

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Національний технічний університет України «Київський  
політехнічний  
інститут ім. Ігоря Сікорського»  
ФІОТ  
Кафедра обчислювальної техніки

Звіт  
до лабораторної роботи №4-10  
з предмету «Розроблення програм для Android»

Виконали:  
Студентки групи ІІ-84  
Коломієць Євгенія Валеріївна  
Шахова Поліна Миколаївна

Перевірив:  
Шимкович Володимир  
Миколайович

Київ 2021

**Тема:** Додаток для читання електронних книг

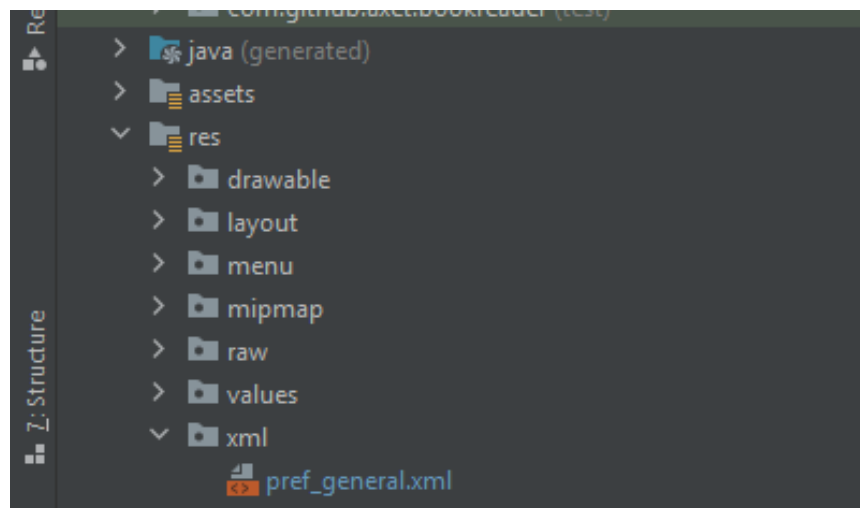
## **Опис**

1. Основні формати електронної книги, що підтримуються додатком:
  - ePub (включаючи основні можливості ePub3)
  - fb2 (.zip)
  - інші ( RTF, doc (MS Word), HTML, звичайний текст).
  - PDF, DjVu і комікси
2. Можливість використовувати різні шрифти і вбудовані стилі; яскравість екрана можна регулювати під час читання, а також можна вибрати різні кольорові теми (денна/нічна)
3. Автоматичне знаходження книг та документів різних форматів
4. Можливість ручного пошуку книжок у своїй бібліотеці та сортування їх за назвою або автором, в алфавітному порядку або навпаки

## Ресурси та стилі

В Android прийнято тримати деякі об'єкти - зображення, рядкові константи, кольору, анімацію, стилі і т.п. за межами вихідного коду. Система підтримує зберігання ресурсів в окремих файлах. Ресурси легше підтримувати, оновлювати, редагувати.

Кожна програма на Android містить каталог для ресурсів `res`. Інформація в каталозі ресурсів буде доступна в додатку через клас `R`, який автоматично генерується середовищем розробки. Тобто зберігання файлів і даних в ресурсах (в каталозі `res`) робить їх легкодоступними для використання в коді програми.



```
<resources>
    <style name="AppThemeLight" parent="AppThemeLightLib">
        <item name="side_nav_bar">@drawable/side_nav_bar_light</item>
    </style>

    <style name="AppThemeLight.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>

    <style name="AppBarOverlay"
parent="ThemeOverlay.AppCompat.Dark.ActionBar" />

    <style name="AppThemeLight.PopupOverlay"
parent="ThemeOverlay.AppCompat.Light" />

    <style name="AppThemeDark" parent="AppThemeDarkLib">
        <item name="side_nav_bar">@drawable/side_nav_bar_dark</item>
        <item name="android:navigationBarColor">@color/black</item>
    </style>

    <style name="AppThemeDark.NoActionBar">
        <item name="windowActionBar">false</item>
        <item name="windowNoTitle">true</item>
    </style>

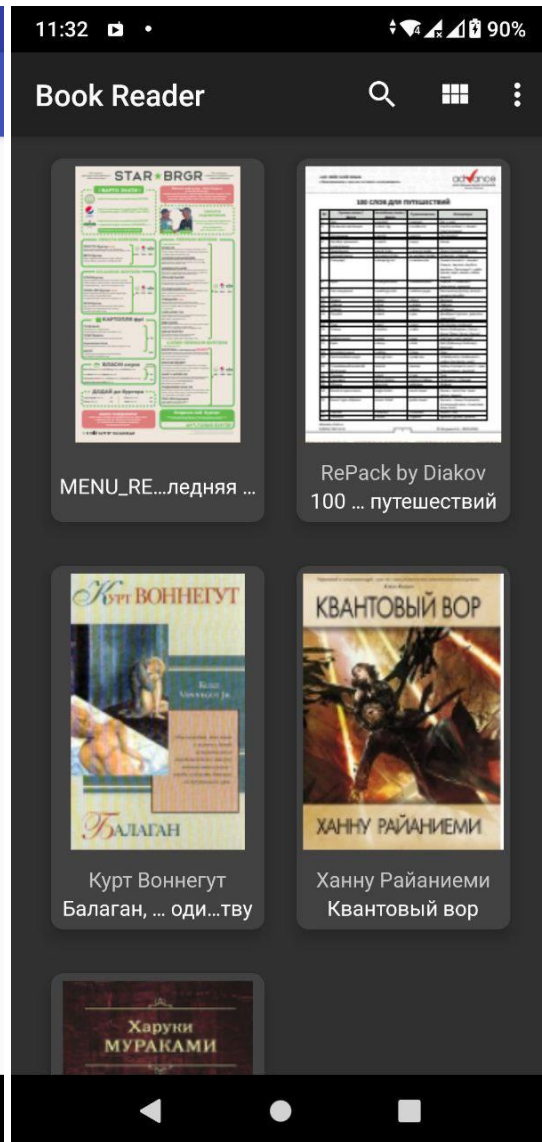
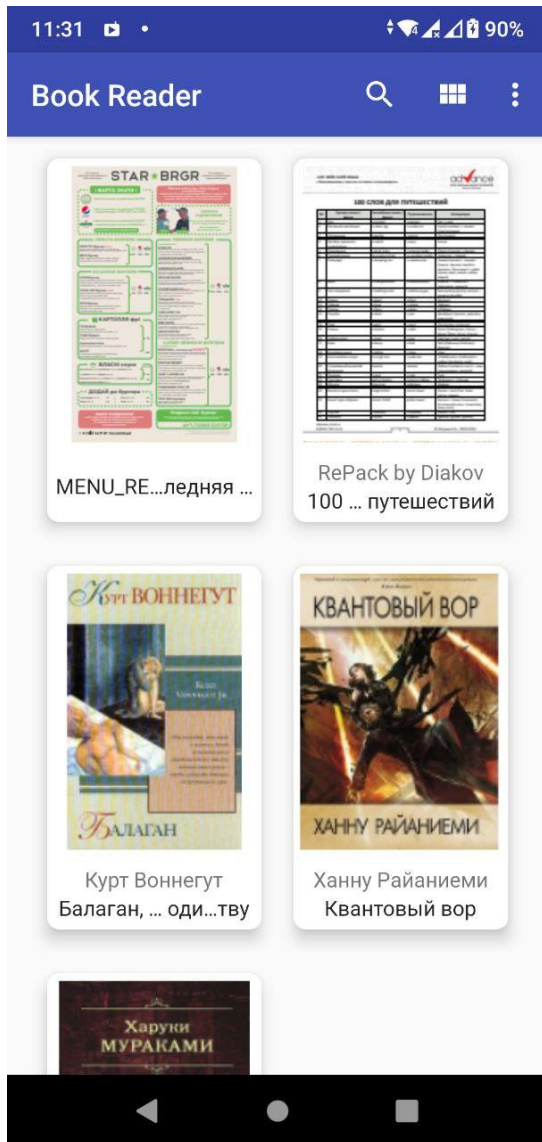
    <style name="AppThemeDark.PopupOverlay"
parent="ThemeOverlay.AppCompat.Dark" />
```

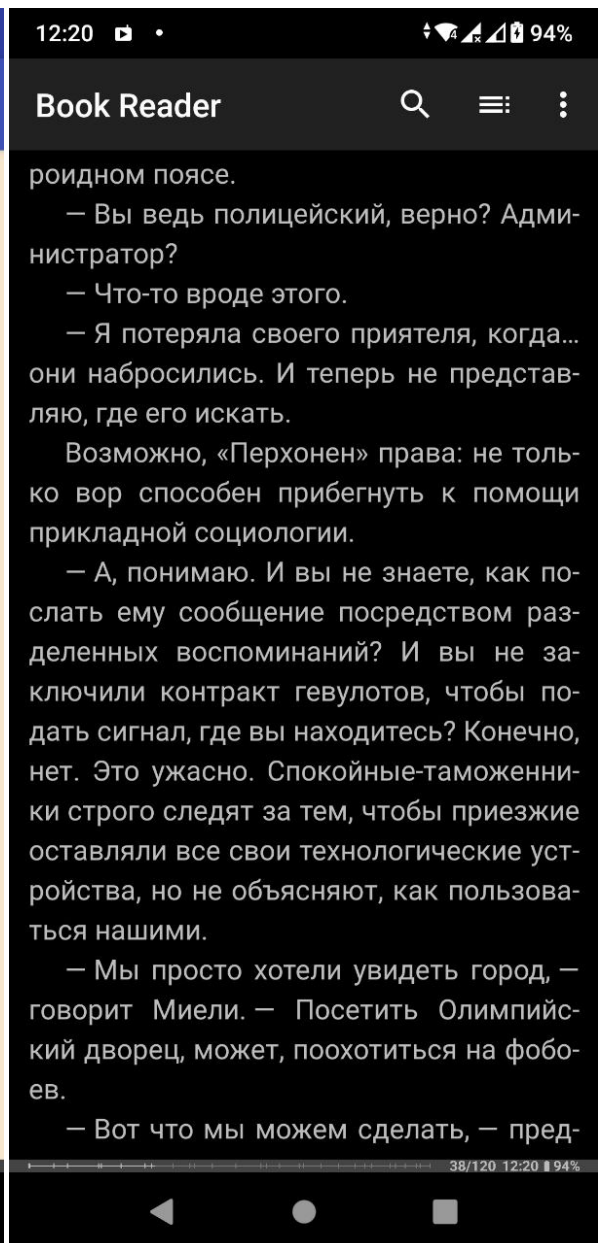
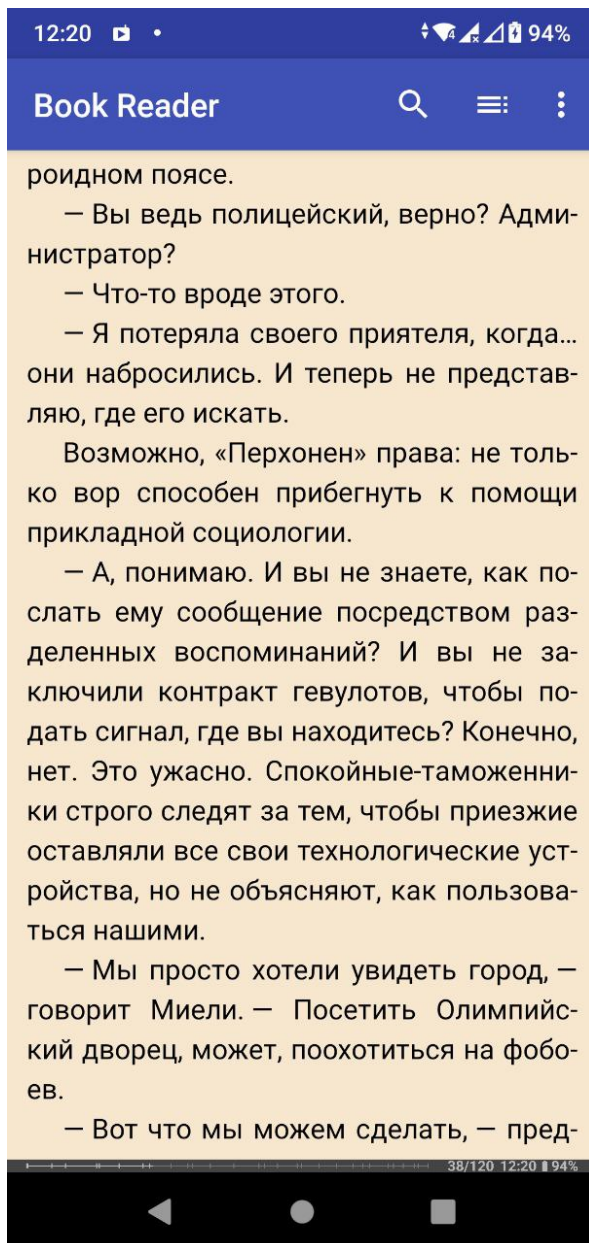
```

<style name="toolbar_bottom_icon_text" parent="toolbar_icon_text">
    <item name="android:layout_marginBottom">7dp</item>
    <item name="android:maxWidth">60dp</item>
    <item name="android:singleLine">true</item>
</style>
</resources>

```

Приклад світлої та темної теми





## Зміни візуального стану елемента

From `widgets/FBReaderView.java`

```
public BrightnessGesture (FBReaderView view) {
    fb = view;
    areaWidth = ThemeUtils.dp2px (fb.getContext (), 36); // 24dp -
icon; 48dp - button
}

public boolean onTouchEvent (MotionEvent e) {
    int x = (int) e.getX ();
    int y = (int) e.getY ();
    switch (e.getActionMasked ()) {
        case MotionEvent.ACTION_DOWN:
            if
(fb.app.MiscOptions.AllowScreenBrightnessAdjustment.getValue () && x <
areaWidth) {
                myIsBrightnessAdjustmentInProgress = true;
                myStartY = y;
            }
        }
    }
}
```

```

        myStartBrightness = fb.widget.getScreenBrightness();
        return true;
    }
    break;
case MotionEvent.ACTION_MOVE:
    if (myIsBrightnessAdjustmentInProgress) {
        if (x >= areaWidth * 2) {
            myIsBrightnessAdjustmentInProgress = false;
            return false;
        } else {
            final int delta = (myStartBrightness + 30) *
(myStartY - y) / fb.getHeight();
            fb.widget.setScreenBrightness(myStartBrightness +
delta);

            return true;
        }
    }
    break;
case MotionEvent.ACTION_UP:
case MotionEvent.ACTION_CANCEL:
    if (myIsBrightnessAdjustmentInProgress) {
        myIsBrightnessAdjustmentInProgress = false;
        return true;
    }
    break;
}
return false;
}

public Integer setScreenBrightness(int percent) {
    if (percent < 1)
        percent = 1;
    else if (percent > 100)
        percent = 100;

    final float level;
    final Integer oldColorLevel = myColorLevel;
    if (percent >= 25) {
        // 100 => 1f; 25 => .01f
        level = .01f + (percent - 25) * .99f / 75;
        myColorLevel = null;
    } else {
        level = .01f;
        myColorLevel = 0x60 + (0xFF - 0x60) * Math.max(percent, 0) /
25;
    }

    final WindowManager.LayoutParams attrs = fb.w.getAttributes();
    attrs.screenBrightness = level;
    fb.w.setAttributes(attrs);

    return myColorLevel;
}

public int getScreenBrightness() {
    if (myColorLevel != null)
        return (myColorLevel - 0x60) * 25 / (0xFF - 0x60);

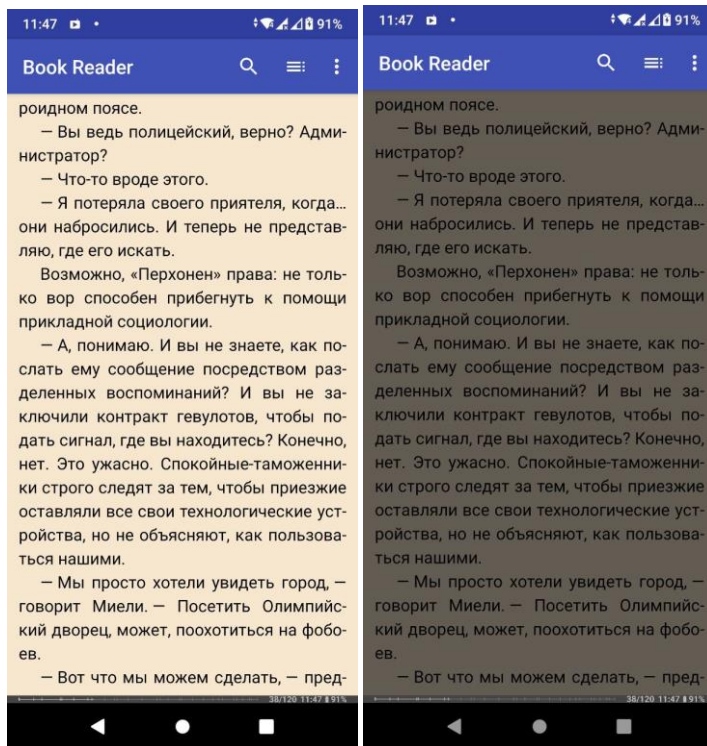
    float level = fb.w.getAttributes().screenBrightness;
    level = level >= 0 ? level : .5f;

    // level = .01f + (percent - 25) * .99f / 75;
    return 25 + (int) ((level - .01f) * 75 / .99f);
}

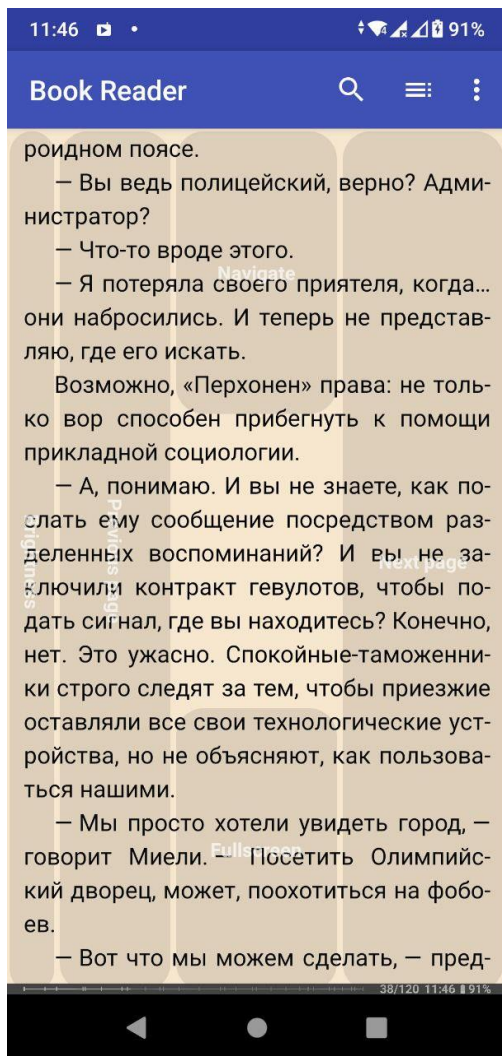
```

}

## Приклад зміни яскравості







## Прив'язка об'єктів

From `layout/book_item.xml`

```
<?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:layout_width="wrap_content"
    android:layout_height="wrap_content">

    <androidx.cardview.widget.CardView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:cardCornerRadius="8dp"
        app:cardElevation="8dp"
        app:cardUseCompatPadding="true">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:orientation="vertical">

            <FrameLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:layout_margin="2dp">
```



```

<ImageView
    android:id="@+id/book_cover"
    android:layout_width="@dimen/image_size"
    android:layout_height="190dp"
    android:layout_gravity="center"
    android:padding="3dp"
    android:src="@mipmap/ic_launcher" />

<ProgressBar
    android:id="@+id/book_progress"
    android:layout_width="15dp"
    android:layout_height="15dp"
    android:layout_gravity="right|bottom"
    android:layout_margin="5dp"
    android:indeterminate="true" />
</FrameLayout>

<FrameLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_margin="5dp">

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="vertical"
        android:visibility="invisible">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="" />
    </LinearLayout>

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:orientation="vertical">

        <com.github.axet.androidlibrary.widgets.TextMax
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center">

            <TextView
                android:id="@+id/book_authors"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="@string/app_name" />
        </com.github.axet.androidlibrary.widgets.TextMax>

        <com.github.axet.androidlibrary.widgets.TextMax

```

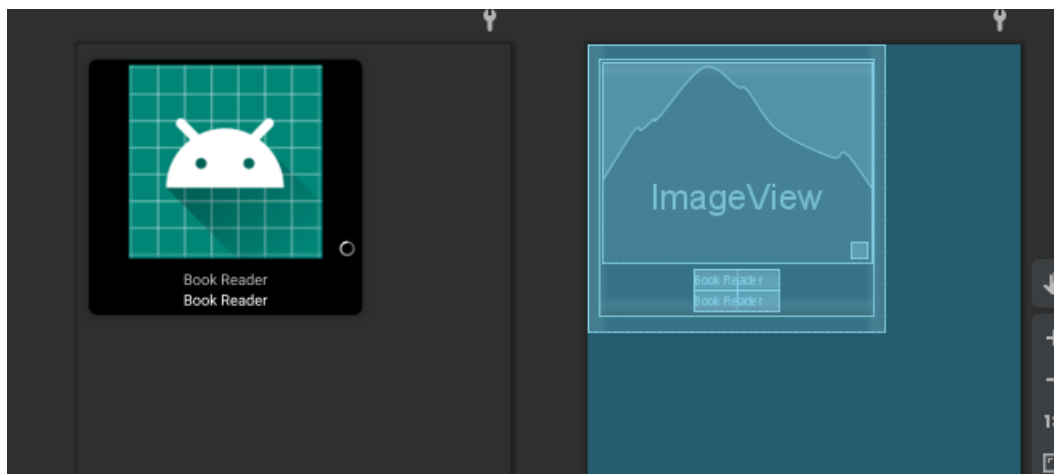
```

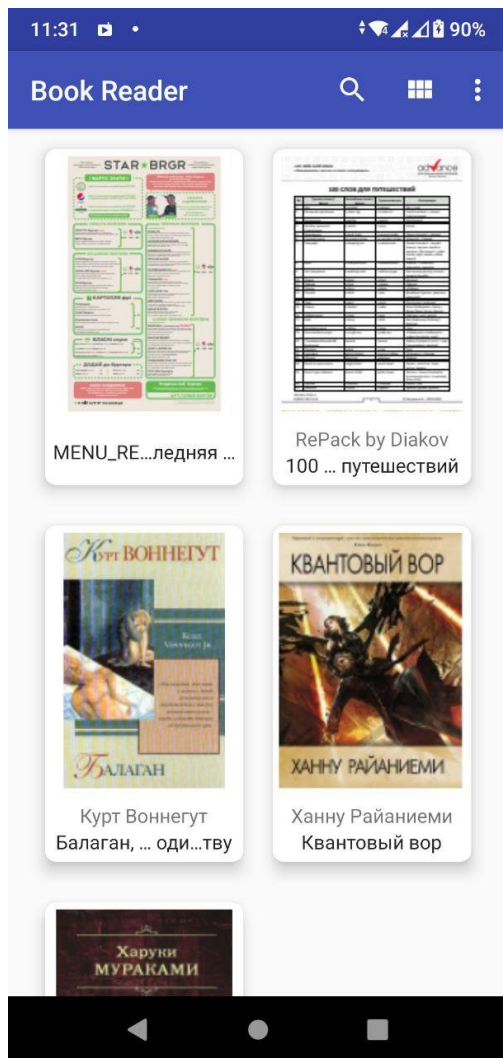
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center">

        <TextView
            android:id="@+id/book_title"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text="@string/app_name"

            android:textAppearance="@style/TextAppearance.AppCompat.Body1" />
        </com.github.axet.androidlibrary.widgets.TextMax>
    </LinearLayout>
</FrameLayout>
</LinearLayout>
</androidx.cardview.widget.CardView>
</LinearLayout>

```





From `layout/book_list_item.xml`

```
<?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:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <androidx.cardview.widget.CardView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        app:cardCornerRadius="8dp"
        app:cardElevation="8dp"
        app:cardUseCompatPadding="true">

        <FrameLayout
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_margin="2dp">

            <ImageView
                android:id="@+id/book_cover"
                android:layout_width="64dp"
                android:layout_height="95dp"
                android:layout_gravity="center"
```

```

        android:padding="5dp"
        android:src="@mipmap/ic_launcher" />

<ProgressBar
    android:id="@+id/book_progress"
    android:layout_width="15dp"
    android:layout_height="15dp"
    android:layout_gravity="right|bottom"
    android:layout_margin="5dp"
    android:indeterminate="true" />

</FrameLayout>
</androidx.cardview.widget.CardView>
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="2dp"
    android:orientation="vertical"
    android:layout_gravity="center">

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text=""
            android:visibility="invisible" />

        <com.github.axet.androidlibrary.widgets.TextMax
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TextView
                android:id="@+id/book_authors"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"
                android:text="Authors" />
        </com.github.axet.androidlibrary.widgets.TextMax>
    </FrameLayout>

    <FrameLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:text=""
            android:visibility="invisible" />

        <com.github.axet.androidlibrary.widgets.TextMax
            android:layout_width="wrap_content"
            android:layout_height="wrap_content">

            <TextView
                android:id="@+id/book_title"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center"

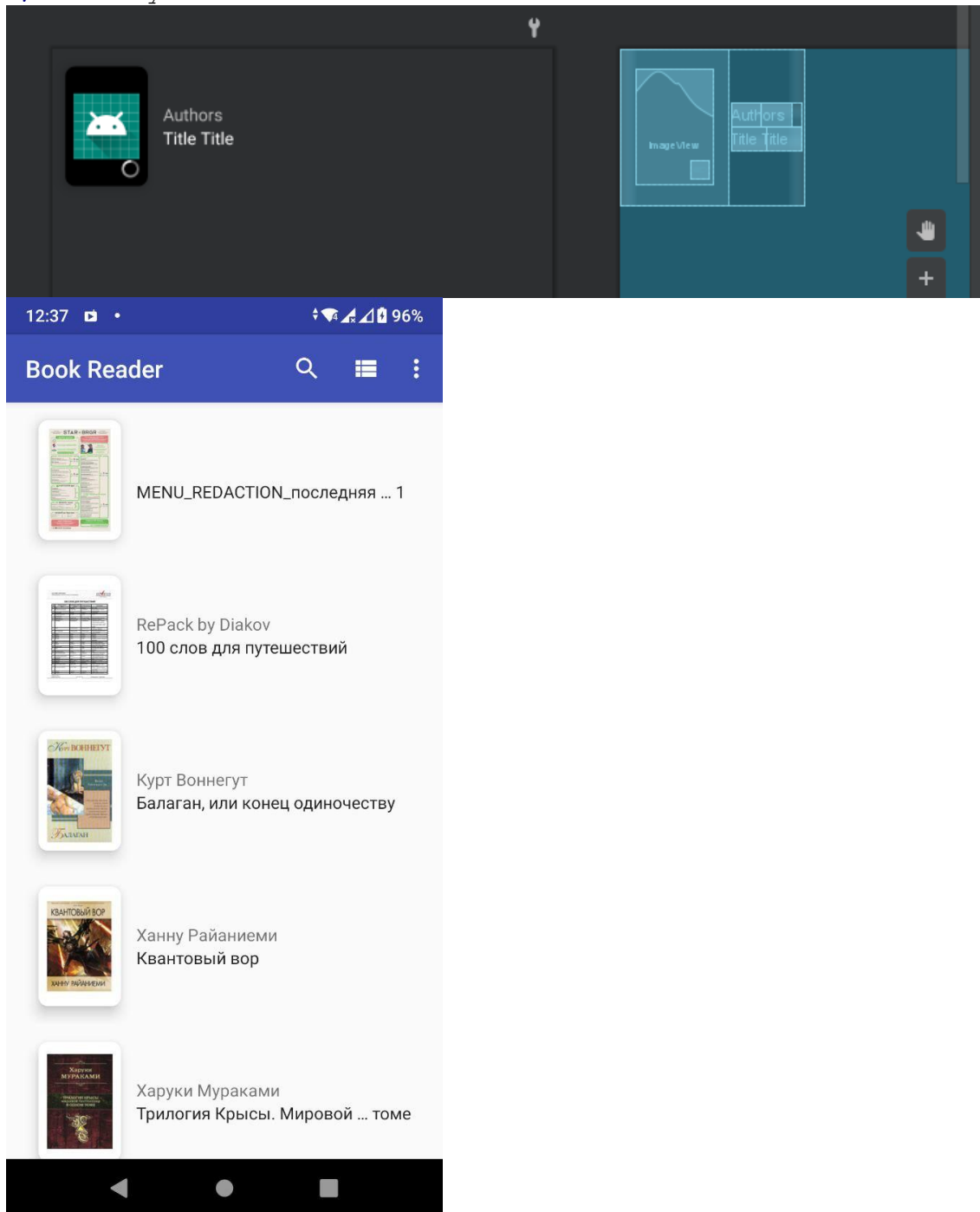
```

```

        android:text="Title Title"

        android:textAppearance="@style/TextAppearance.AppCompat.Body1" />
        </com.github.axet.androidlibrary.widgets.TextMax>
    </FrameLayout>
</LinearLayout>
</LinearLayout>

```



У файлі `book_item.xml` розмітка зроблена лише для одного екземпляру книжки. Загальний зовнішній вигляд сторінки відображення книжок описується у `LibraryFragment.java`:

```

public void updateGrid() {
    final SharedPreferences shared =
    PreferenceManager.getDefaultSharedPreferences(context);

```

```

        if (shared.getString(BookApplication.PREFERENCE_LIBRARY_LAYOUT +
getLayout(), "").equals("book_list_item")) {
            setNumColumns(1);

            layout = R.layout.book_list_item;
        } else {
            setNumColumns(2);
            layout = R.layout.book_item;
        }
    }
}

```

## ListView

### From xml/pref\_general.xml

```

<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
    <com.github.axet.bookreader.widgets.StoragePathPreferenceCompat
        android:defaultValue="My Readings"
        android:key="storage_path"
        android:summary="@string/pref_storagepath_summary"
        android:title="@string/pref_storagepath_title" />

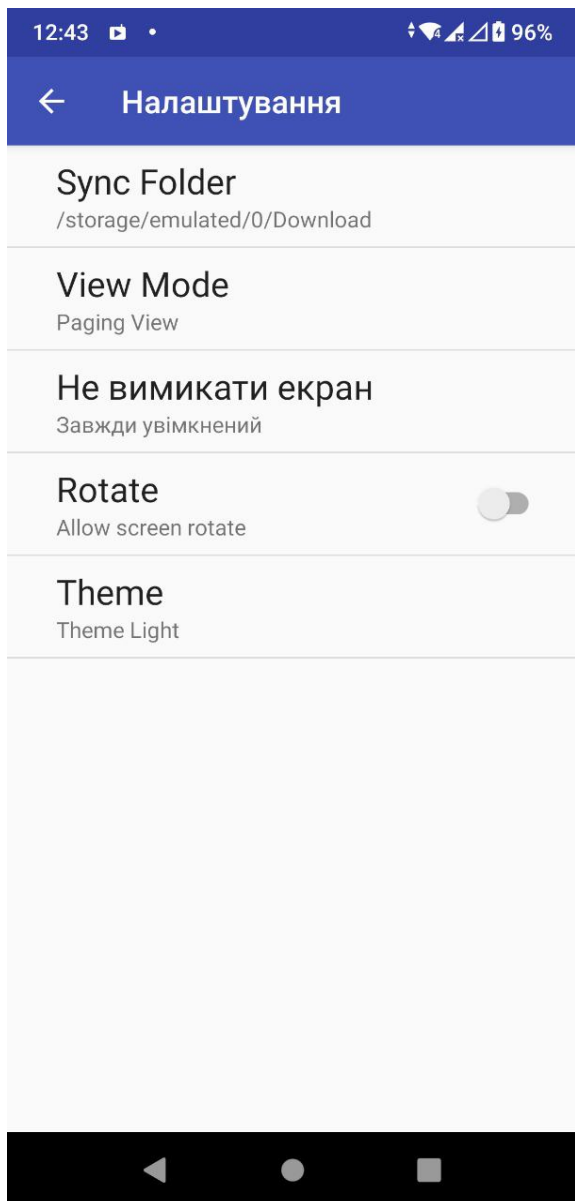
    <ListPreference
        android:defaultValue="PAGING"
        android:entries="@array/viewmode_text"
        android:entryValues="@array/viewmode_value"
        android:key="view_mode"
        android:negativeButtonText="@null"
        android:positiveButtonText="@null"
        android:title="@string/view_mode" />

    <com.github.axet.androidlibrary.preferences.ScreenlockPreference
        android:defaultValue="0"
        android:entries="@array/screen_lock_text"
        android:entryValues="@array/screen_lock_values"
        android:key="screen_lock"
        android:summary=""
        android:title="@string/pref_screenlock_title" />

    <com.github.axet.androidlibrary.preferences.RotatePreferenceCompat
        android:defaultValue="false"
        android:key="rotate"
        android:summary="@string/pref_rotate_summary"
        android:title="@string/pref_rotate_title" />

    <ListPreference
        android:defaultValue="@string/Theme_Light"
        android:entries="@array/theme_text"
        android:entryValues="@array/theme_value"
        android:key="theme"
        android:negativeButtonText="@null"
        android:positiveButtonText="@null"
        android:title="@string/pref_theme_title" />
</PreferenceScreen>

```

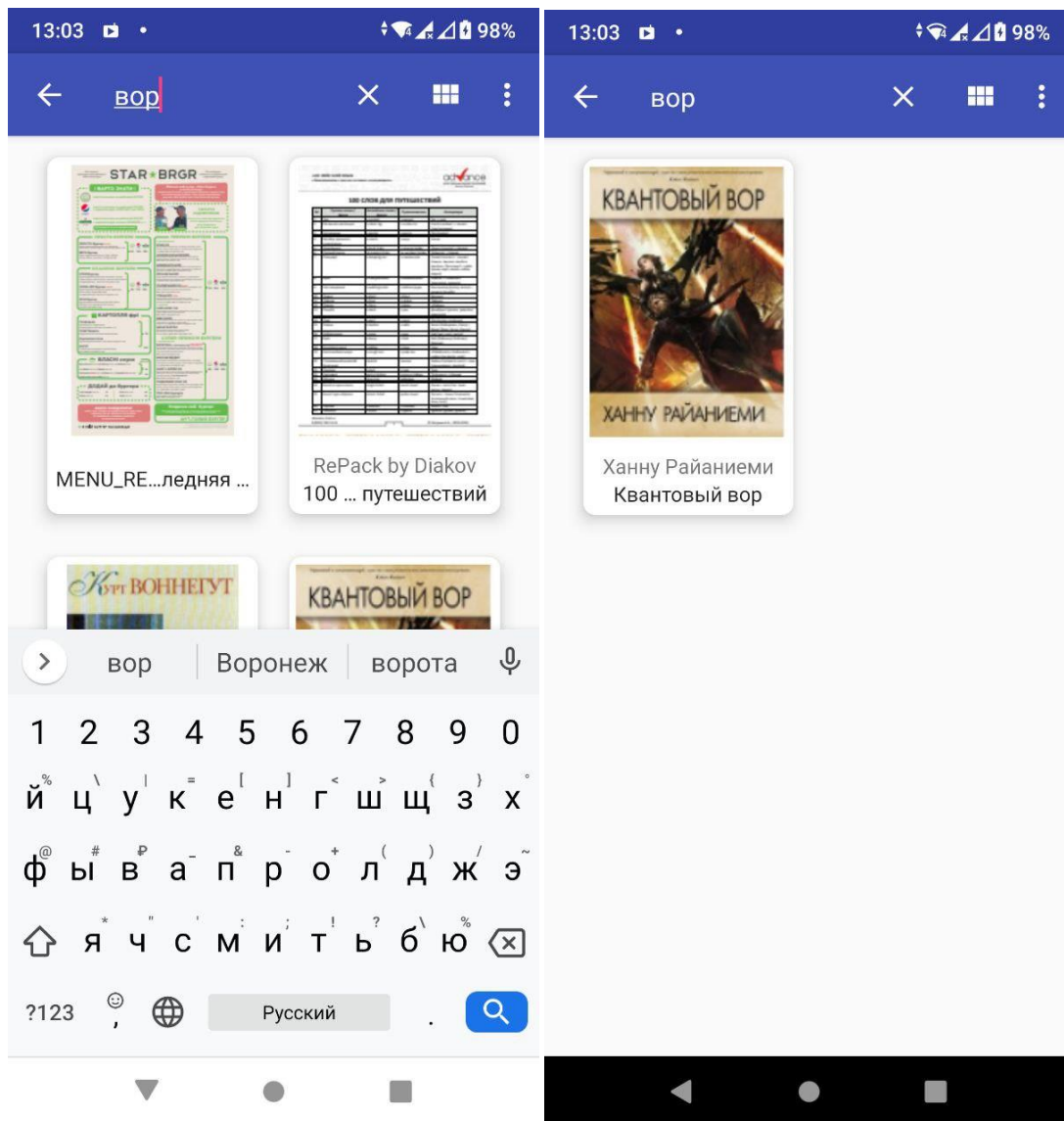


## Навігація

`NavigationView.OnNavigationItemSelectedListener` - Лісєнер для обробки подій на елементах навігації. Викликається, коли вибрано елемент у навігаційному меню.

Приклад перемикання між фрагментами (реалізація пошуку файлів)





Самі фрагменти успадковуються від `androidx.fragment.app.Fragment`. Існує підкласи фрагментів: `ListFragment`, `DialogFragment`, `PreferenceFragment`, `WebViewFragment` і ін.

Для взаємодії між фрагментами використовується клас `android.app.FragmentManager` - спеціальний менеджер по фрагментах.

```
public class MainActivity extends FullscreenActivity implements
NavigationView.OnNavigationItemSelectedListener {
    ...
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);

        MenuItem searchMenu = menu.findItem(R.id.action_search);

        final SharedPreferences shared =
        PreferenceManager.getDefaultSharedPreferences(this);
        ...

        final SearchView searchView = (SearchView)
        MenuItemCompat.getActionView(searchMenu);
```

```

        searchView.setOnQueryTextListener(new
SearchView.OnQueryTextListener() {
    @SuppressWarnings("RestrictedApi")
    @Override
    public boolean onQueryTextSubmit(String query) {
        lastSearch = query;
        searchView.clearFocus();
        FragmentManager fm = getSupportFragmentManager();
        for (Fragment f : fm.getFragments()) {
            if (f != null && f.isVisible() && f instanceof
SearchListener) {
                SearchListener s = (SearchListener) f;
                s.search(searchView.getQuery().toString());
            }
        }
        return true;
    }

    @Override
    public boolean onQueryTextChange(String newText) {
        return false;
    }
});
searchView.setOnSearchClickListener(new View.OnClickListener() {
    @SuppressWarnings("RestrictedApi")
    @Override
    public void onClick(View v) {
        if (lastSearch != null && !lastSearch.isEmpty())
            searchView.setQuery(lastSearch, false);
        FragmentManager fm = getSupportFragmentManager();
        for (Fragment f : fm.getFragments()) {
            if (f != null && f.isVisible() && f instanceof
SearchListener) {
                SearchListener s = (SearchListener) f;
                searchView.setQueryHint(s.getHint());
            }
        }
    }
});
searchView.setOnCollapsedListener(new
SearchView.OnCollapsedListener() {
    @SuppressWarnings("RestrictedApi")
    @Override
    public void onCollapsed() {
        FragmentManager fm = getSupportFragmentManager();
        for (Fragment f : fm.getFragments()) {
            if (f != null && f.isVisible() && f instanceof
SearchListener) {
                SearchListener s = (SearchListener) f;
                s.searchClose();
            }
        }
    }
});
searchView.setOnCloseButtonListener(new
SearchView.OnCloseButtonListener() {
    @Override
    public void onClosed() {
        lastSearch = "";
    }
});

return true;
}

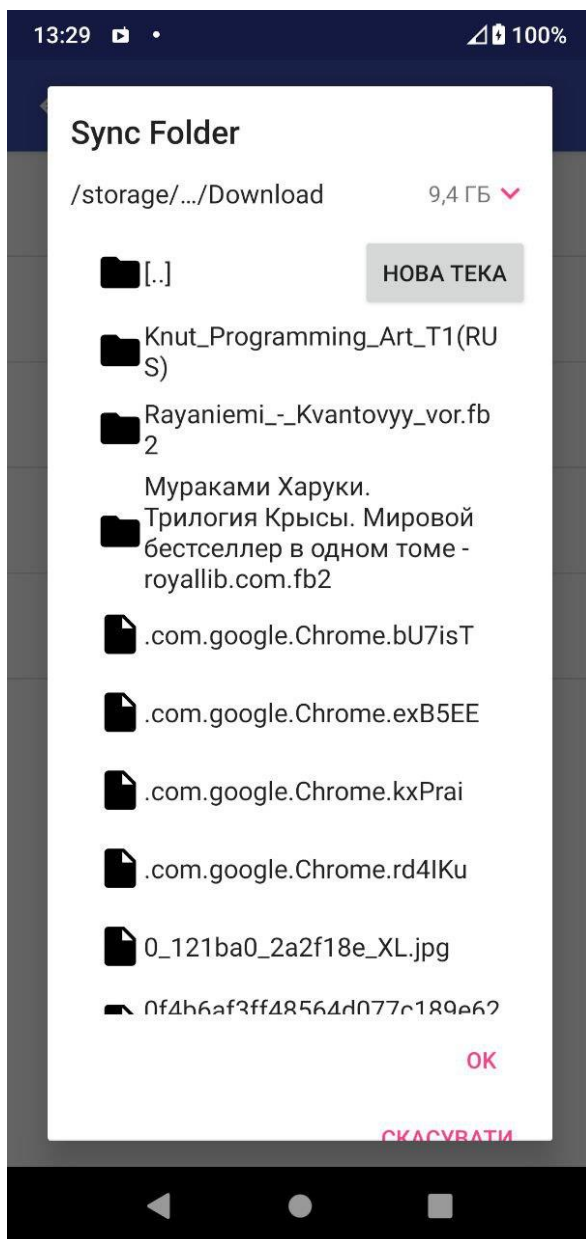
```

```
}
```

## Збереження даних. Робота з файлами

### Отримання шляху до файлу

```
public Uri getStoragePath() {  
    SharedPreferences shared =  
    PreferenceManager.getDefaultSharedPreferences(context);  
    String path = shared.getString(BookApplication.PREFERENCE_STORAGE,  
    null);  
    if (path == null)  
        return Uri.fromFile(getLocalStorage());  
    else  
        return getStoragePath(path);  
}
```



## Розпакування книжки

```
public Book load(InputStream is, Uri u) {
    Uri storage = getStoragePath();

    String s = storage.getScheme();
    ...
    FileTypeDetector.Detector[] dd = supported();

    book.md5 = FileTypeDetector.detecting(context, dd, is, os, u);

    for (FileTypeDetector.Detector d : dd) {
        if (d.detected) {
            book.ext = d.ext;
            if (d instanceof
FileTypeDetector.FileTypeDetectorZipExtract.Handler) {
                FileTypeDetector.FileTypeDetectorZipExtract.Handler e
= (FileTypeDetector.FileTypeDetectorZipExtract.Handler) d;
                if (!tmp) { // !tmp
                    File z = file;
                    file = createTempBook("tmp");
                    book.md5 = e.extract(z, file);
                    tmp = true; // force to delete 'fbook.file'
                } else { // tmp
                    File tt = createTempBook("tmp");
                    book.md5 = e.extract(file, tt);
                    file.delete(); // delete old
                    file = tt; // tmp = true
                }
            }
            break; // priority first - more important
        }
    }

    if (book.ext == null)
        throw new RuntimeException("Unsupported format");

    if (book.ext.equals(ComicsPlugin.CBR)) { // handling cbz solid
archives
        File cbz = null;
        try {
            final Archive archive = new Archive(new
NativeStorage(file));
            if (archive.getMainHeader().isSolid()) {
                cbz = createTempBook("tmp");
                OutputStream zos = new FileOutputStream(cbz);
                zos = new BufferedOutputStream(zos);
                ZipOutputStream out = new ZipOutputStream(zos);
                List<FileHeader> list = archive.getFileHeaders();
                for (FileHeader h : list) {
                    if (h.isDirectory())
                        continue;

                    ZipEntry entry = new
ZipEntry(RarSAF.getRarFileName(h));
                    out.putNextEntry(entry);

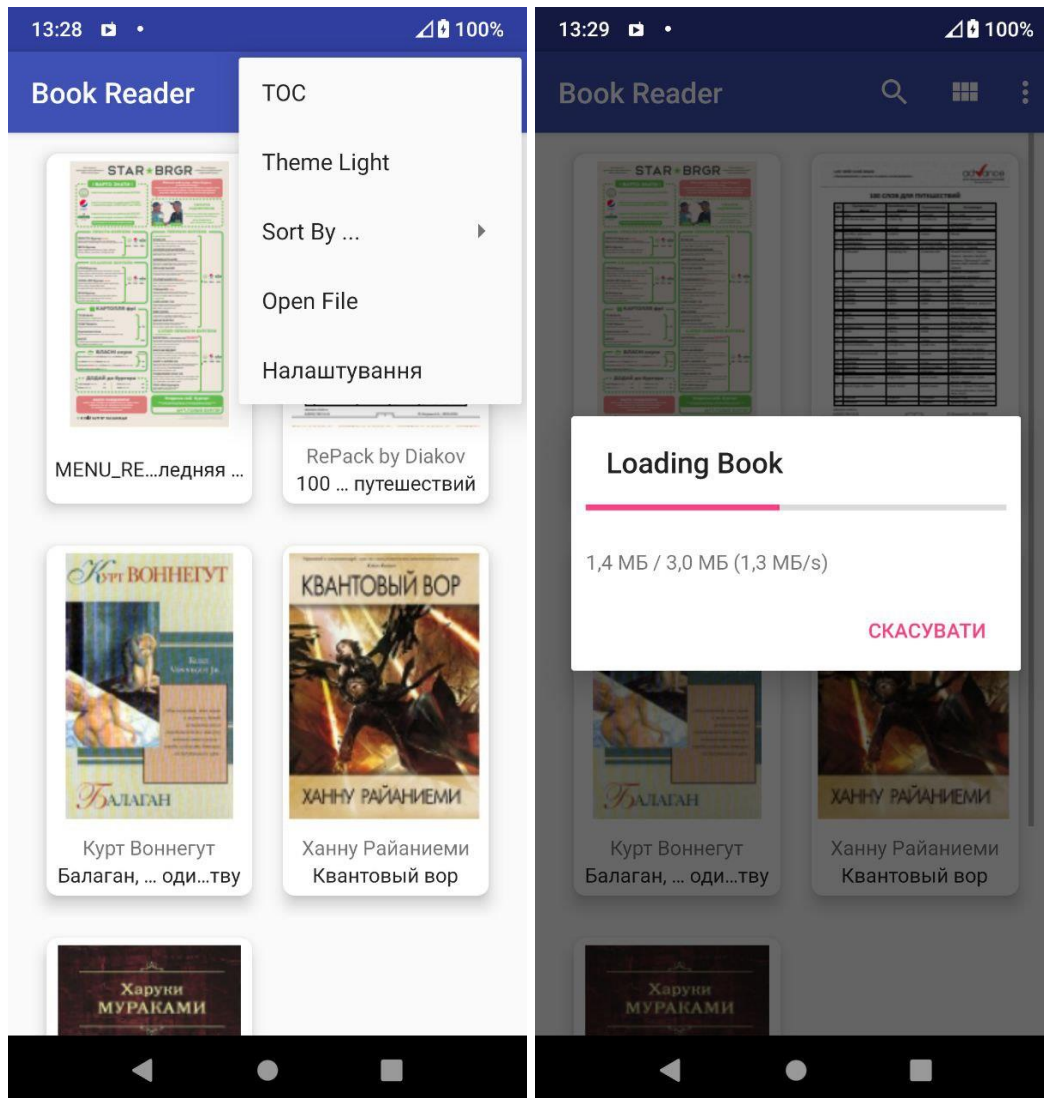
                    archive.extractFile(h, out);
                }
                out.close();
                if (tmp)
                    file.delete();
                book.ext = ComicsPlugin.CBZ;
                file = cbz;
            }
        }
    }
}
```

```

        tmp = true;
    }
} catch (Exception e) {
    if (cbz != null)
        cbz.delete();
    throw new RuntimeException("unsupported rar", e);
}
}

...
return book;
}

```



## Fonts

From ReaderFragment.java

```

public static class FontAdapter extends RecyclerView.Adapter<FontHolder> {
    Context context;
    public ArrayList<FontView> ff = new ArrayList<>();
    public int selected;
    public AdapterView.OnItemClickListener clickListener;

    public FontAdapter(Context context) {

```

```

        this.context = context;
    }

    public void addBasics() {
        add("sans-serif"); // "normal"
        add("serif");
        add("monospace");
    }

    ...
}

```

