GitHub Username: soo7ru

Comicser

Description

A pocket helper for all comic lovers! The application contains a plenty of features which can help users to manage their personal comic collection, browse the latest published issues and track the release dates of all favorite volumes. Also Comicser has a built-in search feature which allows users to load and browse some additional comic info like volume descriptions, character profiles and so on. All comic related data provided by Comic Vine API. Application will be written solely in the Java Programming Language.

Intended User

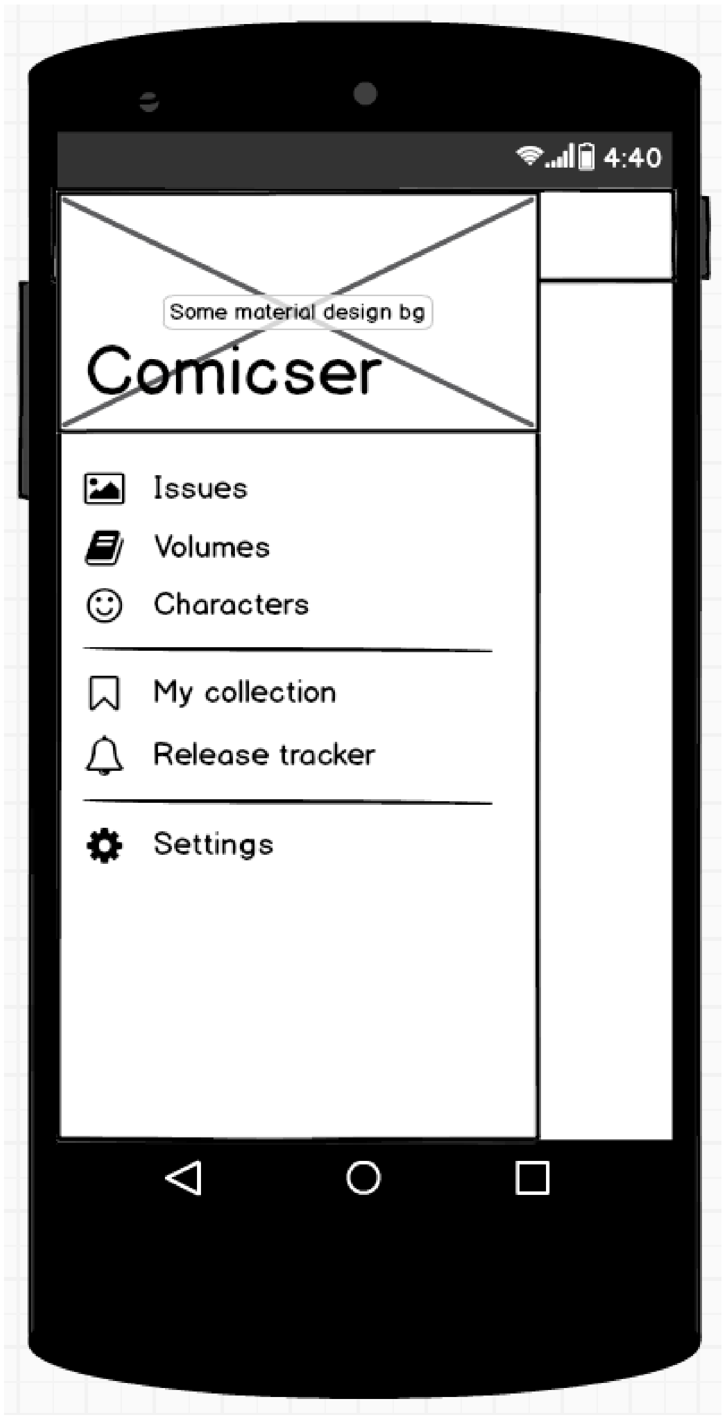
This application is aimed to comic fans who want to organize their own comic collection, track release dates for chosen volumes and get Online access to Comic Vine search engine.

Features

- Browse the newest released issues list
- Browse the upcoming issues list
- Bookmark chosen issues to create your collection of comics which you already have
- Get full volume info and check which issues still missed in your collection
- Search for characters and volumes info
- Get notification when a new issue from your favorite volume released

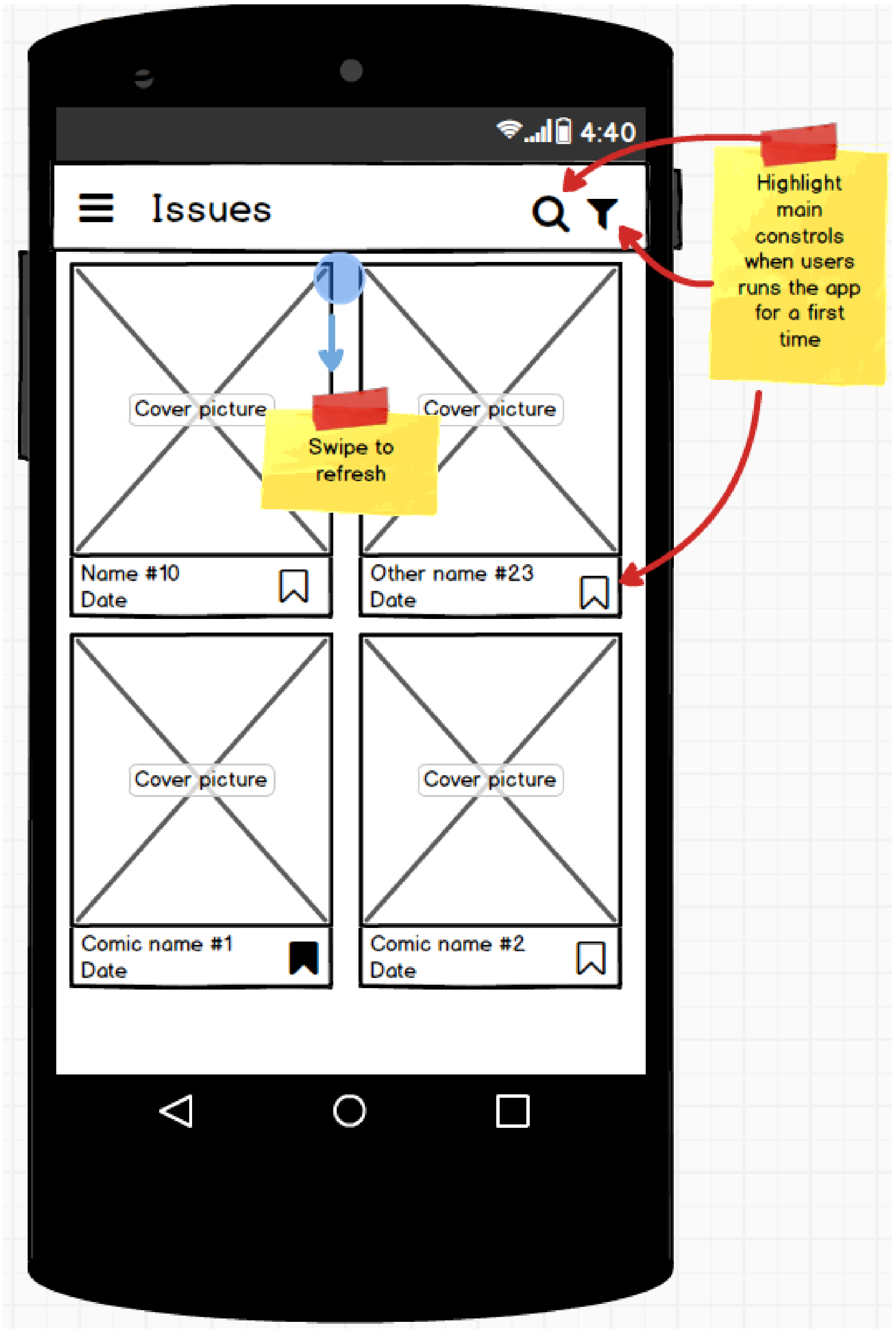
User Interface Mocks

Navigation bar drawer



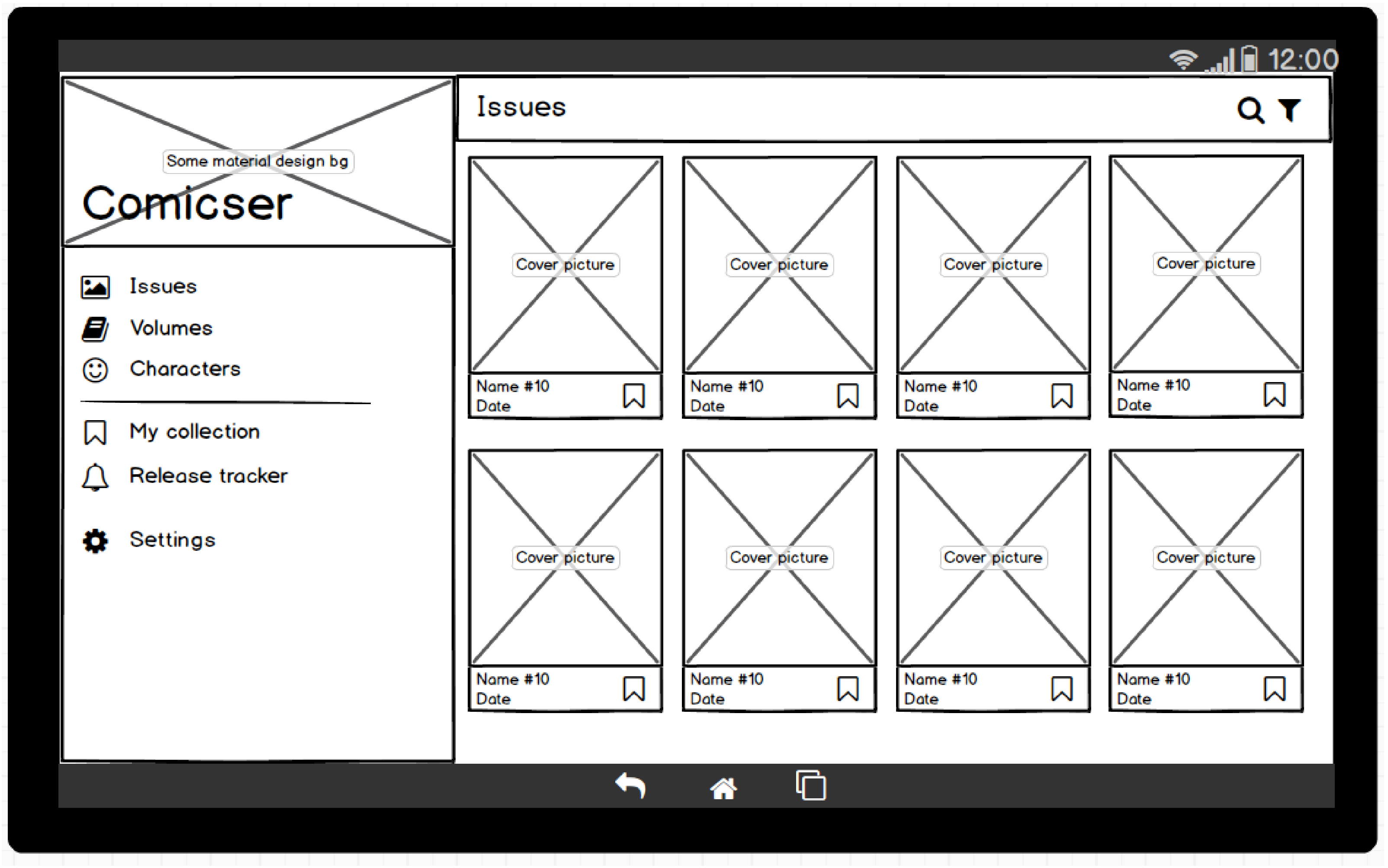
Displays main navigation bar with all available app features.

Issues



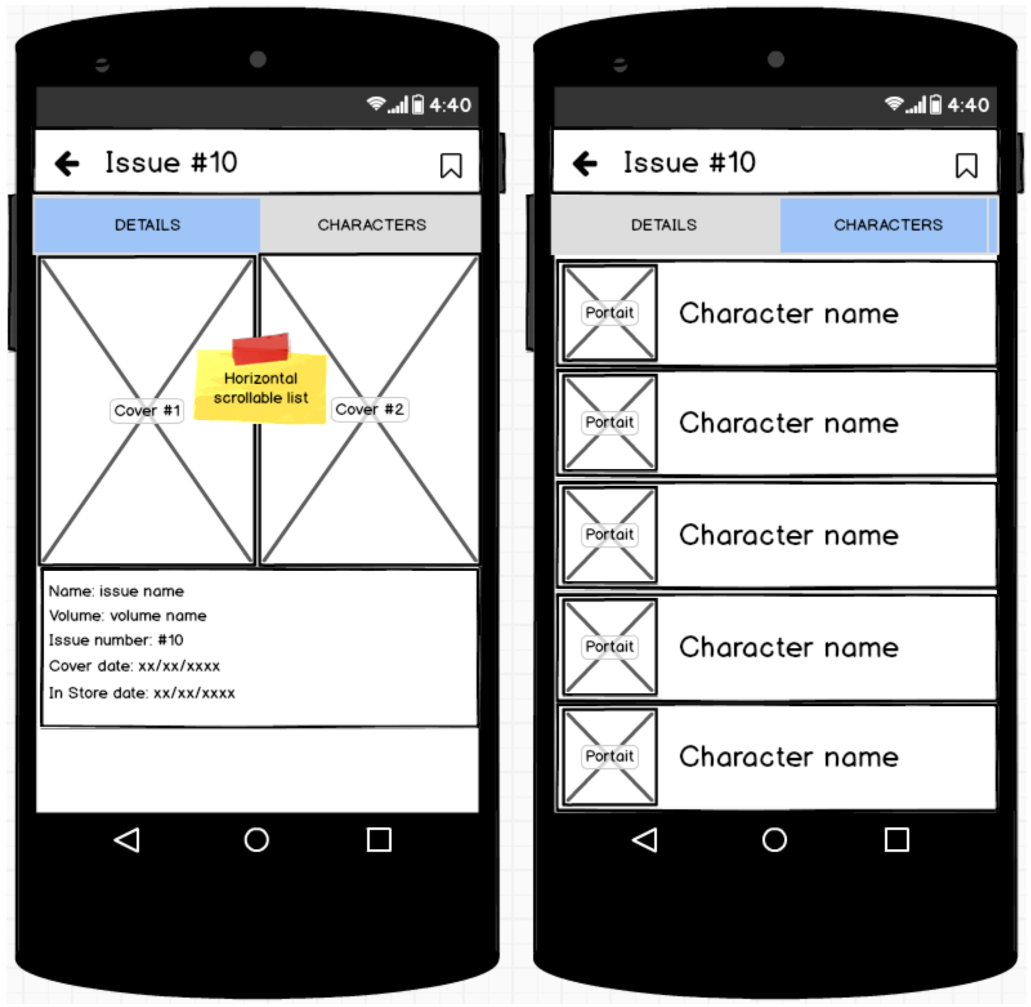
Main application page which shows today's issue releases. Contains filtering and search options, also each issue card has a bookmark option which allows user to mark selected issue as owned. When user launches the app for a first time, main UI controls highlighted with ShowcaseView.

Tablet variation



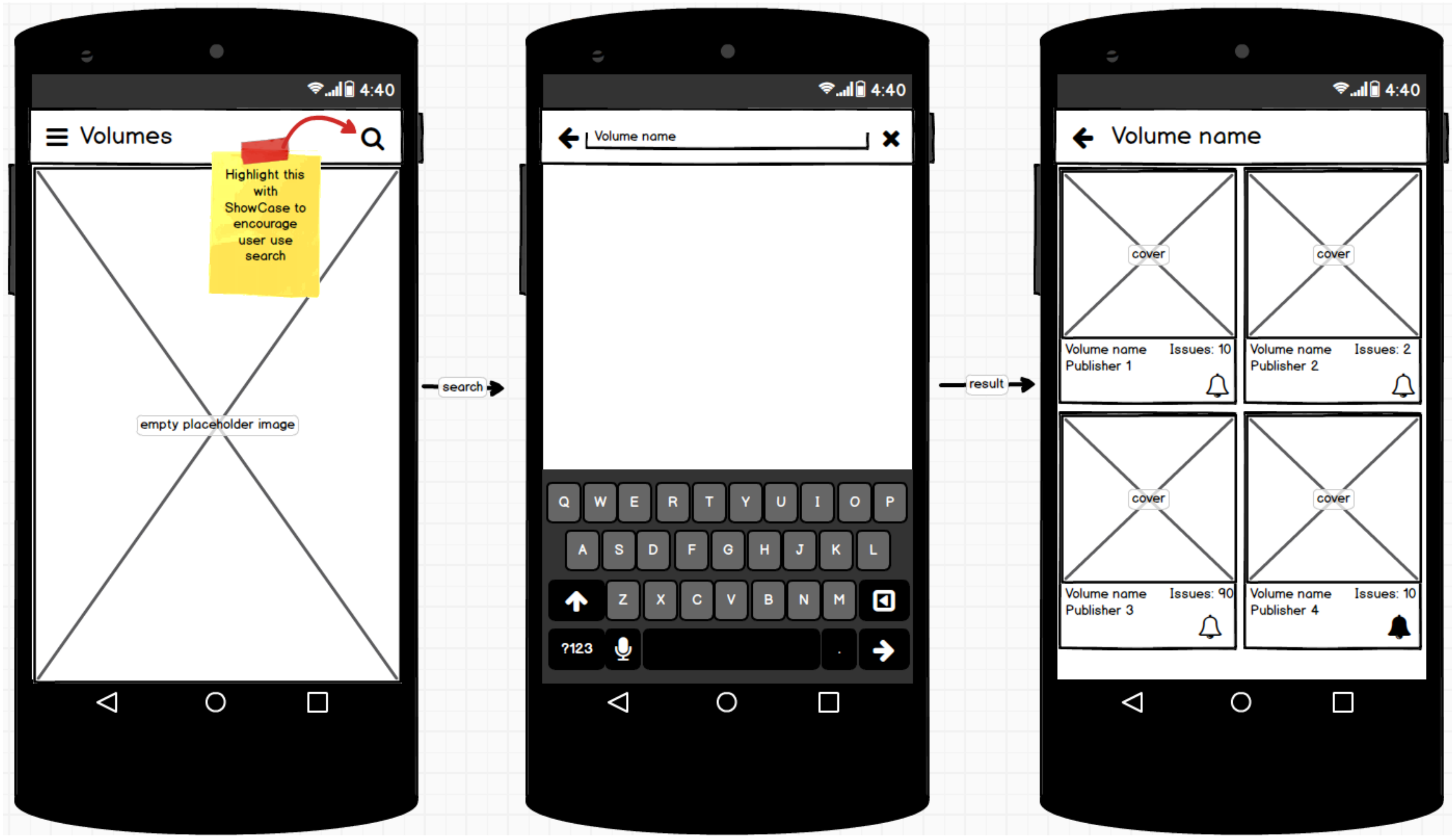
Tablet layout variations will have constantly visible left navigation bar.

Issue details



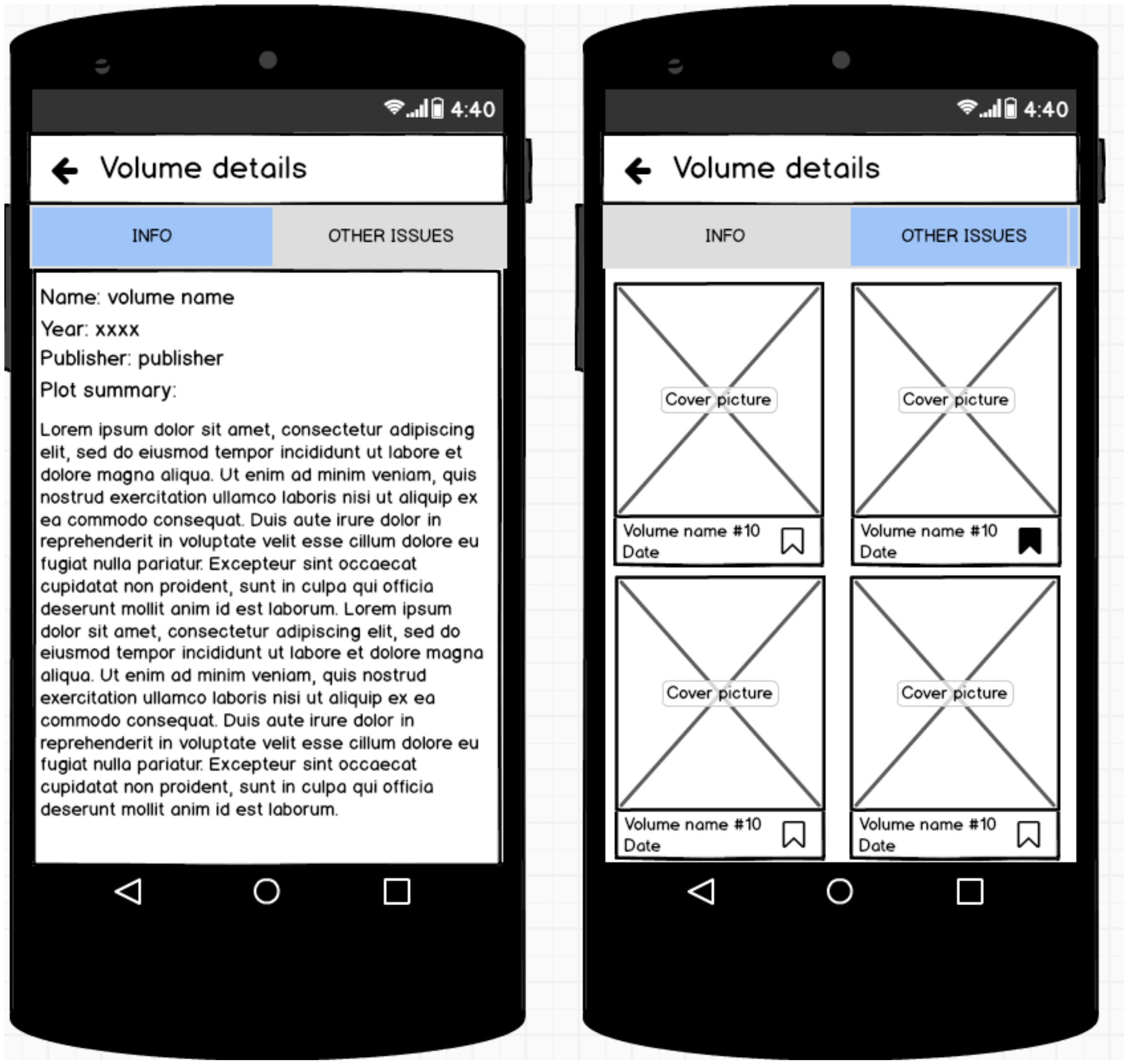
Issue details activity, allows to check the main issue info as well as browse issue characters list. Click on any character item launches a new activity which displays chosen character info.

Volumes



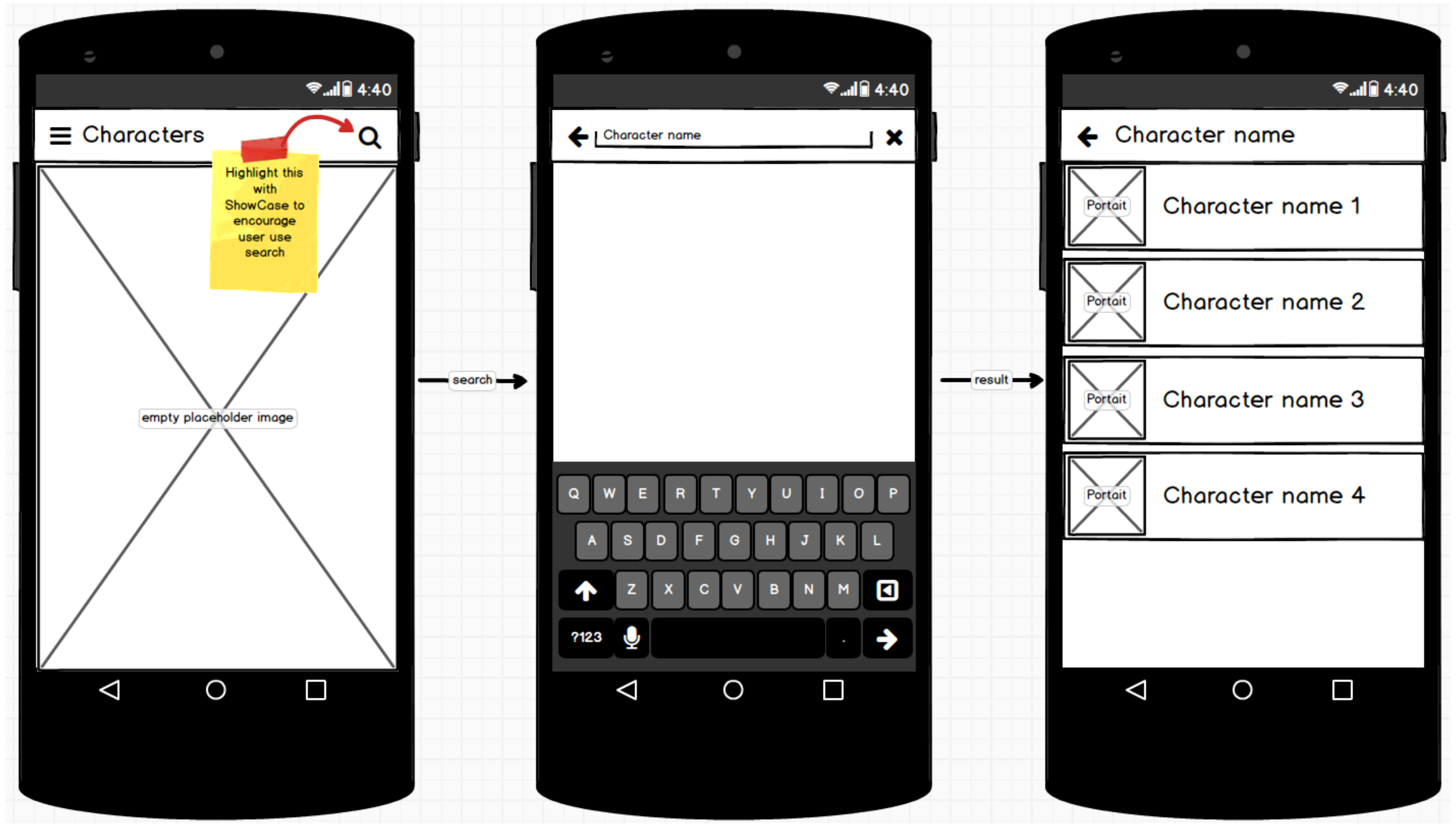
Volumes activity with a search option. Search results contain obtained volume list, each item in the list has an option which allows to toggle release tracking feature. When users sees this activity for a first time, ShowcaseView highlights search button and explains its purpose.

Volume details


4:40
Volume details
INFO
OTHER ISSUES
Name: volume name
Year: xxxx
Publisher: publisher
Plot summary:
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.
Cover picture
Volume name #10
Date

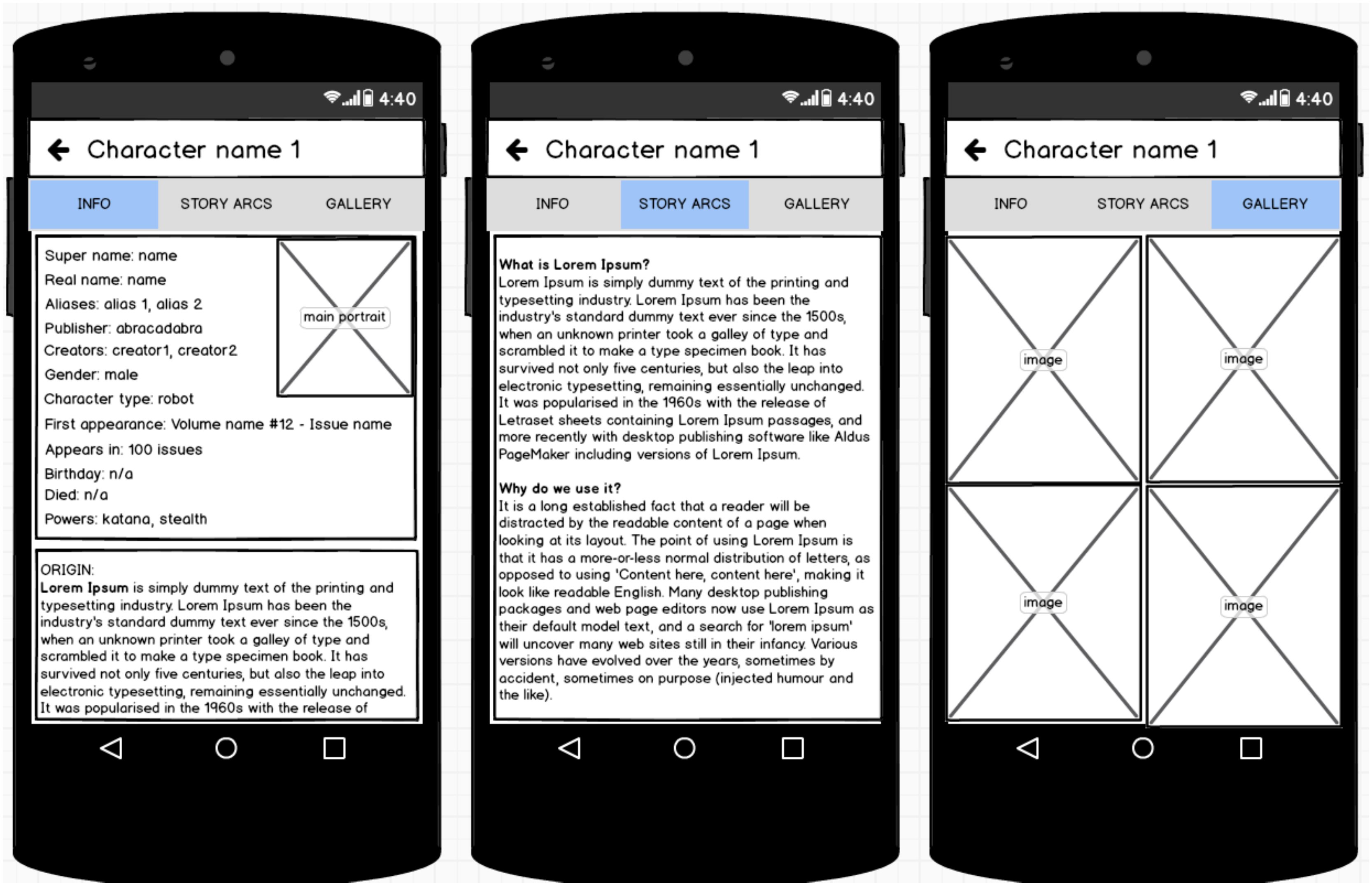
Volume details activity displays primary volume info and all issues related to this volume. Each issue card has bookmark option which allows user to mark selected issue as owned.

Characters



Characters activity with a search option. Search results contain obtained characters list, click on any character item launches a new activity which displays chosen character info. When users sees this activity for a first time, ShowcaseView highlights search button and explains its purpose.

Character details activity



Super name: name
Real name: name
Aliases: alias 1, alias 2
Publisher: abracadabra
Creators: creator1, creator2
Gender: male
Character type: robot
First appearance: Volume name #12 - Issue name
Appears in: 100 issues
Birthday: n/a
Died: n/a
Powers: katana, stealth

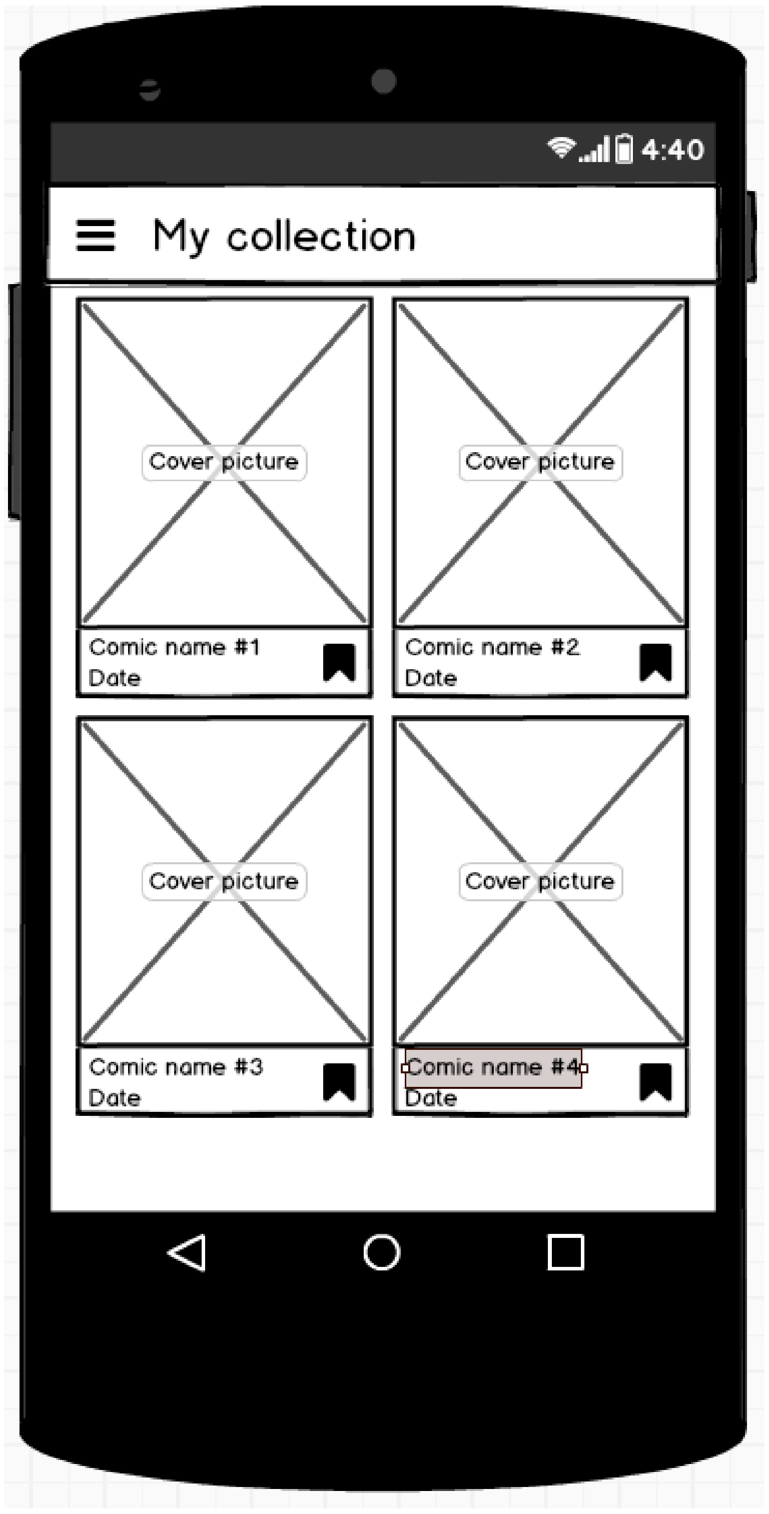
ORIGIN:
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of

What is Lorem Ipsum?
Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem Ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem Ipsum.

Why do we use it?
It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout. The point of using Lorem Ipsum is that it has a more-or-less normal distribution of letters, as opposed to using 'Content here, content here', making it look like readable English. Many desktop publishing packages and web page editors now use Lorem Ipsum as their default model text, and a search for 'lorem ipsum' will uncover many web sites still in their infancy. Various versions have evolved over the years, sometimes by accident, sometimes on purpose (injected humour and the like).

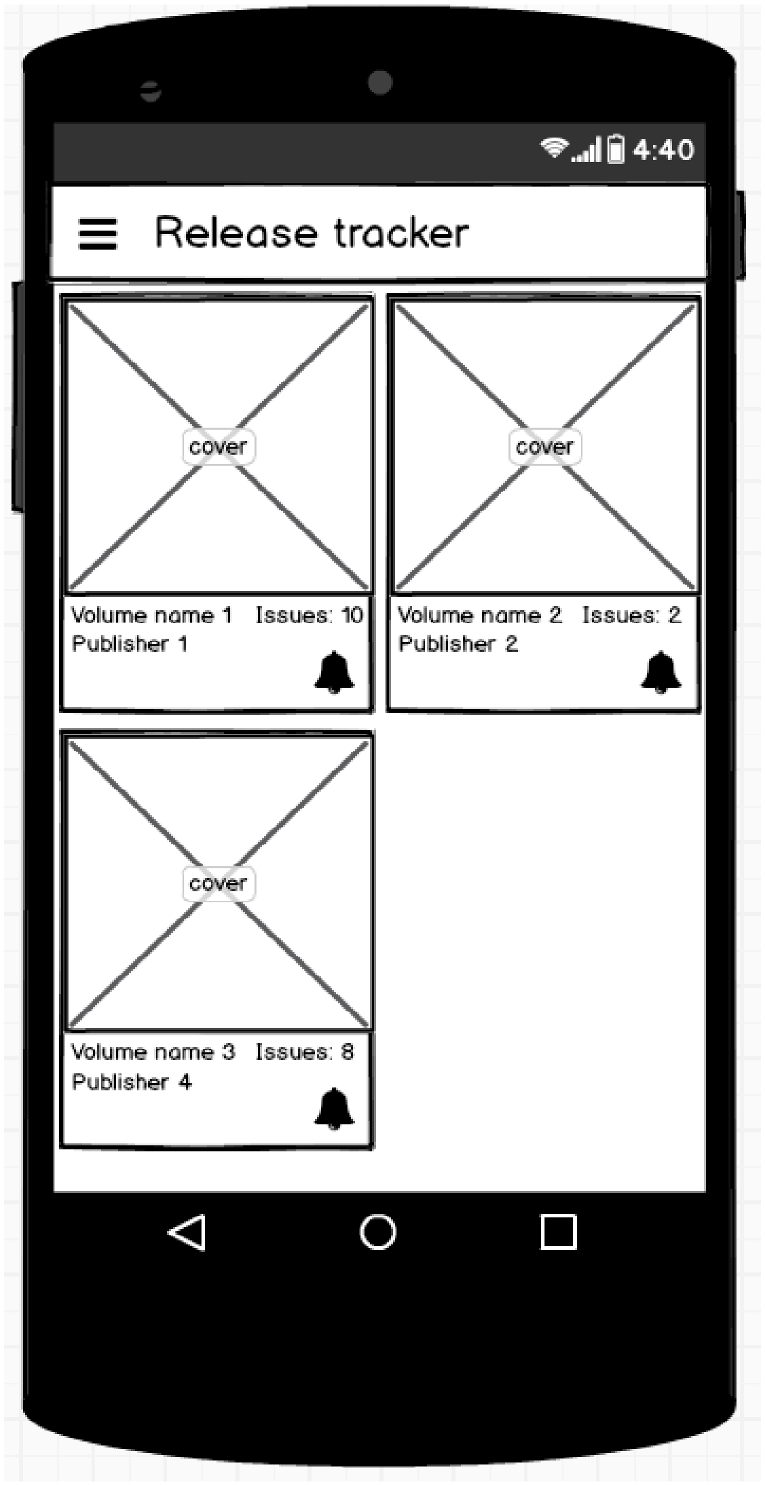
Character details activity displays primary character info, character story arcs and images gallery.

My collection



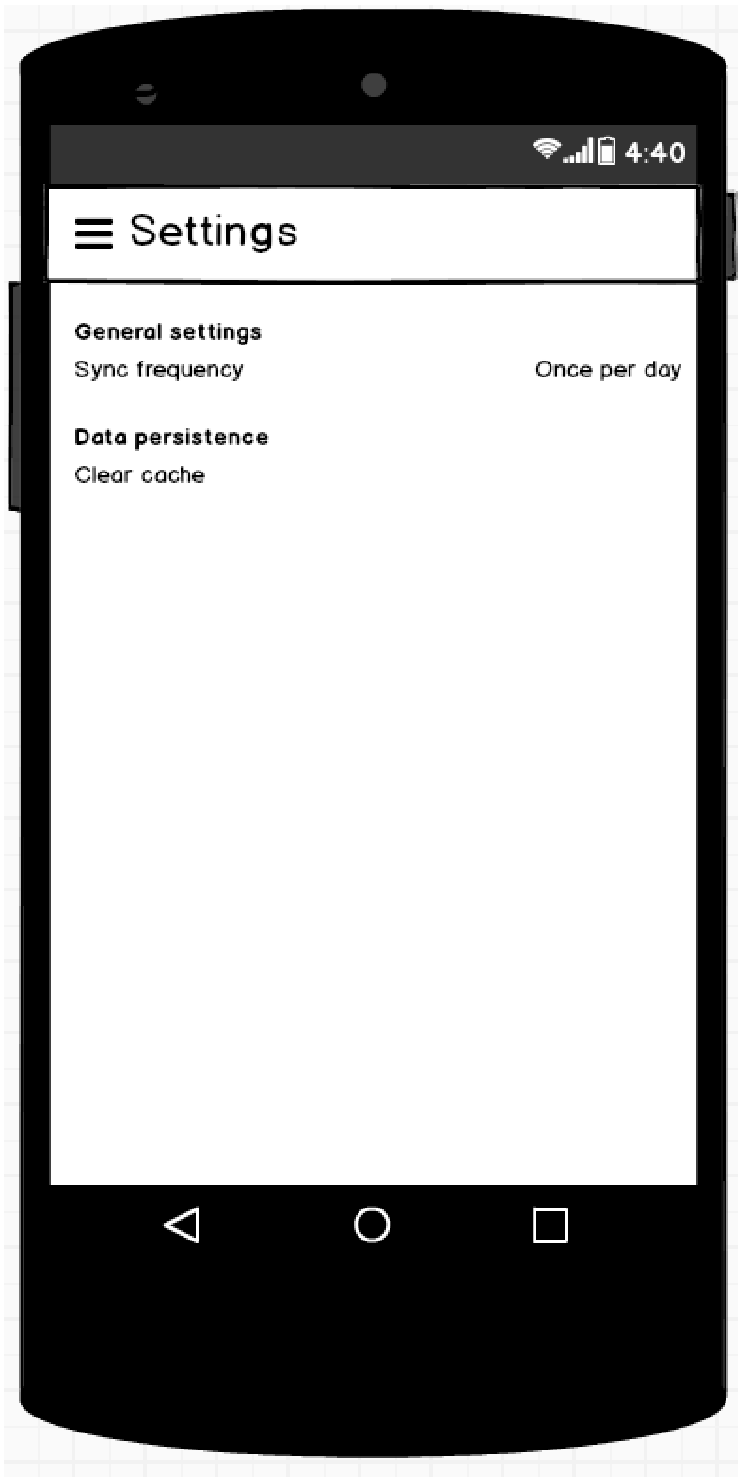
Collection list activity allows user to manage owned comic collection.
Click on any item opens chosen issue details activity.

Release tracker



Release tracker allows user to chose which volumes must be tracked. If volume tracking enabled, user gets a notification every time when any issue from chosen volume appears in the stores (In Store date compared with current date).

Settings



General settings

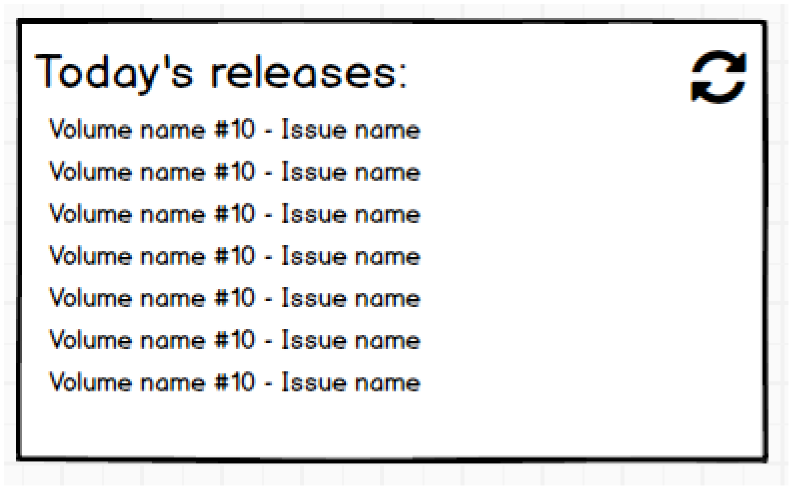
Sync frequency Once per day

Data persistence

Clear cache

Settings screen allows user to chose sync period as well as clean all images stored in the local cache.

Widget


Today's releases:
Volume name #10 - Issue name
Volume name #10 - Issue name
Volume name #10 - Issue name
Volume name #10 - Issue name
Volume name #10 - Issue name
Volume name #10 - Issue name
Volume name #10 - Issue name

App Widget displays today's released issues list. Click on any item list opens issue details activity.

Key Considerations

How will your app handle data persistence?

App will use a Content Provider and a Shared Preferences to maintain the local data.

Describe any corner cases in the UX.

- **Unstable or missed network connection:** the application must not crash in that cases
- **Device orientation change**: the application must handle all long-running operations correctly considering possible configuration changes
- **UI freezes:** the application must not use the main thread for any resource consuming operations

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

- **Android Studio version 3.5.1:**(IDE) for for Android development
- **Android Gradle version 5.6:**build system (open source) which is used to automate building, testing, deployment etc
- **Dagger version 2.24**: for dependency injections
- **Retrofit version 2.6.2**: for network API requests
- **Moshi version 1.8.0**: for JSON parsing
- **AutoValue version 1.7.0**: for data model classes generation
- **Butter Knife version 10.2.0**: for boilerplate code reducing
- **Glide version 4.10.0**: for images loading
- **ShowcaseView version 5.4.3**: to highlight some UI elements and guide user
- **Material Values version1.1.1**: for handy Material Design dimensions access
- **Stetho version 1.5.1**: for debugging purposes
- **Leak Canary version 1.5.1**: for memory leaks detection
- **Timber version 4.7.1**: for logging purposes
- **Firebase version 17.2.0**: for analytics and crash reports

Describe how you will implement Google Play Services.

The application will use Firebase Crash Reporting and Google Analytics for Firebase which both depend on Google Play Services.

Required Tasks

Task 1: Project Setup

Create and setup a new project. This task includes:

- creating a new project in Android Studio
- configuring libraries by adding all necessary dependencies

Task 2: Data model classes

Create data classes which help to handle all response data provided by Comic Vine API calls.
Required classes:

- Response
- Issue
- Volume
- Character
- StoryArc

Task 3: Data model tests

Write unit tests for data model classes if necessary.

Task 4: API

Implement a service which provides all necessary network API requests. Service based on Retrofit 2. Required API calls:

- load issues data (with different filter options)
- load volumes data
- load characters data
- load story arcs data for selected character

Task 5: API testing

Write unit tests for API service calls if necessary.

Task 6: Data persistence

Add a content provider and shared preferences helper class to handle all locally stored data.

Required steps:

- create a sharedpreferences helper
- create a database helper
- create a content provider
- implement database access functions using Loaders and created content provider, Loader finishing callback should update the related views

Task 7: Content provider testing

Write tests for content provider testing if necessary. Content provider testing includes tests for the standard provider interactions and for incorrect URIs.

Task 8: Create the UI

Implement all necessary fragments and activities.
Following UI parts will be created:

- left side navigation bar drawer
- issues list [activity]
    - issue details [fragment]
    - issue characters list [fragment]
- volumes [activity]
    - volume details [fragment]
    - other volume related issues [fragment
- characters [activity]
    - character info [fragment]
    - character story arcs [fragment]
    - character gallery [fragment]
- comic collection management [activity]
- release tracker [activity]
- settings screen [activity]

Task 9: Responsive design

Adapt the application layouts for tablets.

Task 10: UI testing

Write espresso tests for automated UI testing.

Task 11: Synchronization and notifications

Add sync service which loads the newest issue releases list. Sync period depends on the app settings which can be adjusted by the user. Also implement notification system which notifies user when any issue from his favorite volume appears in today's releases list.

Task 12: Google Play Services

Implement chosen services (Firebase Crash Reports and Firebase Analytics).

Task 13: Make the app production ready

- Improve the app visuals using some Material Design techniques (shared element transitions, parallax scrolling, animations etc.)
- Provide RTL-layout support
- Provide accessibility support
- Provide content descriptions for meaningful UI elements
- Plan for localization (move all strings into strings.xml)

Task 14: Prepare for release

- Create app icon (square and round variations)
- Remove or disable debug messages and info
- Follow steps from Android Developers Launch Checklist