

Moodr

By IBRice101

For CMP309

Submitted 17/05/2022



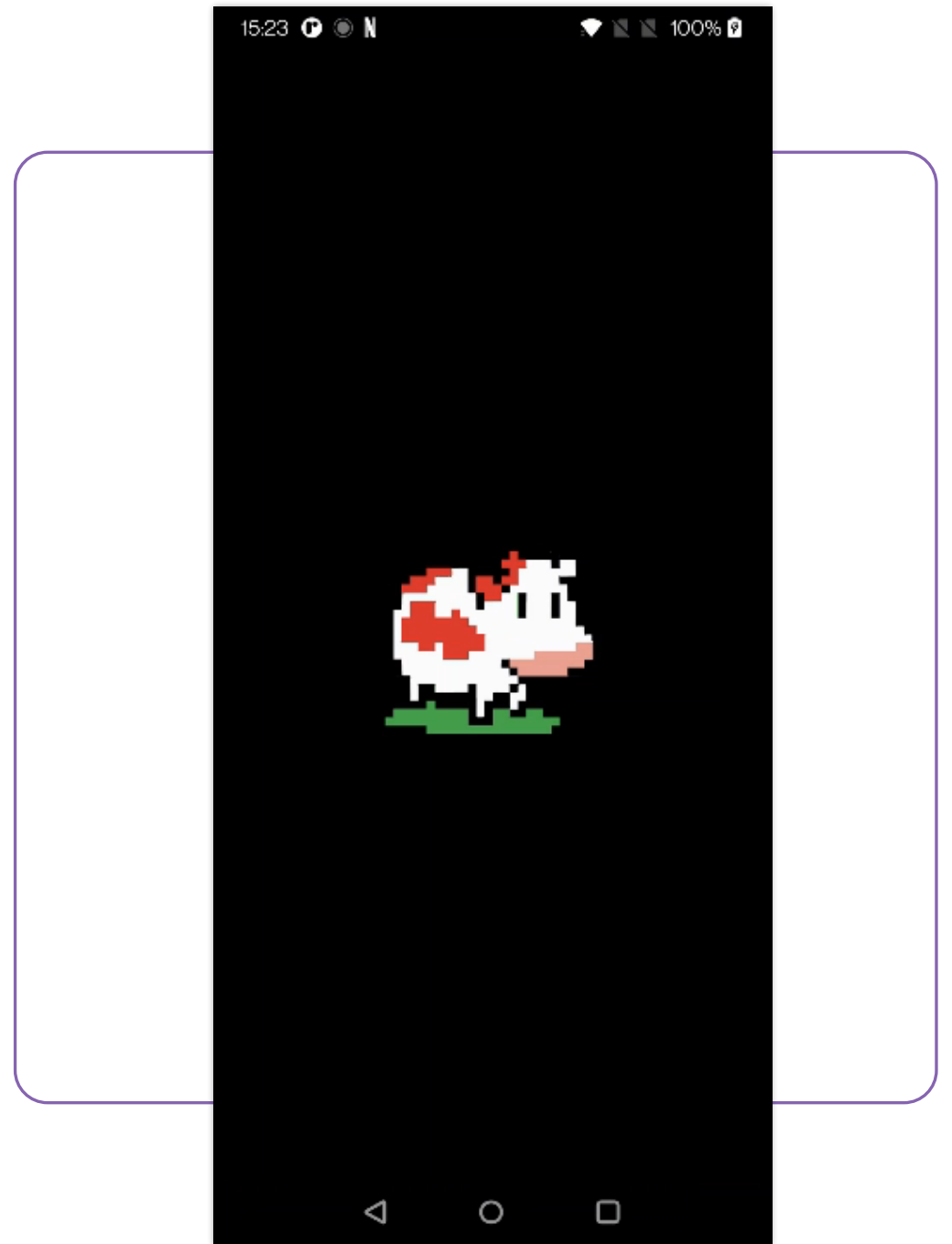
Overview of the App

- Moodr is a mood improvement tool with:
 - A Diary
 - Three Good Things
 - A habit-forming tool
- A section that allows the user to reflect on these



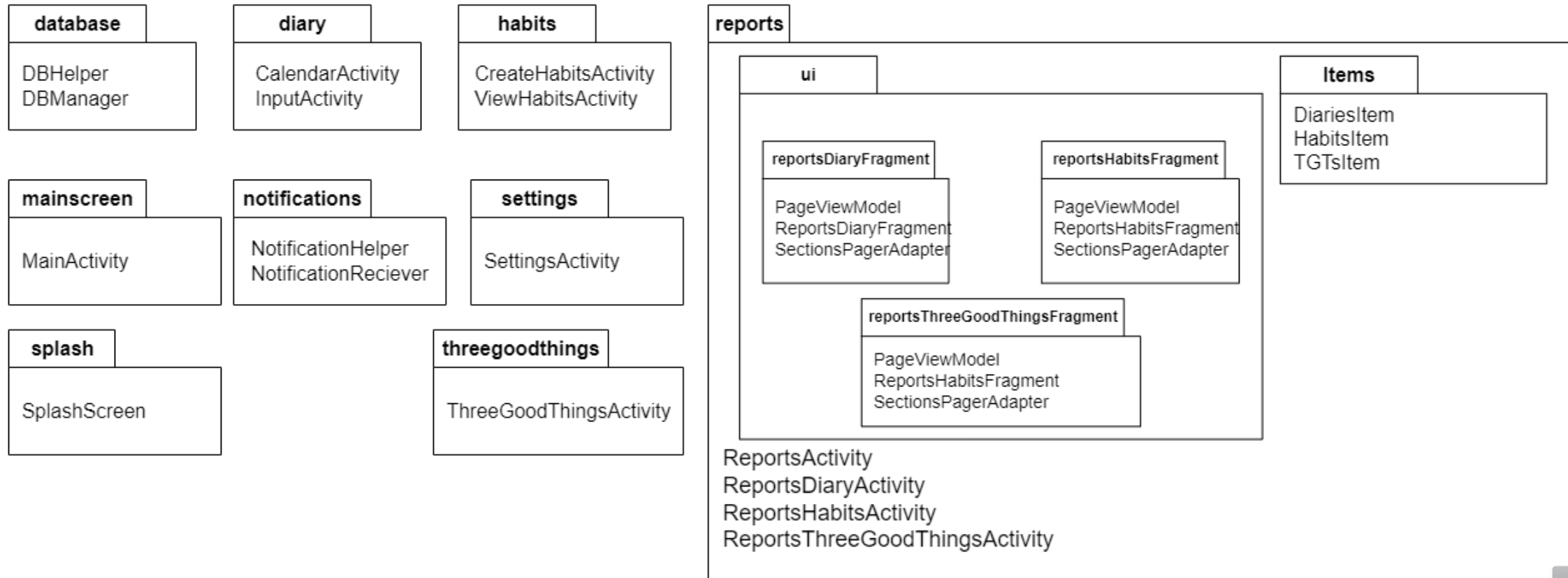
Moodr

- Video also available at <https://youtu.be/2C-dJKbjT4I>

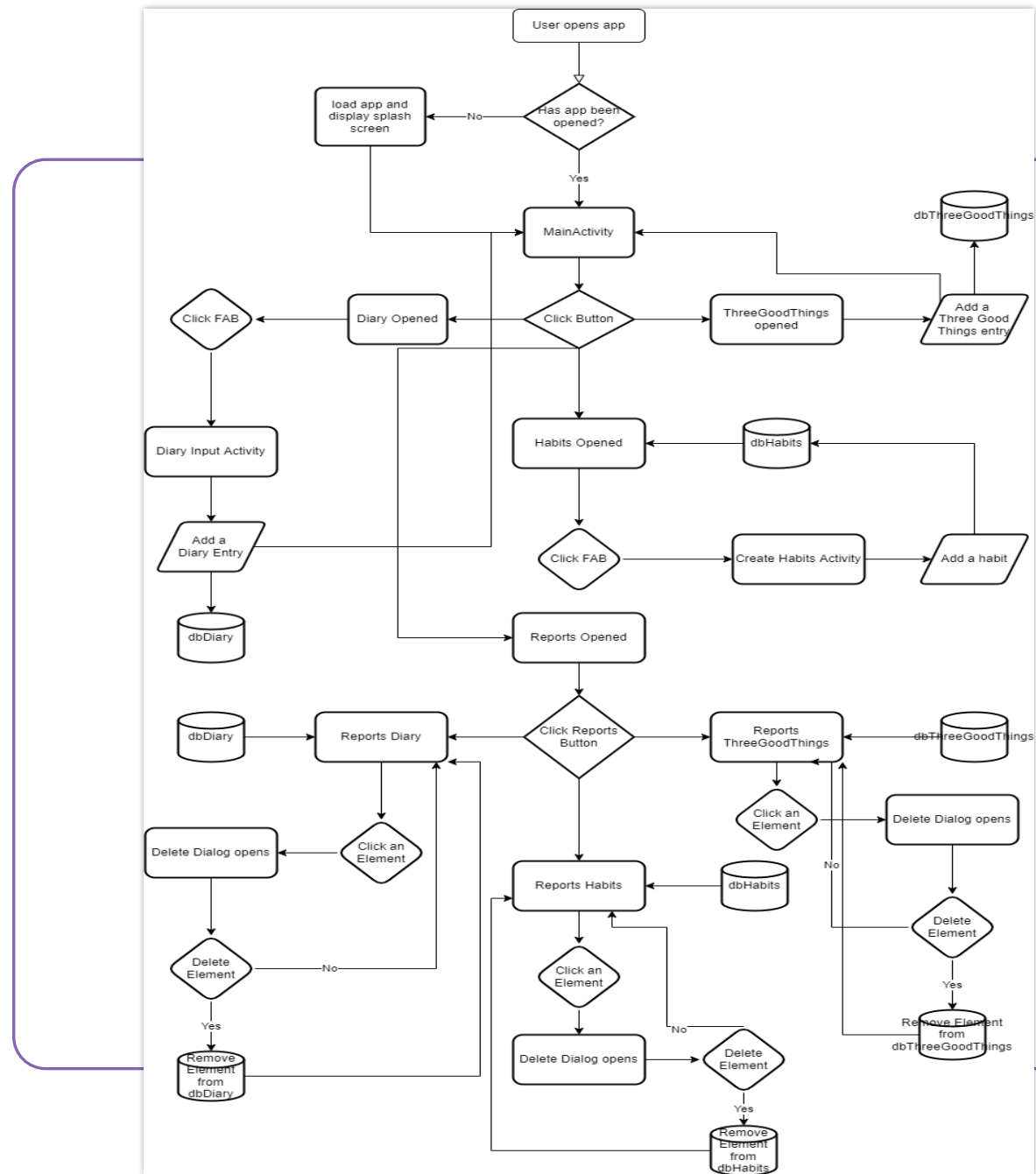


Package Structure

com.ibrice.moodr



Flow Chart



Main Features

- SQLite Database
- Notifications
- Splash Screen
- Alert Dialogs
- Theme Switching in-app (Shared Preferences)
- Look back on previous submissions using Tabbed Activities
- Set time for Habits using Google's time setter



SQLite Database

- Serverless
- One database with three tables
 - Diary
 - Habits
 - TGT
- Multithreaded and displayed using ViewModels and RecyclerViews

```
private void refreshListContents() {  
    new Thread(() -> {  
        DBManager dbManager = new DBManager(getApplicationContext().getApplicationContext());  
        dbManager.open();  
        Cursor cursorHabits = dbManager.fetchHabits();  
        List<HabitsItem> habitsItems = new ArrayList<>();  
        if (cursorHabits.moveToFirst()) {  
            do {  
                HabitsItem item = parseItem(cursorHabits);  
                habitsItems.add(item);  
            } while (cursorHabits.moveToNext());  
        }  
        cursorHabits.close();  
        habitsList.set(habitsItems);  
        notifyDataSetChanged();  
    }).start();  
}
```



```
// set notification content
NotificationCompat.Builder notificationBuilder =
    new NotificationCompat.Builder(notificationContext, NOTIFICATION_CHANNEL_ID);
notificationBuilder.setSmallIcon(R.drawable.ic_logo);
notificationBuilder.setContentIntent(pendingIntent)
    .setContentTitle("Come improve your mood!")
    .setContentText("Add entries to your diary, three good things, or habits")
    .setAutoCancel(false)
    .setSound(Settings.System.DEFAULT_NOTIFICATION_URI)
    .setContentIntent(resultPendingIntent);

// set importance level
int importance = NotificationManager.IMPORTANCE_DEFAULT;

// set notification channel behaviour
NotificationChannel notificationChannel =
    new NotificationChannel(NOTIFICATION_CHANNEL_ID,
        name "NOTIFICATION_CHANNEL_NAME",
        importance);
notificationChannel.enableLights(true);
notificationChannel.setLightColor(androidx.appcompat.R.attr.colorPrimary);
notificationChannel.enableVibration(true);
notificationChannel.setVibrationPattern(new long[]{100, 200, 100});

// create notification manager
assert notificationManager != null;
notificationBuilder.setChannelId(NOTIFICATION_CHANNEL_ID);
notificationManager.createNotificationChannel(notificationChannel);

// send notification
notificationManager.notify(0, notificationBuilder.build());
```

```
public class NotificationReceiver extends BroadcastReceiver {
    @SuppressWarnings("UnspecifiedImmutableFlag")
    @Override
    public void onReceive(Context context, Intent intent) {
        if (SettingsActivity.notificationsOn) {
            NotificationHelper notificationHelper = new NotificationHelper(context);

            Intent notificationIntent = new Intent(context, MainActivity.class);
            notificationIntent.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);

            PendingIntent.getActivity(context, requestCode 0, notificationIntent,
                PendingIntent.FLAG_UPDATE_CURRENT);

            notificationHelper.createNotification();
        }
    }
}
```

```
// set time
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.HOUR_OF_DAY, 17); // set to be at 5 PM
calendar.set(Calendar.MINUTE, 0);
calendar.set(Calendar.SECOND, 0);

if (calendar.getTime().compareTo(new Date()) < 0) {
    calendar.add(Calendar.DAY_OF_MONTH, amount 1);
}

// get notification receiver, pending intent
Intent intent = new Intent(getApplicationContext(), NotificationReceiver.class);
@SuppressWarnings("UnspecifiedImmutableFlag")
PendingIntent pendingIntent = PendingIntent.getBroadcast(context MainActivity.this,
    requestCode 100, intent, PendingIntent.FLAG_UPDATE_CURRENT);

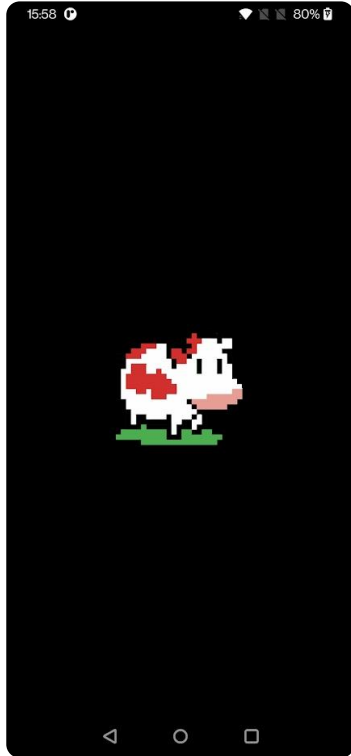
// get alarm manager for time
AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
if (alarmManager != null){
    alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, calendar.getTimeInMillis(),
        AlarmManager.INTERVAL_DAY, pendingIntent);
}
```

```
testNotification.setOnClickListener(v -> {
    if (notificationsOn) {
        // call notification function
        NotificationHelper notificationHelper = new NotificationHelper(context this);
        notificationHelper.createNotification();
    } // else do nothing
});
```

Notifications

- Settings Notification and Daily Notification
- NotificationHelper and NotificationReceiver classes
 - Linked to SettingsActivity and MainActivity





```
<activity
    android:name=".splash.SplashScreen"
    android:exported="true"
    android:theme="@style/SplashScreen">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:drawable="?attr/colorOnPrimary">
    </item>

    <item
        android:drawable="@drawable/ic_logo" android:gravity="center">
    </item>
</layer-list>
```

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    new Handler().postDelayed(() -> {
        Intent splashIntent = new Intent(packageContext, SplashScreen.this, MainActivity.class);
        startActivity(splashIntent);
        finish();
    }, delayMillis: 1000);
}
```

Splash Screen



```
// take id (for if just one entry requires deletion), and bool to check if all should be deleted or not
public void deleteDiary(long id, boolean deleteAll) {
    if (deleteAll) {
        database.delete(DBHelper.DIARY_TABLE_NAME, whereClause: null, whereArgs: null);
        dbHelper.close();
    } else {
        database.delete(DBHelper.DIARY_TABLE_NAME, whereClause: DBHelper._ID + " = " + id, whereArgs: null);
    }
}
```

```
// delete items from the list on click via a dialog/alert
viewHolder.itemView.setOnClickListener(v -> {
    DBManager dbDeleteManager = new DBManager(getContext());
    dbDeleteManager.open();
    DiariesItem currentItem = localDataset.get(position);

    AlertDialog.Builder alert = new AlertDialog.Builder(v.getContext());
    alert.setTitle("Delete Diary Entry?");
    alert.setMessage("Are you sure you want to delete this diary entry?");

    alert.setPositiveButton(text: "Yes", (dialog, which) -> {
        dbDeleteManager.deleteDiary(currentItem.ID, deleteAll: false);

        dbDeleteManager.close();

        Toast.makeText(v.getContext(), text: "Diary Entry Deleted", Toast.LENGTH_SHORT).show();

        Intent homeIntent = new Intent(v.getContext(), ReportsActivity.class).setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
        startActivity(homeIntent);
    });

    alert.setNegativeButton(text: "No", (dialog, which) -> dialog.cancel());

    alert.show();
});
```

Alert Dialogs

- Allows user to delete content from the database
- One function for deleting a single element and clearing the database



```
// get theme switch preference toggle
SwitchPreferenceCompat themeSwitch = findPreference(key: "darkTheme");

if (themeSwitch != null) {

    // when switch is toggled
    themeSwitch.setOnPreferenceChangeListener((preference, newValue) -> {
        boolean isChecked = false; // set whether the switch is checked
        if (newValue instanceof Boolean)
            isChecked = (Boolean) newValue;

        // if checked, dark theme, else, light theme
        if (isChecked) {
            Objects.requireNonNull(getPreferenceManager().getSharedPreferences())
                .edit().putBoolean("Toggle Dark Theme", true).apply();
            AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_YES);
        } else {
            Objects.requireNonNull(getPreferenceManager().getSharedPreferences())
                .edit().putBoolean("Toggle Dark Theme", false).apply();
            AppCompatActivity.setDefaultNightMode(AppCompatActivity.MODE_NIGHT_NO);
        }
    });

    return true;
}
}
```

```
SwitchPreferenceCompat notificationSwitch = findPreference(key: "notificationsToggle");

if (notificationSwitch != null) {
    notificationSwitch.setOnPreferenceChangeListener((preference, newValue) -> {
        boolean isChecked = true;
        if (newValue instanceof Boolean)
            isChecked = (Boolean) newValue;

        notificationsOn = isChecked;

        return true;
    });
}
```

Shared Preferences

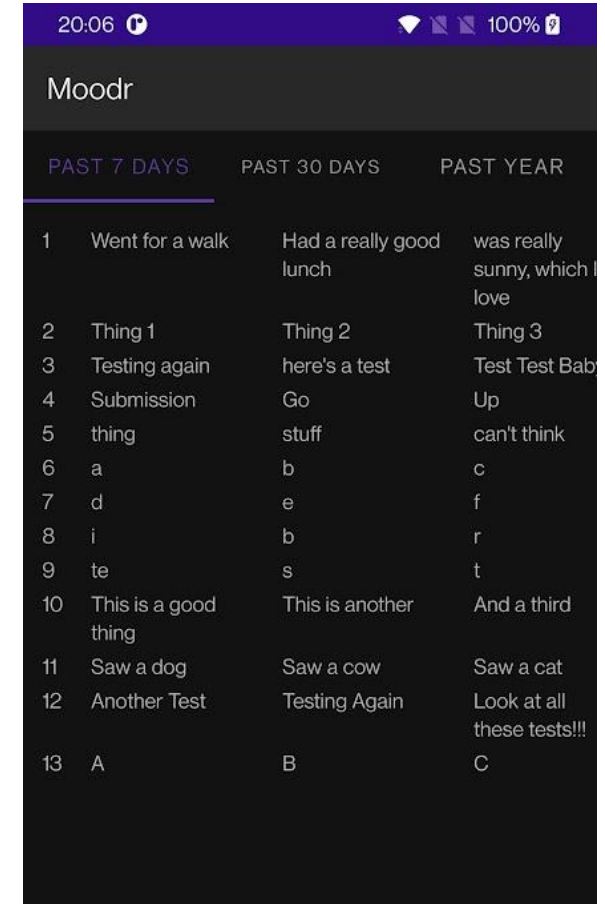
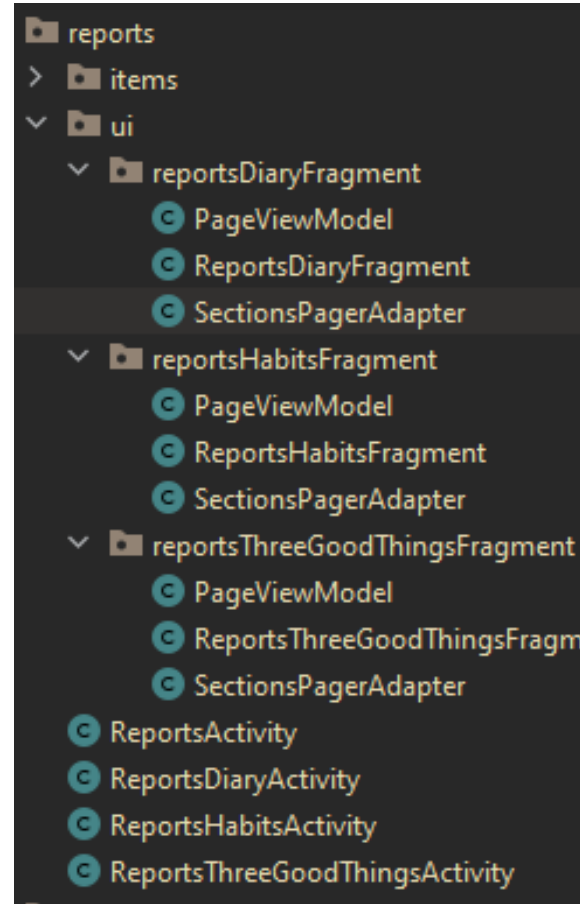
Changes both theme and enables/disables notifications

Also counts the number of ThreeGoodThings submissions



Tabbed Activities

- 3 Activities
 - AndroidViewModel
 - Fragment
 - SectionsPagerAdapter



```

<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/listReportsDiaries"
    diariesList="@{viewModelReportDiary.diariesList}"
    android:layout_width="406dp"
    android:layout_height="683dp"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="48dp"
    android:dividerHeight="1dp"
    android:padding="10dp"
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    tools:listitem="@layout/activity_reports_diary"/>

```

```

<?xml version="1.0" encoding="utf-8"?>
<layout 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"
    tools:context=".reports.ui.reportsDiaryFragment.ReportsDiaryFragment">

    <!-- grab data from ViewModel (where fetch function is called in a thread) -->
    <data>
        <variable
            name="viewModelReportDiary"
            type="com.librice.moodr.reports.ui.reportsDiaryFragment.PageViewModel" />
    </data>

```

1	Went for a walk	Had a really good lunch	was really sunny, which I love
2	Thing 1	Thing 2	Thing 3
3	Testing again	here's a test	Test Test Bab
4	Submission	Go	Up
5	thing	stuff	can't think
6	a	b	c
7	d	e	f
8	i	b	r
9	te	s	t
10	This is a good thing	This is another	And a third
11	Saw a dog	Saw a cow	Saw a cat
12	Another Test	Testing Again	Look at all these tests!!!
13	A	B	C

RecyclerViews

- Dynamic list that “can be scrolled by maintaining a limited number of views”



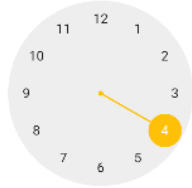
```
int hour = timePicker.getHour();  
int minute = timePicker.getMinute();  
  
@SuppressWarnings("DefaultLocale")  
String timePicked = String.format("%02d", hour) + ":" + String.format("%02d", minute);
```

Add a Habit!

Give your Habit a title!

Give your habit a short description

4:26^{AM}
PM



SUBMIT

Time Picker

- Used to pick time for Habits



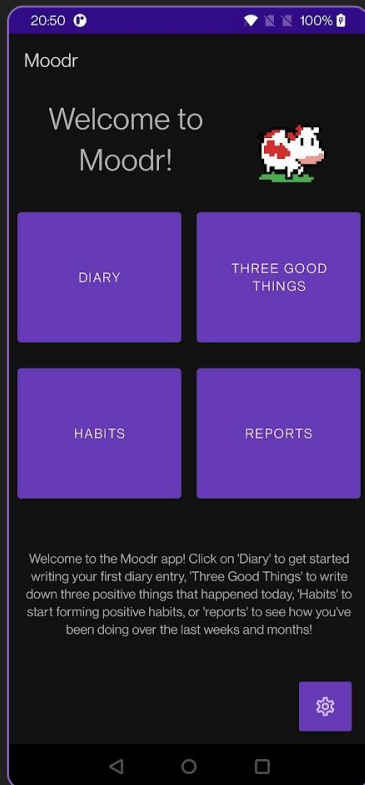
Critical Analysis



Overall & Compatibility



What Went Well?



- Light, simple, accessible, and attractive design
- Core functionality works as intended
- App is extremely fast without compromising usability
- App is compatible with older versions of android, down to SDK26 as well as Android 12



What Could Be Improved?

- Ability to add images to entries
- Notifications at specific times
- A timer for specific habits
- Graphical improvements and other tweaks
 - Element IDs resetting when one is deleted etc
 - Automatic capitalisation



Usability



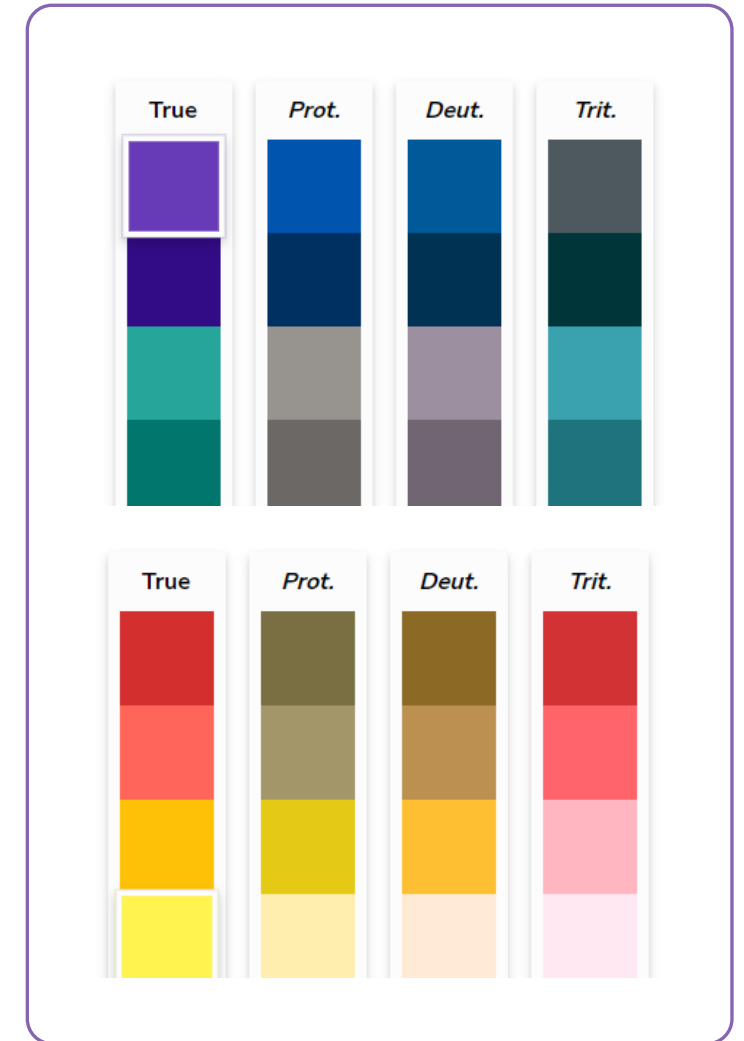
What Went Well?

- Light and dark mode
- Large buttons in many places
- Validation on the EditText fields to ensure nobody submits blank content
- Plain and easily navigable application generally



What Could Be Improved?

- More colour themes
 - Colourblindness
 - High-Contrast
- Decouple theme from system theme



Security



What Went Well?

- All user data is stored locally rather than in a server
- Coding practises and libraries up to date
- No personal data is taken

Issue: Object deserialization is used in androidx.versionedparcelable.VersionedParcel.readSerializable()

Severity: 10: Medium

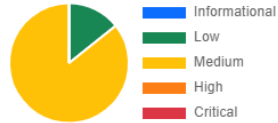
Confidence: 1: High

Summary: Object deserialization is used in androidx.versionedparcelable.VersionedParcel.readSerializable()

View
Details



Severity of Issues



Report Confidence



Issue: Potential Path Traversal (file read)

Severity: 12: Medium

Confidence: 2: Normal

Summary: java/nio/file/Paths.get(Ljava/lang/String;Ljava/lang/String;)Ljava/nio/file/Path; reads a file whose location might be specified by user input

View
Details



What Could Be Improved?

- REX scanner was used against the app
- Main issues:
 - Object deserialization
 - Potential Path Traversal



Performance



What Went Well?

- Multithreaded SQLite read
- RecyclerView used



What Could Be Improved?

- Would have used Room API instead of SQLite
- Reduce background notification/time checking service



Thank you for listening



References

- Abdi, J. (2019) 'Answer to "How to show a notification everyday at a certain time even when the app is closed?"', *Stack Overflow*. Available at: <https://stackoverflow.com/a/55910596/12207605> (Accessed: 3 May 2022).
- *Android - Time Picker* (no date). Available at: https://www.tutorialspoint.com/android/android_timepicker_control.htm (Accessed: 23 April 2022).
- 'Android SQLite Database Example Tutorial' (2015) *JournalDev*, 21 October. Available at: <https://www.journaldev.com/9438/android-sqlite-database-example-tutorial> (Accessed: 17 April 2022).
- *Android SQLite ListView with Examples - Tutlane* (no date). Available at: <https://www.tutlane.com/tutorial/android/android-sqlite-listview-with-examples> (Accessed: 9 May 2022).
- *AndroidViewModel* (no date) *Android Developers*. Available at: <https://developer.android.com/reference/androidx/lifecycle/AndroidViewModel> (Accessed: 17 May 2022).
- Cleave, P. (2021) *Examining the Importance of High Contrast Accessibility in Accommodating Low Vision Users*, *SmartSurvey*. Available at: <https://www.smartsurvey.co.uk/blog/examining-the-importance-of-high-contrast-accessibility-in-accommodating-low-vision-users> (Accessed: 17 May 2022).
- Dekanski, D. (2021) *BottomNavigationView switching between activities*. Available at: <https://github.com/ddekanski/BottomNavigationViewBetweenActivities> (Accessed: 12 April 2022).

References

- *Displaying dialogs with DialogFragment* (no date) Android Developers. Available at: <https://developer.android.com/guide/fragments/dialogs> (Accessed: 25 April 2022).
- Jha, A. (2016) 'Answer to "How to Store an int in SharedPreferences"', *Stack Overflow*. Available at: <https://stackoverflow.com/a/38099562/12207605> (Accessed: 11 May 2022).
- Kuzubasioglu, E. (2016) 'How to create a day counter for android application?', *Stack Overflow*. Available at: <https://stackoverflow.com/a/39179974/12207605> (Accessed: 17 April 2022).
- major (2017) 'Displaying ALL data from sqlite database into listview in tabbed activity', *Stack Overflow*. Available at: <https://stackoverflow.com/a/47954066/12207605> (Accessed: 17 April 2022).
- Nichols, D. (no date) *Coloring for Colorblindness*. Available at: <http://www.davidmathlogic.com/colorblind/> (Accessed: 17 May 2022).
- Ochieng, C. (2016) *Android Tabbed Data Ep.02 : ListView- Swipe Fragments SQLite CRUD*. Available at: <https://www.youtube.com/watch?v=O7noFYLxP14> (Accessed: 9 May 2022).
- Patel, V. (2016) 'Answer to "How to set confirmation delete message (AlertDialog) for my list view for android studio"', *Stack Overflow*. Available at: <https://stackoverflow.com/a/36106306/12207605> (Accessed: 27 April 2022).
- *Pixel cow image for game assets vector image on VectorStock* (no date) VectorStock. Available at: <https://www.vectorstock.com/royalty-free-vector/pixel-cow-image-for-game-assets-vector-38165012> (Accessed: 22 April 2022).

References

- REX (no date). Available at: <https://app.rexscan.com/dashboard/reports/a46f2ca8-cd83-4674-8243-c94aca9ce4e4> (Accessed: 17 May 2022).
- Rippstein-Leuenberger, K. et al. (2017) 'A qualitative analysis of the Three Good Things intervention in healthcare workers', *BMJ Open*, 7(5), p. e015826. doi:[10.1136/bmjopen-2017-015826](https://doi.org/10.1136/bmjopen-2017-015826).
- Sanath14 (2021) *Sanath14/to-do-list-using-sqlite*. Available at: <https://github.com/Sanath14/to-do-list-using-sqlite> (Accessed: 13 April 2022).
- *Save data in a local database using Room* | Android Developers (no date). Available at: <https://developer.android.com/training/data-storage/room> (Accessed: 17 May 2022).
- *SharedPreferences* | Android Developers (no date). Available at: <https://developer.android.com/reference/android/content/SharedPreferences> (Accessed: 17 April 2022).
- *SVG to Vector Drawable Converter – Convert SVG, PNG, JPEG, GIF images to Android VectorDrawable XML resource files online* (no date). Available at: <https://svg2vector.com/> (Accessed: 22 April 2022).
- *SwitchPreferenceCompat* | Android Developers (no date). Available at: <https://developer.android.com/reference/androidx/preference/SwitchPreferenceCompat> (Accessed: 17 April 2022).

References

- Taioli, F. (2016) 'Answer to "How to create a day counter for android application?"', *Stack Overflow*. Available at: <https://stackoverflow.com/a/39180033/12207605> (Accessed: 17 April 2022).
- Thompson, B. (2020) *Android RecyclerView: What is, Learn with Simple Examples*. Available at: <https://www.guru99.com/recyclerview-android-list.html> (Accessed: 17 May 2022).
- *ViewModel | Android Developers* (no date). Available at: <https://developer.android.com/reference/android/arch/lifecycle/ViewModel> (Accessed: 14 May 2022).
- Vujovic, F. (2016) *Daily Repeating Local Notification In Android*. Available at: <https://www.youtube.com/watch?v=1fV9NmvxXJo> (Accessed: 12 May 2022).
- Yousaf (2017) 'Answer to "Displaying ALL data from sqlite database into listview in tabbed activity"', *Stack Overflow*. Available at: <https://stackoverflow.com/a/47954360/12207605> (Accessed: 17 April 2022).