GitHub Username: Abdallah-Abdelazim

Star Wars Wallpapers

Description

A wallpapers app for Star Wars fans! This application allows users to show best wallpapers from the Star Wars franchise. Users can easily choose one of the offered wallpapers and set it on their home screen or lock screen. Also, they can download the wallpaper on their device storage or share it with a friend.

Instead of having to search the web for wallpapers, this application brings a curated set of wallpapers that are categorized and organized to all the Star Wars fans around the world.

Intended User

This app is aimed to the Star Wars franchise fans. Any person who have watched Star Wars movies or series or have read Star Wars comics is a target audience.

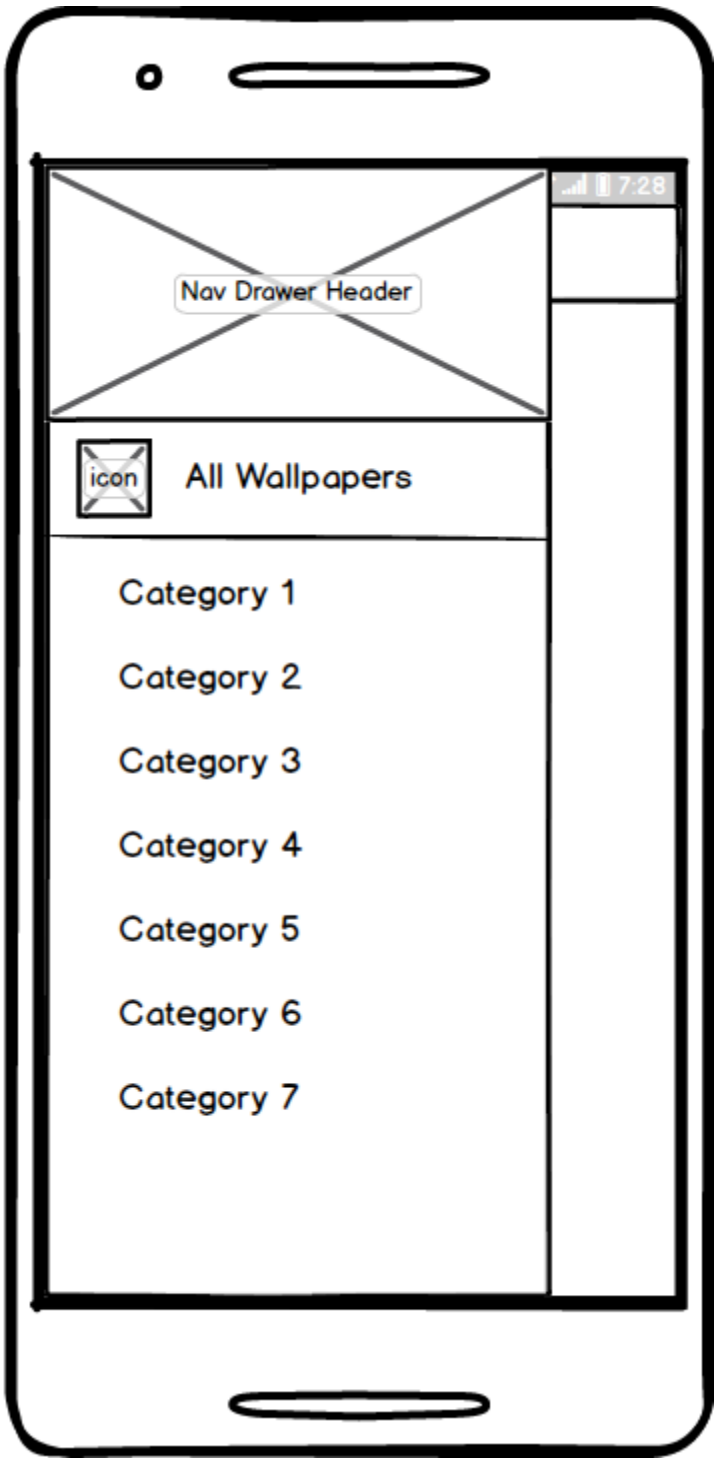
Features

- Display wallpapers in a grid so that the user can easily scroll between them.
- Allow the user to choose a specific category to show only the wallpapers belonging to it or simply display all the wallpapers from all categories.
- Display single wallpaper details which includes:
    a. The wallpaper itself allowing him to zoom in it using touch gestures and when clicked it goes full screen.
    b. Allow the user to set the wallpaper on the home screen, lock screen or both (lock screen wallpapers require Android Nougat 7.0 or higher).
    c. Allow the user to download the wallpaper as an image to his device storage.
    d. Allow the user to share the wallpaper through other apps.
    e. Display title & some description for the wallpaper.
    f. Display the wallpaper author/creator.

User Interface Mocks

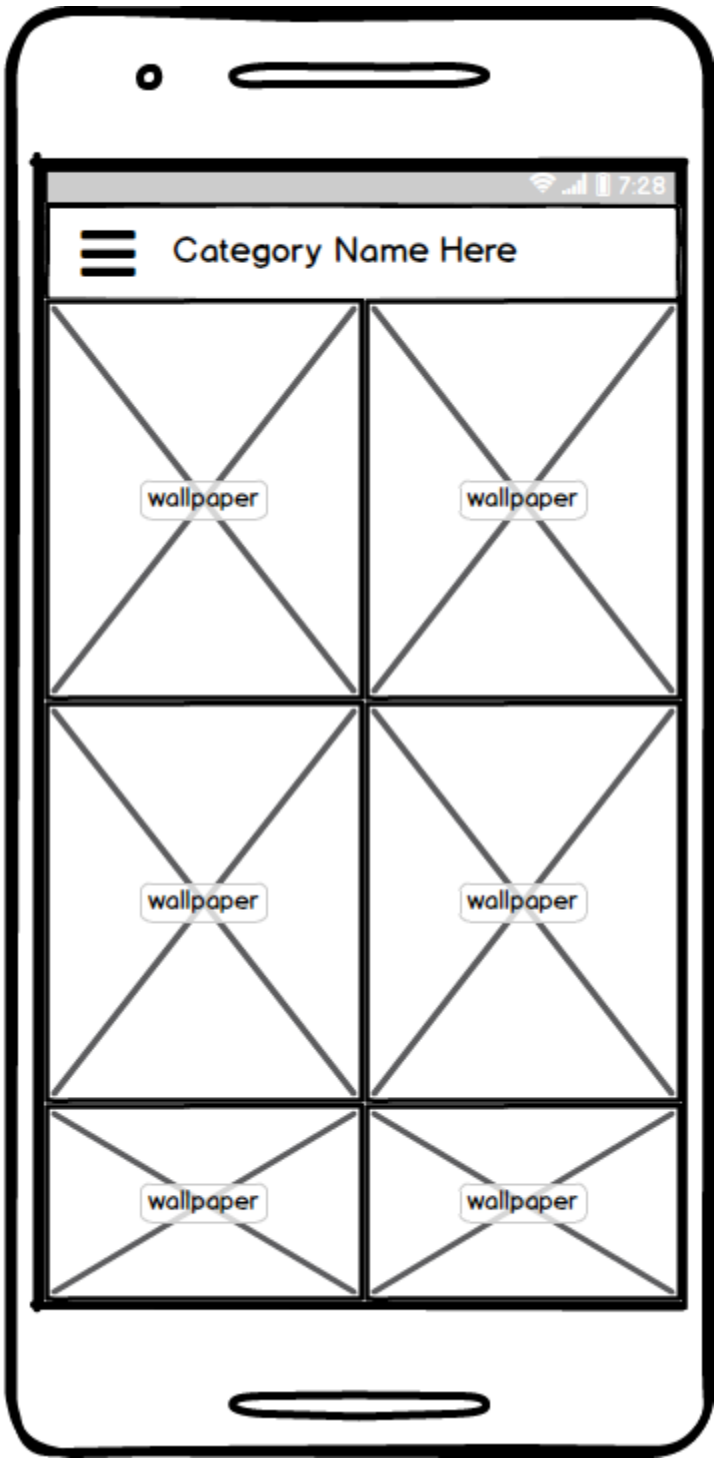
Screen 1

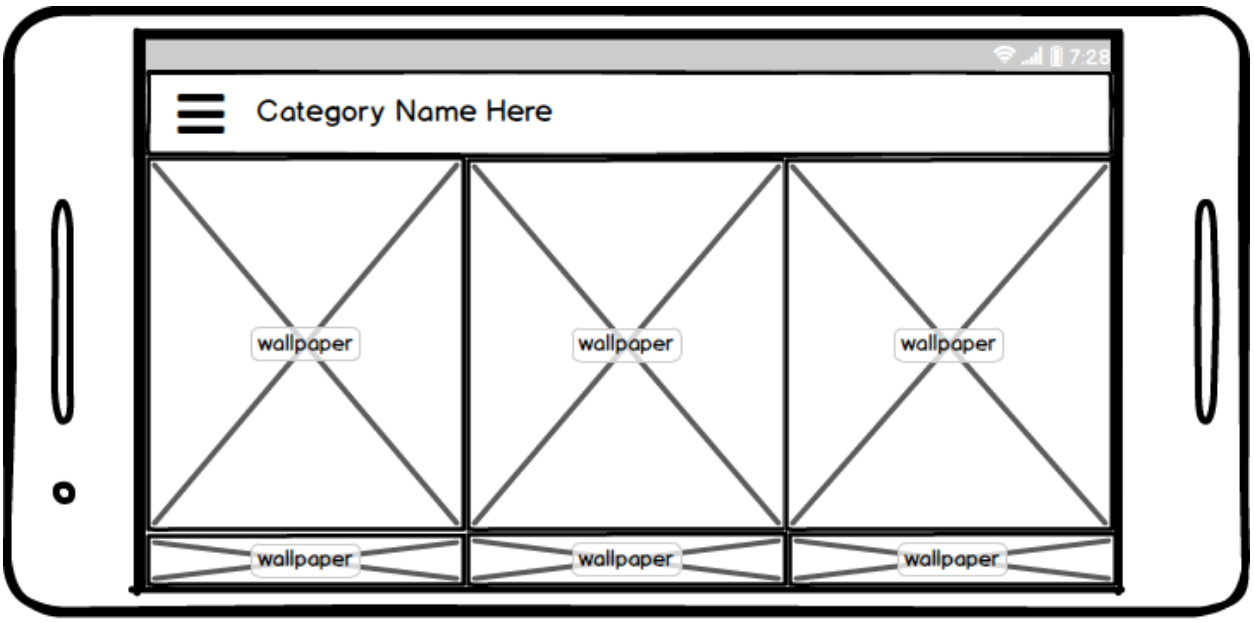
The app's navigation drawer.



Screen 2

The wallpapers screen in portrait and landscape orientations.

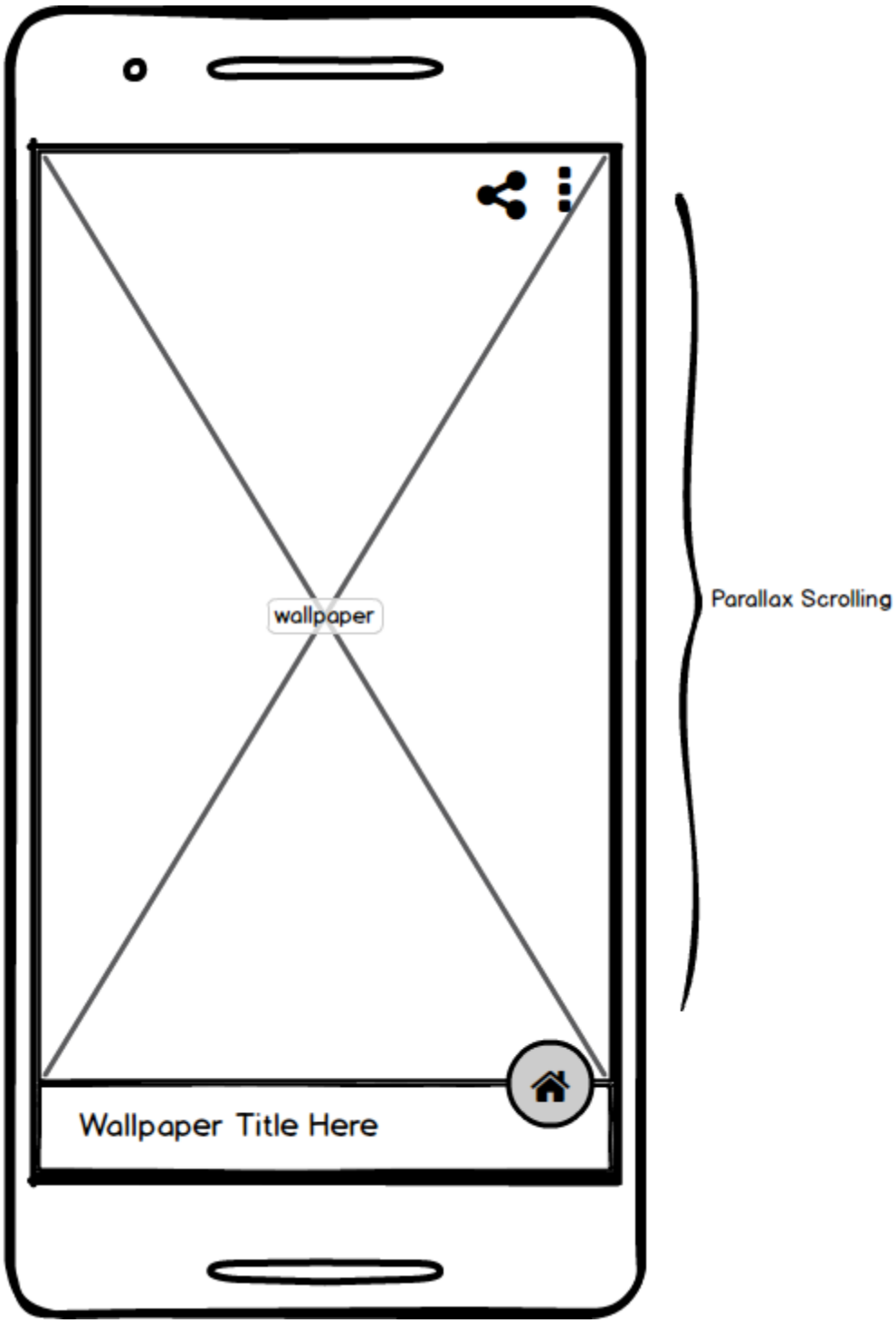


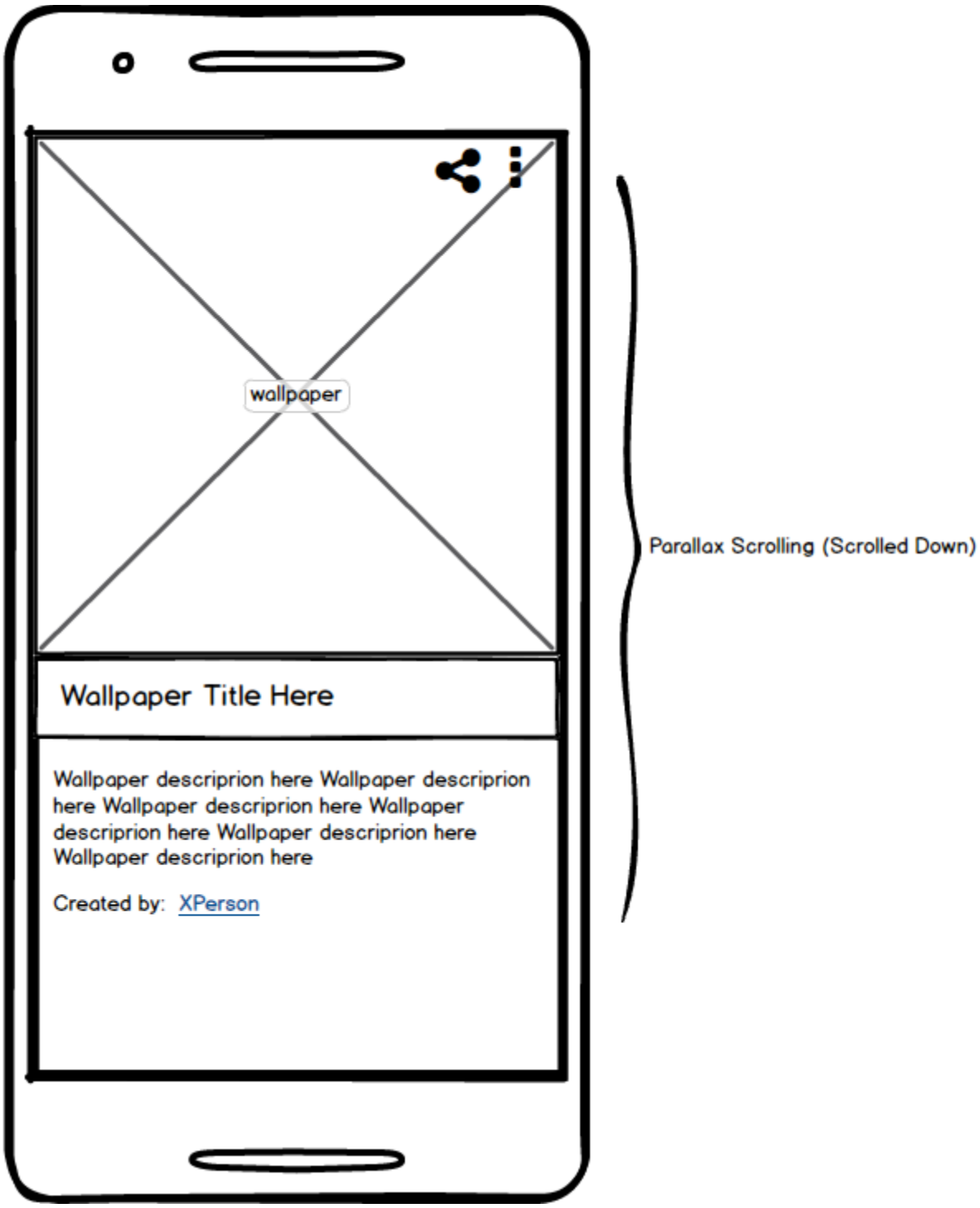


Displaying 3 or more wallpapers in landscape to take advantage of the extra space

Screen 3

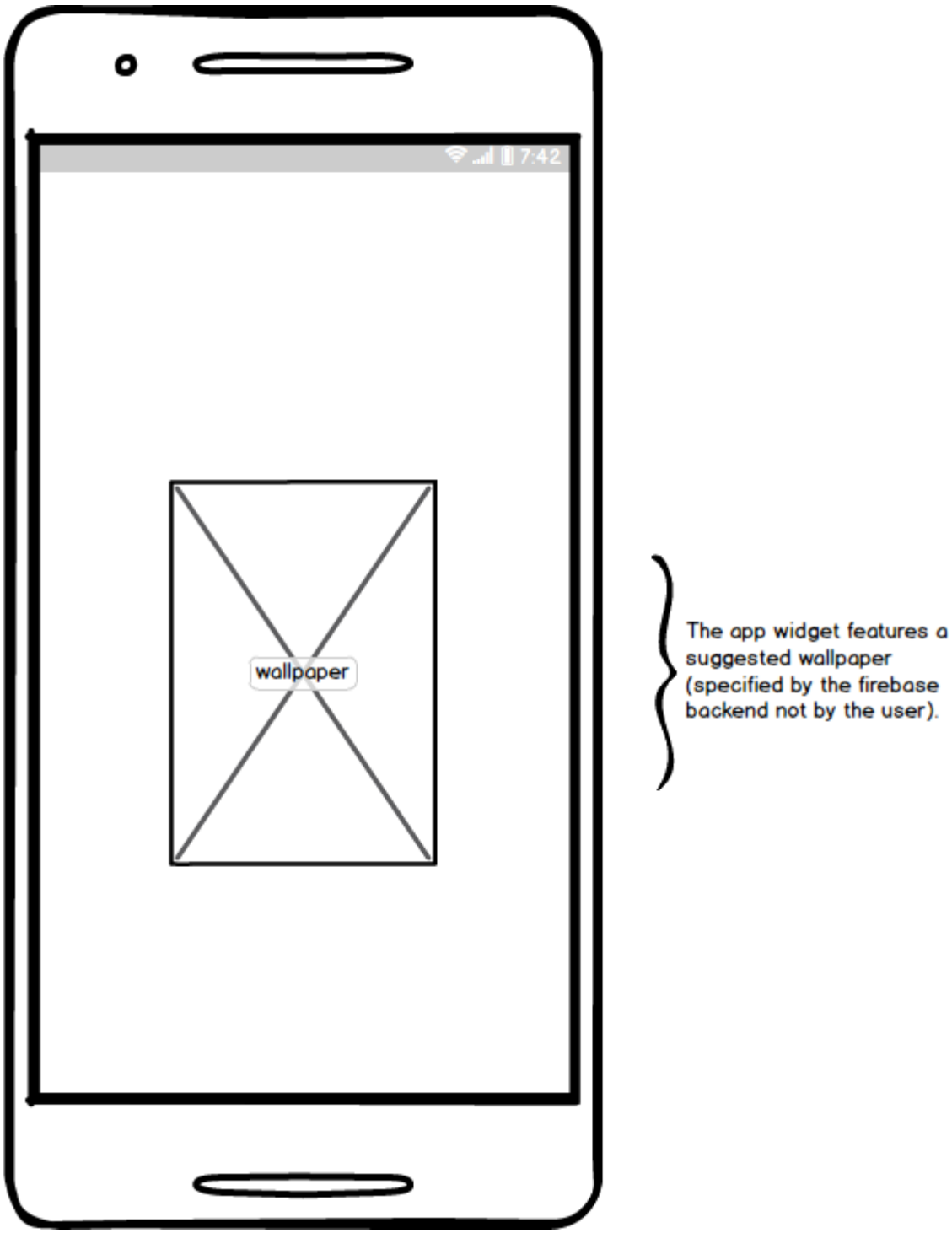
Wallpaper details screen.




wallpaper
Wallpaper Title Here
Wallpaper descriprion here Wallpaper descriprion here Wallpaper descriprion here Wallpaper descriprion here Wallpaper descriprion here Wallpaper descriprion here
Created by: XPerson
Parallax Scrolling (Scrolled Down)

Screen 3

The app's widget.



Key Considerations

Which technologies & programming languages will be used in development?

The app will be built with the native Android SDK. App is written solely in the Java Programming Language.

The stable version of Android Studio (ver. 3.1.3) will be used as the development IDE. The stable Gradle build tool (ver. 4.4) will be used to manage the build process.

How will this app handle data persistence?

App will use Firebase Realtime Database. App will cache data to disk by enabling the disk persistence feature of the Firebase Realtime Database.

Describe any edge or corner cases in the UX.

- The app will handle the case in which the device has no network connectivity. It will ask the user to turn on Wi-Fi or cellular data and display a RETRY button.
- The app will offer the user to set the wallpaper on the lock screen only if the user's device runs Android Nougat 7.0 or higher. This feature is not available in Android versions below than that and will be disabled. However, all devices can set the wallpaper on the home screen.
- The app will utilize dialogs to ask the user for extended actions. For example, the 'Set as wallpaper' button will open a dialog asking the user whether he wants to set the wallpaper on the home screen, lock screen or both.
- Setting the wallpaper is done in an Async Task away from the main thread to avoid app freeze. So, for example when the user chooses a wallpaper and clicks on 'Set as wallpaper' button, this will start an Async Task which will set the wallpaper on the device.
- For higher security of the user and the application, DEBUG & VERBOSE log levels will be disabled on release builds.
- App keeps all strings in the strings.xml file.
- The app enables RTL layout switching on all layouts.
- The app includes support for accessibility. That includes content descriptions & navigation using a D-pad.

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

- Butter Knife (ver. 8.8.1): for field and method binding for Android views.
- Glide (ver. 4.7.1): for handling the loading and caching of images.
- Parceler (ver. 1.1.11): for generating the Android *Parcelable* boilerplate source code.
- Timber (ver. 4.7.1): for better logging utilities.
- PhotoView (ver. 2.1.4): ImageView for Android that supports zooming, by various touch gestures.

All library versions used are stable versions (the latest at time of writing this document).

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

The application will use:

- Firebase Analytics: For various analytics data collected about the users of the app.
- Firebase Performance Monitoring: To collect data about app performance such as start time & loading time.
- Firebase Crashlytics: For collecting the app's crash reports.

The application also uses:

- Firebase Realtime Database: Serves as the app's backend database that stores the app's data.
- Firebase Storage: Serves as host for the wallpaper image files.
- Firebase Cloud Messaging: For user engagement through notifications.

Next Steps: Required Tasks

Task 1: Project Setup

Create and setup a new project. This task includes:

- Creating a new project in Android Studio
- Creating a GitHub Repository for the project and setup Git to use this remote repository.
- Configuring libraries by adding all necessary dependencies

Task 2: Create Model Classes

Create model classes which will be used to deserialize the Firebase responses.

This include:

- Create 'Wallpaper' model class.
- Create 'Author' model class.
- Create 'Category' model class.

Task 3: Implement UI for Each Activity and Fragment

This task includes the following subtasks:

- Build UI for MainActivity and its navigation drawer.
- Build UI for WallpapersFragment to display wallpapers of a specific category in an adaptive grid layout.
- Build UI for WallpaperDetailsActivity.
- Build UI for WallpapersDetailsFragmet which will be added dynamically to WallpaperDetailsActivity.

Task 4: Implement Firebase Services

This include the following subtasks:

- Add the required Firebase dependencies for the used services.
- Add the functionality to get the category, wallpapers data from the Firebase Realtime Database.
- Add the functionality to display images from the Firebase Storage as well as downloading them.
- Configure Firebase disk caching to enable data persistence.

Task 5: Implement App Features

Implement the features as described earlier. Create utility functions for setting a specific image as wallpaper, downloading a specific wallpaper & sharing a wallpaper and then use these utility in there right places to enable these features on the app.

Task 6: Handle Error Cases

Handle the corner cases that may cause an error. This includes:

- Handle opening the app while no network connectivity. Or losing network connectivity while using the app.
- Display error messages to inform the user that an error has occurred and why.

Task 7: Create Build Variants

Configure Gradle for the app's release build and generate a signed APK.

Task 8: Test the App and Solve the Errors & Bugs Noticed

Take time to use the app and try all of its functionality to determine bugs and errors and then try to solve them.

Task 9: Prepare the App for Publishing on The Google Play Store

This include:

- Setup Timber logging library to disable DEBUG & VERBOSE log levels on release builds.
- Create the app's icon.
- Create the app's graphics and screenshots.
- Publish the app's release APK on the Google Play Store to allow other people to reach it and use it.