***פרויקט Notes and quotes***

**עבודת גמר תכנון ותכנות מערכות 883589**

**חלופה: מערכות טלפונים ניידים**

**מגיש: אדר פלג**

**ת.ז. 214731333**

**מנחה: הייתם**

**אורט למדעים ואומנויות**

**תאריך הגשה: 19/05/23**

**תוכן העניינים**

[**מענה לדרישות משרד החינוך לפרויקט 5 יח"ל 3**](#_heading=h.gjdgxs)

[**מבוא 6**](#_heading=h.30j0zll)

[הרקע לפרויקט 6](#_heading=h.1fob9te)

[מטרת המערכת 6](#_heading=h.3znysh7)

[יכולות המערכת 6](#_heading=h.2et92p0)

[תהליך המחקר 7](#_heading=h.tyjcwt)

[אתגרים מרכזיים 7](#_heading=h.3dy6vkm)

[**מדריך למשתמש 8**](#_heading=h.1t3h5sf)

[תרשים מסכים 9](#_heading=h.4d34og8)

[תיאור המסכים 10](#_heading=h.2s8eyo1)

[מסך פתיחה 10](#_heading=h.17dp8vu)

[חלון Splash. 11](#_heading=h.vrf2lfhxxboj)

[חלון יצירת חשבון חדש. 12](#_heading=h.38w2ru697qkq)

[חלון Transfer. 13](#_heading=h.kjffclh72ile)

[חלון עם בחירה. 14](#_heading=h.vyu6l5ldwvxr)

[חלון Notes. 15](#_heading=h.xwj5lmyw8yyj)

[חלון create note. 16](#_heading=h.as4c6fnl7ds)

[דיאלוג URL. 18](#_heading=h.x3dxvl6k1qf0)

[חלון result activity. 19](#_heading=h.en2b5h7wo7jn)

[הרשאות 20](#_heading=h.26in1rg)

[דרישות מיוחדות ומגבלות 20](#_heading=h.lnxbz9)

[גירסת Android מינימלית 22](#_heading=h.35nkun2)

[מכשירים עליהם נבדקה המערכת 22](#_heading=h.1ksv4uv)

[**תיאור תחום הידע 23**](#_heading=h.44sinio)

[אובייקטים נחוצים 23](#_heading=h.2jxsxqh)

[סוגי נתונים 23](#_heading=h.z337ya)

[ייצוג מידע 23](#_heading=h.3j2qqm3)

[**מבנה/ארכיטקטורה 24**](#_heading=h.4i7ojhp)

[מחלקות הפרויקט וקשרים ביניהם 24](#_heading=h.2xcytpi)

[תרשים מחלקות 29](#_heading=h.1ci93xb)

[מחלקות 29](#_heading=h.3whwml4)

[מחלקה A 29](#_heading=h.2bn6wsx)

[מחלקה B 30](#_heading=h.q7tkbwwyd45y)

[מחלקה C 30](#_heading=h.j5wtr5uo8kk7)

[מחלקה D 30](#_heading=h.nwecunvn78cc)

[שמירת נתונים 33](#_heading=h.3as4poj)

[קבצים 33](#_heading=h.1pxezwc)

[Shared Preference 35](#_heading=h.2p2csry)

[סכמת Database 36](#_heading=h.147n2zr)

[**רפלקציה אישית 37**](#_heading=h.23ckvvd)

[נספחים: 39](#_heading=h.al4r3s3pb5fy)

[קוד המקור (source code) 39](#_heading=h.41mghml)

[מחלקה A 39](#_heading=h.2grqrue)

[מחלקה B 42](#_heading=h.vx1227)

[מחלקה C 43](#_heading=h.h1e7qev8pjk4)

[מחלקה D 44](#_heading=h.p14zy962bi4w)

[מחלקה E 47](#_heading=h.6hdvqrt7auyx)

[מחלקה F 48](#_heading=h.xe2w3lcoqrqj)

[מחלקה G 49](#_heading=h.juorxy4eflh7)

[מחלקה H 50](#_heading=h.l5t42zco80am)

[מחלקה I 51](#_heading=h.pnb10fxgs654)

[מחלקה J 52](#_heading=h.xsc7epxi4hs6)

[מחלקה K 55](#_heading=h.6fu1q4pn4l3n)

[Activities (XML & java) 57](#_heading=h.yz0u86i3wsd)

[CreateNoteActivity: 57](#_heading=h.4qn1smwf3mv5)

[xml: 69](#_heading=h.kp2c31n438yf)

[main activity 74](#_heading=h.l8ketemj3qep)

[main activity xml: 77](#_heading=h.v37jbb5wl22e)

[notes page java: 79](#_heading=h.p2or0zllc5ru)

[notes page Xml: 83](#_heading=h.y6q7yquz71m5)

[splash screen: 86](#_heading=h.mdfii93an6bs)

[splash screen xml: 87](#_heading=h.v1ie5tg9jfd2)

[Register page java: 88](#_heading=h.76lkp7vt21qh)

[Register page XML: 90](#_heading=h.ucls9nqni0h1)

[Result activity Java: 93](#_heading=h.rk57qe6foem8)

[Transfer page java: 96](#_heading=h.5aveatzdxnsz)

[Transfer page XML: 99](#_heading=h.vdex1780sujv)

[Circle menu view: 100](#_heading=h.9kaxyggvyu1c)

[item container note: 100](#_heading=h.rzjue7tfnlqy)

[Dialog Url view; 102](#_heading=h.i7syjx1i2i9s)

[layout miscellaneous: 105](#_heading=h.ccaakjwumvz4)

[111](#_heading=h.6hb4an58s1h3)

[android:layout\_height="w](#_heading=h.ko3ndisg90ca)[" 113](#_heading=h.tsrcotayfzaf) 114

[Row item: 114](#_heading=h.qfs74sbtu7eo)

[other: 115](#_heading=h.4qjwqb331mgi)

[drawable: 117](#_heading=h.vx0hll7slter)

[Manifest 118](#_heading=h.noic4ocpme8)

[thread page: 120](#_heading=h.qog8rpes9i46)

[xml: 122](#_heading=h.j7y0wqhdu0w9)

# מענה לדרישות משרד החינוך לפרויקט 5 יח"ל

| **פרק במחוון** | **נושאים שמומשו** | **מחלקה/ות בהן מומש הנושא** |
| --- | --- | --- |
| 6 | * AlarmManager + Notification |  |
|  | * Job Scheduler |  |
|  | * כתיבת מחלקת Service | myService |
|  | * הורדת נתוני מידע (לא תמונות/ מוזיקה) מהאינטרנט (API של אתר) ושימוש בנתונים שהורדו )דוגמה: שווי מטבע חוץ, רשימת בתי עסק באזור). | X |
|  | * שימוש ב – AcvtivityForResult בפורמט החדש | X |
|  | * שימוש ב-) ContentProvider לא כולל שימוש ב-Intent מרומז( | X |
|  | * יצירת אנימציה (לא כולל שימוש במחלקת אנימציה( | activity\_splash\_screen |
|  | הערה: תלמיד שבחר ביותר מנושא אחד מהרשימה, יוכל לבחור הרחבה אחת בלבד מסעיף 9 |  |
| 7 | אחסון וטיפול בנתונים (חובה שמירה ושליפה) | Room - SQllite |
|  | אחסון ועדכון תוכן | CreateNoteActivity - ROOM |
|  | עדכון ואחסון משתמשים | SQLhelper - SQLlite |
| 9 | נושאים מתקדמים: |  |
|  | * SurfaceView | X |
|  | * ירושה ממחלקת View |  |
|  | * אפליקציית רב משתתפים | User.Java+ sharedpref notes |
|  | * Fragment | X |
|  | * GoogleMaps (לפחות הצגת מיקום ע"י כתובת / נקודות אורך רוחב) | X |
|  | * NFC | X |
|  | * מאזיני תנועה | X |
|  | * Bluetooth | X |
|  | * Threads | ExampleTH.Java |
|  | * פיתוח ושימוש ב API בצד שרת (אפשר בכל שפת תכנות) | X |
|  | * בינה מלאכותית | X |
|  | * רשתות מחשבים | X |
|  | * למידת מכונה | X |
|  | * טכנולוגיה חדשה (כגון עיבוד תמונה( | X |
|  | * Firebase / Firestore | X |
| 10 | שילוב של 2 נושאים מרשימה זו יחשבו ביחד לנושא מתקדם: |  |
|  | * Broadcast Receiver | NetworkChangeReciver |
|  | * Shared Preference | MainActivity |
|  | * Camera & Gallery | X |
|  | * Async Task | CreateNoteActivity |
|  | * Handler | Splash\_screen\_book |
|  | * Microphone | X |
|  | * GPS | X |
|  | * Count Down Timer | X |
|  | * TextToSpeech | X |
|  | * SpeechToText | X |

# מבוא

## הרקע לפרויקט

1. שם הפרויקט - Notes & quotes

2. תיאור קצר של הפרויקט

אפליקציה לפרסום לוקלי של הערות בנושאים שונים. מטרת הפרויקט היא לאפשר לאנשים לפרסם מעין "טור דעה" לחברים שלומדים יחד איתם או קוראים איתם באותה ספרייה על נושאים משותפים.

3. קהל היעד

ספריות או מקומות מרובות משתמשים לוקליות לתקשרות פנימית.

4. הסיבות לבחירת הנושא

אני מתחבר מאוד לקיראת ספרים ואוהב מאוד לכתוב את דעתי, ציטוטים אהובים ועוד. בשל כך רציתי לפתור את הבעיה של לחלוק את החוייה עם אחרים.

## מטרת המערכת

* המערכת משמשת מעין מדף אלקטרוני של "טורי דעה". כל אחד שמעוניין יכול לפתוח חשבון ולהעלות מה שעולה על נפשו שאחרים יוכלו לקרוא.
* מנהלי הספריה\ בית הספר\ המערכת הם בעלי יכולות admin . הם יכולים למחוק (ולערוך) הערות שאינן מתאימות בפורם של המערכת.
* בנוסף admin יכול להקצות יכולות admin או editor "עורך" למשתמשים אחרים, שימוש לדוגמא הוא שמנהל נותן יכולות לצוות המורים לנהל את המערכת. או להקצות לתלמיד יכולת לערוך את הפתק שלו.
* מטרה נוספת זה להשמיע מוסיקה להנעים את זמן הקיראה או הכתיבה.

## יכולות המערכת

תיאור האופציות שנותנת המערכת למשתמש (Feature List).

* ליצור notes חדשים כולם.
* להוסיף לפתק תמונה שתוצג בדף המרכזי.
* להוסיף לפתק קישור.
* לשנות את צבע הפתק.
* להוסיף לפתק כותרת שתוצג בדף המרכזי.
* להוסיף לפרק תת כותרת שתוצג בדף המרכזי.
* לערוך את הפתק לאחר הכתיבה ע"י editor וע"י admin.
* למחוק את הפתק ע"י אדמין בלבד.
* להשמיע מוזיקה.
* לבדוק האם המערכת מחוברת לאינטרנט (נעשה אוטומטית).
* ליצור חשבון חדש.
* לבדוק את הרשאות החשבון.
* להיכנס לחשבון קיים.
* להיכנס לחשבון admin קבוע מראש.

## תהליך המחקר

כל האפליקציות שמצאתי תחת הרעיון של כתיבת פתקים הם מוגבלות לאדם או קבוצה ובדרכ לא המוניות. בעיני התרומה של כל האנשים ברשת היא שמאפשרת יצירה טובה יותר.

בעיני יצרתי מעין קובץ שיתופי ליצירה שבו השמיים הם הגבול!

כל אחד יכול לתרום לרעיון, כמו "פרויקט קוד פתוח" שמאפשר לכולם לתרום.

### אתגרים מרכזיים

1. אחת הבעיות המרכזיות שבהם התעסקתי במהלך הפרויקט זה לאפשר יכולת קריאה וכתיבה. העיסוק עם שמירת נתונים והדפסתם הייתה קשה. עם זאת בעיני הצלחתי ליצור תוצאה מתפקדת ונוחה.

2. על איזה צורך הפרויקט עונה? איזה פתרון הפרויקט הזה בא לתת?

הפרויקט בא לענות על צורך של יצירה משותפת. הכתיבה יחד מאפשר בעיני אפשרויות חסרות סוף ליצירה משותפת.

# מדריך למשתמש

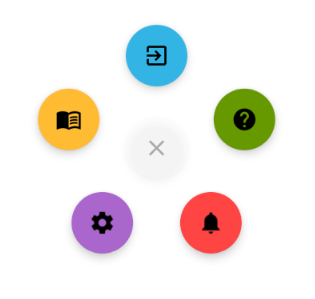
כניסת משתמש חדש:

משתמש חדש צריך ללחוץ על יצירת משתמש חדש והכניס את פרטיו המזהים לאחר הכנסת הפרטים תיכנס לאפליקציה.

כניסת משתמש ישן:

על משתמש ישן להכניס את השם והסיסמא.

הערות: לא ניתן שיהיו 2 משתמשים עם אותו שם.



העמוד הראשי (אפשרויות):

אדום: לחיצה קצרה להפעלת מוזיקה

אדום: לחיצה ארוכה לעצור מוזיקה

ירוק: בדיקת הרשאות (admin;default;editor)

סגול (לאדמינים בלבד): דף שינוי הרשאות למשתמשים רשומים.

כחול: לצאת לדף כניסה.

צהוב מעבר לדף הפתקים!

דף הדפים:

בראש הדף רשום ההרשאות שפתוחות לך.

* משתמש חדש: יכול להוסיף פתק חדש לא יכול לערוך ולא יכול למחוק.
* משתמש עם אישור לערוך: יכול להוסיף פתק חדש + לערוך פתקים קיימים. (אם אתה מעוניין צור קשר עם מנהל)
* אדמין (מנהל) יכול לערוך למחוק ולהוסיף.

## 

## תרשים מסכים

תרשים מסכים המתאר את היררכיית המסכים והמעברים ביניהם Screen flow diagram

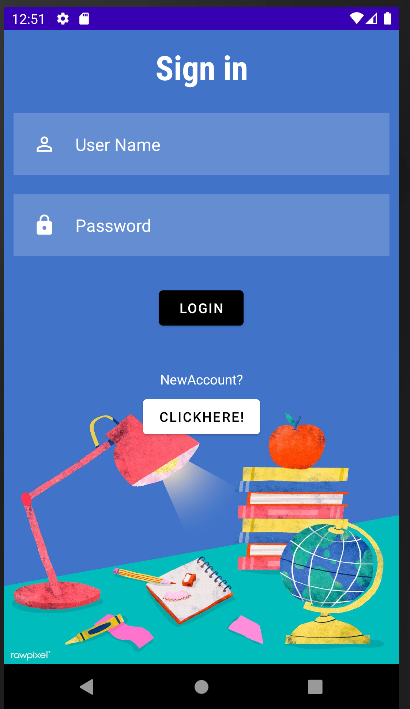
**מתאר את אופן הניווט בין המסכים השונים**

**דוגמא:**

## 

## תיאור המסכים

### מסך פתיחה



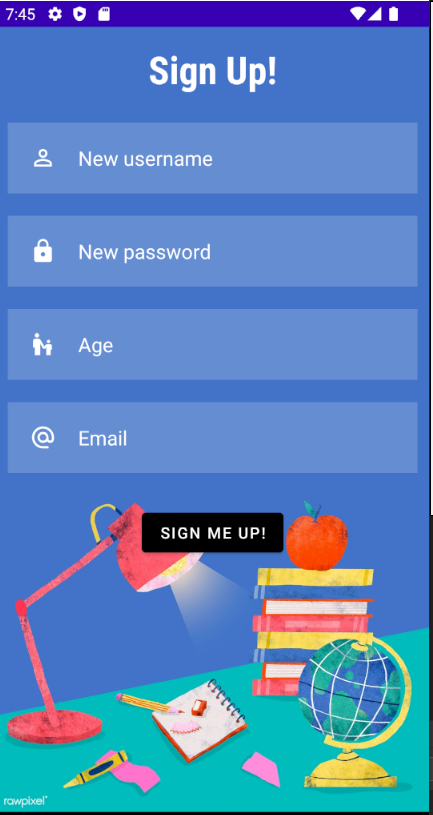
* מסך כניסה
* ניתן להיכנס לחשבון או ליצור חשבון חדש.

### חלון Splash.

### 

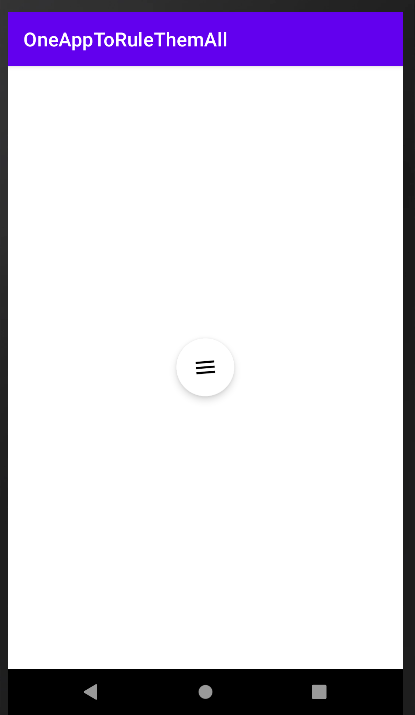
* מסך אנימציה מעבר
* מראה אנימציה ועובר לפי האקסטרה שהועבר לו.

### חלון יצירת חשבון חדש.



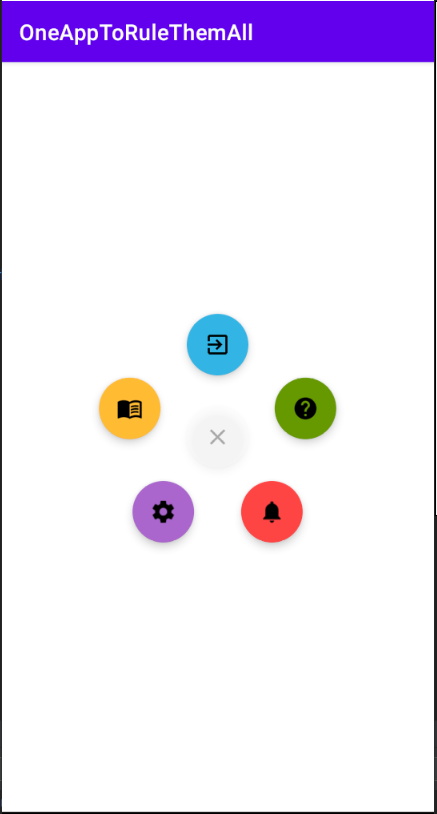
* מסך הכנסת פרטים
* אחרי הכנסת פרטים נשמר החשבון + נכנס לעמוד הראשי.

### חלון Transfer.



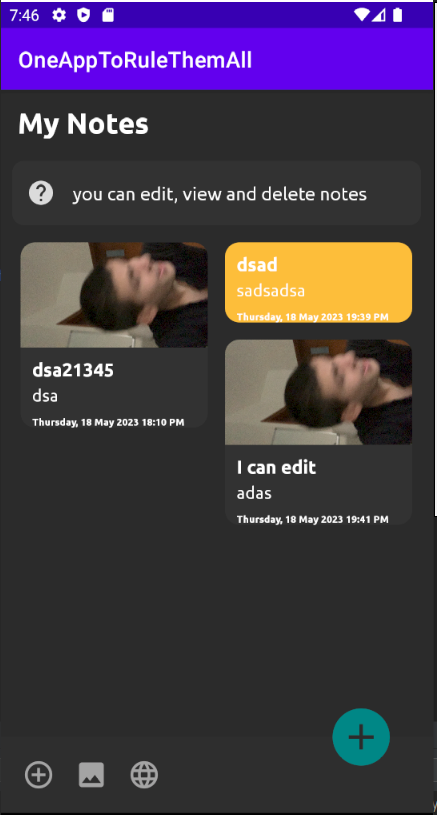
* מסך ביניים תפריט ראשי
* כפתור בודד לפתיחת גלגל בחירה.

### חלון עם בחירה.



* גלגל הבחירה ע"פ הוראות הפעלה.

### חלון Notes.

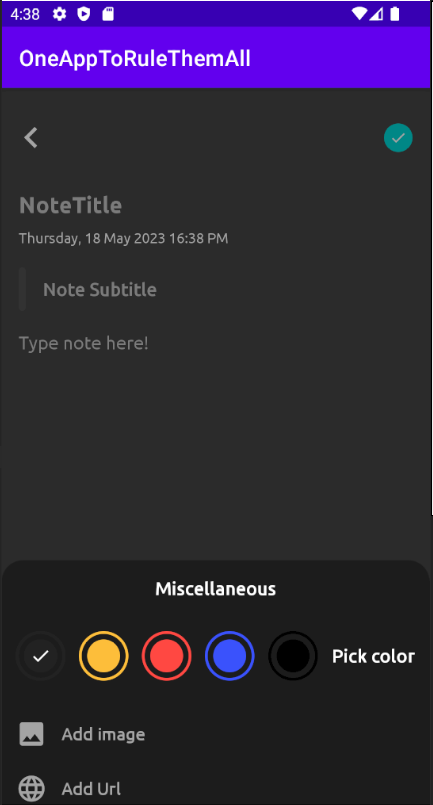


* חלון ההצגה של הפתקים הקיימים.
* בראש החלון יש פירוט הרשאה לכל משתמש מותאם אישית. בצד ימין למטה יש כפתור ליצירת פתק חדש.

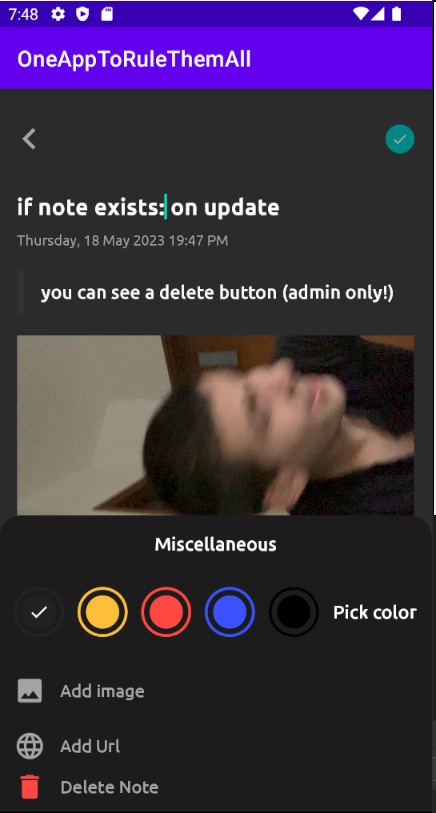
### חלון create note.

* לאחר לחיצה על +. חלון יצירת note חדש. (משמש גם לעדכון).

עבור לא אדמין:



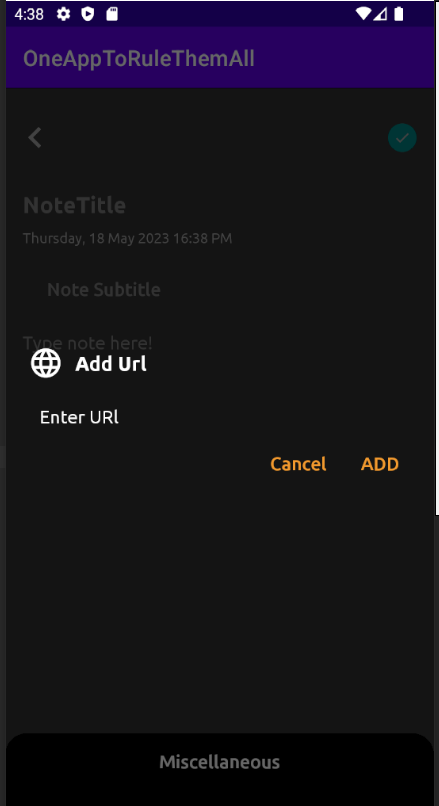
עבור אדמין:



**הערה חשובה:**

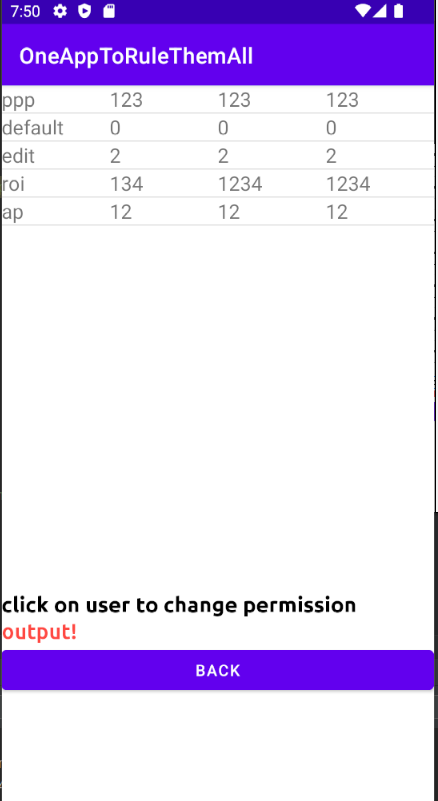
אותו דף של יצירת פתק משמש גם לצפייה ולעדכון בהתאם להוראות השימוש.

### דיאלוג URL.



* מקבל URL תקין בלבד.

### חלון result activity.



* בעמוד זה ניתן לשנות את ההרשאות של המשתמשים בלחיצה. מרגיל לעורך לאדמין וחוזר חלילה.

### הרשאות

פירוט ההרשאות שהאפליקציה דורשת והסבר על כל הרשאה.

העתיקו מה-AndroidManifest את השמות והסבירו מה כל אחד עושה.

דוגמא:

android.permission.VIBRATE – נדרשת כדי לאפשר שימוש ברטט במכשיר.

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE" />

.ACCESS\_NETWORK\_STATE -נדרש כדי לגשת למצב החיבור האינטרנט

READ\_EXTERNAL\_STORAGE- נדרש כדי לגשת לתמונות בטלפון

התאמה של תמונות לגרסה חדשה.

<queries>

<intent>

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

</intent>

</queries>

### דרישות מיוחדות ומגבלות

יחד עם הקבצים המוגשים מצורף קובץ מוסיקה שאותה משמיע MyService.

על מנת לצרף תמונות לפתק יש לצלם תמונות לגלריה של הטלפון או להעלות אותם ניתן לצלם גם באמולטור.

בנוסף, במהלך השימוש על מנת לגשת למצלמה יש דרישת אישור מהמשתמש. בקוד שלהלן:

// handling permissions

layoutMiscellaneous.findViewById(R.id.*layoutAddImage*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

bottomSheetBehavior.setState(BottomSheetBehavior.*STATE\_COLLAPSED*);

// requesting runtime storage permission

if(ContextCompat.*checkSelfPermission*(

getApplicationContext(), Manifest.permission.*READ\_EXTERNAL\_STORAGE*

)!= PackageManager.*PERMISSION\_GRANTED*){ // if access denied

ActivityCompat.*requestPermissions*(

CreateNoteActivity.this,

new String[] {Manifest.permission.*READ\_EXTERNAL\_STORAGE*},

*REQUEST\_CODE\_STORAGE\_PERMISSION*

);

}else { // access accepted

selectImage(); // func to pick image

}

}

});

// handling permission result

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if(requestCode == *REQUEST\_CODE\_STORAGE\_PERMISSION* && grantResults.length > 0){

if(grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){

selectImage();

}else{

Toast.*makeText*(this, "Permission Denied!",Toast.*LENGTH\_SHORT*).show();

}

}

}

#### גירסת Android מינימלית

Android \_\_\_\_\_ גירסת API 31

#### מכשירים עליהם נבדקה המערכת

ציינו על איזה מכשירים / אמולטורים נבדקה המערכת.

Pixel 3 API 31.

# תיאור תחום הידע

### אובייקטים נחוצים

ROOM

SQL

SHAREDPREFERENCES

### סוגי נתונים

USER

האובייקט הנ"ל מכיל את ההרשאה של כל משתמש ופרטים אישיים.

NOTE

אובייקט שמכיל את הפרטים:

תמונה

תאריך

טקסט (כותרת ולא כותרת)

צבע של הדף

כתובת URL

### ייצוג מידע

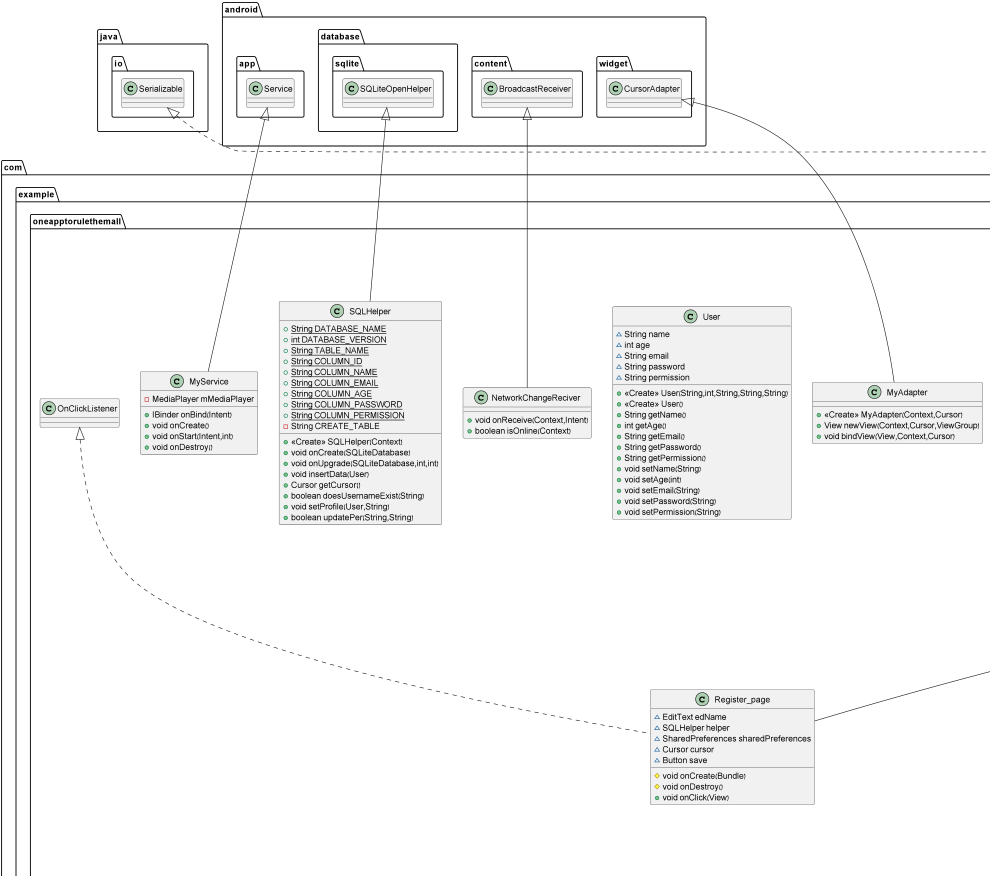
1. משתמשים שימוש במערכת שמירת מידע SQLite , שומר בטלפון. האפליקציה מתכוננת לשימוש לוקאלי ועל כן שימוש בSQLite מתאים. השימוש הוא בטבלה (דוגמאה בהמשך)
2. פתקים שימוש בROOM שהיא הרחבה של SQLite שמאפשר עידכון קל יותר של פרטים וזרימה של מידע. גם כן מוצג בטבלה דוגמא בהמשך.

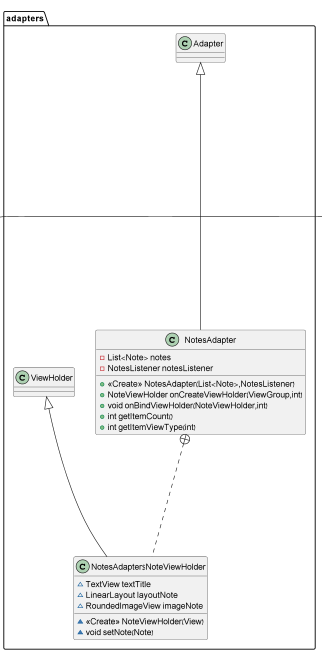
את המידע על הפתקים אני מציג דף הפתקים הנראה למעלה, ואת המידע של המשתמשים אני מראה לאדמינים בדף שינוי הרשאות ברשימה.

* הוספת מידע יש ב הוספת פתק חדש ובמשתמש חדש
* שינוי מידע שינוי הרשאה למשתמש וגם בשינוי מידע בפתק קיים ( ניתן גם להוסיף).
* מחיקה: ניתן למחוק פתקים.

# מבנה/ארכיטקטורה

## מחלקות הפרויקט וקשרים ביניהם





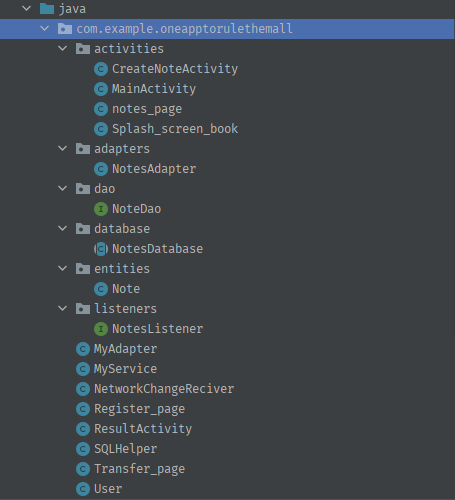
### 

### 

### 

### תרשים מחלקות

### מחלקות



### מחלקה A

* NotesAdaptor
* תפקיד ביניים בעיבוד נתוני הפתק.
* הסבר עבור כל תכונה ושימושה (כולל תכונות סטטיות)
* הסבר עבור כל פעולה ותפקידה (למעט get/set).

// creates holder for notes

public NoteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

@Override

public void onBindViewHolder(@NonNull NoteViewHolder holder, int position) {

holder.setNote(notes.get(position));

// listener

holder.layoutNote.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

notesListener.onNotesClicked(notes.get(position), position);

}

});

}

משמש עבור האזנה ללחיצות על פתקים.

void setNote

פונקציה שמגדירה את האובייקט NOTE.

הקוד מלא בנספחים.

### מחלקה B

* NoteDao
* קביעת הQuery עבור השימוש בROOM ולהגדיר התנהגות לקונפליקטים.

### מחלקה C

* NoteDatabase

המחלקה שמגדירה את בסיס הנתונים של הפתקים. יורשת את ROOM ומאפשרת שימוש בו.

### מחלקה D

* Note
* המחלקה מגדירה אובייקט חדש: פתק. ומגדיר את כל הנתונים שלעתיד יהפכו עמודות בטבלה של שמירת הנתונים.
* המחלקה Note היא מעין תת טבלה, כל אובייקט מכיל את כל הנתונים שלו.

**מחלקה E**

* NoteListener
* מגדירה מאזין לדף הפתקים.

**מחלקה F**

* MyAdapter
* המחלקה מהווה חלק בעיבוד נתונים של המשתמש. משמש על מנת לשלוף נתונים.

**מחלקה G**

* MyService
* מחלקה שיורשת את מחלקת האב Service, ומממשת באצאותה ניגון מוזיקה.
* פונקצית יצירה מתחילה את המוזיקה
* פונקצית הריסה מפסיקה את המוזיקה.

**מחלקה H**

* NetworkChangeReciver
* מחלקה שיורדת את BroadcastReceiver
* המחלקה ממשת בדיקת חיבור לאינטרנט.
* הפונקציה OnRecive קוראת ל isOnline וממשת טוסט שמציג את הנתונים על אינטרנט.

**~~מחלקה I~~**

**מחלקה J**

* SQLHelper
* נתונים המוגדרים בהתחלה הם לשימוש קל בשמות העמודות בטבלה של SQL.
* onUpgrade
* בעת עדכון טבלה מוחקים את הטבלה ויוצרים מחדש.

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int newVersion) {

sqLiteDatabase.execSQL("DROP TABLE IF EXISTS "+ *TABLE\_NAME*);

onCreate(sqLiteDatabase);

}

* insertData
* מגדיר hash map מכניס אלייה את הנתונים ופורש אותם בטבלה בעזרת פונק של SQL.

public void insertData(User user){

SQLiteDatabase sqLiteDatabase = getWritableDatabase();

ContentValues values = new ContentValues(); //hash map

values.put(*COLUMN\_NAME*,user.getName());

values.put(*COLUMN\_EMAIL*,user.getEmail());

values.put(*COLUMN\_AGE*,user.getAge());

values.put(*COLUMN\_PASSWORD*,user.getPassword());

values.put(*COLUMN\_PERMISSION*,user.getPermission());

sqLiteDatabase.insert(*TABLE\_NAME*,null,values); //nullcolumnhack default value

}

* doesUsernameExist
* בהינתן שם בודק קיימות (כדי לא לנסות לשלוף משהו לא קיים וכדי לא ליצור כפילויות)

public boolean doesUsernameExist(String name) {

SQLiteDatabase db = this.getReadableDatabase();

String query = "SELECT \* FROM " + *TABLE\_NAME* + " WHERE "+ *COLUMN\_NAME* + " = '" + name + "';";

Cursor userCursor = db.rawQuery(query , null); // applying the query

boolean ans = userCursor.getCount() > 0;

userCursor.close();

return ans;

}

* setProfile
* משתמש בקלאס קיים של USER לממש בעזרתו "פרופיל" שליפה מלאה של נתונים לשימוש קל בקוד.
* updatePer
* מעדכן את עמודת ההרשאות של משתמש נתון ועדכון הטבלה.

public boolean updatePer(String username, String permission){

SQLiteDatabase db = this.getReadableDatabase();

String query = "SELECT \* FROM " + *TABLE\_NAME* + " WHERE " + *COLUMN\_NAME* + " = '" + username + "';";

Cursor userCursor = db.rawQuery(query , null);

userCursor.moveToFirst();

ContentValues cv = new ContentValues();

cv.put(*COLUMN\_NAME*, username);

cv.put(*COLUMN\_EMAIL*,userCursor.getString(2));

cv.put(*COLUMN\_PASSWORD*,userCursor.getString(3) );

cv.put(*COLUMN\_PERMISSION*, permission);

cv.put(*COLUMN\_AGE*, userCursor.getString(5));

db.update(*TABLE\_NAME*, cv, "name=?", new String[]{username});

return true;

}

**מחלקה K**

* USER
* מחלקה שמגדירה את USER, טיפוס נתונים בעל:
* שם, גיל , סיסמא , מייל, הרשאה.

## שמירת נתונים

מערכת המשתמשים:

את הנתונים על המשתמש אני שמרתי בSQLite , בחרתי באופציה זו מכיוון שכוונתי ליצור אפליקציה לוקלית (לשימוש במכשיר אחד). את הנתונים אני שומר בטבלה USERS ואני שולף את הנותנים על פי הצורך. הדבר המפריד בין המשתמשים הוא שמם, לא יכול להיות שם כפול בטבלה. על פי השם אני יצרתי פונקציה לקריאת נתונים מהטבלה. על פי השם יצרתי גם פונקציה לעדכן את ההרשאות בטבלה. את ההרשאות אני שומר ב shared preferences כך שהמשתמש האקטיבי ( זה שהכניס את הסיסמא אחרון) הרשאתו רשומה ב SP וקריאה לי בעת פעולת באפליקציה.

אני משתמש בSP כדי לקבוע איזה פעולות יכול או לא יכול משתמש לעשות או למנוע כניסה לדפים "מסווגים" , לדוגמא:

// sp for admin actions

sharedPreferences = getSharedPreferences("active\_user",0);

String name = sharedPreferences.getString("name", null);

if(name.equals("default")){

Toast.*makeText*(this, "editor or admin action!", Toast.*LENGTH\_SHORT*).show();

return;

}

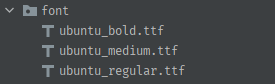
על מנת לשמור את הפתקים רציתי לנסות הרחבה של SQLite שנקראת ROOM, חקרתי באינטרנט וגיליתי שROOM מאפשר עדכון קל יותר של פרטים וזרימה של מידע. על כן בחרתי להשתמש בו. אחד מהפיצ'רים השימושים ביותר הוא דרכי עבודה עם קונפליקטים. יכלתי להשתמש בקונפליקט על מנת לעדכן את הטבלה בקלות רבה יותר ויעילה יותר.

על מנת להציג את הדפים השתמשתי בדף הכנסת המידע ויצרתי פונקציות שיטענו את המידע על המסך יאפשרו שינוים ויקראו מחדש את הנתונים לעדכון. את המחיקה אפשרתי רק לאדמינים,ב ROOM המימוש היה יחסית קל וקריא בקוד.

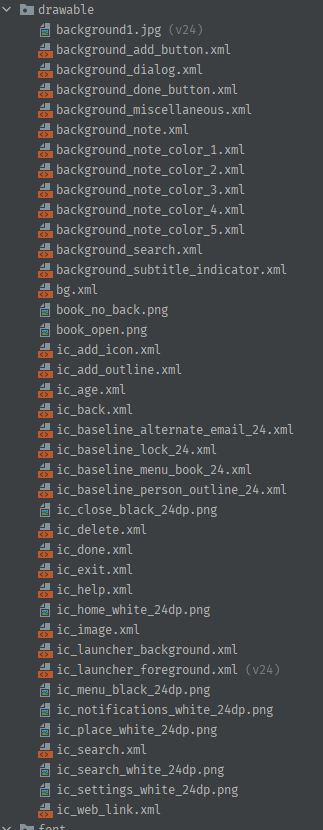
## קבצים

כאמור, יש להוסיף קובץ שמע על מנת להפעיל את ה Service.

בנוסף, ניתן להוסיף קבצים לספריית הטלפון ולהציגם בפתקים.







### Shared Preference

<string name="name">admin</string>

// sp for admin actions

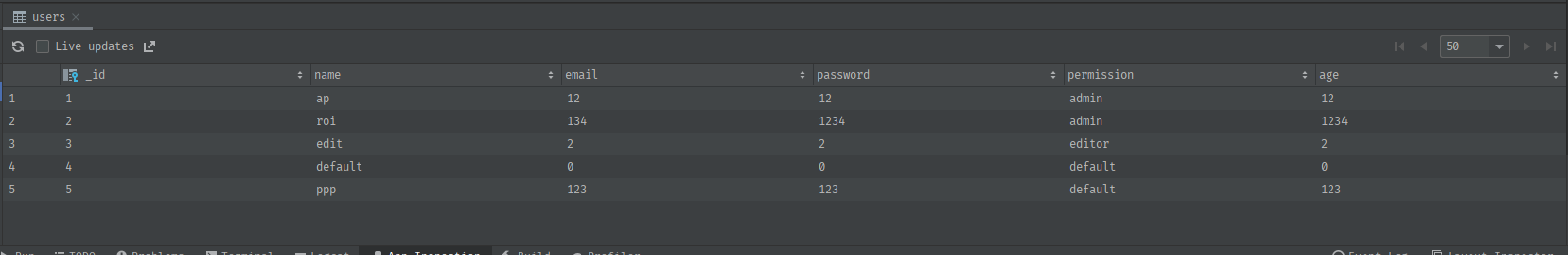
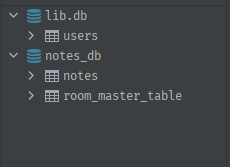
sharedPreferences = getSharedPreferences("active\_user",0);

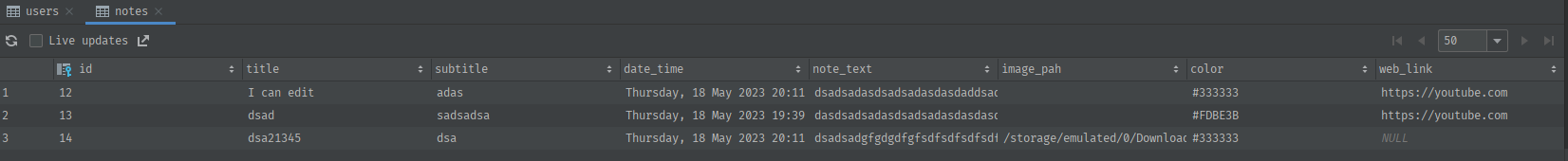
String name = sharedPreferences.getString("name", null);

// in this code name holds the permission!

השם: name הוא כולל את ההרשאה של המשתמש האקטיבי. (האחרון)

## סכמת Database



**

# רפלקציה אישית

* על הפרויקט עבדתי במהלך תקופה ארוכה מאוד, כאשר כל הזמן עמדו בפניי אתגרים שונים כמו בגרויות וכיוצא באלו. עם זאת הצלחתי להקדיש מעט זמן מכל שבוע כדי לעבוד על הפרויקט או לחשוב על רעיונות או לצפות בסרטוני הסברים.
* במהלך הלמידה למדתי כיצד להשתמש ב dependencies ליבא קוד וגם הרבה מאוד לדבג!
* את מערכת הנתונים ROOM לא למדנו בכיתה ולכן היה אתגר נוסף של ללמוד את המערכת הזו.
* למדתי מהפרויקט עצמאות גדולה יותר בכתיבת קוד וגם בשיתפו עם חברים שעזרו לי בדרך.
* הופתעתי מאוד מכמה שלדבר עם חברים אפילו ל30 שניות עזר לי לשפר את הקוד להפוך אותו לקריא יותר או לפתור באג.
* אם היה ברשותי עוד זמן הייתי מוסיף עוד פיצ'רים שרציתי לעשות, כמו חיפוש פתק או סימון פתק בכוכבית. כמו כן רעיון שחשבתי עליו בדיעבד הוא לייצר עוד חלוקה בין הפתקים מלבד שיטת הצבעים.
* אני כן מעוניין להמשיך את הפרויקט אחרי הלמידה בביהס. אני חושב שלנסות להמיר את הפרויקט להיות אינטרנטי הוא אתגר טוב להמשך הניסיון שלי כמתכנת.
* אני חושב שלעשות את הפרויקט היה חוויה מאוד מלמדת ובונה שאפשרה לי יצירתיות ולמידה גם יחד.

## נספחים:

<https://stackoverflow.com/questions/32431723/read-external-storage-permission-for-android>

<https://stackoverflow.com/questions/60679685/what-does-query-all-packages-permission-do>

<https://developer.android.com/guide/topics/manifest/queries-element#intent>

<https://github.com/JimSeker/threads/blob/main/SimpleThreadDemo/app/src/main/java/edu/cs4730/simplethreaddemo/MainActivity.java>

## קוד המקור (source code)

### מחלקה A

package com.example.oneapptorulethemall.adapters;

import android.graphics.BitmapFactory;

import android.graphics.Color;

import android.graphics.drawable.GradientDrawable;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.LinearLayout;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.example.oneapptorulethemall.R;

import com.example.oneapptorulethemall.entities.Note;

import com.example.oneapptorulethemall.listeners.NotesListener;

import com.makeramen.roundedimageview.RoundedImageView;

import java.util.List;

public class NotesAdapter extends RecyclerView.Adapter<NotesAdapter.NoteViewHolder>{

private List<Note> notes;

// listener

private NotesListener notesListener;

public NotesAdapter(List<Note> notes, NotesListener notesListener) {

this.notes = notes;

this.notesListener = notesListener;

}

// methods

@NonNull

@Override

// creates holder for notes

public NoteViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

return new NoteViewHolder(

LayoutInflater.*from*(parent.getContext()).inflate(

R.layout.*item\_container\_note*,

parent,

false

)

);

}

@Override

public void onBindViewHolder(@NonNull NoteViewHolder holder, int position) {

holder.setNote(notes.get(position));

// listener

holder.layoutNote.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

notesListener.onNotesClicked(notes.get(position), position);

}

});

}

@Override

public int getItemCount() {

return notes.size();

}

@Override

public int getItemViewType(int position) {

return position;

}

static class NoteViewHolder extends RecyclerView.ViewHolder{

TextView textTitle, textSubtitle, textDateTime;

LinearLayout layoutNote;

RoundedImageView imageNote;

NoteViewHolder(@NonNull View itemView) {

super(itemView);

textTitle = itemView.findViewById(R.id.*textTitle*); // needed to import R class

textSubtitle = itemView.findViewById(R.id.*textSubtitle*);

textDateTime = itemView.findViewById(R.id.*textDateTime*);

layoutNote = itemView.findViewById(R.id.*layoutNote*); // from notes container

imageNote = itemView.findViewById(R.id.*imageNote*);

}

void setNote(Note note){

textTitle.setText(note.getTitle());

if(note.getSubtitle().trim().isEmpty()){

// no subtitle available

textSubtitle.setVisibility(View.*GONE*);

}else{

textSubtitle.setText(note.getSubtitle());

}

textDateTime.setText(note.getDateTime());

// color

GradientDrawable gradientDrawable = (GradientDrawable) layoutNote.getBackground();

if(note.getColor() != null){

gradientDrawable.setColor(Color.*parseColor*(note.getColor())); // if selected update

}else{

gradientDrawable.setColor(Color.*parseColor*("#333333")); // default

}

if( note.getImage\_pah() != null){

imageNote.setImageBitmap(BitmapFactory.*decodeFile*(note.getImage\_pah()));

imageNote.setVisibility(View.*VISIBLE*);

}else{

imageNote.setVisibility(View.*GONE*);

}

}

}

}

### מחלקה B

@Dao

public interface NoteDao {

@Query("SELECT \* FROM notes ORDER BY id DESC")

List<Note> getAllNotes();

// onConflict = onConflictStratagy.REPLACE

@Insert(onConflict = OnConflictStrategy.*REPLACE*)

void insertNote(Note note);

@Delete

void deleteNote(Note note);

}

### מחלקה C

@Database( entities = Note.class, version = 1, exportSchema = false)

// inherits from Room database

public abstract class NotesDatabase extends RoomDatabase {

private static NotesDatabase *notesDatabase*;

public static synchronized NotesDatabase getDataBase(Context context){

if(*notesDatabase* == null){

*notesDatabase* = Room.*databaseBuilder*(

context,

NotesDatabase.class,

"notes\_db"

).build();

}

return *notesDatabase*;

}

public abstract NoteDao noteDao();

}

### מחלקה D

@Entity (tableName = "notes")

public class Note implements Serializable {

// table

@PrimaryKey(autoGenerate = true)

private int id;

@ColumnInfo(name= "title")

private String title;

@ColumnInfo(name= "subtitle")

private String subtitle;

@ColumnInfo(name= "date\_time")

private String dateTime;

@ColumnInfo(name= "note\_text")

private String note\_text;

@ColumnInfo(name= "image\_pah")

private String image\_pah;

@ColumnInfo(name= "color")

private String color;

@ColumnInfo(name= "web\_link")

private String web\_link;

// getters and setters

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getSubtitle() {

return subtitle;

}

public void setSubtitle(String subtitle) {

this.subtitle = subtitle;

}

public String getDateTime() {

return dateTime;

}

public void setDateTime(String dateTime) {

this.dateTime = dateTime;

}

public String getNote\_text() {

return note\_text;

}

public void setNote\_text(String note\_text) {

this.note\_text = note\_text;

}

public String getImage\_pah() {

return image\_pah;

}

public void setImage\_pah(String image\_pah) {

this.image\_pah = image\_pah;

}

public String getColor() {

return color;

}

public void setColor(String color) {

this.color = color;

}

public String getWeb\_link() {

return web\_link;

}

public void setWeb\_link(String web\_link) {

this.web\_link = web\_link;

}

@NonNull

@Override

public String toString() {

return title + " : " + dateTime;

}

}

### מחלקה E

public interface NotesListener {

void onNotesClicked(Note note, int position);

}

### מחלקה F

class MyAdapter extends CursorAdapter {

public MyAdapter(Context context, Cursor c) {

super(context, c, true);

}

@Override

public View newView(Context context, Cursor cursor, ViewGroup viewGroup) {

return LayoutInflater.*from*(context).inflate(R.layout.*row\_item*, null);

}

@SuppressLint("Range")

@Override

public void bindView(View view, Context context, Cursor cursor) {

TextView tvName = view.findViewById(R.id.*tvName*);

TextView tvAge = view.findViewById(R.id.*tvAge*);

TextView tvEmail = view.findViewById(R.id.*tvEmail*);

TextView tvPass = view.findViewById(R.id.*tvPass*);

tvName.setText(cursor.getString(cursor.getColumnIndex(SQLHelper.*COLUMN\_NAME*)));

tvAge.setText(cursor.getInt(cursor.getColumnIndex(SQLHelper.*COLUMN\_AGE*)) +"");

tvEmail.setText(cursor.getString(cursor.getColumnIndex(SQLHelper.*COLUMN\_EMAIL*)));

tvPass.setText(cursor.getString(cursor.getColumnIndex(SQLHelper.*COLUMN\_PASSWORD*)));

}

}

### מחלקה G

public class MyService extends Service {

private MediaPlayer mMediaPlayer;

@Nullable

@Override

public IBinder onBind(Intent intent) {

return null;

}

@Override

public void onCreate() {

mMediaPlayer = MediaPlayer.*create*(getApplicationContext(), R.raw.*music1* );

mMediaPlayer.setLooping(false);

Toast.*makeText*(this, "Service created", Toast.*LENGTH\_SHORT*).show();

}

@Override

public void onStart(Intent intent, int startId) {

//super.onStart(intent, startId);

mMediaPlayer.start();

Toast.*makeText*(this, "Service start", Toast.*LENGTH\_SHORT*).show();

}

public void onDestroy(){

mMediaPlayer.stop();

Toast.*makeText*(this, "Service stopped", Toast.*LENGTH\_SHORT*).show();

}

}

### מחלקה H

public class NetworkChangeReciver extends BroadcastReceiver {

@Override

public void onReceive(Context context, Intent intent) {

try {

if (isOnline(context)) {

Toast.*makeText*(context, "Internet Connected", Toast.*LENGTH\_SHORT*).show();

} else {

Toast.*makeText*(context, "Internet NOT Connected", Toast.*LENGTH\_SHORT*).show();

}

} catch (NullPointerException e) {

e.printStackTrace();

}

}

public boolean isOnline(Context context){

try {

ConnectivityManager cm = (ConnectivityManager) context.getSystemService(context.*CONNECTIVITY\_SERVICE*);

NetworkInfo info = cm.getActiveNetworkInfo();

boolean up = false;

if(info != null && info.isConnected()){

up = true;

}

return (info!=null&&up);

}

catch (NullPointerException e){

e.printStackTrace();

return false;

}

}

}

### ~~מחלקה I~~

### מחלקה J

public class SQLHelper extends SQLiteOpenHelper {

public static final String *DATABASE\_NAME* ="lib.db";

public static final int *DATABASE\_VERSION* = 1;

public static final String *TABLE\_NAME*="users";

public static final String *COLUMN\_ID*="\_id";

public static final String *COLUMN\_NAME*="name";

public static final String *COLUMN\_EMAIL*="email";

public static final String *COLUMN\_AGE*="age";

public static final String *COLUMN\_PASSWORD*="password";

public static final String *COLUMN\_PERMISSION*="permission";

private final String CREATE\_TABLE="CREATE TABLE " +*TABLE\_NAME*+ " ( "

+ *COLUMN\_ID* +" INTEGER PRIMARY KEY AUTOINCREMENT, "

+ *COLUMN\_NAME* + " TEXT, "

+ *COLUMN\_EMAIL* + " TEXT, "

+ *COLUMN\_PASSWORD* + " TEXT, "

+ *COLUMN\_PERMISSION* + " TEXT, "

+ *COLUMN\_AGE* + " INTEGER );";

public SQLHelper(Context context) {

super(context, *DATABASE\_NAME*, null, *DATABASE\_VERSION*);

}

@Override

public void onCreate(SQLiteDatabase sqLiteDatabase) {

sqLiteDatabase.execSQL(CREATE\_TABLE);

}

@Override

public void onUpgrade(SQLiteDatabase sqLiteDatabase, int oldVersion, int newVersion) {

sqLiteDatabase.execSQL("DROP TABLE IF EXISTS "+ *TABLE\_NAME*);

onCreate(sqLiteDatabase);

}

public void insertData(User user){

SQLiteDatabase sqLiteDatabase = getWritableDatabase();

ContentValues values = new ContentValues(); //hash map

values.put(*COLUMN\_NAME*,user.getName());

values.put(*COLUMN\_EMAIL*,user.getEmail());

values.put(*COLUMN\_AGE*,user.getAge());

values.put(*COLUMN\_PASSWORD*,user.getPassword());

values.put(*COLUMN\_PERMISSION*,user.getPermission());

sqLiteDatabase.insert(*TABLE\_NAME*,null,values); //nullcolumnhack default value

}

public Cursor getCursor(){

Cursor cursor = getWritableDatabase().query(

*TABLE\_NAME*,

null,

null,

null,

null,

null,

*COLUMN\_ID*+ " DESC "

);

return cursor;

}

public boolean doesUsernameExist(String name) {

SQLiteDatabase db = this.getReadableDatabase();

String query = "SELECT \* FROM " + *TABLE\_NAME* + " WHERE "+ *COLUMN\_NAME* + " = '" + name + "';";

Cursor userCursor = db.rawQuery(query , null); // applying the query

boolean ans = userCursor.getCount() > 0;

userCursor.close();

return ans;

}

public void setProfile(User profile,String username){

SQLiteDatabase db = this.getReadableDatabase();

String query = "SELECT \* FROM " + *TABLE\_NAME* + " WHERE " + *COLUMN\_NAME* + " = '" + username + "';";

Cursor userCursor = db.rawQuery(query , null);

userCursor.moveToFirst();

profile.setName(userCursor.getString(1));

profile.setEmail(userCursor.getString(2));

profile.setPassword(userCursor.getString(3));

profile.setPermission(userCursor.getString(4));

profile.setAge(userCursor.getInt(5));

userCursor.close();

}

public boolean updatePer(String username, String permission){

SQLiteDatabase db = this.getReadableDatabase();

String query = "SELECT \* FROM " + *TABLE\_NAME* + " WHERE " + *COLUMN\_NAME* + " = '" + username + "';";

Cursor userCursor = db.rawQuery(query , null);

userCursor.moveToFirst();

ContentValues cv = new ContentValues();

cv.put(*COLUMN\_NAME*, username);

cv.put(*COLUMN\_EMAIL*,userCursor.getString(2));

cv.put(*COLUMN\_PASSWORD*,userCursor.getString(3) );

cv.put(*COLUMN\_PERMISSION*, permission);

cv.put(*COLUMN\_AGE*, userCursor.getString(5));

db.update(*TABLE\_NAME*, cv, "name=?", new String[]{username});

return true;

}

}

### מחלקה K

public class User {

String name;

int age;

String email;

String password;

String permission;

//ctrl + insert creates this constructor automatically

public User(String name, int age, String email, String password, String permission) {

this.name = name;

this.age = age;

this.email = email;

this.password = password;

this.permission = permission;

}

public User() {

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

public String getEmail() {

return email;

}

public String getPassword() {

return password;

}

public String getPermission() {

return permission;

}

public void setName(String name) {

this.name = name;

}

public void setAge(int age) {

this.age = age;

}

public void setEmail(String email) {

this.email = email;

}

public void setPassword(String password) {

this.password = password;

}

public void setPermission(String permission) {

this.permission = permission;

}

}

### Activities (XML & java)

#### CreateNoteActivity:

public class CreateNoteActivity extends AppCompatActivity {

// access CONST

private static final int *REQUEST\_CODE\_STORAGE\_PERMISSION* =555;

private static final int *REQUEST\_CODE\_SELECT\_IMAGE* = 2;

// input

private EditText inputNoteTitle, inputNoteSubtitle, inputNoteText;

// local

private TextView textDateTime;

// color

private View viewSubtitleIndicator;

private String selectedNoteColor;

// image

private ImageView imageNote;

private String selectedImagePath;

// url

private TextView textWebURL;

private LinearLayout layoutWebUrl;

private AlertDialog dialogAddURL;

// update note

private Note alreadyAvailableNote;

// for admin actions

SharedPreferences sharedPreferences;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_create\_note*);

ImageView imageBack = findViewById(R.id.*imageBack*);

imageBack.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

onBackPressed();

}

});

//defult note color

selectedNoteColor = "#333333";

selectedImagePath ="";

// admin actions

sharedPreferences = getSharedPreferences("active\_user",0);

// input

inputNoteTitle = findViewById(R.id.*inputNoteTitle*);

inputNoteSubtitle = findViewById(R.id.*inputNoteSubtitle*);

inputNoteText = findViewById(R.id.*inputNote*);

textDateTime = findViewById(R.id.*textDateTime*);

// color Indicator

viewSubtitleIndicator = findViewById(R.id.*viewSubtitleIndicator*);

// image integration

imageNote = findViewById(R.id.*imageNote*);

// url

textWebURL = findViewById(R.id.*textWebURL*);

layoutWebUrl =findViewById(R.id.*layoutWebURL*);

// Time and date management

textDateTime.setText(

new SimpleDateFormat("EEEE, dd MMMM yyyy HH:mm a", Locale.*getDefault*())

.format(new Date()) // using java util

);

ImageView imageSave = findViewById(R.id.*imageSave*);

imageSave.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

saveNote();

}

});

if(getIntent().getBooleanExtra("isViewOrUpdate", false)){

alreadyAvailableNote = (Note) getIntent().getSerializableExtra("note");

setViewOrUpdateNote();

}

// miscellaneous tab support for tap and drag

initMiscellaneous();

// update color Indicator

setSubtitleIndicatorColor();

}

// update note

private void setViewOrUpdateNote(){

inputNoteTitle.setText(alreadyAvailableNote.getTitle());

inputNoteSubtitle.setText(alreadyAvailableNote.getSubtitle());

inputNoteText.setText(alreadyAvailableNote.getNote\_text());

if(alreadyAvailableNote.getImage\_pah() != null &&

!alreadyAvailableNote.getImage\_pah().trim().isEmpty()){

Log.*d*("ee", "pic");

imageNote.setImageBitmap(BitmapFactory.*decodeFile*(alreadyAvailableNote.getImage\_pah()));

imageNote.setVisibility(View.*VISIBLE*);

selectedImagePath = alreadyAvailableNote.getImage\_pah();

}

if(alreadyAvailableNote.getWeb\_link() != null &&

!alreadyAvailableNote.getWeb\_link().trim().isEmpty()){

textWebURL.setText(alreadyAvailableNote.getWeb\_link());

layoutWebUrl.setVisibility(View.*VISIBLE*);

}

}

private void saveNote(){

// sp for admin actions

sharedPreferences = getSharedPreferences("active\_user",0);

String name = sharedPreferences.getString("name", null);

if(name.equals("default")){

Toast.*makeText*(this, "editor or admin action!", Toast.*LENGTH\_SHORT*).show();

return;

}

// validations

// trims removes spaces!

if(inputNoteTitle.getText().toString().trim().isEmpty()){

Toast.*makeText*(this, "Note title can't be empty!", Toast.*LENGTH\_SHORT*).

show();

return;

}else if(inputNoteSubtitle.getText().toString().trim().isEmpty() &&

inputNoteText.getText().toString().trim().isEmpty()){

Toast.*makeText*(this, "Note can't be empty!", Toast.*LENGTH\_SHORT*).

show();

return;

}

// preparing note object to save in data base

final Note note = new Note();

note.setTitle(inputNoteTitle.getText().toString());

note.setSubtitle(inputNoteSubtitle.getText().toString());

note.setNote\_text(inputNoteText.getText().toString());

note.setDateTime(textDateTime.getText().toString());

note.setColor(selectedNoteColor); // saves color selected

note.setImage\_pah(selectedImagePath);

/\*

Room doesn't allow data base operation on the main thread

therefore we use async task to save notes

\*/

if(layoutWebUrl.getVisibility() == View.*VISIBLE*){ // if true URL has been added

note.setWeb\_link(textWebURL.getText().toString());

}

// update note

if(alreadyAvailableNote != null){

note.setId(alreadyAvailableNote.getId());

}

/\*

\* setting new id of new note from an already available note

\* onConflictStrategy is "REPLACE" so its allowed! (NotesDAO)

\*

\* \*/

@SuppressLint("StaticFieldLeak") // ?

class SaveNoteTask extends AsyncTask<Void, Void, Void>{

@Override

protected Void doInBackground(Void... voids) {

// note dao requires input of class note

NotesDatabase.*getDataBase*(getApplicationContext()).noteDao().insertNote(note);

return null;

}

@Override

protected void onPostExecute(Void unused) {

// returns when done to update data base

super.onPostExecute(unused);

Intent intent = new Intent();

setResult(*RESULT\_OK*, intent);

finish();

}

}

new SaveNoteTask().execute();

}

// notes colors + url

private void initMiscellaneous(){

final LinearLayout layoutMiscellaneous = findViewById(R.id.*layoutMiscellaneous*);

final BottomSheetBehavior<LinearLayout> bottomSheetBehavior = BottomSheetBehavior.*from*(layoutMiscellaneous);

layoutMiscellaneous.findViewById(R.id.*textMiscellaneous*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(bottomSheetBehavior.getState() != BottomSheetBehavior.*STATE\_EXPANDED*){

bottomSheetBehavior.setState(BottomSheetBehavior.*STATE\_EXPANDED*);

}else{

bottomSheetBehavior.setState(BottomSheetBehavior.*STATE\_COLLAPSED*);

}

}

});

// sp for admin actions

String name = sharedPreferences.getString("name", null);

// buttons

final ImageView imageColor1 = layoutMiscellaneous.findViewById(R.id.*imageColor1*);

final ImageView imageColor2 = layoutMiscellaneous.findViewById(R.id.*imageColor2*);

final ImageView imageColor3 = layoutMiscellaneous.findViewById(R.id.*imageColor3*);

final ImageView imageColor4 = layoutMiscellaneous.findViewById(R.id.*imageColor4*);

final ImageView imageColor5 = layoutMiscellaneous.findViewById(R.id.*imageColor5*);

layoutMiscellaneous.findViewById(R.id.*viewColor1*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

selectedNoteColor = "#333333";

// display selection

imageColor1.setImageResource(R.drawable.*ic\_done*);

imageColor2.setImageResource(0);

imageColor3.setImageResource(0);

imageColor4.setImageResource(0);

imageColor5.setImageResource(0);

setSubtitleIndicatorColor();

}

});

layoutMiscellaneous.findViewById(R.id.*viewColor2*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

selectedNoteColor = "#FDBE3B";

// display selection

imageColor1.setImageResource(0);

imageColor2.setImageResource(R.drawable.*ic\_done*);

imageColor3.setImageResource(0);

imageColor4.setImageResource(0);

imageColor5.setImageResource(0);

setSubtitleIndicatorColor();

}

});

layoutMiscellaneous.findViewById(R.id.*viewColor3*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

selectedNoteColor = "#FF4842";

// display selection

imageColor1.setImageResource(0);

imageColor2.setImageResource(0);

imageColor3.setImageResource(R.drawable.*ic\_done*);

imageColor4.setImageResource(0);

imageColor5.setImageResource(0);

setSubtitleIndicatorColor();

}

});

layoutMiscellaneous.findViewById(R.id.*viewColor4*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

selectedNoteColor = "#3A52Fc";

// display selection

imageColor1.setImageResource(0);

imageColor2.setImageResource(0);

imageColor3.setImageResource(0);

imageColor4.setImageResource(R.drawable.*ic\_done*);

imageColor5.setImageResource(0);

setSubtitleIndicatorColor();

}

});

layoutMiscellaneous.findViewById(R.id.*viewColor5*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

selectedNoteColor = "#000000";

// display selection

imageColor1.setImageResource(0);

imageColor2.setImageResource(0);

imageColor3.setImageResource(0);

imageColor4.setImageResource(0);

imageColor5.setImageResource(R.drawable.*ic\_done*);

setSubtitleIndicatorColor();

}

});

if(alreadyAvailableNote != null && alreadyAvailableNote.getColor() != null &&

!alreadyAvailableNote.getColor().trim().isEmpty()){

switch (alreadyAvailableNote.getColor()){

case "#FDBE3B":

layoutMiscellaneous.findViewById(R.id.*viewColor2*).performClick();

break;

case "#FF4842":

layoutMiscellaneous.findViewById(R.id.*viewColor3*).performClick();

break;

case "#3A52Fc":

layoutMiscellaneous.findViewById(R.id.*viewColor4*).performClick();

break;

case "#000000":

layoutMiscellaneous.findViewById(R.id.*viewColor5*).performClick();

break;

}

}

// handling permissions

layoutMiscellaneous.findViewById(R.id.*layoutAddImage*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

bottomSheetBehavior.setState(BottomSheetBehavior.*STATE\_COLLAPSED*);

// requesting runtime storage permission

if(ContextCompat.*checkSelfPermission*(

getApplicationContext(), Manifest.permission.*READ\_EXTERNAL\_STORAGE*

)!= PackageManager.*PERMISSION\_GRANTED*){ // if access denied

ActivityCompat.*requestPermissions*(

CreateNoteActivity.this,

new String[] {Manifest.permission.*READ\_EXTERNAL\_STORAGE*},

*REQUEST\_CODE\_STORAGE\_PERMISSION*

);

}else { // access accepted

selectImage(); // func to pick image

}

}

});

// url:

layoutMiscellaneous.findViewById(R.id.*layoutAddUrl*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

bottomSheetBehavior.setState(BottomSheetBehavior.*STATE\_COLLAPSED*);

showAddURLDialog();

}

});

if(alreadyAvailableNote != null){ // and if user is admin!!

layoutMiscellaneous.findViewById(R.id.*layoutDeleteNote*).setVisibility(View.*VISIBLE*);

layoutMiscellaneous.findViewById(R.id.*layoutDeleteNote*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

@SuppressLint("StaticFieldLeak")

class DeleteNoteTask extends AsyncTask <Void, Void, Void>{

@Override

protected Void doInBackground(Void... voids) {

NotesDatabase.*getDataBase*(getApplicationContext()).noteDao()

.deleteNote(alreadyAvailableNote);

return null;

}

@Override

protected void onPostExecute(Void aVoid) {

super.onPostExecute(aVoid);

Intent intent = new Intent();

intent.putExtra("isNoteDeleted", true);

setResult(*RESULT\_OK*, intent);

finish();

}

}

new DeleteNoteTask().execute();

}

});

}

}

// set Indicator color

private void setSubtitleIndicatorColor(){

GradientDrawable gradientDrawable = (GradientDrawable) viewSubtitleIndicator.getBackground();

gradientDrawable.setColor(Color.*parseColor*(selectedNoteColor));

}

// picking image

private void selectImage(){

Log.*d*("test\_selectImage", "enter");

Intent intent = new Intent(Intent.*ACTION\_PICK*, MediaStore.Images.Media.*EXTERNAL\_CONTENT\_URI*);

if(intent.resolveActivity(getPackageManager()) != null){ // logic?

Log.*d*("test\_selectImage", "enter2");

startActivityForResult(intent, *REQUEST\_CODE\_SELECT\_IMAGE*);

}

}

// handling permission result

@Override

public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

super.onRequestPermissionsResult(requestCode, permissions, grantResults);

if(requestCode == *REQUEST\_CODE\_STORAGE\_PERMISSION* && grantResults.length > 0){

if(grantResults[0] == PackageManager.*PERMISSION\_GRANTED*){

selectImage();

}else{

Toast.*makeText*(this, "Permission Denied!",Toast.*LENGTH\_SHORT*).show();

}

}

}

// result from selected image

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if(requestCode == *REQUEST\_CODE\_SELECT\_IMAGE* && resultCode == *RESULT\_OK*){

if(data != null){

Uri selectedImageUri = data.getData();

if(selectedImageUri != null){

try {

InputStream inputStream = getContentResolver().openInputStream(selectedImageUri);

Bitmap bitmap = BitmapFactory.*decodeStream*(inputStream);

imageNote.setImageBitmap(bitmap);

imageNote.setVisibility(View.*VISIBLE*);

selectedImagePath = getPathFromUri( selectedImageUri);

}catch (Exception exception){

Toast.*makeText*(this, exception.getMessage(), Toast.*LENGTH\_SHORT*).show();

}

}

}

}

}

private String getPathFromUri(Uri contentUri){

String filePath;

Cursor cursor = getContentResolver()

.query(contentUri,null,null,null,null);

if(cursor == null){

filePath = contentUri.getPath();

} else{

cursor.moveToFirst();

int index = cursor.getColumnIndex("\_data");

filePath = cursor.getString(index);

cursor.close();

}

return filePath;

}

private void showAddURLDialog(){

if(dialogAddURL == null){

AlertDialog.Builder builder = new AlertDialog.Builder(CreateNoteActivity.this);

View view = LayoutInflater.*from*(this).inflate(

R.layout.*layout\_add\_url*,

(ViewGroup) findViewById(R.id.*layoutAddUrlContainer*)

);

builder.setView(view);

dialogAddURL = builder.create();

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

dialogAddURL.getWindow().setBackgroundDrawable(new ColorDrawable(0));

}

final EditText inputURL = view.findViewById(R.id.*inputURL*);

inputURL.requestFocus();

view.findViewById(R.id.*textAdd*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

if(inputURL.getText().toString().trim().isEmpty()){

Toast.*makeText*(CreateNoteActivity.this, "Enter URL!", Toast.*LENGTH\_SHORT*).show();

}else if(!Patterns.*WEB\_URL*.matcher(inputURL.getText().toString()).matches()){ // non valid URL

Toast.*makeText*(CreateNoteActivity.this, "Enter valid URL!", Toast.*LENGTH\_SHORT*).show();

}else{

textWebURL.setText(inputURL.getText().toString());

layoutWebUrl.setVisibility(View.*VISIBLE*);

Log.*d*("vis", "visible!");

dialogAddURL.dismiss();

}

}

});

view.findViewById(R.id.*textCancel*).setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

dialogAddURL.dismiss();

}

});

}

dialogAddURL.show();

}

}

#### 

#### xml:

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

<androidx.coordinatorlayout.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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorPrimary"

tools:context=".activities.CreateNoteActivity">

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginBottom="@dimen/\_40sdp"

android:clipToPadding="false"

android:paddingBottom="@dimen/\_10sdp"

>

<androidx.constraintlayout.widget.ConstraintLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:animateLayoutChanges="true"

>

<ImageView

android:id="@+id/imageBack"

android:layout\_width="@dimen/\_30sdp"

android:layout\_height="@dimen/\_30sdp"

android:layout\_marginStart="@dimen/\_6sdp"

android:layout\_marginTop="@dimen/\_20sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

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

android:src="@drawable/ic\_back"

app:tint="@color/colorIcon"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

/>

<ImageView

android:id="@+id/imageSave"

android:layout\_width="@dimen/\_20sdp"

android:layout\_height="@dimen/\_20sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:background="@drawable/background\_add\_button"

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

android:padding="@dimen/\_4sdp"

android:src="@drawable/ic\_done"

app:layout\_constraintBottom\_toBottomOf="@+id/imageBack"

app:tint="@color/colorIcon"

app:layout\_constraintTop\_toTopOf="@+id/imageBack"

app:layout\_constraintEnd\_toEndOf="parent"

/>

<EditText

android:id="@+id/inputNoteTitle"

android:layout\_width="match\_parent"

android:layout\_height="@dimen/\_35sdp"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:background="@null"

android:fontFamily="@font/ubuntu\_bold"

android:hint="@string/notetitle"

android:imeOptions="actionDone"

android:importantForAutofill="no"

android:includeFontPadding="false"

android:inputType="text"

android:textColor="@color/colorWhite"

android:textColorHint="@color/colorTextHint"

android:textSize="@dimen/\_16ssp"

app:layout\_constraintTop\_toBottomOf="@id/imageBack"

/>

<TextView

android:id="@+id/textDateTime"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:fontFamily="@font/ubuntu\_regular"

android:includeFontPadding="false"

android:textColor="@color/colorIcon"

android:textSize="@dimen/\_10ssp"

app:layout\_constraintEnd\_toEndOf="@+id/inputNoteTitle"

app:layout\_constraintStart\_toStartOf="@+id/inputNoteTitle"

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

/>

<EditText

android:id="@+id/inputNoteSubtitle"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:background="@null"

android:fontFamily="@font/ubuntu\_medium"

android:hint="@string/note\_subtitle"

android:importantForAutofill="no"

android:textColorHint="@color/colorTextHint"

android:includeFontPadding="false"

android:inputType="textMultiLine"

android:paddingTop="@dimen/\_8sdp"

android:paddingBottom="@dimen/\_8sdp"

android:textColor="@color/colorNoteSubtitleText"

android:textSize="@dimen/\_13ssp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/viewSubtitleIndicator"

app:layout\_constraintTop\_toBottomOf="@id/textDateTime"

/>

<View

android:id="@+id/viewSubtitleIndicator"

android:layout\_width="@dimen/\_5sdp"

android:layout\_height="0dp"

android:layout\_marginStart="@dimen/\_12sdp"

android:background="@drawable/background\_subtitle\_indicator"

app:layout\_constraintBottom\_toBottomOf="@id/inputNoteSubtitle"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="@id/inputNoteSubtitle"/>

<ImageView

android:id="@+id/imageNote"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:adjustViewBounds="true"

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

android:visibility="gone"

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

/>

<LinearLayout

android:id="@+id/layoutWebURL"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:gravity="center\_vertical"

android:orientation="horizontal"

android:visibility="gone"

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

>

<TextView

android:id="@+id/textWebURL"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:autoLink="web"

android:fontFamily="@font/ubuntu\_regular"

android:includeFontPadding="false"

android:textSize="@dimen/\_13ssp"

android:textColor="@color/colorAccent"

/>

</LinearLayout>

<EditText

android:id="@+id/inputNote"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:layout\_marginBottom="@dimen/\_15sdp"

android:background="@null"

android:fontFamily="@font/ubuntu\_regular"

android:gravity="top"

android:hint="@string/type\_note\_here"

android:importantForAutofill="no"

android:includeFontPadding="false"

android:inputType="textMultiLine"

android:minHeight="@dimen/\_100sdp"

android:textColor="@color/colorWhite"

android:textColorHint="@color/colorTextHint"

android:textSize="@dimen/\_13ssp"

app:layout\_constraintBottom\_toBottomOf="parent"

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

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>

<include layout="@layout/layout\_miscellaneous"/>

</androidx.coordinatorlayout.widget.CoordinatorLayout>

#### 

#### main activity

public class MainActivity extends AppCompatActivity {

// SharedPreferences sp;

User user;

SharedPreferences sharedPreferences;

NetworkChangeReciver broadcastReceiver;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

//cosmetic

getSupportActionBar().hide();

sharedPreferences = getSharedPreferences("active\_user",0);

user=new User();

SQLHelper helper= new SQLHelper(this);

TextView username = (TextView) findViewById(R.id.*username*);

TextView password = (TextView) findViewById(R.id.*password*);

MaterialButton loginbtn = (MaterialButton) findViewById(R.id.*loginbtn*);

MaterialButton registerbtn = (MaterialButton) findViewById(R.id.*registerbtn*);

loginbtn.setOnClickListener(new View.OnClickListener() {

@SuppressLint("ResourceType")

@Override

public void onClick(View view) {

if (username.getText().toString().equals("admin") && password.getText().toString().equals("admin")){

Toast.*makeText*(MainActivity.this,"Success!", Toast.*LENGTH\_LONG*).show();

// give admin permission

String name = "admin";

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("name",name);

editor.commit();

// original

// Intent intent = new Intent(getApplicationContext(), TheShelf.class);

// startActivity(intent);

Intent intent = new Intent(getApplicationContext(), Splash\_screen\_book.class);

intent.putExtra("activity", 4);

startActivity(intent);

}else {

if(helper.doesUsernameExist(username.getText().toString())){

helper.setProfile(user,username.getText().toString()); // setting profile!

if(user.getPassword().equals(password.getText().toString())){

Log.*d*("test", "user found: ");

String name = user.getPermission();

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("name",name);

editor.commit();

Intent intent = new Intent(getApplicationContext(), Splash\_screen\_book.class);

intent.putExtra("activity", 4);

startActivity(intent);

finish();

}

}else{

Toast.*makeText*(MainActivity.this,"Login failed",

Toast.*LENGTH\_LONG*).show();

}

}

}

});

registerbtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent register = new Intent(getApplicationContext(), Register\_page.class);

startActivity(register);

}

});

broadcastReceiver = new NetworkChangeReciver();

registerNetworkBroadcastReciver();

}

protected void registerNetworkBroadcastReciver(){

if(Build.VERSION.*SDK\_INT*>=Build.VERSION\_CODES.*N*) {

registerReceiver(broadcastReceiver, new IntentFilter(ConnectivityManager.*CONNECTIVITY\_ACTION*));

}

if(Build.VERSION.*SDK\_INT*>=Build.VERSION\_CODES.*M*) {

registerReceiver(broadcastReceiver, new IntentFilter(ConnectivityManager.*CONNECTIVITY\_ACTION*));

}

}

protected void unregisterNetwork(){

try{

unregisterReceiver(broadcastReceiver);

}

catch (IllegalArgumentException e){

e.printStackTrace();

}

}

@Override

protected void onDestroy() {

super.onDestroy();

unregisterNetwork();

}

}

#### 

#### main activity xml:

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

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".activities.MainActivity"

android:background="@drawable/background1"

>

<!--

Login page

!-->

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/SignIn"

android:text="Sign in"

android:textColor="@color/white"

android:textSize="35dp"

android:layout\_margin="15dp"

android:gravity="center"

android:fontFamily="sans-serif-condensed"

android:textStyle="bold"

/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/username"

android:layout\_below="@id/SignIn"

android:hint="User Name"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:layout\_margin="10dp"

android:padding="20dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_person\_outline\_24"

android:drawablePadding="20dp"

/>

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/password"

android:layout\_below="@id/username"

android:hint="Password"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:layout\_margin="10dp"

android:padding="20dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_lock\_24"

android:drawablePadding="20dp"

android:inputType="textPassword"

/>

<com.google.android.material.button.MaterialButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/loginbtn"

android:layout\_below="@id/password"

android:text="login"

android:layout\_centerHorizontal="true"

android:backgroundTint="@color/black"

android:layout\_margin="20dp"

/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="NewAccount?"

android:id= "@+id/Register"

android:layout\_below="@id/loginbtn"

android:layout\_centerHorizontal="true"

android:textColor="@color/white"

android:layout\_margin="20dp"

/>

<com.google.android.material.button.MaterialButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/registerbtn"

android:layout\_below="@id/Register"

android:layout\_centerHorizontal="true"

android:text="clickhere!"

android:backgroundTint="@color/white"

android:textColor="@color/black"

android:layout\_marginTop="-15dp"

/>

</RelativeLayout>

#### notes page java:

public class notes\_page extends AppCompatActivity implements NotesListener {

public static final int *REQUEST\_CODE\_ADD\_NOTE* = 1; // new note!!

public static final int *REQUEST\_CODE\_UPDATE\_NOTE* = 2; // update

public static final int *REQUEST\_CODE\_SHOW\_NOTES* = 3; // display all notes

// for admin actions

SharedPreferences sharedPreferences;

private RecyclerView notesRecyclerView;

private List<Note> noteList;

private NotesAdapter notesAdapter;

private TextView textView;

private int noteClickedPosition = -1;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_notes\_page*);

// sp for admin actions

sharedPreferences = getSharedPreferences("active\_user",0);

String name = sharedPreferences.getString("name", null);

textView = findViewById(R.id.*inputSearch*);

if(name.equals("default")){

textView.setText("you can view only!");

}else if(name.equals("editor")){

textView.setText("you can edit and view");

}else{

textView.setText("you can edit, view and delete notes");

}

ImageView imageAddNoteMain = findViewById(R.id.*imageAddNoteMain*);

**if**( name.equals(**"default"**)){

imageAddNoteMain.setVisibility(View.***GONE***);

}

// on click

imageAddNoteMain.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivityForResult(

new Intent(getApplicationContext(), CreateNoteActivity.class),

*REQUEST\_CODE\_ADD\_NOTE*

);

}

});

// notes recycler view

notesRecyclerView = findViewById(R.id.*notesRecyclerView*);

notesRecyclerView.setLayoutManager(

new StaggeredGridLayoutManager(2, StaggeredGridLayoutManager.*VERTICAL*)

);

noteList= new ArrayList<>();

notesAdapter = new NotesAdapter(noteList, this);

notesRecyclerView.setAdapter(notesAdapter);

// render notes

getNotes(*REQUEST\_CODE\_SHOW\_NOTES*, false);

}

// func for note update

@Override

public void onNotesClicked(Note note, int position) {

noteClickedPosition = position;

Intent intent = new Intent(getApplicationContext(), CreateNoteActivity.class);

intent.putExtra("isViewOrUpdate", true);

intent.putExtra("note", note);

startActivityForResult(intent, *REQUEST\_CODE\_UPDATE\_NOTE*);

}

// using async

private void getNotes(final int requestCode, final boolean isNoteDeleted) {

@SuppressLint("StaticFieldLeak")

class GetNoteTask extends AsyncTask<Void, Void, List<Note>>{

@Override

protected List<Note> doInBackground(Void... voids) {

return NotesDatabase

.*getDataBase*(getApplicationContext())

.noteDao().getAllNotes();

}

/\*

\* if note list is empty it means the app just started because it's globle variable

\* and we need to update(add all notes) from db to list.

\* else that means that we need to add only one note

\* and scroll recycler view to the top

\* \*/

@Override

protected void onPostExecute(List<Note> notes) {

super.onPostExecute(notes);

Log.*d*("my\_notes", notes.toString()); // debugging

Log.*d*("my\_notes\_size", String.*valueOf*(noteList.size())); // debugging

// sp for admin actions

sharedPreferences = getSharedPreferences("active\_user",0);

String name = sharedPreferences.getString("name", null);

if(requestCode == *REQUEST\_CODE\_SHOW\_NOTES*){

noteList.addAll(notes);

notesAdapter.notifyDataSetChanged();

}else if(requestCode == *REQUEST\_CODE\_ADD\_NOTE*){

noteList.add(0, notes.get(0));

notesAdapter.notifyItemInserted(0);

notesRecyclerView.smoothScrollToPosition(0);

}else if(requestCode == *REQUEST\_CODE\_UPDATE\_NOTE*){

noteList.remove(noteClickedPosition);

if(isNoteDeleted){

notesAdapter.notifyItemRemoved(noteClickedPosition);

}else{

noteList.add(noteClickedPosition, notes.get(noteClickedPosition));

notesAdapter.notifyItemChanged(noteClickedPosition);

}

}

// old version:

// if(noteList.size() ==0){

// noteList.addAll(notes);

// Log.d("my\_notes\_size", String.valueOf(noteList.size())); // debugging

// notesAdapter.notifyDataSetChanged();

// }else{

// noteList.add(0, notes.get(0));

// notesAdapter.notifyItemInserted(0);

// }

// notesRecyclerView.smoothScrollToPosition(0);

}

}

new GetNoteTask().execute();

}

// update note real time (after save button pressed!)

@Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

// sp for admin actions

sharedPreferences = getSharedPreferences("active\_user",0);

String name = sharedPreferences.getString("name", null);

// result code checks if createnote activity is done

if(requestCode == *REQUEST\_CODE\_ADD\_NOTE* && resultCode == *RESULT\_OK*){

getNotes(*REQUEST\_CODE\_ADD\_NOTE*, false);

}else if(requestCode == *REQUEST\_CODE\_UPDATE\_NOTE* && resultCode == *RESULT\_OK*

){// && !name.equals("default")

if(data != null){

getNotes(*REQUEST\_CODE\_UPDATE\_NOTE*, data.getBooleanExtra("isNoteDeleted", false));

}

}

}

}

#### notes page Xml:

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/colorPrimary"

tools:context=".activities.notes\_page">

<TextView

android:id="@+id/text\_MyNotes"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_20sdp"

android:layout\_marginEnd="@dimen/\_12sdp"

android:fontFamily="@font/ubuntu\_bold"

android:includeFontPadding="false"

android:text="@string/all\_notes"

android:textColor="@color/colorWhite"

android:textSize="@dimen/\_20ssp"

app:layout\_constraintTop\_toTopOf="parent"

/>

<LinearLayout

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

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_8sdp"

android:layout\_marginTop="@dimen/\_15sdp"

android:layout\_marginEnd="@dimen/\_8sdp"

android:background="@drawable/background\_search"

android:gravity="center\_vertical"

android:orientation="horizontal"

android:paddingStart="@dimen/\_10sdp"

android:paddingEnd="@dimen/\_10sdp"

app:layout\_constraintTop\_toBottomOf="@+id/text\_MyNotes"

>

<ImageView

android:layout\_width="@dimen/\_20sdp"

android:layout\_height="@dimen/\_20sdp"

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

android:src="@drawable/ic\_help"

app:tint="@color/colorSearchIcon"

/>

<TextView

android:id="@+id/inputSearch"

android:layout\_width="match\_parent"

android:layout\_height="@dimen/\_35sdp"

android:layout\_marginStart="@dimen/\_12sdp"

android:background="@null"

android:fontFamily="@font/ubuntu\_regular"

android:hint="help"

android:textColor="@color/colorWhite"

android:textColorHint="@color/colorTextHint"

android:textSize="@dimen/\_13ssp"

android:layout\_marginTop="@dimen/\_10sdp"

/>

</LinearLayout>

<LinearLayout

android:id="@+id/layoutQuickActions"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="@color/colorQuickActionBackground"

android:gravity="center\_vertical"

android:orientation="horizontal"

android:padding="@dimen/\_15sdp"

app:layout\_constraintBottom\_toBottomOf="parent"

>

<ImageView

android:id="@+id/imageAddNote"

android:layout\_width="@dimen/\_23sdp"

android:layout\_height="@dimen/\_23sdp"

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

android:src="@drawable/ic\_add\_outline"

app:tint="@color/colorIcon"

/>

<ImageView

android:id="@+id/imageAddImage"

android:layout\_width="@dimen/\_23sdp"

android:layout\_height="@dimen/\_23sdp"

android:layout\_marginStart="@dimen/\_14sdp"

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

android:src="@drawable/ic\_image"

app:tint="@color/colorIcon"

/>

<ImageView

android:id="@+id/imageAddWebLink"

android:layout\_width="@dimen/\_23sdp"

android:layout\_height="@dimen/\_23sdp"

android:layout\_marginStart="@dimen/\_14sdp"

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

android:src="@drawable/ic\_web\_link"

app:tint="@color/colorIcon"

/>

</LinearLayout>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/notesRecyclerView"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_marginStart="@dimen/\_2sdp"

android:layout\_marginEnd="@dimen/\_2sdp"

android:clipToPadding="false"

android:paddingStart="0dp"

android:paddingEnd="@dimen/\_12sdp"

android:paddingBottom="@dimen/\_12sdp"

app:layout\_constraintBottom\_toTopOf="@id/layoutQuickActions"

app:layout\_constraintTop\_toBottomOf="@id/layout\_search"

/>

<ImageView

android:id="@+id/imageAddNoteMain"

android:layout\_width="@dimen/\_40sdp"

android:layout\_height="@dimen/\_40sdp"

android:layout\_marginEnd="@dimen/\_30sdp"

android:background="@drawable/background\_add\_button"

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

android:padding="@dimen/\_5sdp"

android:src="@drawable/ic\_add\_icon"

app:layout\_constraintBottom\_toTopOf="@id/layoutQuickActions"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toTopOf="@id/layoutQuickActions"

app:tint="@color/colorPrimary"

/>

</androidx.constraintlayout.widget.ConstraintLayout>

#### splash screen:

public class Splash\_screen\_book extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

Intent intent = getIntent();

getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*,

WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_splash\_screen\_book*);

new Handler().postDelayed(new Runnable() {

@Override

public void run() {

// gets a value what activity to go to

int id = getIntent().getIntExtra("activity", 0);

// use of hashmap for easy search

Map<Integer, Class<?>> activityMap = new HashMap<>();

activityMap.put(2, Register\_page.class);

activityMap.put(3, notes\_page.class);

activityMap.put(4, Transfer\_page.class);

activityMap.put(5, MainActivity.class);

// Add more entries to the map as needed

// Log.d("", String.valueOf(id)); // for debugg

Class<?> cls = activityMap.get(id);

if (cls == null) {

cls = MainActivity.class;

}

Intent intent = new Intent(getApplicationContext(), cls);

startActivity(intent);

finish();

/\*

old code

//int p= intent.getExtras().getInt("activity");

//@SuppressLint("ResourceType") int x = findViewById(R.layout.activity\_the\_shelf).getId();

//Log.d("sssssssssss",String.valueOf(x));

//if(p ==x){startActivity(new Intent(getApplicationContext(),TheShelf.class));

//}

\*/

}

},3000);

}

}

#### splash screen xml:

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

<androidx.constraintlayout.motion.widget.MotionLayout 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"

app:layoutDescription="@xml/activity\_splash\_screen\_book\_scene"

tools:context=".activities.Splash\_screen\_book">

<androidx.constraintlayout.motion.widget.MotionLayout

android:id="@+id/motionLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

app:layoutDescription="@xml/activity\_splash\_screen\_book\_scene"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

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

<androidx.constraintlayout.widget.ConstraintLayout

android:id="@+id/backgroundLayout"

android:layout\_width="150dp"

android:layout\_height="150dp"

android:background="@drawable/bg"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

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

<ImageView

android:id="@+id/brownBook"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/book\_no\_back"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

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

<ImageView

android:id="@+id/whiteBook"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:src="@drawable/book\_no\_back"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:tint="@color/white" />

</androidx.constraintlayout.motion.widget.MotionLayout>

</androidx.constraintlayout.motion.widget.MotionLayout>

#### Register page java:

public class Register\_page extends AppCompatActivity implements View.OnClickListener {

EditText edName,edPass,edAge,edEmail;

SQLHelper helper;

SharedPreferences sharedPreferences;

Cursor cursor;

Button save;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_register\_page*);

getSupportActionBar().hide();

edName = (EditText) findViewById(R.id.*username\_input*);

edPass = (EditText) findViewById(R.id.*password\_input1*);

edAge = (EditText) findViewById(R.id.*age\_input*);

edEmail = (EditText) findViewById(R.id.*email\_input*);

helper = new SQLHelper(this);

cursor = helper.getCursor();

cursor.moveToFirst();

save = findViewById(R.id.*savebtn*);

save.setOnClickListener(this);

sharedPreferences = getSharedPreferences("active\_user",0);

}

// public void save(View view) {

//

// }

@Override

protected void onDestroy() {

super.onDestroy();

helper.getWritableDatabase().close(); //close the data base

}

@Override

public void onClick(View view) {

// data from filed

// can't get the same name!

if(helper.doesUsernameExist(edName.getText().toString())){

Toast.*makeText*(this, "name is taken!", Toast.*LENGTH\_SHORT*).show();

return;

}

String name = edName.getText().toString();

int age = Integer.*parseInt*(edAge.getText().toString());

String email = edEmail.getText().toString();

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

// sheared pref

String per = "default"; // default permission

// active login

SharedPreferences.Editor editor = sharedPreferences.edit();

editor.putString("name",name);

editor.commit();

// int interger= cursor.getInt(cursor.getInt(0));

// editor.putInt("id",interger);

MyAdapter adapter = new MyAdapter(this,helper.getCursor());

helper.insertData(new User(name,age,email,pass, per));

Toast.*makeText*(this, "data saved", Toast.*LENGTH\_SHORT*).show();

Intent intent = new Intent(getApplicationContext(), Splash\_screen\_book.class);

intent.putExtra("activity", 4);

startActivity(intent);

finish();

}

}

#### Register page XML:

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

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".Register\_page"

android:background="@drawable/background1"

>

<!--

Register page

!-->

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/SignUp"

android:text="Sign Up!"

android:textColor="@color/white"

android:textSize="35dp"

android:layout\_margin="15dp"

android:gravity="center"

android:fontFamily="sans-serif-condensed"

android:textStyle="bold"

/>

<!--

inputs:

!-->

<EditText

android:id="@+id/username\_input"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/SignUp"

android:layout\_marginStart="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginBottom="10dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_person\_outline\_24"

android:drawablePadding="20dp"

android:hint="New username"

android:padding="20dp"

android:textColor="@color/white"

android:textColorHint="@color/white" />

<EditText

android:id="@+id/password\_input1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/username\_input"

android:layout\_marginStart="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginBottom="10dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_lock\_24"

android:drawablePadding="20dp"

android:hint="New password"

android:inputType="textPassword"

android:padding="20dp"

android:textColor="@color/white"

android:textColorHint="@color/white" />

<EditText

android:id="@+id/age\_input"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/password\_input1"

android:layout\_marginStart="10dp"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="10dp"

android:layout\_marginBottom="10dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_age"

android:drawablePadding="20dp"

android:hint="Age"

android:inputType="number"

android:padding="20dp"

android:textColor="@color/white"

android:textColorHint="@color/white" />

<EditText

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

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

android:layout\_below="@id/age\_input"

android:hint="Email"

android:textColor="@color/white"

android:textColorHint="@color/white"

android:layout\_margin="10dp"

android:padding="20dp"

android:background="#30ffffff"

android:drawableLeft="@drawable/ic\_baseline\_alternate\_email\_24"

android:drawablePadding="20dp"

android:inputType="text"

/>

<com.google.android.material.button.MaterialButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/savebtn"

android:layout\_below="@id/email\_input"

android:text="Sign me up!"

android:layout\_centerHorizontal="true"

android:backgroundTint="@color/black"

android:layout\_margin="20dp"

/>

</RelativeLayout>

#### Result activity Java:

package com.example.oneapptorulethemall;

import android.annotation.SuppressLint;

import android.content.Intent;

import android.database.Cursor;

import android.database.sqlite.SQLiteCursor;

import android.os.Bundle;

import android.util.Log;

import android.view.View;

import android.widget.AdapterView;

import android.widget.EditText;

import android.widget.ListView;

import android.widget.TextView;

import androidx.appcompat.app.AppCompatActivity;

import com.example.oneapptorulethemall.activities.MainActivity;

import java.util.jar.Attributes;

public class ResultActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_users\_list*);

// output data

SQLHelper helper = new SQLHelper(this);

ListView listView = (ListView) findViewById(R.id.*listview*);

MyAdapter adapter = new MyAdapter(this,helper.getCursor());

listView.setAdapter(adapter);

TextView textView = (TextView) findViewById(R.id.*output*);

listView.setClickable(true);

listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {

@Override

public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

Cursor item= (Cursor) adapter.getItem(position);

String Name= "";

item.moveToPosition(position);

if(item.getColumnIndex("name") != -1){

@SuppressLint("Range") String text = item.getString(item.getColumnIndex("name"));

Log.*d*("PP", text);

Name = text;

}

if(helper.doesUsernameExist(Name)){

User user;

user = new User();

helper.setProfile(user,Name);

if(user.permission.equals("default")){

Log.*d*("PP", "default to editor:");

helper.updatePer(Name, "editor");

textView.setText(Name + " is now editor");

}else if(user.permission.equals("editor")){

Log.*d*("PP", "editor to admin:");

helper.updatePer(Name, "admin");

textView.setText(Name + " is now admin");

}else{

Log.*d*("PP", "admin to default:");

helper.updatePer(Name, "default");

textView.setText(Name + " is now default");

}

}

}

});

}

public void back(View view) {

startActivity(new Intent(this, Transfer\_page.class));

finish();

}

}

Result activity XML:

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

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context="com.example.oneapptorulethemall.ResultActivity">

<ListView

android:id="@+id/listview"

android:layout\_width="match\_parent"

android:layout\_height="460dp"

tools:layout\_editor\_absoluteX="8dp"

tools:layout\_editor\_absoluteY="8dp" />

<TextView

android:id="@+id/EditTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/listview"

android:layout\_above="@id/EditTitle"

android:text="click on user to change permission"

android:textSize="@dimen/\_15ssp"

android:textColor="@color/black"

android:fontFamily="@font/ubuntu\_bold"

/>

<TextView

android:id="@+id/output"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/EditTitle"

android:text="output!"

android:textSize="@dimen/\_15ssp"

android:textColor="@color/colorDelete"

android:fontFamily="@font/ubuntu\_medium"

/>

<Button

android:onClick="back"

android:layout\_below="@id/output"

android:text="back"

android:layout\_width="match\_parent"

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

</RelativeLayout>

#### Transfer page java:

package com.example.oneapptorulethemall;

import android.content.Intent;

import android.content.SharedPreferences;

import android.os.Bundle;

import android.util.Log;

import android.view.WindowManager;

import android.widget.Toast;

import com.example.oneapptorulethemall.activities.MainActivity;

import com.example.oneapptorulethemall.activities.Splash\_screen\_book;

import com.example.oneapptorulethemall.activities.notes\_page;

import com.ramotion.circlemenu.CircleMenuView;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

public class Transfer\_page extends AppCompatActivity {

SharedPreferences sharedPreferences;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_transfer\_page*);

sharedPreferences = getSharedPreferences("active\_user",0);

String name = sharedPreferences.getString("name", null);

getWindow().setFlags(WindowManager.LayoutParams.*FLAG\_FULLSCREEN*,

WindowManager.LayoutParams.*FLAG\_FULLSCREEN*);

final CircleMenuView menu = findViewById(R.id.*circle\_menu*);

menu.setEventListener(new CircleMenuView.EventListener() {

@Override

public void onMenuOpenAnimationStart(@NonNull CircleMenuView view) {

Log.*d*("D", "onMenuOpenAnimationStart");

}

@Override

public void onMenuOpenAnimationEnd(@NonNull CircleMenuView view) {

Log.*d*("D", "onMenuOpenAnimationEnd");

}

@Override

public void onMenuCloseAnimationStart(@NonNull CircleMenuView view) {

Log.*d*("D", "onMenuCloseAnimationStart");

}

@Override

public void onMenuCloseAnimationEnd(@NonNull CircleMenuView view) {

Log.*d*("D", "onMenuCloseAnimationEnd");

}

@Override

public void onButtonClickAnimationStart(@NonNull CircleMenuView view, int index) {

Log.*d*("D", "onButtonClickAnimationStart| index: " + index);

}

@Override

public void onButtonClickAnimationEnd(@NonNull CircleMenuView view, int index) {

Log.*d*("D", "onButtonClickAnimationEnd| index: " + index);

String name = sharedPreferences.getString("name", null);

switch (index){

case 0: // Main/ login

//SharedPreferences.Editor editor = sharedPreferences.edit();

//editor.putString("name","");

//editor.commit();

Intent intent0 = new Intent(getApplicationContext(), Splash\_screen\_book.class);

intent0.putExtra("activity", 5);

startActivity(intent0);

break;

case 1:

Intent intent1 = new Intent(getApplicationContext(), ExampleTH.class);

startActivity(intent1);

break;

case 4: // notes

Intent intent = new Intent(getApplicationContext(), Splash\_screen\_book.class);

intent.putExtra("activity", 3);

startActivity(intent);

break;

case 3: // settings

if(name.equals("admin")){

Intent intent3 = new Intent(getApplicationContext(), ResultActivity.class);

startActivity(intent3);

finish();

break;

}else{

Toast.*makeText*(Transfer\_page.this,"Admin only page!",

Toast.*LENGTH\_LONG*).show();

break;

}

case 2: // service music

Intent intent2 = new Intent(getApplicationContext(), MyService.class);

startService(intent2);

break;

}

}

@Override

public boolean onButtonLongClick(@NonNull CircleMenuView view, int index) {

Log.*d*("D", "onButtonLongClick| index: " + index);

return true;

}

@Override

public void onButtonLongClickAnimationStart(@NonNull CircleMenuView view, int index) {

Log.*d*("D", "onButtonLongClickAnimationStart| index: " + index);

}

@Override

public void onButtonLongClickAnimationEnd(@NonNull CircleMenuView view, int index) {

Log.*d*("D", "onButtonLongClickAnimationEnd| index: " + index);

switch (index){

case 4:

Intent intent = new Intent(getApplicationContext(), notes\_page.class);

startActivity(intent);

finish();

break;

case 2: // service music

Intent intent2 = new Intent(getApplicationContext(), MyService.class);

stopService(intent2);

break;

}

}

});

}

}

#### Transfer page XML:

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".Transfer\_page">

<com.ramotion.circlemenu.CircleMenuView

android:id="@+id/circle\_menu"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:button\_colors="@array/colors"

app:button\_icons="@array/icons"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

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

</androidx.constraintlayout.widget.ConstraintLayout>

#### Circle menu view:

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

<merge

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="match\_parent">

<com.ramotion.circlemenu.RingEffectView

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

android:layout\_gravity="center"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"/>

<com.google.android.material.floatingactionbutton.FloatingActionButton

android:id="@+id/circle\_menu\_main\_button"

android:layout\_gravity="center"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_menu\_black\_24dp"

app:backgroundTint="@android:color/white"/>

</merge>

#### item container note:

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

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

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

android:id="@+id/layoutNote"

android:orientation="vertical"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_12sdp"

android:layout\_marginTop="@dimen/\_12sdp"

android:background="@drawable/background\_note">

<com.makeramen.roundedimageview.RoundedImageView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:id="@+id/imageNote"

android:adjustViewBounds="true"

android:visibility="gone"

app:riv\_corner\_radius\_top\_left="@dimen/\_10sdp"

app:riv\_corner\_radius\_top\_right="@dimen/\_10sdp"

/>

<TextView

android:id="@+id/textTitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_8sdp"

android:layout\_marginEnd="@dimen/\_8sdp"

android:layout\_marginStart="@dimen/\_8sdp"

android:fontFamily="@font/ubuntu\_bold"

android:includeFontPadding="false"

android:textColor="@color/colorWhite"

android:textSize="@dimen/\_13ssp"/>

<TextView

android:id="@+id/textSubtitle"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_4sdp"

android:layout\_marginEnd="@dimen/\_8sdp"

android:layout\_marginStart="@dimen/\_8sdp"

android:fontFamily="@font/ubuntu\_regular"

android:includeFontPadding="false"

android:textColor="@color/colorWhite"

android:textSize="@dimen/\_12ssp"/>

<TextView

android:id="@+id/textDateTime"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="@dimen/\_8sdp"

android:layout\_marginEnd="@dimen/\_8sdp"

android:layout\_marginStart="@dimen/\_8sdp"

android:fontFamily="@font/ubuntu\_bold"

android:includeFontPadding="false"

android:textColor="@color/colorWhite"

android:textSize="@dimen/\_7ssp"/>

</LinearLayout>

#### Dialog Url view;

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

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

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

android:id="@+id/layoutAddUrlContainer"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_margin="@dimen/\_10sdp"

android:background="@drawable/background\_dialog"

>

<ImageView

android:id="@+id/imageAddURL"

android:layout\_width="@dimen/\_25sdp"

android:layout\_height="@dimen/\_25sdp"

android:layout\_marginStart="@dimen/\_8sdp"

android:layout\_marginTop="@dimen/\_8sdp"

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

android:src="@drawable/ic\_web\_link"

app:tint="@color/colorWhite"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

/>

<TextView

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="@dimen/\_8sdp"

android:layout\_marginEnd="@dimen/\_8sdp"

android:fontFamily="@font/ubuntu\_bold"

android:includeFontPadding="false"

android:text="@string/add\_url"

android:textColor="@color/colorWhite"

android:textSize="@dimen/\_14ssp"

app:layout\_constraintBottom\_toBottomOf="@+id/imageAddURL"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toEndOf="@+id/imageAddURL"

app:layout\_constraintTop\_toTopOf="@+id/imageAddURL"

/>

<EditText

android:id="@+id/inputURL"

android:layout\_width="match\_parent"

android:layout\_height="@dimen/\_35sdp"

android:layout\_marginStart="@dimen/\_8sdp"

android:layout\_marginEnd="@dimen/\_8sdp"

android:layout\_marginTop="@dimen/\_8sdp"

android:background="@null"

android:fontFamily="@font/ubuntu\_regular"

android:gravity="center\_vertical"

android:hint="@string/enter\_url"

android:textColor="@color/colorWhite"

android:imeOptions="actionDone"

android:importantForAutofill="no"

android:includeFontPadding="false"

android:inputType="textUri"

android:paddingStart="@dimen/\_8sdp"

android:paddingEnd="@dimen/\_8sdp"

android:textColorHint="@color/colorWhite"

android:textSize="@dimen/\_13ssp"

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

/>

<TextView

android:id="@+id/textAdd"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="@dimen/\_8sdp"

android:layout\_marginBottom="@dimen/\_8sdp"

android:fontFamily="@font/ubuntu\_medium"

android:includeFontPadding="false"

android:padding="@dimen/\_8sdp"

android:text="@string/add"

android:textColor="@color/colorAccent"

android:textSize="@dimen/\_13ssp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintTop\_toBottomOf="@id/inputURL"

/>

<TextView

android:id="@+id/textCancel"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="@dimen/\_8sdp"

android:layout\_marginBottom="@dimen/\_8sdp"

android:fontFamily="@font/ubuntu\_medium"

android:includeFontPadding="false"

android:padding="@dimen/\_8sdp"

android:text="Cancel"

android:textColor="@color/colorAccent"

android:textSize="@dimen/\_13ssp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toStartOf="@+id/textAdd"

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

/>

</androidx.constraintlayout.widget.ConstraintLayout>

#### layout miscellaneous:

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

#### <LinearLayout 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/layoutMiscellaneous"

#### android:orientation="vertical"

#### android:layout\_width="match\_parent"

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

#### android:background="@drawable/background\_miscellaneous"

#### app:behavior\_peekHeight="@dimen/\_50sdp"

#### app:layout\_behavior="@string/bottom\_sheet\_behavior">

#### 

#### <TextView

#### android:id="@+id/textMiscellaneous"

#### android:layout\_width="match\_parent"

#### android:layout\_height="@dimen/\_40sdp"

#### android:fontFamily="@font/ubuntu\_medium"

#### android:gravity="center"

#### android:includeFontPadding="false"

#### android:text="@string/miscellaneous"

#### android:textColor="@color/colorWhite"

#### android:textSize="@dimen/\_13ssp"

#### />

#### 

#### <LinearLayout

#### android:id="@+id/layoutNoteColor"

#### android:layout\_width="match\_parent"

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

#### android:layout\_margin="@dimen/\_10sdp"

#### android:gravity="center\_vertical"

#### android:orientation="horizontal">

#### 

#### 

#### <FrameLayout

#### android:layout\_width="0dp"

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

#### android:layout\_weight="1">

#### 

#### <View

#### android:id="@+id/viewColor1"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:background="@drawable/background\_note\_color\_1"/>

#### <ImageView

#### android:id="@+id/imageColor1"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:contentDescription="@string/app\_name"

#### android:padding="@dimen/\_10sdp"

#### android:src="@drawable/ic\_done"

#### app:tint="@color/colorWhite" />

#### </FrameLayout>

#### 

#### <FrameLayout

#### android:layout\_width="0dp"

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

#### android:layout\_weight="1">

#### 

#### <View

#### android:id="@+id/viewColor2"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:background="@drawable/background\_note\_color\_2"/>

#### <ImageView

#### android:id="@+id/imageColor2"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:contentDescription="@string/app\_name"

#### android:padding="@dimen/\_10sdp"

#### app:tint="@color/colorWhite" />

#### </FrameLayout>

#### 

#### <FrameLayout

#### android:layout\_width="0dp"

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

#### android:layout\_weight="1">

#### 

#### <View

#### android:id="@+id/viewColor3"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:background="@drawable/background\_note\_color\_3"/>

#### <ImageView

#### android:id="@+id/imageColor3"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:contentDescription="@string/app\_name"

#### android:padding="@dimen/\_10sdp"

#### app:tint="@color/colorWhite" />

#### </FrameLayout>

#### 

#### <FrameLayout

#### android:layout\_width="0dp"

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

#### android:layout\_weight="1">

#### 

#### <View

#### android:id="@+id/viewColor4"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:background="@drawable/background\_note\_color\_4"/>

#### <ImageView

#### android:id="@+id/imageColor4"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:contentDescription="@string/app\_name"

#### android:padding="@dimen/\_10sdp"

#### app:tint="@color/colorWhite" />

#### </FrameLayout>

#### 

#### <FrameLayout

#### android:layout\_width="0dp"

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

#### android:layout\_weight="1">

#### 

#### <View

#### android:id="@+id/viewColor5"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:background="@drawable/background\_note\_color\_5"/>

#### <ImageView

#### android:id="@+id/imageColor5"

#### android:layout\_width="@dimen/\_35sdp"

#### android:layout\_height="@dimen/\_35sdp"

#### android:contentDescription="@string/app\_name"

#### android:padding="@dimen/\_10sdp"

#### app:tint="@color/colorWhite" />

#### </FrameLayout>

#### 

#### <TextView

#### android:layout\_width="wrap\_content"

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

#### android:fontFamily="@font/ubuntu\_medium"

#### android:includeFontPadding="false"

#### android:text="@string/pick\_color"

#### android:textColor="@color/colorWhite"

#### android:textSize="@dimen/\_13ssp"

#### />

#### 

#### </LinearLayout>

#### 

#### 

#### <LinearLayout

#### android:id="@+id/layoutAddImage"

#### android:layout\_width="match\_parent"

#### android:layout\_height="@dimen/\_35sdp"

#### android:layout\_margin="@dimen/\_10sdp"

#### android:gravity="center\_vertical"

#### android:orientation="horizontal"

#### >

#### 

#### <ImageView

#### android:layout\_width="@dimen/\_22sdp"

#### android:layout\_height="@dimen/\_22sdp"

#### android:contentDescription="@string/app\_name"

#### android:src="@drawable/ic\_image"

#### app:tint="@color/colorIcon" />

#### <TextView

#### android:layout\_width="match\_parent"

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

#### android:layout\_marginStart="@dimen/\_10sdp"

#### android:fontFamily="@font/ubuntu\_medium"

#### android:includeFontPadding="false"

#### android:text="Add image"

#### android:textColor="@color/colorIcon"

#### android:textSize="@dimen/\_12ssp"/>

#### 

#### 

#### </LinearLayout>

#### 

#### 

#### <LinearLayout

#### android:id="@+id/layoutAddUrl"

#### android:layout\_width="match\_parent"

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

#### android:layout\_marginStart="@dimen/\_10sdp"

#### android:layout\_marginEnd="@dimen/\_10sdp"

#### android:gravity="center\_vertical"

#### android:orientation="horizontal"

#### tools:ignore="UseCompoundDrawables">

#### 

#### <ImageView

#### android:layout\_width="@dimen/\_22sdp"

#### android:layout\_height="@dimen/\_22sdp"

#### android:contentDescription="@string/app\_name"

#### android:src="@drawable/ic\_web\_link"

#### app:tint="@color/colorIcon"

#### />

#### 

#### <TextView

#### android:layout\_width="match\_parent"

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

#### android:layout\_marginStart="@dimen/\_10sdp"

#### android:fontFamily="@font/ubuntu\_medium"

#### android:includeFontPadding="false"

#### android:text="@string/add\_url"

#### android:textColor="@color/colorIcon"

#### android:textSize="@dimen/\_12ssp"

#### />

#### 

#### </LinearLayout>

#### 

#### 

#### <LinearLayout

#### android:id="@+id/layoutDeleteNote"

#### android:layout\_width="match\_parent"

#### android:layout\_height="@dimen/\_35sdp"

#### android:layout\_marginStart="@dimen/\_10sdp"

#### android:layout\_marginEnd="@dimen/\_10sdp"

#### android:gravity="center\_vertical"

#### android:orientation="horizontal"

#### android:visibility="gone"

#### tools:ignore="UseCompoundDrawables">

#### 

#### <ImageView

#### android:layout\_width="@dimen/\_22sdp"

#### android:layout\_height="@dimen/\_22sdp"

#### android:contentDescription="@string/app\_name"

#### android:src="@drawable/ic\_delete"

#### app:tint="@color/colorDelete"

#### />

#### 

#### <TextView

#### android:layout\_width="match\_parent"

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

#### android:layout\_marginStart="@dimen/\_10sdp"

#### android:fontFamily="@font/ubuntu\_medium"

#### android:includeFontPadding="false"

#### android:text="@string/delete\_note"

#### android:textColor="@color/colorIcon"

#### android:textSize="@dimen/\_12ssp"

#### />

#### 

#### </LinearLayout>

#### 

#### </LinearLayout>

#### Row item:

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

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

android:orientation="horizontal" android:layout\_width="match\_parent"

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

<TextView

android:id="@+id/tvName"

android:textSize="18sp"

android:text="bla bla"

android:layout\_weight="1"

android:layout\_width="0dp"

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

<TextView

android:id="@+id/tvEmail"

android:textSize="18sp"

android:text="bla bla"

android:layout\_weight="1"

android:layout\_width="0dp"

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

<TextView

android:id="@+id/tvAge"

android:textSize="18sp"

android:text="bla bla"

android:layout\_weight="1"

android:layout\_width="0dp"

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

<TextView

android:id="@+id/tvPass"

android:textSize="18sp"

android:text="bla bla"

android:layout\_weight="1"

android:layout\_width="0dp"

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

</LinearLayout>

#### 

#### other:

colors:

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

<resources>

<color name="purple\_200">#FFBB86FC</color>

<color name="purple\_500">#FF6200EE</color>

<color name="purple\_700">#FF3700B3</color>

<color name="teal\_200">#FF03DAC5</color>

<color name="teal\_700">#FF018786</color>

<color name="black">#FF000000</color>

<color name="white">#FFFFFFFF</color>

<!--

for notes centralized coloring

!-->

<color name = "colorPrimary">#292929</color>

<color name = "colorPrimaryDark">#292929</color>

<color name = "colorAccent">#F1992D</color>

<color name = "colorWhite">#FFFFFF</color>

<color name = "colorSearchIcon">#DBDBDB</color>

<color name = "colorTextHint">#7B7B7B</color>

<color name = "colorQuickActionBackground">#2F2D2E</color>

<color name = "colorIcon">#A4A4A4</color>

<color name = "colorTextSubtitle">#CECECE</color>

<color name = "colorDefaultNoteColor">#3333</color>

<color name = "colorNoteSubtitleText">#FAFAFA</color>

<color name = "colorMiscellaneousBackground">#1F1F1F</color>

<color name = "colorNoteColor2">#FDBE3B</color>

<color name = "colorNoteColor3">#FF4842</color>

<color name = "colorNoteColor4">#3A52Fc</color>

<color name = "colorNoteColor5">#000000</color>

<color name = "colorDelete">#FF4842</color>

<color name = "coloSearchBackground">#333333</color>

<array name="icons">

<item>@drawable/ic\_exit</item>

<item>@drawable/ic\_help</item>

<item>@drawable/ic\_notifications\_white\_24dp</item>

<item>@drawable/ic\_settings\_white\_24dp</item>

<item>@drawable/ic\_baseline\_menu\_book\_24</item>

</array>

<array name="colors">

<item>@android:color/holo\_blue\_light</item>

<item>@android:color/holo\_green\_dark</item>

<item>@android:color/holo\_red\_light</item>

<item>@android:color/holo\_purple</item>

<item>@android:color/holo\_orange\_light</item>

</array>

</resources>

strings:

<resources>

<string name="app\_name">OneAppToRuleThemAll</string>

<string name="notetitle">NoteTitle</string>

<string name="note\_subtitle">Note Subtitle</string>

<string name="type\_note\_here">Type note here!</string>

<string name="miscellaneous">Miscellaneous</string>

<string name="pick\_color">Pick color</string>

<string name="add\_url">Add Url</string>

<string name="enter\_url">Enter URl</string>

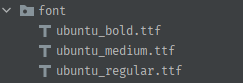
<string name="add">ADD</string>

<string name="delete\_note">Delete Note</string>

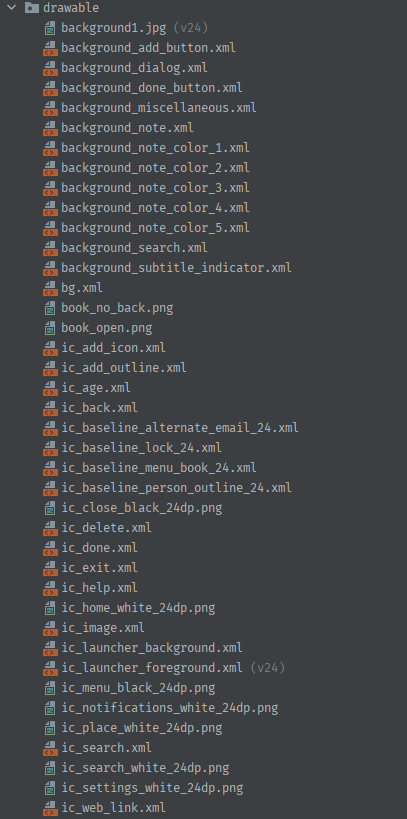
<string name="all\_notes">All Notes</string>

</resources>

fonts:



#### drawable:



#### 

#### Manifest

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

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

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

**package="com.example.oneapptorulethemall"**>

*<!--*

*for pictures*

*!*

*<permission android:name="android.permission.QUERY\_ALL\_PACKAGES" />*

*-->*

<**queries**>

<**intent**>

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

</**intent**>

</**queries**>

<**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />

<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** />

<**application**

**android:allowBackup="true"**

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

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

**android:requestLegacyExternalStorage="true"**

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

**android:supportsRtl="true"**

**android:theme="@style/Theme.OneAppToRuleThemAll"**>

<**service android:name=".MyService"**/>

<**activity**

**android:name=".Transfer\_page"**

**android:exported="false"**

**tools:ignore="Instantiatable"**>

<**meta-data**

**android:name="android.app.lib\_name"**

**android:value=""** />

</**activity**>

<**activity**

**android:name=".ExampleTH"**

**android:exported="false"**

**android:screenOrientation="portrait"**

**android:windowSoftInputMode="stateHidden"** />

<**activity**

**android:name=".activities.CreateNoteActivity"**

**android:exported="false"**

**android:screenOrientation="portrait"**

**android:windowSoftInputMode="stateHidden"** />

<**activity**

**android:name=".activities.notes\_page"**

**android:exported="false"**

**android:screenOrientation="portrait"**

**android:windowSoftInputMode="stateHidden"** />

<**activity**

**android:name=".Register\_page"**

**android:exported="false"** />

<**activity**

**android:name=".ResultActivity"**

**android:exported="false"** />

<**activity**

**android:name=".activities.Splash\_screen\_book"**

**android:exported="false"** />

<**activity**

**android:name=".activities.MainActivity"**

**android:exported="true"**>

<**intent-filter**>

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

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

</**intent-filter**>

</**activity**>

</**application**>

</**manifest**>

#### thread page:

**package** com.example.oneapptorulethemall;

**import** androidx.appcompat.app.AppCompatActivity;

**import** android.content.SharedPreferences;

**import** android.os.Bundle;

**import** android.os.Handler;

**import** android.os.Message;

**import** android.os.SystemClock;

**import** androidx.appcompat.app.AppCompatActivity;

**import** android.os.Bundle;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.TextView;

*/\*\**

*\* A very simple thread demo. It matches what happens in aSyncTaskDemo.*

*\* basically starts a thread to display the progress from 0 to 100 (in increments of 5).*

*\*/*

**public class** ExampleTH **extends** AppCompatActivity {

TextView **Progress**;

**int ProgressValue**;

Button **Button1**;

SharedPreferences **sharedPreferences**;

*//for threading and communication,*

**protected** Handler **handler**;

*//for the thread*

Thread **myThread**;

@Override

**protected void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.***activity\_example\_th***);

**Progress** = findViewById(R.id.***textView1***);

**Button1** = findViewById(R.id.***button1***);

**Button1**.setOnClickListener(**new** Button.OnClickListener() {

*// starts the Thread.*

@Override

**public void** onClick(View view) {

**myThread** = **new** Thread(**new** CountingThread(0));

**myThread**.start();

}

});

*//message handler for the animation.*

**handler** = **new** Handler(**new** Handler.Callback() {

@Override

**public boolean** handleMessage(Message msg) {

**if** (msg.**what** == 0) { *// update progress.*

**Progress**.setText(**"Progress: "** + **ProgressValue** + **"%"**);

} **else if** (msg.**what** == 1) { *//finished.*

*// Progress.setText("Completed: " + ProgressValue + "%");*

*// sp for admin actions*

**sharedPreferences** = getSharedPreferences(**"active\_user"**,0);

String name = **sharedPreferences**.getString(**"name"**, **null**);

**Progress**.setText(**" your permission is "** + name);

}

**return true**;

}

});

}

**class** CountingThread **implements** Runnable {

**int i** = 0; *//default value of zero.*

CountingThread(**int** start) {

**i** = start; *//starting value for the count.*

}

@Override

**public void** run() {

**while** (**i** < 100) {

SystemClock.*sleep*(250);

**i**++;

**if** (**i** % 5 == 0) {

*//update UI*

**ProgressValue** = **i**;

*//send message to update the screen.*

**handler**.sendEmptyMessage(0);

}

}

**ProgressValue** = **i**;

**handler**.sendEmptyMessage(1); *//we are done!*

}

}

}

#### xml:

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

<**androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"**

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

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

**android:layout\_width="match\_parent"**

**android:layout\_height="match\_parent"**

**tools:context=".ExampleTH"**>

<**Button**

**android:id="@+id/button1"**

**android:layout\_width="0dp"**

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

**android:layout\_marginLeft="8dp"**

**android:layout\_marginTop="8dp"**

**android:layout\_marginRight="8dp"**

**android:text="What's my permission?"**

**app:layout\_constraintLeft\_toLeftOf="parent"**

**app:layout\_constraintRight\_toRightOf="parent"**

**app:layout\_constraintTop\_toTopOf="parent"**

**tools:layout\_constraintLeft\_creator="1"**

**tools:layout\_constraintTop\_creator="1"** />

<**TextView**

**android:id="@+id/textView1"**

**android:layout\_width="0dp"**

**android:layout\_height="0dp"**

**android:layout\_marginLeft="0dp"**

**android:layout\_marginTop="0dp"**

**android:layout\_marginRight="8dp"**

**android:layout\_marginBottom="8dp"**

**android:text="nothing yet."**

**android:textAppearance="?android:attr/textAppearanceLarge"**

**app:layout\_constraintBottom\_toBottomOf="parent"**

**app:layout\_constraintHorizontal\_bias="0.0"**

**app:layout\_constraintLeft\_toLeftOf="@+id/button1"**

**app:layout\_constraintRight\_toRightOf="parent"**

**app:layout\_constraintTop\_toBottomOf="@+id/button1"**

**app:layout\_constraintVertical\_bias="0.0"**

**tools:layout\_constraintLeft\_creator="1"**

**tools:layout\_constraintTop\_creator="1"** />

</**androidx.constraintlayout.widget.ConstraintLayout**>