

# Majáles festival application

## **Final Documentation**

Kutac, Pastuszek

## Application development

We have spent 78 hours working on the project. We used [www.toggl.com](http://www.toggl.com) web application to measure time we actually spent working on the project. The summary downloaded from the web application is in the separate file when submitting this documentation.

We also used [www.Trello.com](http://www.Trello.com) for project management and git for easy cooperation together. We hosted our repository privately on [www.BitBucket.com](http://www.BitBucket.com) and made regular commits.

## Development progress

We have not been able to finish the application, but we made very big part of it, but the application has to be ready for the festival, which is coming in May so we have to finish it till February. It will still require a lot of testing, since it will go into production.

## Application structure and navigation

The application consist of one loading activity and one main activity. The main activity is filled with fragments. These fragments are replaced in the main activity, when the user navigates to a different page. For navigation through main pages, we used Drawer navigation. Additional detail information can be seen in the News, Practical information and Support program detail pages. In few pages, there is also an Overflow menu used for another actions related to the content (usually option to open the content on the website).

## Data downloading

This was the most difficult part of our back-end. The data are editable through web interface. This web application login is based on WordPress and it generates JSON files, which are then acquired by the mobile application. The structure of those files are the following:

- File with data hash
- File with news hash
- File with data
- File with news

The process is the following: first, the current data and news hashes are compared to the hashes, which are available online. If the hash is the same, the mobile application has the current data available. If the hash of either of those is the same, the data are renewed. We divided the data and news, because we don't suppose the data different than news will be changed often.

When the application is installed, there are no data, which are editable through the web interface. The data has to be downloaded before the application is able to be functional. If the user is using the Wi-Fi, the data will be downloaded automatically, if the user is using mobile data, the application will ask the user, if he wants to download data or postpone downloading for later.

We use and we download 2 types of pictures – Thumbnail and Cover pictures. All of these are downloaded immediately asynchronously with the data.

For downloading we have used two libraries:

- Retrofit – downloading requests from web and transforming JSON to data model using GSON.
- Picasso – downloading images from web and loading images into views

## Data storage

JSON files are parsed with GSON library and then stored in Shared preferences. The mapping of the data to our model classes are done using the same GSON library automatically.

Thumbnail images are saved inside Local storage. Cover images are saved inside External cash storage.



