GitHub Username: PanGargamel

DailyWallpaper

Description

DailyWallpaper is an application that will make your phone surprise you every single day. Select category you like the most, and let the app change your wallpaper every 24 hours!

Intended User

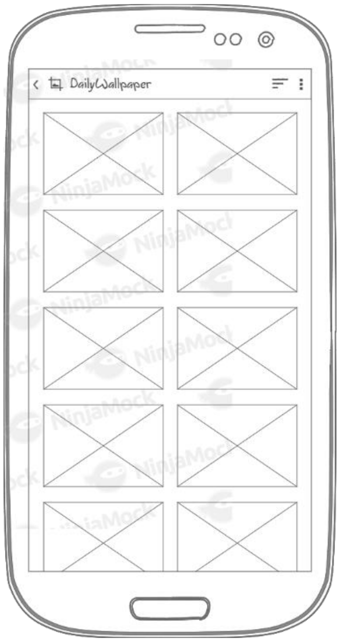
DailyWallpaper is an application for anyone who wants their phone to become more alive.

Features

- Scroll through over 1.4 million of wallpapers
- Mark any wallpaper as favorite, and access them later again
- Let your phone surprise you by automatically changing your wallpaper very single day

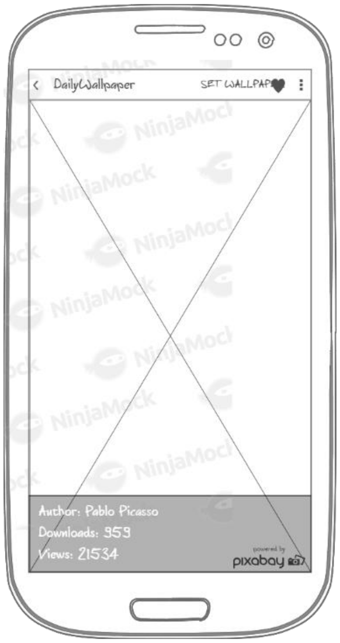
User Interface Mocks

Screen 1



Main view – the images will be displayed in two-column grid. The sorting order may be changed using a menu button. Clicking on any of the images will take user to a detail view.

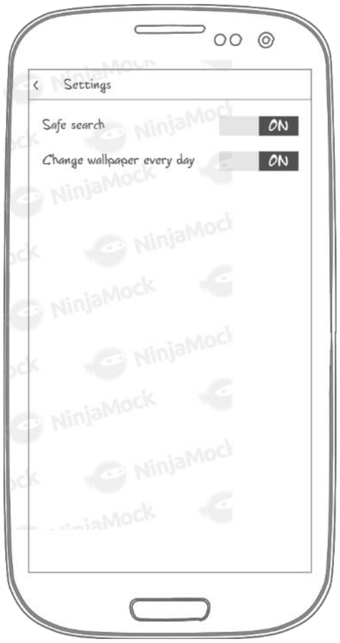
Screen 2



Author: Pablo Picasso
Downloads: 959
Views: 21534

Detail view – the image takes all of the application space. On the bottom, over the image, there will be displayed basic info about the image – author, downloads, etc. Using the menu, user can mark the image as favorite or set is a wallpaper

Screen 3



Safe search ON
Change wallpaper every day ON

Settings view – it will allow user to change basic application configuration, such as using the “SafeSearch” filter, or enabling/disabling daily wallpaper feature.

Screen 4

1x1 widget, allowing user to easily change wallpaper to another, random one.

Key Considerations

How will your app handle data persistence?

DailyWallpaper will include Room Persistence Library. It will store images that user has marked as favorite. User will be able to access them later using a sorting button in main activity. Application settings (such as “SafeSearch” filter being enabled/disabled) will be stored using SharedPreferences.

Describe any edge or corner cases in the UX.

In the main view, user can click any of the images to check its details, mark it as favorite, or set it as a wallpaper. User can also access settings menu using a menu button in right top corner.

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

Picasso – for images loading
Recyclerview – for displaying images efficiently
Butterknife – for easier data binding
Gson – for JSON to POJO conversion
Firebase – for Firebase Crashlytics and Google Analytics for Firebase

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

Firebase Crashlytics will be used to track stability issues in realtime.
Google Analytics for Firebase will be used to analyze users behavior.

Next Steps: Required Tasks

Task 1: Project Setup

- Create a new project in Android Studio
- Configure libraries
- Add project to Github
- Setup a Pixabay account and get an API key

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity
- Build UI for SettingsActivity

Task 3: Implement data loading in MainActivity

- Setup a RecyclerView
- Create data model classes
- Setup a Retrofit client to properly load data
- Wire everything up, so images in RecyclerView will be loaded using Retrofit

Task 4: Implement data loading in DetailActivity

- Implement opening DetailActivity after clicking on image in MainActivity
- Make MainActivity pass extra data about selected image to DetailActivity
- Make DetailActivity parse and properly display extra data

Task 5: Implement Room Persistence Library

- Implement marking wallpaper as favorite using the heart button in DetailActivity
- Implement loading favorite wallpapers after switching the sorting order in MainActivity

Task 6: Configure Settings

- Store all data from settings in SharedPreferencese
- Make Retrofit include the “SafeSearch” filter while making an API call

Task 7: Implement widget

- Add a 1x1 widget with one button
- Make the button change phone wallpaper to an another random one

Task 8: Implement Google Play Services

- Implement Firebase Crashlytics
- Implement Google Analytics for Firebase

Other technical information

Application will be written solely in the Java Programming Language. It will also utilize stable release versions of all libraries, Gradle, and Android Studio. All strings will be kept in a strings.xml file, and RTL layout switching will be enabled on all layouts.

To provide good accessibility support, content descriptions will be provided on all ImageViews.

All images will be loaded using external Application Programming Interface, provided by Pixabay.com. Loading data will be performed using AsyncTask, so the requests will not be executed on a main thread.

API documentation: https://pixabay.com/api/docs