****

**NoteSH!**

**A Simple, yet effective notes app**

**A PROJECT REPORT**

***Submitted by***

**SHARAN SHANKAR A K (715519104046)**

**SHUBHAM GHOSH (715519104047)**

**CS8662 – MOBILE APPLICATION DEVELOPMENT LAB**

**B.E. COMPUTER SCIENCE AND ENGINEERING**

**ACADEMIC YEAR: 2021 – 2022 (EVEN SEMESTER)**

**PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH,**

**COIMBATORE-641062.**

**ABSTRACT**

We tend to forget many important things when we have a busy schedule, and in order to remember them, we must jot them down on a piece of paper. People in such hectic schedules require a personal assistant or a reminder to remind them of the crucial tasks that must be completed. NoteSH is an Android application that assists in the reminder of important tasks. It assists us in keeping track of the everyday tasks that must be completed. Notes app is used for making short text notes, update when you need them, and trash when you are done. It can be used for various functions as you can add your to-do list in this app, some important notes for future reference, etc.

In some situations, such as when you need quick access to your notes, the app is really beneficial. Important meetings, events, and other activities can be easily recorded with NoteSH. This software provides a number of functions, including the ability to add notes. Along with the message, it can be given a title. The time and date of creation are indicated beside it. There is a redo option while typing. You can also remove the memo from he app. The text version of the memo can be shared through other applications. It's also possible to search through your notes.

The notes can be sorted in alphabetical order or by date using the sorting option. The user can alter the accent color, font color, background color, and color navigation bar using the settings feature. It is incredibly user-friendly and easy to use. This application will help users to organize data in a more straightforward and user-friendly manner. It offers numerous advantages, such as ensuring that crucial meetings or assignments are not missed. The color of the application can be adjusted accordingly. At any time, you can contribute new ideas or critical information.

**CODE – JAVA and GRADLE**

**Build.gradle:**

apply plugin: 'com.android.application'

android {

compileSdkVersion 28

defaultConfig {

applicationId "com.appsh.notesh"

minSdkVersion 21

targetSdkVersion 28

versionCode 6

versionName "2.0"

testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"

}

buildTypes {

debug {

applicationIdSuffix = ".debug"

}

release {

minifyEnabled false

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

}

dependencies {

implementation fileTree(dir: 'libs', include: ['\*.jar'])

implementation 'com.android.support:appcompat-v7:28.0.0'

implementation 'com.android.support.constraint:constraint-layout:1.1.3'

implementation 'com.android.support:design:28.0.0'

implementation 'com.github.enricocid:cpl:1.0.3'

implementation 'com.android.support:cardview-v7:28.0.0'

testImplementation 'junit:junit:4.12'

androidTestImplementation 'com.android.support.test.espresso:espresso-contrib:2.2.2', {

exclude group: 'com.android.support', module: 'support-annotations'

exclude group: 'com.android.support', module: 'support-v4'

exclude group: 'com.android.support', module: 'design'

exclude group: 'com.android.support', module: 'recyclerview-v7'

}

androidTestImplementation('com.android.support.test.espresso:espresso-core:2.2.2', {

exclude group: 'com.android.support', module: 'support-annotations'

})

}

**NotesListActivity.java:**

package com.appsh.notesh;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.res.ColorStateList;

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.drawable.Animatable;

import android.graphics.drawable.ColorDrawable;

import android.graphics.drawable.Drawable;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.support.annotation.ColorInt;

import android.support.annotation.NonNull;

import android.support.design.widget.FloatingActionButton;

import android.support.v4.content.ContextCompat;

import android.support.v7.app.ActionBar;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.support.v7.widget.DividerItemDecoration;

import android.support.v7.widget.LinearLayoutManager;

import android.support.v7.widget.RecyclerView;

import android.support.v7.widget.SearchView;

import android.support.v7.widget.helper.ItemTouchHelper;

import android.util.DisplayMetrics;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.WindowManager;

import android.view.inputmethod.EditorInfo;

import android.widget.TextView;

public class NotesListActivity extends AppCompatActivity implements SearchView.OnQueryTextListener {

public static String PREFERENCE\_SORT\_ALPHABETICAL = "sortAlphabetical";

private boolean colourNavbar, sortAlphabetical;

private TextView emptyText;

private NotesListAdapter notesListAdapter;

private FloatingActionButton fab;

private SharedPreferences preferences;

private AlertDialog dialog;

private @ColorInt

int colourPrimary, colourFont, colourBackground;

@Override

protected void onCreate(Bundle savedInstanceState) {

setTheme(R.style.AppTheme);

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_notes\_list);

preferences = PreferenceManager.getDefaultSharedPreferences(NotesListActivity.this);

getSettings(preferences);

fab = findViewById(R.id.fab);

emptyText = findViewById(R.id.tv\_empty);

RecyclerView recyclerView = findViewById(R.id.recycler\_view);

LinearLayoutManager linearLayoutManager = new LinearLayoutManager(NotesListActivity.this);

recyclerView.setLayoutManager(linearLayoutManager);

DividerItemDecoration itemDecorator = new DividerItemDecoration(NotesListActivity.this, DividerItemDecoration.VERTICAL);

Drawable divider = getDrawable(R.drawable.divider);

if (divider != null) {

divider.setTint(colourPrimary);

itemDecorator.setDrawable(divider);

recyclerView.addItemDecoration(itemDecorator);

}

notesListAdapter = new NotesListAdapter(colourFont, colourBackground);

recyclerView.setAdapter(notesListAdapter);

recyclerView.addOnScrollListener(new RecyclerView.OnScrollListener() {

@Override

public void onScrolled(@NonNull RecyclerView recyclerView, int dx, int dy) {

if (dy > 0 || dy < 0 && fab.isShown())

fab.hide();

}

@Override

public void onScrollStateChanged(@NonNull RecyclerView recyclerView, int newState) {

if (newState == RecyclerView.SCROLL\_STATE\_IDLE) {

fab.show();

}

super.onScrollStateChanged(recyclerView, newState);

}

});

setItemTouchHelper(recyclerView);

applySettings();

}

@Override

public void onResume() {

super.onResume();

// Hide keyboard

getWindow().setSoftInputMode(

WindowManager.LayoutParams.SOFT\_INPUT\_STATE\_ALWAYS\_HIDDEN

);

// Close search

SearchView searchView = findViewById(R.id.btn\_search);

if (searchView != null) {

if (!searchView.isIconified()) {

searchView.onActionViewCollapsed();

}

}

// Update the list

notesListAdapter.updateList(HelperUtils.getFiles(NotesListActivity.this), sortAlphabetical);

showEmptyListMessage();

findViewById(R.id.layout\_coordinator).clearFocus();

}

@Override

public void onPause() {

SharedPreferences.Editor editor = preferences.edit();

editor.putBoolean(PREFERENCE\_SORT\_ALPHABETICAL, sortAlphabetical);

editor.apply();

if (dialog != null && dialog.isShowing())

dialog.dismiss();

dialog = null;

super.onPause();

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_notes\_list, menu);

final MenuItem searchItem = menu.findItem(R.id.btn\_search);

final SearchView searchView = (SearchView) searchItem.getActionView();

searchView.setOnQueryTextListener(NotesListActivity.this);

searchView.setMaxWidth(Integer.MAX\_VALUE);

searchView.setImeOptions(EditorInfo.IME\_FLAG\_NO\_EXTRACT\_UI);

if (sortAlphabetical)

menu.findItem(R.id.btn\_sort).setIcon(R.drawable.ic\_sort\_alphabetical);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.btn\_settings:

startActivity(new Intent(NotesListActivity.this, SettingsActivity.class));

return (true);

case R.id.btn\_sort:

if (sortAlphabetical) {

item.setIcon(R.drawable.alphabetical\_to\_numerical);

sortAlphabetical = false;

} else {

item.setIcon(R.drawable.numeric\_to\_alphabetical);

sortAlphabetical = true;

}

notesListAdapter.sortList(sortAlphabetical);

Drawable drawable = item.getIcon();

if (drawable instanceof Animatable)

((Animatable) drawable).start();

case R.id.btn\_search:

return (true);

}

return (super.onOptionsItemSelected(item));

}

@Override

public void onBackPressed() {

SearchView searchView = findViewById(R.id.btn\_search);

if (!searchView.isIconified()) {

searchView.onActionViewCollapsed();

} else {

super.onBackPressed();

}

}

@Override

public boolean onQueryTextChange(String query) {

notesListAdapter.filterList(query.toLowerCase());

return true;

}

@Override

public boolean onQueryTextSubmit(String query) {

return false;

}

private void getSettings(SharedPreferences preferences) {

colourPrimary = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_PRIMARY, ContextCompat.getColor(NotesListActivity.this, R.color.colorPrimary));

colourFont = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_FONT, Color.BLACK);

colourBackground = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_BACKGROUND, Color.WHITE);

colourNavbar = preferences.getBoolean(HelperUtils.PREFERENCE\_COLOUR\_NAVBAR, false);

sortAlphabetical = preferences.getBoolean(PREFERENCE\_SORT\_ALPHABETICAL, false);

}

private void applySettings() {

HelperUtils.applyColours(NotesListActivity.this, colourPrimary, colourNavbar);

findViewById(R.id.layout\_coordinator).setBackgroundColor(colourBackground);

emptyText.setTextColor(colourFont);

fab.setBackgroundTintList(ColorStateList.valueOf(colourPrimary));

ActionBar actionBar = getSupportActionBar();

if (actionBar != null) {

actionBar.setBackgroundDrawable(new ColorDrawable(colourPrimary));

}

}

private void showEmptyListMessage() {

if (notesListAdapter.getItemCount() == 0) {

emptyText.setVisibility(View.VISIBLE);

} else if (emptyText.getVisibility() == View.VISIBLE) {

emptyText.setVisibility(View.GONE);

}

}

private void setItemTouchHelper(RecyclerView recyclerView) {

ItemTouchHelper.SimpleCallback simpleItemTouchCallback = new ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT | ItemTouchHelper.RIGHT) {

@Override

public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {

return false;

}

public void onChildDraw(@NonNull Canvas canvas, @NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder, float dX, float dY, int actionState, boolean isCurrentlyActive) {

if (actionState == ItemTouchHelper.ACTION\_STATE\_SWIPE) {

View itemView = viewHolder.itemView;

Paint p = new Paint();

p.setColor(ContextCompat.getColor(NotesListActivity.this, R.color.colorDelete));

Bitmap icon = BitmapFactory.decodeResource(getResources(), R.drawable.ic\_delete\_white\_24dp);

if (dX > 0) {

canvas.drawRect((float) itemView.getLeft(), (float) itemView.getTop(), dX, (float) itemView.getBottom(), p);

canvas.drawBitmap(icon,

(float) itemView.getLeft() + Math.round(16 \* (getResources().getDisplayMetrics().xdpi / DisplayMetrics.DENSITY\_DEFAULT)),

(float) itemView.getTop() + ((float) itemView.getBottom() - (float) itemView.getTop() - icon.getHeight()) / 2, p);

} else {

canvas.drawRect((float) itemView.getRight() + dX, (float) itemView.getTop(), (float) itemView.getRight(), (float) itemView.getBottom(), p);

canvas.drawBitmap(icon,

(float) itemView.getRight() - Math.round(16 \* (getResources().getDisplayMetrics().xdpi / DisplayMetrics.DENSITY\_DEFAULT)) - icon.getWidth(),

(float) itemView.getTop() + ((float) itemView.getBottom() - (float) itemView.getTop() - icon.getHeight()) / 2, p);

}

super.onChildDraw(canvas, recyclerView, viewHolder, dX, dY, actionState, isCurrentlyActive);

}

}

@Override

public void onSwiped(@NonNull final RecyclerView.ViewHolder viewHolder, int direction) {

dialog = new AlertDialog.Builder(NotesListActivity.this, R.style.AlertDialogTheme)

.setTitle(getString(R.string.confirm\_delete))

.setMessage(getString(R.string.confirm\_delete\_text))

.setPositiveButton(getString(R.string.yes), new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

notesListAdapter.deleteFile(viewHolder.getAdapterPosition());

showEmptyListMessage();

}

})

.setNegativeButton(getString(R.string.no), new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

notesListAdapter.cancelDelete(viewHolder.getAdapterPosition());

}

})

.setOnCancelListener(new DialogInterface.OnCancelListener() {

@Override

public void onCancel(DialogInterface dialog) {

notesListAdapter.cancelDelete(viewHolder.getAdapterPosition());

}

})

.setIcon(ContextCompat.getDrawable(getApplicationContext(), R.drawable.ic\_delete\_white\_24dp))

.show();

if (dialog.getWindow() != null) {

dialog.getWindow().getDecorView().setBackgroundColor(colourPrimary);

}

dialog.getButton(DialogInterface.BUTTON\_POSITIVE).setTextColor(Color.WHITE);

dialog.getButton(DialogInterface.BUTTON\_NEGATIVE).setTextColor(Color.WHITE);

}

};

new ItemTouchHelper(simpleItemTouchCallback).attachToRecyclerView(recyclerView);

}

public void newNote(View view) {

startActivity(NoteActivity.getStartIntent(NotesListActivity.this, ""));

}

}

**NoteActivity.java:**

package com.appsh.notesh;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.res.ColorStateList;

import android.graphics.Color;

import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.support.annotation.ColorInt;

import android.support.v4.content.ContextCompat;

import android.support.v4.graphics.ColorUtils;

import android.support.v7.app.AlertDialog;

import android.support.v7.app.AppCompatActivity;

import android.text.TextUtils;

import android.view.Menu;

import android.view.MenuItem;

import android.widget.EditText;

public class NoteActivity extends AppCompatActivity {

private static final String EXTRA\_NOTE\_TITLE = "EXTRA\_NOTE\_TITLE";

private boolean colourNavbar;

private String title, note;

private EditText noteText, titleText;

private AlertDialog dialog;

private @ColorInt

int colourPrimary, colourFont, colourBackground;

public static Intent getStartIntent(Context context, String title) {

Intent intent = new Intent(context, NoteActivity.class);

intent.putExtra(EXTRA\_NOTE\_TITLE, title);

return intent;

}

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_note);

titleText = findViewById(R.id.et\_title);

noteText = findViewById(R.id.et\_note);

Intent intent = getIntent();

String action = intent.getAction();

String type = intent.getType();

if (Intent.ACTION\_SEND.equals(action) && type != null) {

if ("text/plain".equals(type)) {

String sharedText = intent.getStringExtra(Intent.EXTRA\_TEXT);

noteText.setText(sharedText);

note = sharedText;

title = "";

}

} else {

title = intent.getStringExtra(EXTRA\_NOTE\_TITLE);

if (title == null || TextUtils.isEmpty(title)) {

title = "";

note = "";

noteText.requestFocus();

if (getSupportActionBar() != null)

getSupportActionBar().setTitle(getString(R.string.new\_note));

} else {

titleText.setText(title);

note = HelperUtils.readFile(NoteActivity.this, title);

noteText.setText(note);

if (getSupportActionBar() != null)

getSupportActionBar().setTitle(title);

}

}

getSettings(PreferenceManager.getDefaultSharedPreferences(NoteActivity.this));

applySettings();

}

@Override

public void onRestart() {

super.onRestart();

note = noteText.getText().toString().trim();

if (getCurrentFocus() != null)

getCurrentFocus().clearFocus();

}

@Override

public void onPause() {

if (!isChangingConfigurations()) {

saveFile();

}

if (dialog != null && dialog.isShowing())

dialog.dismiss();

dialog = null;

super.onPause();

}

@Override

public void onBackPressed() {

super.onBackPressed();

}

@Override

public boolean onSupportNavigateUp() {

onBackPressed();

return true;

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.menu\_note, menu);

return true;

}

@Override

public boolean onOptionsItemSelected(MenuItem item) {

switch (item.getItemId()) {

case R.id.btn\_undo:

noteText.setText(note);

noteText.setSelection(noteText.getText().length());

return (true);

case R.id.btn\_share:

Intent sendIntent = new Intent();

sendIntent.setAction(Intent.ACTION\_SEND);

sendIntent.putExtra(Intent.EXTRA\_TEXT, noteText.getText().toString());

sendIntent.setType("text/plain");

startActivity(Intent.createChooser(sendIntent, getString(R.string.share\_to)));

return (true);

case R.id.btn\_delete:

dialog = new AlertDialog.Builder(NoteActivity.this, R.style.AlertDialogTheme)

.setTitle(getString(R.string.confirm\_delete))

.setMessage(getString(R.string.confirm\_delete\_text))

.setPositiveButton(getString(R.string.yes), new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

if (HelperUtils.fileExists(NoteActivity.this, title)) {

deleteFile(title + HelperUtils.TEXT\_FILE\_EXTENSION);

}

title = "";

note = "";

titleText.setText(title);

noteText.setText(note);

finish();

}

})

.setNegativeButton(getString(R.string.no), new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

}

})

.setIcon(ContextCompat.getDrawable(getApplicationContext(), R.drawable.ic\_delete\_white\_24dp))

.show();

if (dialog.getWindow() != null) {

dialog.getWindow().getDecorView().setBackgroundColor(colourPrimary);

}

dialog.getButton(DialogInterface.BUTTON\_POSITIVE).setTextColor(Color.WHITE);

dialog.getButton(DialogInterface.BUTTON\_NEGATIVE).setTextColor(Color.WHITE);

return (true);

}

return (super.onOptionsItemSelected(item));

}

private void getSettings(SharedPreferences preferences) {

colourPrimary = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_PRIMARY, ContextCompat.getColor(NoteActivity.this, R.color.colorPrimary));

colourFont = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_FONT, Color.BLACK);

colourBackground = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_BACKGROUND, Color.WHITE);

colourNavbar = preferences.getBoolean(HelperUtils.PREFERENCE\_COLOUR\_NAVBAR, false);

}

private void applySettings() {

HelperUtils.applyColours(NoteActivity.this, colourPrimary, colourNavbar);

noteText.setBackgroundTintList(ColorStateList.valueOf(colourPrimary));

titleText.setBackgroundTintList(ColorStateList.valueOf(colourPrimary));

findViewById(R.id.scroll\_view).setBackgroundColor(colourBackground);

if (getSupportActionBar() != null)

getSupportActionBar().setBackgroundDrawable(new ColorDrawable(colourPrimary));

titleText.setTextColor(colourFont);

noteText.setTextColor(colourFont);

titleText.setHintTextColor(ColorUtils.setAlphaComponent(colourFont, 120));

noteText.setHintTextColor(ColorUtils.setAlphaComponent(colourFont, 120));

}

private void saveFile() {

// Get current title and note

String newTitle = titleText.getText().toString().trim().replace("/", " ");

String newNote = noteText.getText().toString().trim();

if (TextUtils.isEmpty(newTitle) && TextUtils.isEmpty(newNote)) {

return;

}

if (newTitle.equals(title) && newNote.equals(note)) {

return;

}

if (!title.equals(newTitle) || TextUtils.isEmpty(newTitle)) {

newTitle = newFileName(newTitle);

titleText.setText(newTitle);

}

HelperUtils.writeFile(NoteActivity.this, newTitle, newNote);

if (!TextUtils.isEmpty(title) && !newTitle.equals(title)) {

deleteFile(title + HelperUtils.TEXT\_FILE\_EXTENSION);

}

title = newTitle;

}

private String newFileName(String name) {

if (TextUtils.isEmpty(name)) {

name = getString(R.string.note);

}

if (HelperUtils.fileExists(NoteActivity.this, name)) {

int i = 1;

while (true) {

if (!HelperUtils.fileExists(NoteActivity.this, name + " (" + i + ")") || title.equals(name + " (" + i + ")")) {

name = (name + " (" + i + ")");

break;

}

i++;

}

}

return name;

}

}

**NotesDiffCallback.java:**

package com.appsh.notesh;

import android.support.annotation.Nullable;

import android.support.v7.util.DiffUtil;

import java.io.File;

import java.util.List;

public class NotesDiffCallback extends DiffUtil.Callback {

private List<File> oldList;

private List<File> newList;

public NotesDiffCallback(List<File> oldList, List<File> newList) {

this.oldList = oldList;

this.newList = newList;

}

@Override

public int getOldListSize() {

return oldList.size();

}

@Override

public int getNewListSize() {

return newList.size();

}

@Override

public boolean areItemsTheSame(int oldItemPosition, int newItemPosition) {

return oldList.get(oldItemPosition).getName().equals(newList.get(newItemPosition).getName());

}

@Override

public boolean areContentsTheSame(int oldItemPosition, int newItemPosition) {

return oldList.get(oldItemPosition).lastModified() < (System.currentTimeMillis() - 5000);

}

@Nullable

@Override

public Object getChangePayload(int oldItemPosition, int newItemPosition) {

return super.getChangePayload(oldItemPosition, newItemPosition);

}

}

**NotesListAdapter.java:**

package com.appsh.notesh;

import android.support.annotation.NonNull;

import android.support.constraint.ConstraintLayout;

import android.support.v7.util.DiffUtil;

import android.support.v7.widget.RecyclerView;

import android.text.TextUtils;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.TextView;

import java.io.File;

import java.text.DateFormat;

import java.util.ArrayList;

import java.util.Collections;

import java.util.Comparator;

import java.util.List;

class NotesListAdapter extends RecyclerView.Adapter<NotesListAdapter.ViewHolder> {

private List<File> fullList, filesList;

private int colourText, colourBackground;

NotesListAdapter(int colourText, int colourBackground) {

filesList = new ArrayList<>();

fullList = new ArrayList<>();

this.colourText = colourText;

this.colourBackground = colourBackground;

}

@Override

public void onBindViewHolder(@NonNull NotesListAdapter.ViewHolder holder, int position) {

File file = filesList.get(position);

String fileName = file.getName().substring(0, file.getName().length() - 4);

String fileDate = DateFormat.getDateInstance(DateFormat.MEDIUM).format(file.lastModified());

String fileTime = DateFormat.getTimeInstance(DateFormat.SHORT).format(file.lastModified());

holder.setData(fileName, fileDate, fileTime);

}

@NonNull

@Override

public NotesListAdapter.ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View inflatedView = LayoutInflater.from(parent.getContext()).inflate(R.layout.list\_item, parent, false);

return new ViewHolder(inflatedView, colourText, colourBackground);

}

@Override

public int getItemCount() {

return filesList.size();

}

void updateList(List<File> files, boolean sortAlphabetical) {

filesList = files;

sortList(sortAlphabetical);

fullList = new ArrayList<>(filesList);

}

void filterList(String query) {

if (TextUtils.isEmpty(query)) {

DiffUtil.calculateDiff(new NotesDiffCallback(filesList, fullList)).dispatchUpdatesTo(this);

filesList = new ArrayList<>(fullList);

} else {

filesList.clear();

for (int i = 0; i < fullList.size(); i++) {

final File file = fullList.get(i);

final String fileName = file.getName().substring(0, file.getName().length() - 4).toLowerCase();

if (fileName.contains(query)) {

filesList.add(fullList.get(i));

}

}

DiffUtil.calculateDiff(new NotesDiffCallback(fullList, filesList)).dispatchUpdatesTo(this);

}

}

void sortList(boolean sortAlphabetical) {

if (sortAlphabetical) {

sortAlphabetical(filesList);

} else {

sortDate(filesList);

}

DiffUtil.calculateDiff(new NotesDiffCallback(fullList, filesList)).dispatchUpdatesTo(this);

fullList = new ArrayList<>(filesList);

}

private void sortAlphabetical(List<File> files) {

Collections.sort(files, new Comparator<File>() {

public int compare(File f1, File f2) {

return (f1.getName().compareTo(f2.getName()));

}

});

}

private void sortDate(List<File> files) {

Collections.sort(files, new Comparator<File>() {

public int compare(File f1, File f2) {

return Long.compare(f2.lastModified(), f1.lastModified());

}

});

}

void deleteFile(int position) {

File file = filesList.get(position);

fullList.remove(file);

filesList.remove(file);

notifyItemRemoved(position);

file.delete();

}

void cancelDelete(int position) {

notifyItemChanged(position);

}

static class ViewHolder extends RecyclerView.ViewHolder implements View.OnClickListener {

private final TextView noteTitle, noteDate, noteTime;

private String stringTitle;

ConstraintLayout constraintLayout;

ViewHolder(View view, int colourText, int colourBackground) {

super(view);

noteTitle = view.findViewById(R.id.tv\_title);

noteDate = view.findViewById(R.id.tv\_date);

noteTime = view.findViewById(R.id.tv\_time);

noteTitle.setTextColor(colourText);

noteDate.setTextColor(colourText);

noteTime.setTextColor(colourText);

constraintLayout = view.findViewById(R.id.layout\_constraint);

constraintLayout.setBackgroundColor(colourBackground);

view.setOnClickListener(this);

}

@Override

public void onClick(View view) {

itemView.getContext().startActivity(NoteActivity.getStartIntent(itemView.getContext(), stringTitle));

}

void setData(String title, String date, String time) {

stringTitle = title;

noteTitle.setText(title);

noteDate.setText(date);

noteTime.setText(time);

}

}

}

**SettingsActivity.java:**

package com.appsh.notesh;

import android.content.Intent;

import android.content.SharedPreferences;

import android.content.res.ColorStateList;

import android.graphics.Color;

import android.graphics.PorterDuff;

import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;

import android.preference.PreferenceManager;

import android.support.annotation.ColorInt;

import android.support.v4.app.DialogFragment;

import android.support.v4.content.ContextCompat;

import android.support.v4.graphics.ColorUtils;

import android.support.v4.widget.CompoundButtonCompat;

import android.support.v7.app.AppCompatActivity;

import android.view.View;

import android.widget.CheckBox;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

import com.enrico.colorpicker.colorDialog;

public class SettingsActivity extends AppCompatActivity implements colorDialog.ColorSelectedListener {

private boolean colourNavbar;

private ImageView imageAccent, imageFont, imageBackground;

private CheckBox navBox;

private SharedPreferences preferences;

private @ColorInt

int colourPrimary, colourFont, colourBackground;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_settings);

preferences = PreferenceManager.getDefaultSharedPreferences(SettingsActivity.this);

imageAccent = findViewById(R.id.image\_accent);

imageFont = findViewById(R.id.image\_font);

imageBackground = findViewById(R.id.image\_background);

getSettings(preferences);

applySettings();

}

@Override

public boolean onSupportNavigateUp() {

onBackPressed();

return true;

}

@Override

public void onColorSelection(DialogFragment dialogFragment, @ColorInt int selectedColor) {

int tag = Integer.valueOf(dialogFragment.getTag());

switch (tag) {

case 1:

colourPrimary = ColorUtils.setAlphaComponent(selectedColor, 255);

imageAccent.setColorFilter(colourPrimary);

break;

case 2:

colourFont = ColorUtils.setAlphaComponent(selectedColor, 255);

imageFont.setColorFilter(colourFont);

break;

case 3:

colourBackground = ColorUtils.setAlphaComponent(selectedColor, 255);

imageBackground.setColorFilter(colourBackground);

break;

}

}

private void getSettings(SharedPreferences preferences) {

colourPrimary = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_PRIMARY, ContextCompat.getColor(SettingsActivity.this, R.color.colorPrimary));

colourFont = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_FONT, Color.BLACK);

colourBackground = preferences.getInt(HelperUtils.PREFERENCE\_COLOUR\_BACKGROUND, Color.WHITE);

colourNavbar = preferences.getBoolean(HelperUtils.PREFERENCE\_COLOUR\_NAVBAR, false);

}

private void applySettings() {

HelperUtils.applyColours(SettingsActivity.this, colourPrimary, colourNavbar);

if (getSupportActionBar() != null) {

getSupportActionBar().setBackgroundDrawable(new ColorDrawable(colourPrimary));

getSupportActionBar().setTitle(getString(R.string.settings));

}

findViewById(R.id.layout\_constraint).setBackgroundColor(colourBackground);

imageAccent.setColorFilter(colourPrimary);

imageFont.setColorFilter(colourFont);

imageBackground.setColorFilter(colourBackground);

imageAccent.getBackground().setColorFilter(colourPrimary, PorterDuff.Mode.SRC\_ATOP);

imageFont.getBackground().setColorFilter(colourPrimary, PorterDuff.Mode.SRC\_ATOP);

imageBackground.getBackground().setColorFilter(colourPrimary, PorterDuff.Mode.SRC\_ATOP);

((TextView) findViewById(R.id.tv\_accent)).setTextColor(colourFont);

((TextView) findViewById(R.id.tv\_font)).setTextColor(colourFont);

((TextView) findViewById(R.id.tv\_background)).setTextColor(colourFont);

((TextView) findViewById(R.id.tv\_navigationbar)).setTextColor(colourFont);

((LinearLayout) findViewById(R.id.settingsLayout)).getDividerDrawable().setColorFilter(colourPrimary, PorterDuff.Mode.SRC\_ATOP);

findViewById(R.id.btn\_apply).getBackground().setColorFilter(colourPrimary, PorterDuff.Mode.SRC\_ATOP);

navBox = findViewById(R.id.checkbox\_navigationbar);

navBox.setChecked(colourNavbar);

CompoundButtonCompat.setButtonTintList(navBox, ColorStateList.valueOf(colourPrimary));

}

public void saveSettings(View view) {

SharedPreferences.Editor editor = preferences.edit();

editor.putInt(HelperUtils.PREFERENCE\_COLOUR\_PRIMARY, colourPrimary);

editor.putInt(HelperUtils.PREFERENCE\_COLOUR\_FONT, colourFont);

editor.putInt(HelperUtils.PREFERENCE\_COLOUR\_BACKGROUND, colourBackground);

editor.putBoolean(HelperUtils.PREFERENCE\_COLOUR\_NAVBAR, navBox.isChecked());

editor.apply();

startActivity(new Intent(SettingsActivity.this, NotesListActivity.class).addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TOP));

finish();

}

public void showPicker1(View view) {

colorDialog.setPickerColor(SettingsActivity.this, 1, colourPrimary);

colorDialog.showColorPicker(SettingsActivity.this, 1);

}

public void showPicker2(View view) {

colorDialog.setPickerColor(SettingsActivity.this, 2, colourFont);

colorDialog.showColorPicker(SettingsActivity.this, 2);

}

public void showPicker3(View view) {

colorDialog.setPickerColor(SettingsActivity.this, 3, colourBackground);

colorDialog.showColorPicker(SettingsActivity.this, 3);

}

public void toggleCheckBox(View view) {

navBox.toggle();

}

}

**HelperUtils.java:**

package com.appsh.notesh;

import android.app.Activity;

import android.app.ActivityManager;

import android.content.Context;

import android.graphics.BitmapFactory;

import android.graphics.Color;

import android.view.Window;

import android.view.WindowManager;

import android.widget.Toast;

import java.io.BufferedReader;

import java.io.File;

import java.io.FilenameFilter;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import java.util.ArrayList;

import java.util.Arrays;

public class HelperUtils {

public static String TEXT\_FILE\_EXTENSION = ".txt";

public static String PREFERENCE\_COLOUR\_PRIMARY = "colourPrimary";

public static String PREFERENCE\_COLOUR\_FONT = "colourFont";

public static String PREFERENCE\_COLOUR\_BACKGROUND = "colourBackground";

public static String PREFERENCE\_COLOUR\_NAVBAR = "colourNavbar";

public static int darkenColor(int color, double fraction) {

int red = Color.red(color);

int green = Color.green(color);

int blue = Color.blue(color);

red = darken(red, fraction);

green = darken(green, fraction);

blue = darken(blue, fraction);

int alpha = Color.alpha(color);

return Color.argb(alpha, red, green, blue);

}

private static int darken(int color, double fraction) {

return (int) Math.max(color - (color \* fraction), 0);

}

public static void applyColours(Activity activity, int colourPrimary, boolean colourNavbar) {

Window window = activity.getWindow();

if (colourNavbar)

window.setNavigationBarColor(colourPrimary);

window.addFlags(WindowManager.LayoutParams.FLAG\_DRAWS\_SYSTEM\_BAR\_BACKGROUNDS);

window.setStatusBarColor(HelperUtils.darkenColor(colourPrimary, 0.2));

activity.setTaskDescription(new ActivityManager.TaskDescription(activity.getString(R.string.app\_name),

BitmapFactory.decodeResource(activity.getResources(), R.drawable.ic\_note), colourPrimary));

}

public static ArrayList<File> getFiles(Context context) {

File[] files = context.getFilesDir().listFiles(new FilenameFilter() {

@Override

public boolean accept(File dir, String name) {

return name.toLowerCase().endsWith(HelperUtils.TEXT\_FILE\_EXTENSION);

}

});

return new ArrayList<>(Arrays.asList(files));

}

public static boolean fileExists(Context context, String fileName) {

File file = context.getFileStreamPath(fileName + HelperUtils.TEXT\_FILE\_EXTENSION);

return file.exists();

}

public static void writeFile(Context context, String fileName, String fileContent) {

try {

OutputStreamWriter out = new OutputStreamWriter(context.openFileOutput(fileName + HelperUtils.TEXT\_FILE\_EXTENSION, 0));

out.write(fileContent);

out.close();

} catch (Throwable t) {

Toast.makeText(context, context.getString(R.string.exception) + t.toString(), Toast.LENGTH\_LONG).show();

}

}

public static String readFile(Context context, String fileName) {

String content = "";

if (fileExists(context, fileName)) {

try {

InputStream in = context.openFileInput(fileName + HelperUtils.TEXT\_FILE\_EXTENSION);

if (in != null) {

InputStreamReader tmp = new InputStreamReader(in);

BufferedReader reader = new BufferedReader(tmp);

String str;

StringBuilder buf = new StringBuilder();

while ((str = reader.readLine()) != null) {

buf.append(str).append("\n");

}

in.close();

content = buf.toString();

}

} catch (Exception e) {

Toast.makeText(context, context.getString(R.string.exception) + e.toString(), Toast.LENGTH\_LONG).show();

}

}

return content.trim();

}

}

**CODE – XML**

**AndroidManifest.xml:**

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

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

package="com.appsh.notesh">

<application

android:allowBackup="true"

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

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

android:roundIcon="@mipmap/ic\_launcher"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity

android:name="com.appsh.notesh.NoteActivity"

android:parentActivityName="com.appsh.notesh.NotesListActivity">

<intent-filter>

<action android:name="android.intent.action.SEND" />

<category android:name="android.intent.category.DEFAULT" />

<data android:mimeType="text/plain" />

</intent-filter>

</activity>

<activity

android:name="com.appsh.notesh.NotesListActivity"

android:theme="@style/SplashTheme">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity

android:name="com.appsh.notesh.SettingsActivity"

android:parentActivityName="com.appsh.notesh.NotesListActivity" />

</application>

</manifest>

**activity\_notes\_list.xml**

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

<android.support.design.widget.CoordinatorLayout 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:id="@+id/layout\_coordinator"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:focusableInTouchMode="true"

tools:context="com.appsh.notesh.NotesListActivity">

<TextView

android:id="@+id/tv\_empty"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_marginEnd="30dp"

android:layout\_marginStart="30dp"

android:text="@string/empty\_list\_text"

android:textAlignment="center"

android:textSize="18sp"

android:visibility="gone" />

<android.support.v7.widget.RecyclerView

android:id="@+id/recycler\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layoutAnimation="@anim/layout\_animation\_fall\_down"

android:paddingTop="2dp"

android:scrollbars="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<android.support.design.widget.FloatingActionButton

android:id="@+id/fab"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="bottom|end"

android:layout\_margin="16dp"

android:foreground="@android:drawable/ic\_input\_add"

android:onClick="newNote"

android:scaleType="center"

app:fabSize="normal" />

</android.support.design.widget.CoordinatorLayout>

**activity\_note.xml**

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

<android.support.design.widget.CoordinatorLayout 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:id="@+id/layout\_coordinator"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:focusableInTouchMode="true"

tools:context="com.appsh.notesh.NoteActivity">

<ScrollView

android:id="@+id/scroll\_view"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:fillViewport="true"

tools:context="com.appsh.notesh.NoteActivity"

tools:showIn="@layout/activity\_note">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical">

<EditText

android:id="@+id/et\_title"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:ems="10"

android:hint="@string/title"

android:imeOptions="flagNoExtractUi"

android:inputType="textCapWords"

android:maxLength="36"

android:paddingEnd="8dp"

android:paddingStart="8dp"

android:textCursorDrawable="@null"

android:textStyle="bold"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<EditText

android:id="@+id/et\_note"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1"

android:autoLink="web|email"

android:ems="10"

android:gravity="top"

android:hint="@string/note"

android:imeOptions="flagNoExtractUi"

android:inputType="textCapSentences|textMultiLine"

android:linksClickable="true"

android:paddingEnd="8dp"

android:paddingStart="8dp"

android:textCursorDrawable="@null"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/title" />

</LinearLayout>

</ScrollView>

</android.support.design.widget.CoordinatorLayout>

**list\_item.xml**

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

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

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clickable="true"

android:focusable="true"

android:orientation="vertical">

<android.support.constraint.ConstraintLayout

android:id="@+id/layout\_constraint"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@android:color/white"

android:foreground="?attr/selectableItemBackground"

android:minHeight="60dp">

<TextView

android:id="@+id/tv\_title"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:paddingEnd="2dp"

android:paddingStart="16dp"

android:textSize="20sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintRight\_toLeftOf="@id/tv\_date"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintWidth\_default="percent"

app:layout\_constraintWidth\_percent="0.7" />

<TextView

android:id="@+id/tv\_date"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:paddingBottom="2dp"

android:paddingEnd="16dp"

android:paddingStart="2dp"

android:paddingTop="2dp"

android:textAlignment="viewEnd"

android:textSize="15sp"

app:layout\_constraintBottom\_toTopOf="@id/tv\_time"

app:layout\_constraintLeft\_toRightOf="@id/tv\_title"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintWidth\_default="percent"

app:layout\_constraintWidth\_percent="0.3" />

<TextView

android:id="@+id/tv\_time"

android:layout\_width="0dp"

android:layout\_height="match\_parent"

android:paddingEnd="16dp"

android:paddingStart="2dp"

android:textAlignment="viewEnd"

android:textSize="15sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toRightOf="@id/tv\_title"

app:layout\_constraintTop\_toBottomOf="@id/tv\_date"

app:layout\_constraintWidth\_default="percent"

app:layout\_constraintWidth\_percent="0.3" />

</android.support.constraint.ConstraintLayout>

</FrameLayout>

**activity\_settings.xml**

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

<android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/layout\_constraint"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.appsh.notesh.SettingsActivity">

<LinearLayout

android:id="@+id/settingsLayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:divider="@drawable/divider\_settings"

android:orientation="vertical"

android:showDividers="middle|end">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:gravity="center\_vertical"

android:onClick="showPicker1"

android:orientation="horizontal"

android:paddingBottom="4dp"

android:paddingTop="8dp">

<ImageView

android:id="@+id/image\_accent"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_margin="12dp"

android:background="@drawable/square"

android:contentDescription="@string/colour\_indicator"

android:src="@drawable/ic\_bg" />

<TextView

android:id="@+id/tv\_accent"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="8dp"

android:text="@string/accent\_colour"

android:textSize="16sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:gravity="center\_vertical"

android:onClick="showPicker2"

android:orientation="horizontal"

android:paddingBottom="4dp"

android:paddingTop="4dp">

<ImageView

android:id="@+id/image\_font"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_margin="12dp"

android:background="@drawable/square"

android:contentDescription="@string/colour\_indicator"

android:src="@drawable/ic\_bg" />

<TextView

android:id="@+id/tv\_font"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="8dp"

android:text="@string/font\_colour"

android:textSize="16sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:gravity="center\_vertical"

android:onClick="showPicker3"

android:orientation="horizontal"

android:paddingBottom="4dp"

android:paddingTop="4dp">

<ImageView

android:id="@+id/image\_background"

android:layout\_width="40dp"

android:layout\_height="40dp"

android:layout\_margin="12dp"

android:background="@drawable/square"

android:contentDescription="@string/colour\_indicator"

android:src="@drawable/ic\_bg" />

<TextView

android:id="@+id/tv\_background"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="8dp"

android:text="@string/background\_colour"

android:textSize="16sp" />

</LinearLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="60dp"

android:gravity="center\_vertical"

android:onClick="toggleCheckBox"

android:orientation="horizontal"

android:paddingBottom="4dp"

android:paddingTop="4dp">

<CheckBox

android:id="@+id/checkbox\_navigationbar"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="16dp" />

<TextView

android:id="@+id/tv\_navigationbar"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="8dp"

android:text="@string/colour\_navigation\_bar"

android:textSize="16sp" />

</LinearLayout>

</LinearLayout>

<Button

android:id="@+id/btn\_apply"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_marginBottom="7dp"

android:onClick="saveSettings"

android:text="@string/apply\_changes"

android:textColor="@android:color/white"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent" />

<android.support.constraint.Guideline

android:id="@+id/guideline"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

app:layout\_constraintGuide\_begin="20dp" />

<android.support.constraint.Barrier

android:id="@+id/barrier2"

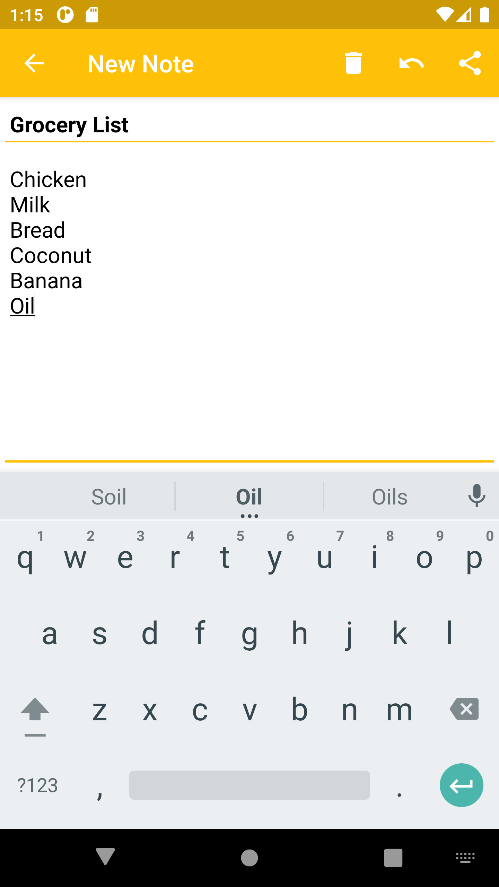
android:layout\_width="wrap\_content"

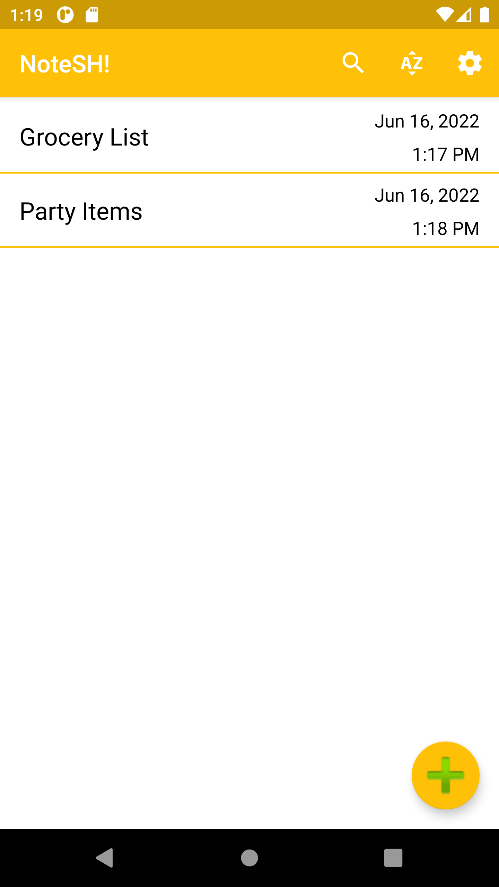
android:layout\_height="wrap\_content"

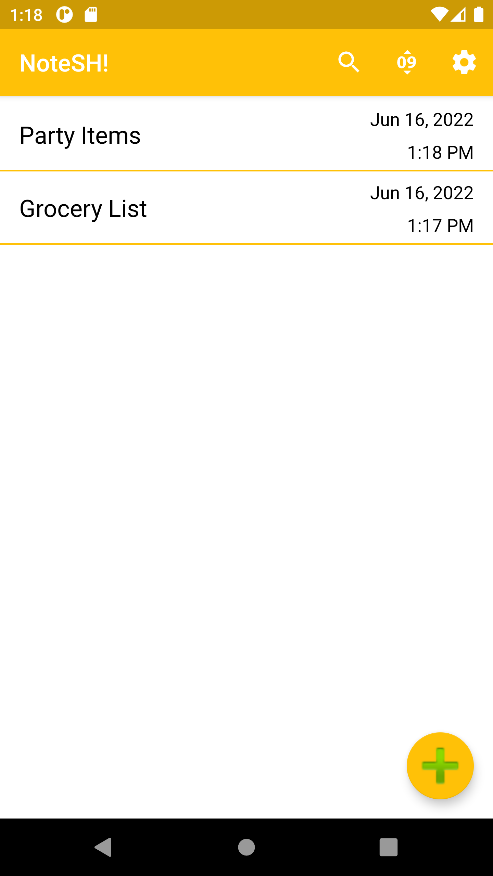
app:barrierDirection="top" />

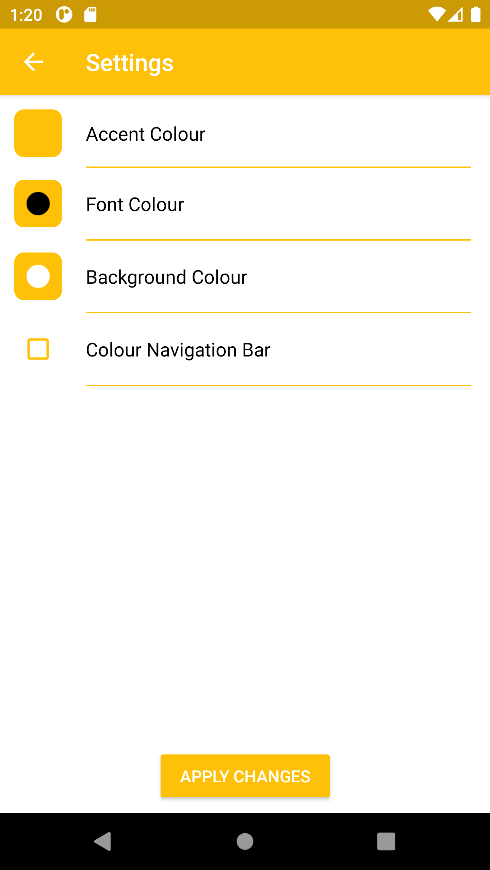
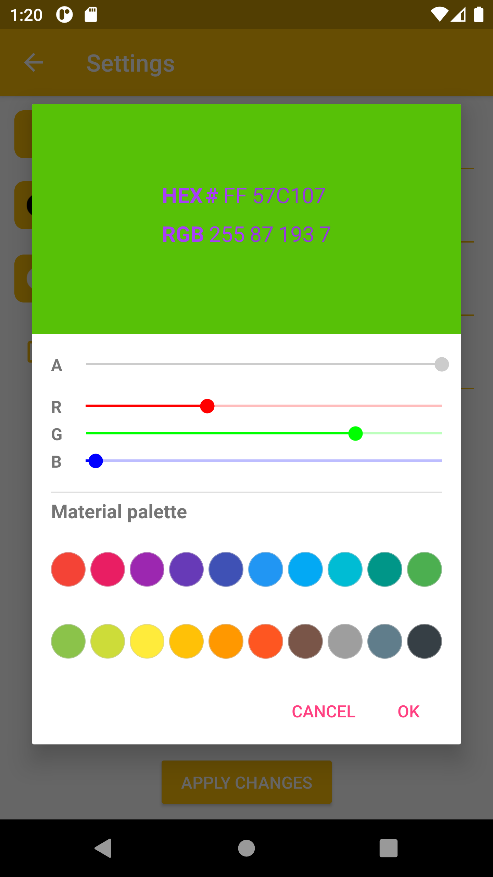
</android.support.constraint.ConstraintLayout>

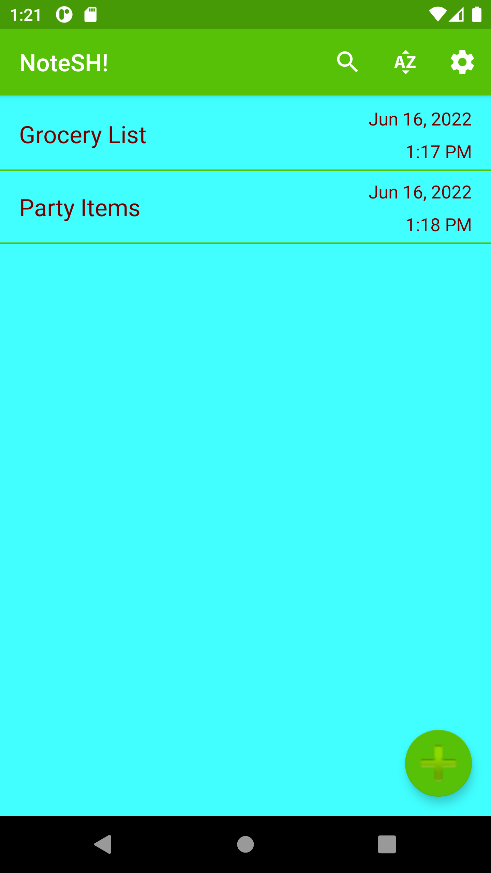
**OUTPUT**

****

****

****

****

****

**CONCLUSION**

Thus, an android application that helps in taking notes effortlessly has been built.