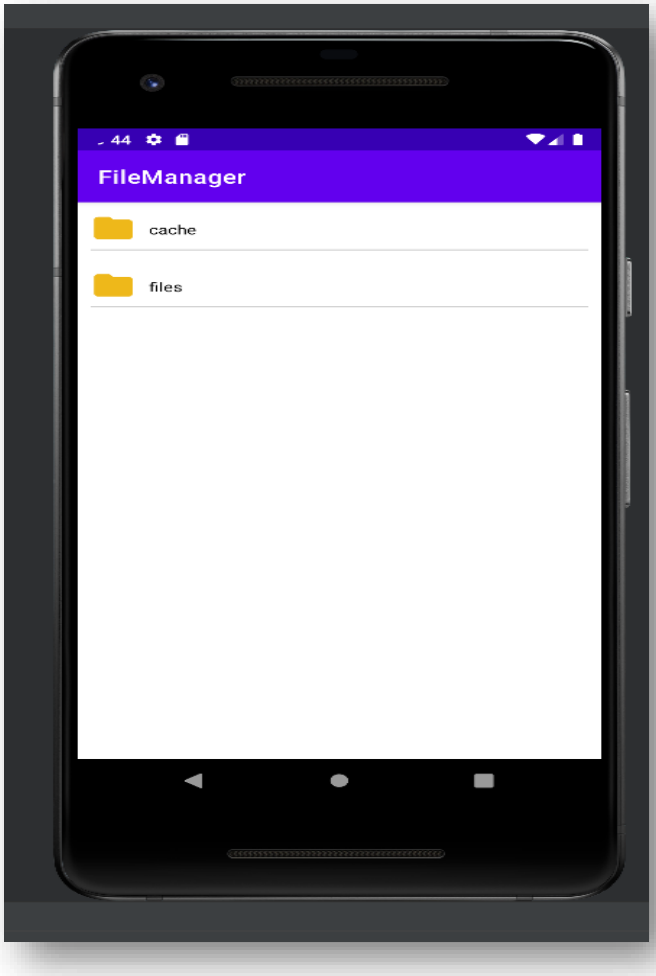

FIG:5.4

Chapter 6

CONCLUSION & FUTURE SCOPE

File management software has been around for at least a decade. In its earliest forms it was concerned mostly with keeping things organized and saving space, today it includes support for multi-national companies and large scale operations that share data globally.

Software companies have also added an element of scalability to their file management software making it easier for smaller businesses to jump on board, and easily transition as they grow.

Future scope of the work:

- ✓ The option to organize files on the basis of time , date etc.
- ✓ The system can be developed in such a way that its existing features can be modified to better versions.

.

BIBLOGRAPHY

- https://www.behance.net/search/projects?search=file%20manager&sort=recommended&time=month
- https://dribbble.com/tags/file_manager_android_app
- https://en.wikipedia.org/wiki/File_manager
- https://www.slideshare.net/Vishalchd11/file-management-32488110