
GitHub Username: Your GitHub username here

Qr4All

Description

Sometimes all of us have many things to sort it out. Where I is my favorite book? Whom I lend my game? When you are about to move to new flat you must keep in order plenty of boxes with valuables. Now everyone keep all thins things under control: print qrcode, stick it on a box and write any notes regards it.

Application includes two parts: android application and a web server. Android application is written solely in Java.

Android Studio 3.1.4 and Gradle 3.1.4 will be used. Actual versions of support libraries are:

- com.android.support:support-v4:27.1.1
- com.android.support:design:27.1.1
- com.android.support:appcompat-v7:27.1.1
- com.android.support:recyclerview-v7:27.1.1
- com.google.android.gms:play-services-analytics:10.2.4
- com.google.maps.android:android-maps-utils:0.5+
- android.arch.persistence.room:runtime:1.1.1

- com.jakewharton:butterknife:8.8.1
- com.google.firebase:firebase-ml-vision:17.0.0

Intended User

It an application for all users, who have some chaos =).

Features

- Saves information
- Scan QR code,
- Add location information to it
- Show description of selected item on main screen
- Sync data with a server

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

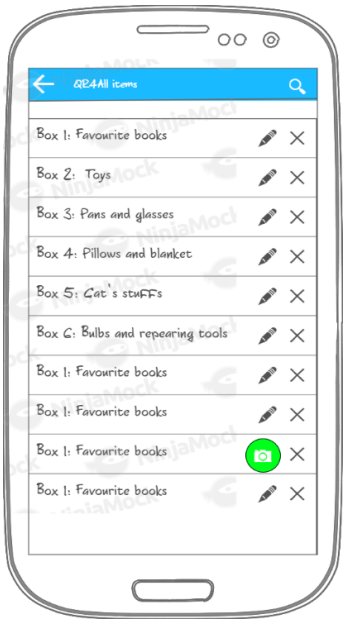
Authorization

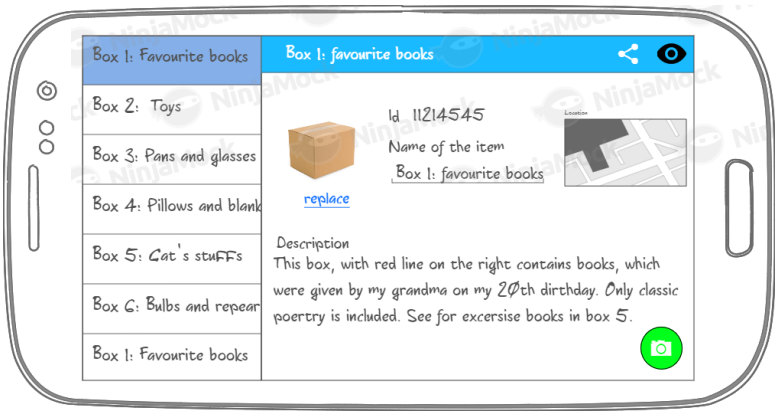
Loading list of items

Wrong credentials.

First screen of the app. You need sign in to be able to synchronize your data.

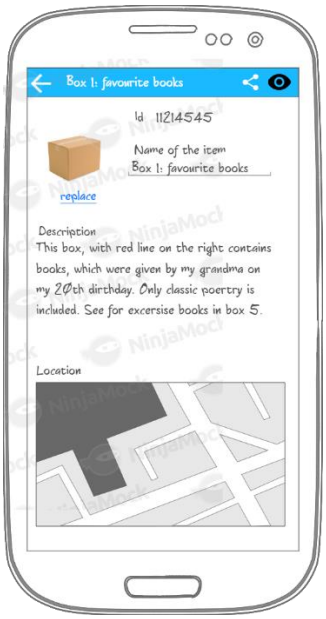
List of items





This is a list of items with FAB “scan code” button. Buttons for search, edit and remove are also available.
Landscape and portrait design list of items. Selected item is highlighted in menu.

Item details



Item details view includes fields to edit name and description of an item, you can upload photo, share a link on item with your friends, or show QR code.

Show QR code

This is QR code of an item. I think background of this page should be black for better readability, but I have not found how to change background in ninjamock ☹/

Widget

This is a widget with a button to scan QR code.

Key Considerations

How will your app handle data persistence?

All data will be stored on server, but a copy will be stored in local SQL database. Probable, I will use Room to access to this data.
REST interface will be provided:

- POST http://qr4all.ru/signin – for authorization,
- POST http://qr4all.ru/signup - for registration
- GET http://qr4all.ru/list - list of items for current user. A “keywords” parameter is available to get search available.
- GET http://qr4all.ru/edit - create new item,
- POST http://qr4all.ru/edit/1 - edit item id 1
- POST http://qr4all.ru/delete/1 - removes item 1
- GET http://qr4all.ru/code/1 - get code for item 1

TalkBack is supported, English and Russian languages both included in package, RTL is supported by margin attribute, but no translation available in first version.
No layout switching needed.
D-pad is not applicable.

Attributes description are stored in strings.xml, application uses standard dimensions and extends AppCompatLight theme.
Because it is neither serious business application, nor a game – primary color is soft: #03A9F4, and accent color is: #FFC107.
Application needs 4 icons: share, scan, remove, edit. All this icons whould be stored in drawable and loaded though ImageView.

Describe any edge or corner cases in the UX.

The UX consider to be simple. I gave back button on all screen. There are two places, where user leave application though explicit intent: scan QR code and upload an image.

- Mobile Vision API for scan QR codes
- *Picasso to show image of an item,*
- *ButterKnife to bind views*

Describe how you will implement Google Play Services or other external services.

- Google analytics to collect data how people use an application
- Google maps to pick point and save location.

Google analytics will collect have often people use application, create items. whether feature of saving location on map is popular or not.

Next Steps: Required Tasks

Application consist of 4 Activities:

- Main activity includes authorization fragment,
- List items activity. Includes recycler view with items,
- Details activity use constraint layout, with a form for editing items,

Project includes 3 AsyncTasks:

- **AuthAsyncTask**, which is triggered by sign in button. This task send request to authorize user in JSON, retrieves results, save accessToken or return to MainActivity with an error,
- **ItemsAsync** – return one item or list from a server,
- **EditSyncTask** – send current item to the server and retrieves error if necessary.

  Background server sync all items in bunch by 50 with the server.

Task 1: Project Setup

Create empty project. Configure gradle: install google and Picasso dependencies, create basic theme, colors, dimension.

Task 2: Implement Authorization

- Mark up and activity,
- Implement Saved Preferences(access token will be stored there),
- Create AsyncTask to authorization user,
- Add validation for incorrect email/password.
- Save token to shared preferences.

Task 3: Sync data with a server.

If there is no internet collection, data should be fetches from local DB.
There I should add background sync with a server.
Local database consist with two tables: **tokens** and **items**,

- Tokens includes 3 fields: id (int), user_id(int, user id from the server), token(string) and expire(date when it should be required again),
- Items includes 7 fields (id, name, description, qrcod_url, image_url, longitude, latitude, sync_date),

Data will be synced only in background, of when element has changed.
ItemDao interfaces has findAll(), findOne(), delete(), update(), insert() methods,
TokensDao interface includes only find() and deleteOld() methods because do not need more then one token.
If there is no items for current user – application suggests create new one.

Task 4: Details activity and show QrCode

- Create basic layout
- Edit and store elements to the server though AuthTask,
- Fetch Qcode from the server and show it on separate activity.

To scan QrCode guide is here: https://firebase.google.com/docs/ml-kit/android/read-barcodes

Task 5: Implement list of items

- Mark up,

- Create RecyclerView and an adapter
- Populate view with data from ItemDao,
- ViewModel should be implemented,
- Recycle view should be an observer on ItemLiveDao

Task 6: Widget

- Implement widget to show data on main screen.
- Widget should be an observer on ItemLiveData,

Task 5: Add analytics and maps library

Add location to Edit activity and analytics to the whole application.
Analytics will be implemented though the guide:
https://developers.google.com/analytics/devguides/collection/android/v4

Task 5: Accessibility and Internationalization

- Check it all text sizes are in sp, but be able to adapt to device font size,
- String for English and Russian packages are included,
- TalkBack is supported, so Buttons and inputs should contain **contentDescription** attribute,
-

Task 6: Implement share intent

- When user clicks on share button – link to the item will be shared in any messager though explicit event.