GitHub Username: danijoo

Notification Utilities

Description

Notification Utilities allows to log and create notifications.
The logging can be used to keep a history of notifications and backup the messages from any app on the phone. It also allows to find detailed information about every notification and the corresponding application.
The ability to create and dispatch custom notifications is a valuable tool for debugging applications using android's accessibility features.

Intended User

The main target group of this applications are other developers. The logging of notifications will assist them in debugging their applications that utilize the android notification api. Also, the ability to dispatch self-crafted notifications is crucial for testing applications that use the NotificationListenerService or android's accessibility features.

Features

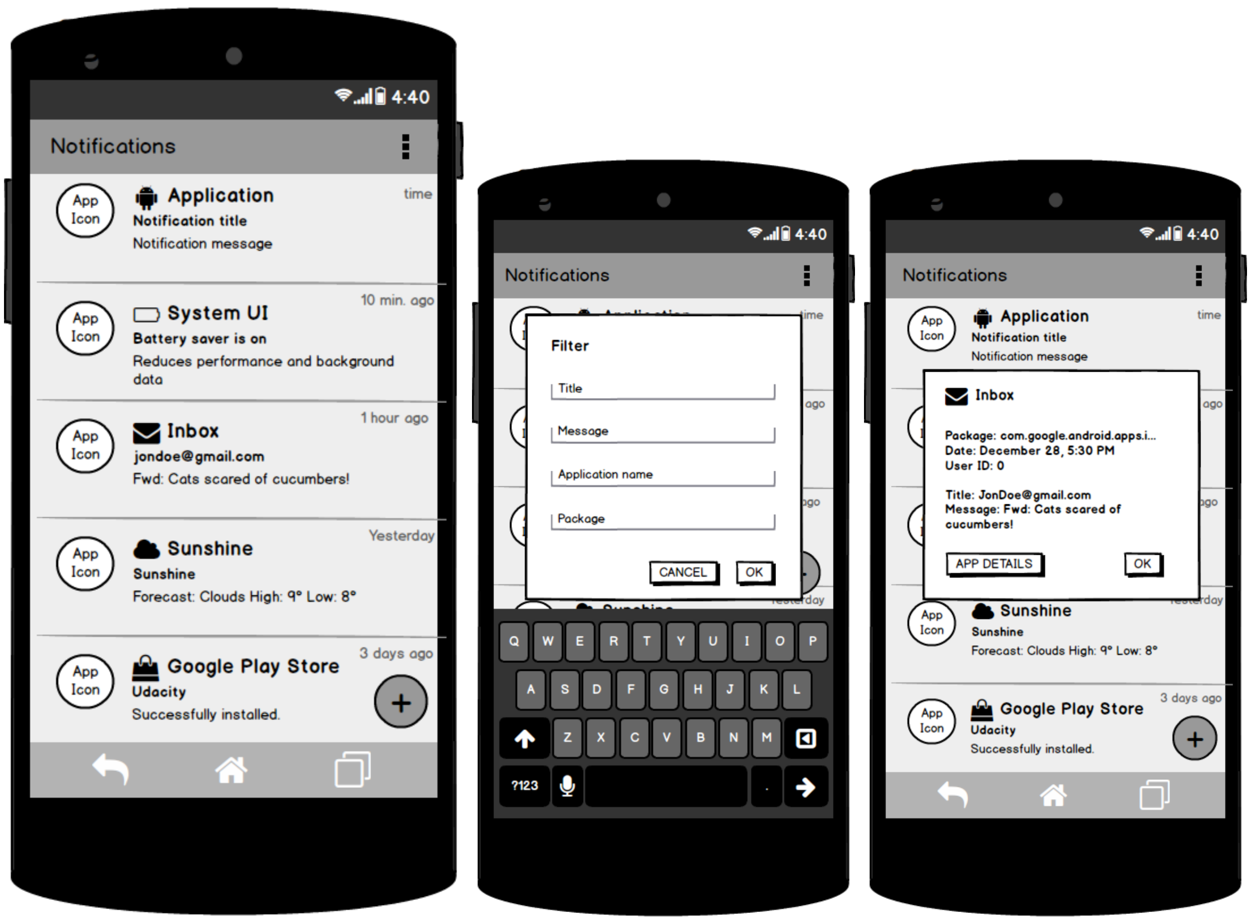
The app has 2 main features:

1) Notification logging
- Notifications of other applications can be logged and reviewed
- Options allow the blacklisting, filtering and searching of logged notifications
- Detailed information about logged notifications can be viewed (application name, package name, exact time, title and message …)
- Logged notifications can be exported to SD card or shared with other applications (For example to send them via mail)
- A widget showing the latest notifications

2) Notification creation
- Custom notifications can be created and dispatched
- Custom notifications can be scheduled

User Interface Mocks

Main Screen



Application time
Notification title
Notification message

System UI 10 min. ago
Battery saver is on
Reduces performance and background data

Inbox 1 hour ago
jondoe@gmail.com
Fwd: Cats scared of cucumbers!

Sunshine Yesterday
Sunshine
Forecast: Clouds High: 9° Low: 8°

Google Play Store 3 days ago
Udacity
Successfully installed.

Filter
Title
Message
Application name
Package

Inbox
Package: com.google.android.apps.i...
Date: December 28, 5:30 PM
User ID: 0
Title: JonDoe@gmail.com
Message: Fwd: Cats scared of cucumbers!

The main screen shows a list of all logged notifications. Each list item consists of the application icon, the small notification icon, the notification title and the message.

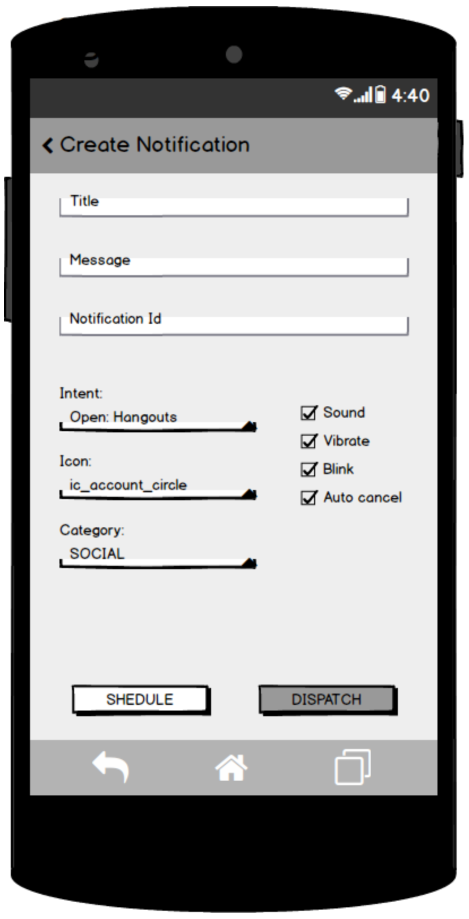
The options menu in the collapsible Toolbar gives access to the settings screen (see below) and allows opening the “Filter” dialog (Main screen screenshot #2).

The filter dialog allows to filter the list of notifications by package name, application name or content of the notification.

Each list element can be clicked and long clicked. On normal click, a details dialog is shown (Main screen screenshot #3). This shows some additional information like package name and user id. The “App Details” button of the details dialog leads to android’s app info activity. On long click, a list dialog with the following options is opened: [Filter package, Blacklist package, Remove notification, Share].

The FAB leads to screen 2, where custom notifications can be dispatched.

Screen 2: Create a notification

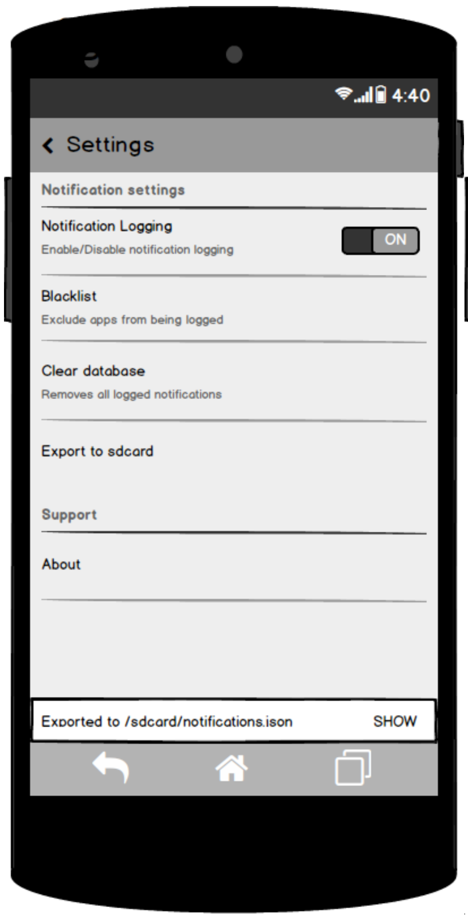


On this screen, custom notification can be created and dispatched. The screen offers various options to customize the dispatched notification and consists of multiple EditTexts, Spinners and Checkboxes.

The “Dispatch” button immediately dispatches the notification and make it shown.

The “Shedule” button opens a DateTimePicker dialog, which allows to set an exact time in the future when the notification will be shown.

Settings



Notification settings

Notification Logging
Enable/Disable notification logging
ON

Blacklist
Exclude apps from being logged

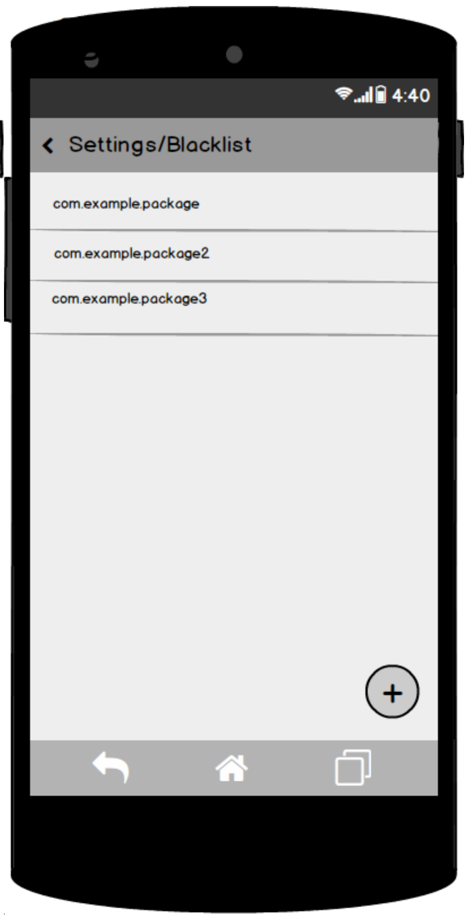
Clear database
Removes all logged notifications

Export to sdcard

Support

About

Exported to /sdcard/notifications.json SHOW



com.example.package

com.example.package2

com.example.package3

Available Options:

- **Notifications Logging**: Allows to enable/disable the logging of notifications
- **Blacklist**: Blacklisting applications will prevent them from being logged. Selecting this option will open the blacklist screen where a list of ignored package names is shown. A package name can be long clicked and removed. The FAB offers the option to add packages to the blacklist.
- **Clear Database**: Removes all logged notifications. Main screen will show an empty list after this is chosen.
- **Export to sdcard**: Opens a DatePicker Dialog to select a date range. All notifications in the date range will be exported to sdcard and saved under /sdcard/notifications.json. A snackbar will indicate the success of the export. The snackbars action button fires an intent to show the file in a file browser.
- **About**: A simple about page created with the „About Libraries“ Library

Key Considerations

How will your app handle data persistence?

Logged notifications will be persisted in an SQLite database. The data can be accessed using a content provider. This content provider will be exported to allow other applications to access the data.

Exported notifications will be saved as json-array in */sdcard/notifications.json*

Scheduled notifications will be dispatched using an alarm manager and don't need to be persisted.

Describe any corner cases in the UX.

Corner Case: Return from screen #2 to main screen:
To return from screen 2 to the main screen, the Up button or back key can be used. If the user has already entered any data in the form AND did not dispatch/schedule the notification, he will be asked If he really wants to dismiss his entered data (yes/no dialog).

Describe any libraries you'll be using and share your reasoning for including them.

Android support libraries: Various android support libraries will be used to backport features of new api versions to older devices and to provide a consistent look&feel across all supported android versions.

SimpleSQLProvider: This library will be used to create the database models and content provider.

Gson: Gson is a library for (de)serializing json. It will be used to export logged notifications to SD card.

Google Play Services/AdMob: Admob will be used for creating banner ads.

Google Play Services/Analytics: Analytics will be used to track the interaction of the user with the ui.

Timber: Timber will be used to enhance the logging features of android in debug mode.

AboutLibraries: About libraries will be used to show an about page in the app. This will include the versioning information of the app as well as licenses of used libraries.

Picasso: Picasso will be used for image loading and resizing.

Next Steps: Required Tasks

Task 1: Project Setup

- Update gradle files and create signing config + keystore
- Add all required permissions
- Add required dependencies

Task 2: Basic UI flow

- Build the MainActivity UI
- Build the Settings UI and About Page
- Build the UI to create notifications and make it accessable via FAB

Task 3: Notification dispatching

- Implement the “Create notifications” feature
- Implement the option to schedule notifications for later

Task 3: Notification logging

- Create the database model and content provider
- Implement the NotificationListenerService to store logged notifications to db
- Implement options to reset database and disable logging

Task 4: Finish Main Activity UI

- Update the main activity to show logged notifications
- Implement the notification details dialog
- Implement the notification filter and search dialogs

Task 5: Blacklist/Exporting

- Implement blacklisting and exporting options

Task 6: Admob and Analytics

- Add banner ads to the main activity
- Add analytics events to log how the user interacts with the UI