

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Implement VK interaction](#)

[Task 3: Implement DB](#)

[Task 4: Implement Utils](#)

[Task 5: My Documents](#)

[Task 6: Community Documents](#)

[Task 7: Dialogs Documents](#)

[Task 8: Global Documents](#)

[Task 9: Make Me Cute](#)

[Task 10: Build](#)

**GitHub Username:** [ashoni](#)

## VK Docs

### Description

VK Docs helps you to work with VK (vk.com) documents with your mobile device. It allows:

- View existing documents
- Share documents with friends
- Upload documents from your device or Google Drive
- Search and add documents from VK library

VK is a perfect platform to share your documents, photos and books. Try it!

### Intended User

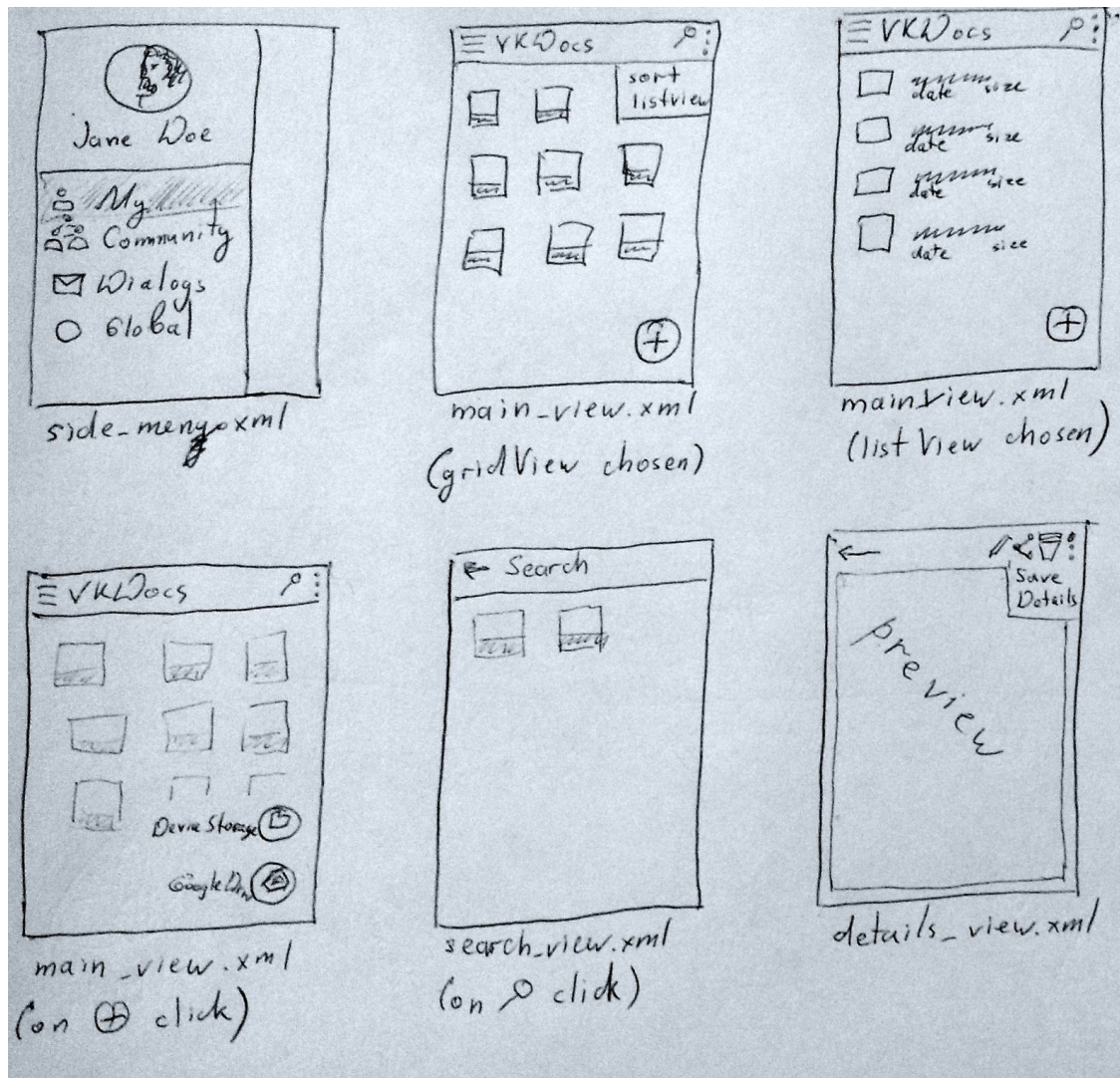
All VK users. Especially students, book lovers, language learners, text games players.

### Features

- View your own documents

- View your community documents
- Search across the whole VK library
- View all dialog attachments
- Rename documents
- Share documents
- Download documents
- Upload documents from your device or from the Google Drive

## User Interface Mocks



## Key Considerations

## How will your app handle data persistence?

DataBase.

MY\_DOCS:

- ID (long) - unique document id
- OWNER\_ID (long) - unique owner id
- TITLE (String) - document title
- SIZE (long) - file size in bytes
- DATE (long) - document creation date (Unix time format)
- URL (String) - document url
- PREVIEW\_URL (String) - only for pictures
- OFFLINE (boolean) - true if document should be accessible in offline mode
- TYPE (int) - type of the document: picture, document, archive, etc.

MY\_DOCS should be periodically updated by SyncAdapter.

If document is accessible from offline, it is stored in internal device memory. In this case, local url looks like BASE\_FOLDER\ID or BASE\_FOLDER\ID\_preview.

## Describe any corner cases in the UX.

Offline access: only My Documents activity should allow it. Offline mode features:

- all image preview pictures are replaced with default icons;
- all icons for documents which are not downloaded should be semi-transparent;
- all downloaded documents and there previews should be stored in internal memory in the special folder.

All documents should be uploaded by small portions if it is possible. For example, we can upload them by ~20 per time when user just looks at them (like Instagram does). However, to complete manual search (VK API provides search by all documents but it doesn't allow filters like 'only user documents') or sort it, we need to get full list.

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

Picasso to handle the loading and caching of images.

Butterknife to make easier views binding.

Retrofit for http calls.

com.vk.androidsdk for VK Api.

## Next Steps: Required Tasks

### Task 1: Project Setup

Initialize VK SDK (see <https://github.com/VKCOM/vk-android-sdk>).

### Task 2: Implement VK interaction

Implement .vk package:

- Create all required enums (scope, errors, etc.)
- Create class which sends requests to VK
- Move requests code to VK stored procedures if possible

### Task 3: Implement DB

Implement data persistence complex:

- MY\_DOCS table
- SyncAdapter for MY\_DOCS

### Task 4: Implement Utils

Implement tasks which are used from different activities:

- Sort documents by type, date, name
- Share document
- Search for document
- Document class

### Task 5: My Documents

Implement My Documents:

- Create new document on FAB pressed
- Implement listview-gridview switch
- Provide sort option
- Provide search option
- Add menu for each preview: delete, share, rename, save-to-offline, details
- Layout

### Task 6: Community Documents

Implement Community Documents:

- Show dialog with communities list on start and on FAB pressed
- Load and show documents previews list of chosen community
- Provide sort option

- Provide search option
- Add menu for each preview: add, delete, share, rename, details. Attention: user may have not enough rights to complete the action
- Layout

## Task 7: Dialogs Documents

Implement Dialogs Documents:

- Show dialog with dialogs list on start and on FAB pressed
- Load and show attached documents previews list
- Provide sort option
- Provide search option
- Add menu for each preview: add, share, details
- Layout

## Task 8: Global Documents

Implement Global Documents:

- Search Activity
- Load and show searched documents previews list
- Add menu for each preview: add, share, details
- Layout

## Task 9: Make Me Cute

Make it real and convenient:

- Add preview-view transition
- Add transition for add document options
- Add transition to make user notice side menu
- Check RTL and content description

## Task 10: Build

Gradle-related:

- Create installRelease task
- Create a separate flavour with advertisement
- Signing configuration