Homework 9: Artist App

Prof. Marco Papa

Developed and Designed by

Kartik Bharatbhai Patel (Android)

Contents

1. OBJECTIVES

- Become familiar with Java, JSON, Android Lifecycle and Android Studio for Android app development.
- Build a good-looking Android app.
- Learn the essentials of Google’s Material design rules for designing Android apps
- Learn to use the Artsy APIs and the Android SDK.
- Get familiar with third party libraries like Picasso, Glide and Volley.

The objective is to create an Android application as specified in the document below and in the reference video.

2. BACKGROUND

2.1 Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android application development, based on IntelliJ IDEA - a powerful Java IDE. On top of the capabilities you expect from IntelliJ, Android Studio offers:

- Flexible Gradle - based build system.
- Build variants and multiple apk file generation.
- Code templates to help you build common app features.
- Rich layout editor with support for drag and drop theme editing.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- ProGuard and app-signing capabilities.
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

More information about Android Studio can be found at:
http://developer.android.com/tools/studio/index.html

2.2 Android

Android is a mobile operating system initially developed by Android Inc. a firm purchased by Google in 2005. Android is based upon a modified version of the Linux kernel. As of Nov 2018, Android is the number 1 mobile OS, in unit sales, surpassing iOS, while iOS was still the most profitable platform.

The Official Android home page is located at:
http://www.android.com/
The Official Android Developer home page is located at:
http://developer.android.com/

2.3 Amazon Web Services (AWS)

AWS is Amazon's implementation of cloud computing. Included in AWS is Amazon Elastic Compute Cloud (EC2), which delivers scalable, pay-as-you-go compute capacity in the cloud, and AWS Elastic Beanstalk, an even easier way to quickly deploy and manage applications in the AWS cloud. You simply upload your application, and Elastic Beanstalk automatically handles the deployment details of capacity provisioning, load balancing, auto-scaling, and application health monitoring. Elastic Beanstalk is built using familiar software stacks such as the Apache HTTP Server, PHP, and Python, Passenger for Ruby, IIS for .NET, and Apache Tomcat for Java.

The Amazon Web Services homepage is available at: http://aws.amazon.com/

2.4 Google App Engine (GAE)

Google App Engine applications are easy to create, easy to maintain, and easy to scale as your traffic and data storage needs change. With App Engine, there are no servers to maintain. You simply upload your application and it's ready to go. App Engine applications automatically scale based on incoming traffic. Load balancing, micro services, authorization, SQL and NoSQL databases, memcache, traffic splitting, logging, search, versioning, roll out and roll backs, and security scanning are all supported natively and are highly customizable.

To learn more about GAE support for Node.js visit this page:

https://cloud.google.com/appengine/docs/nodejs

2.5 Microsoft Azure

The Azure cloud platform is more than 200 products and cloud services designed to help you bring new solutions to life—to solve today's challenges and create the future. Build, run, and manage applications across multiple clouds, on-premises, and at the edge, with the tools and frameworks of your choice.

To learn more about the Azure services, visit this page:

https://azure.microsoft.com/en-us/solutions/

To learn more about Azure support for Node.js visit this page:

https://docs.microsoft.com/en-us/azure/devops/pipelines/targets/webapp?view=azure-devops&tabs=yaml#deploy-a-javascript-nodejs-app.

3. PREREQUISITES

This homework requires the use of the following components:

- Download and install Android Studio. Technically, you may use any IDE other than Android Studio such as Eclipse, but the latest SDKs may not be supported with Eclipse. We will not be providing any help on problems arising due to your choice of alternate IDEs.
- You must use the **emulator, preferably Pixel 5 with API 30**. Everything should just work out of the box.
- If you are new to Android Development, Hints are going to be your best friends!

4. HIGH LEVEL DESIGN

This homework is a mobile app version of Homework 6 and Homework 8.

In this exercise, you will develop an Android application, which allows users to search for different artists and look at the detailed information about them. Users can also save their favorite artist to view them or track their artworks. The App contains 3 screens: Home screen, Search Result screen and Detailed Artist Information screen. However, the App has multiple features on each of these screens.

This homework contains 5 API calls. There are the calls to the Artsy APIs for authentication, search, artist, artwork, gene (category). Each of these 5 API calls are the same as Homework #8. So, you can use the same Node.js backend as Homework #8. In case you need to change something in Node, make sure you do not break your Angular assignment (or deploy a separate copy) as the grading for homework will not be finished at least until 1 week later.

We suggest you to use Java. Kotlin is allowed but will not be supported in piazza.

PS: This app has been designed and implemented in a Pixel 5 emulator by using SDK API 30. It is highly recommended that you use the same virtual device and API to ensure consistency.

Demo will be on an emulator using Zoom, no personal devices allowed, see the rules:

https://csci571.com/courseinfo.html#homeworks

5. IMPLEMENTATION

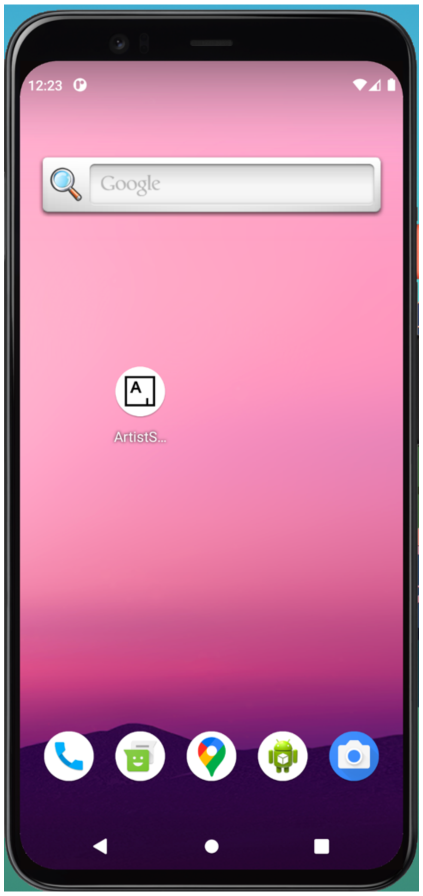


Figure 1.1

App Icon

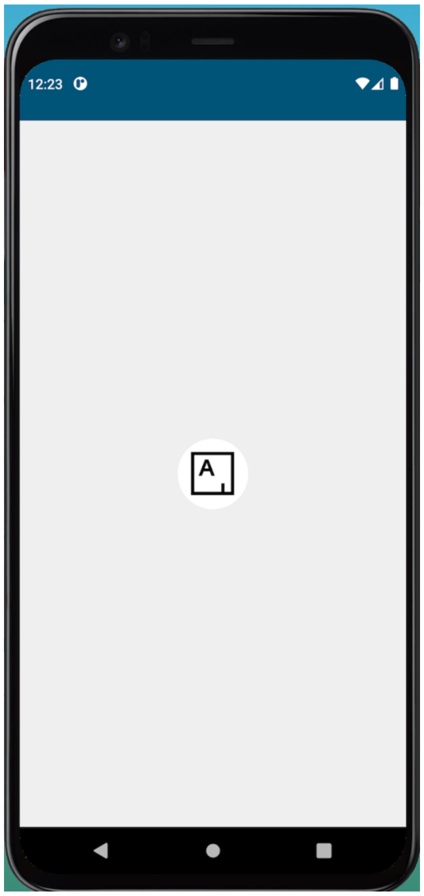


Figure 1.2

Splash Screen

5.1 App Icon and Splash Screen

In order to get the app icon/image, please see the **hints** section.

The app begins with a welcome screen (Figure 1.2) which displays the icon provided in the hint above.

This screen is called Splash Screen and can be implemented using many different methods. The simplest is to create a resource file for the launcher screen and add it as a style to AppTheme.Launcher (see **hints**). Please refer to Figure 1.1 and Figure 1.2.

5.2 Home screen

When you open the app, there will be an initial progress bar while the data is being fetched using volley as shown in Figure 2.1. The home screen will have a toolbar at the top with title "ArtistSearch" and the search icon. Below that, it will show the current date as shown in Figure 2.2.

There is 1 Section on the home screen called Favorites Section which is described below.

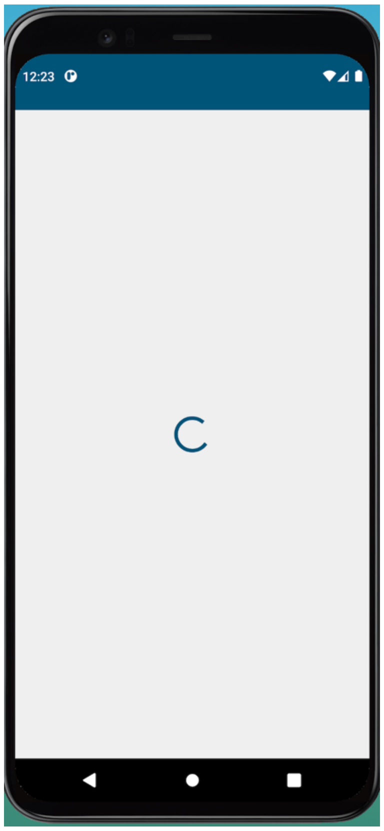


Figure 2.1

Progress Bar

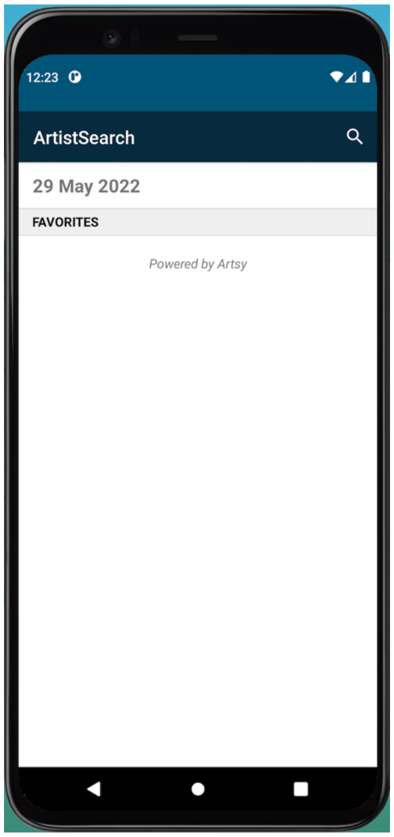


Figure 2.2

Initial Home Screen

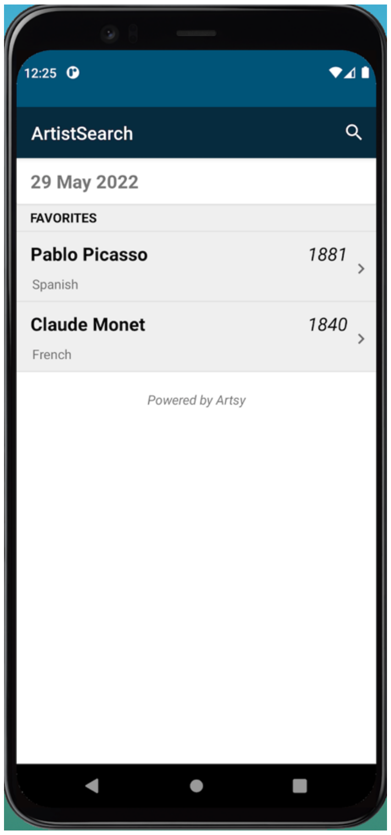


Figure 2.3

Home Screen

<u>**If the user starts the app for the first time, they will not have any artist in the favorites section.**</u>

Please refer to the Hints section for these icons

- ***Favorites Section*** - This section will show all the artists that have been marked as "favorite" by the user. This is the new feature in addition to Homework #8. For each favorited artist show:
    - Artist Name
    - Nationality
    - Birthday

Each artist listing also has a button on the extreme right, next to the birthday field (right arrow (or) a chevron-right button). On clicking the button, the detailed artist information screen will open for the selected artist.

At the bottom of the favorites sections, we have a 'Powered by Artsy' text in italic. On clicking this text, the App should open the Artsy homepage in browser. (Artsy URL: https://www.artsy.net/).

The home screen has been implemented by using a **RecyclerView** with the **SectionedRecyclerViewAdapter.** Each of the artist listings has been implemented using **ConstraintLayout, TextView.**

The **Search** button on the toolbar opens the search bar to type the artist name to search.

5.3 Search Functionality

- On the top right side, there will be a search button which opens a textbox where the user can enter a keyword to search for an artist name. See hints for icon.
- When the user enter the keyword inside the search box and clicks enter/next it takes the user to the search result screen.
- Before you get the data from your backend server, a progress bar should display on the screen as indicated in the search result screen section.

Implementing search functionality requires only searchable – see hints

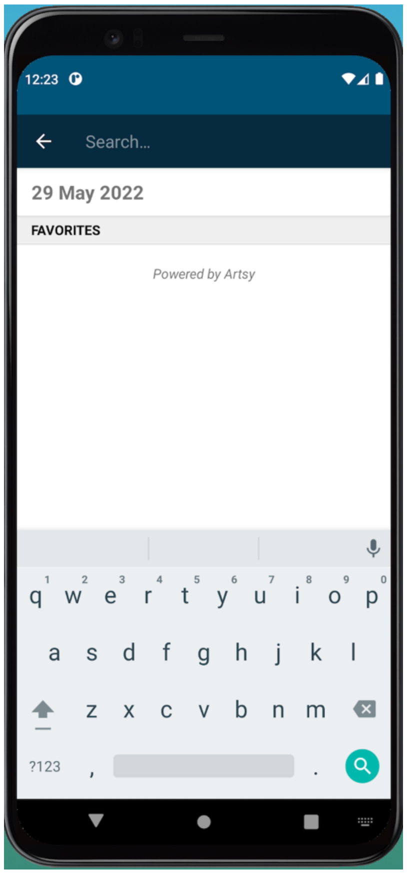


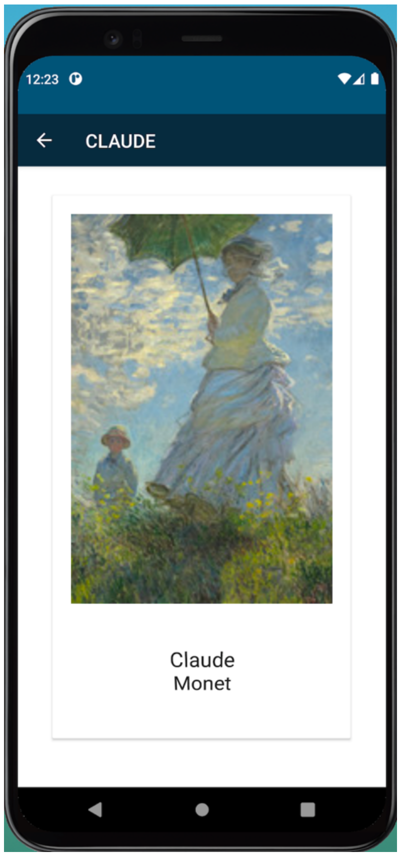
Figure 3.1
Search Bar

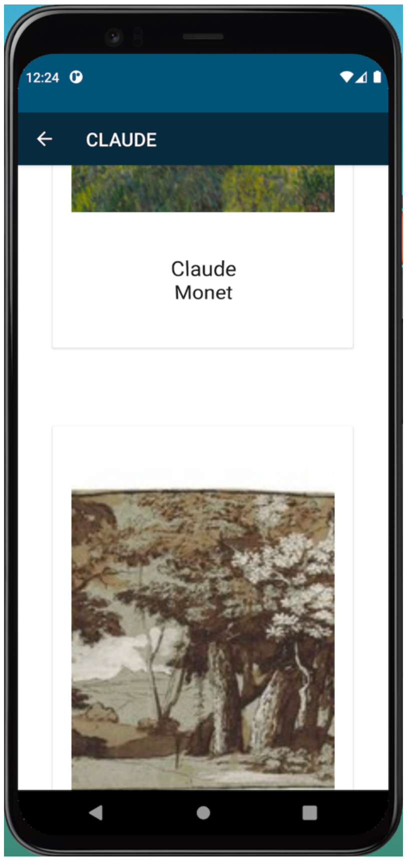
5.4 Search Result Screen

When user performs a search from the home screen through action bar, a progress bar should come up while results from the backend are being fetched. Once, the results are available, the progress bar should disappear and screen containing list of artist with their Name and Image (image might take longer to load sometimes) in Card view should be displayed to the user.

- Search Result screen should be vertically scrollable and uses **RecyclerView.**
- You can load web images with Picasso. (see hints)
- Images should be uniform and can be resized or fitted.

*If there is no result for the given keyword, it should display **'No Results Found'**. See reference video and **Figure 4.****





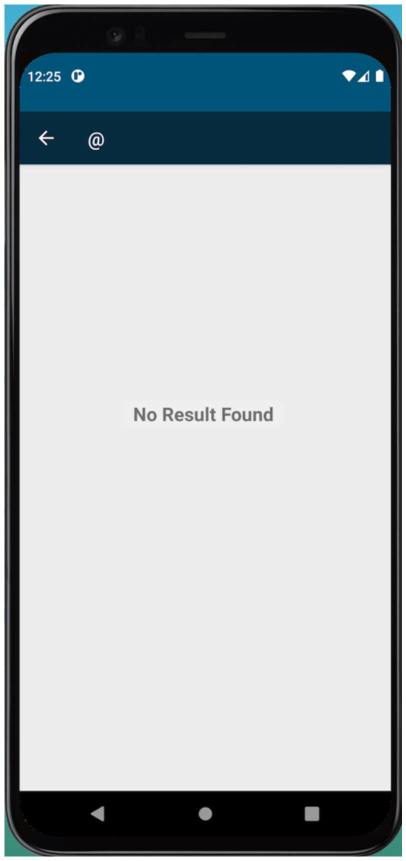


Figure 4.1

Search Result Screen

Figure 4.2

Vertically Scrollable

Figure 4.3

No Result Found

5.5 Detailed Artist Information Screen

On clicking the **chevron-right** button in favorites or **clicking** an artist from search result screen, a progress bar should come up while the artist details are being fetched. Once the data has been fetched the progress bar should disappear and Tabbed Screen with Details Tab (Default) should be available to the user **(Figure 5.*)**.

The top action bar must show the artist name and the back button to go back to the search result screen. The action bar should also contain a favorite icon to add or remove the artist from favorites. The favorite icon will either be filled or bordered depending on whether the artist is marked as favorite or not. Adding/Removing the artist from favorites should also display a toast message as shown in the video. See hints for icons.

Below the Action bar, there should be 2 Tab Headings: Details and Artwork with their respective icons on it. By Default, Details tab should be selected whenever this screen is opened. Each Tab should contain following when selected.

Details Tab:

- Details Tab should contain Artist Name, Artist Nationality, Artist Birthday, Artist Deathday and Artist Biography.

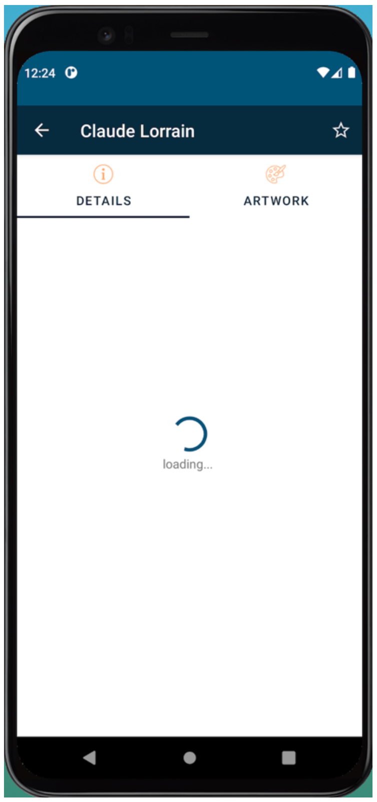
If any of the above mentioned information is null or missing, handle it properly by not displaying its entire part.

Artwork Tab:

- Artwork Tab should contain a list of artworks, each artwork is displayed in a Card view containing Artwork Name and Artwork Image.
- Artwork Image should be clickable.
- When Artwork image is clicked, it should open a modal/dialog box containing Category information.
- Category Dialog box contains category name, category description and category thumbnail.

If there is no artwork for the artist, it should display 'No Artworks Found'.

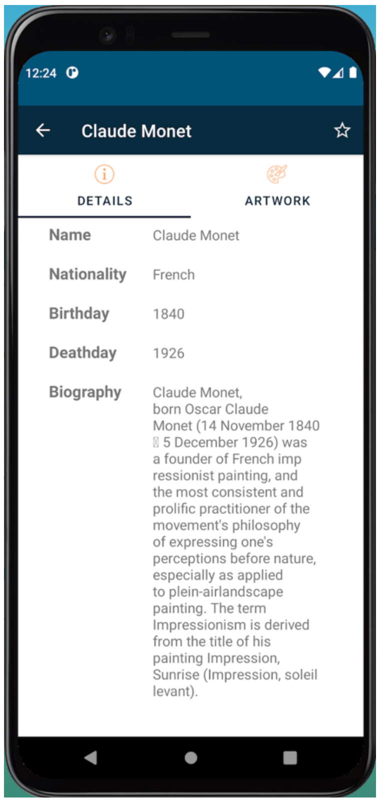
See reference video and **Figure 5.*.**


12:24

Claude Lorrain

Figure 5.1

Initial Progress bar

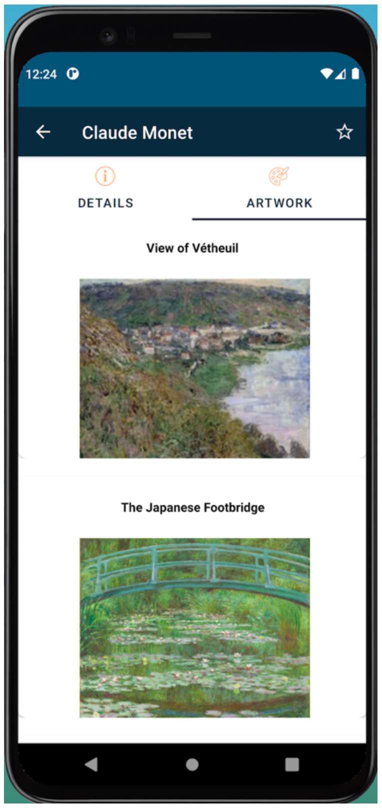

12:24

Claude Monet

Name	Claude Monet
Nationality	French
Birthday	1840
Deathday	1926
Biography	Claude Monet, born Oscar Claude Monet (14 November 1840 � 5 December 1926) was a founder of French imp ressionist painting, and the most consistent and prolific practitioner of the movement's philosophy of expressing one's perceptions before nature, especially as applied to plein-airlandscape painting. The term Impressionism is derived from the title of his painting Impression, Sunrise (Impression, soleil levant).

Figure 5.2

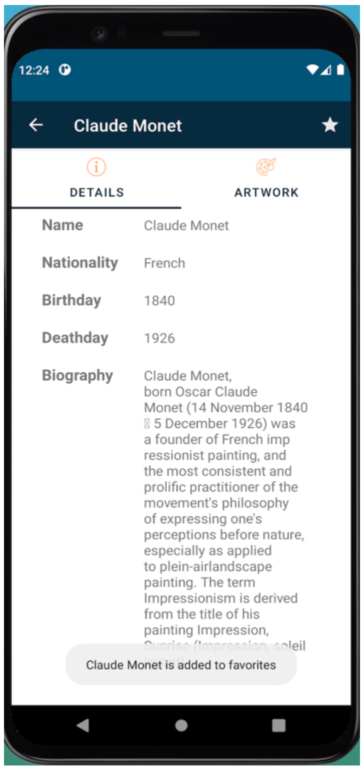
Default Detail tab


12:24

Claude Monet

Figure 5.3

Artwork Tab


12:24

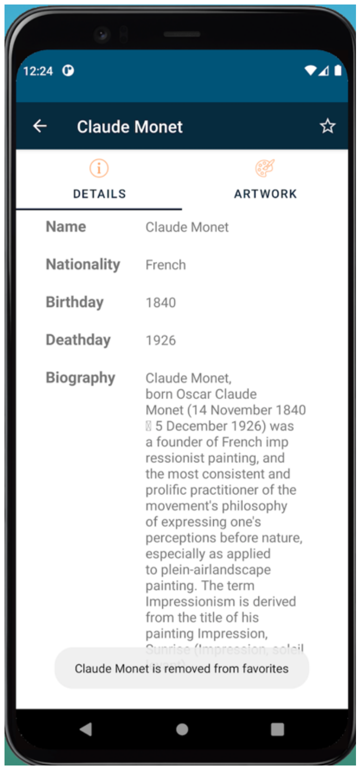
Claude Monet

Name	Claude Monet
Nationality	French
Birthday	1840
Deathday	1926
Biography	Claude Monet, born Oscar Claude Monet (14 November 1840 � 5 December 1926) was a founder of French imp ressionist painting, and the most consistent and prolific practitioner of the movement's philosophy of expressing one's perceptions before nature, especially as applied to plein-airlandscape painting. The term Impressionism is derived from the title of his painting Impression, Sunrise (Impression, soleil

Claude Monet is added to favorites

Figure 5.4

Favorite Added


12:24

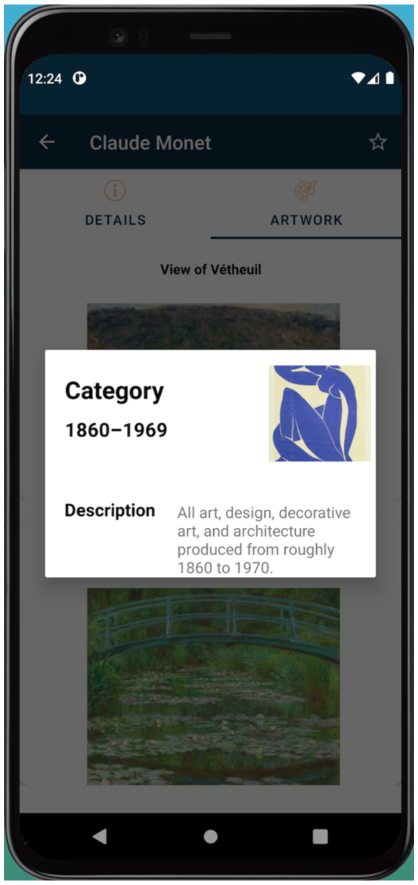
Claude Monet

Name	Claude Monet
Nationality	French
Birthday	1840
Deathday	1926
Biography	Claude Monet, born Oscar Claude Monet (14 November 1840 � 5 December 1926) was a founder of French imp ressionist painting, and the most consistent and prolific practitioner of the movement's philosophy of expressing one's perceptions before nature, especially as applied to plein-airlandscape painting. The term Impressionism is derived from the title of his painting Impression, Sunrise (Impression, soleil

Claude Monet is removed from favorites

Figure 5.5

Favorite Removed



Category

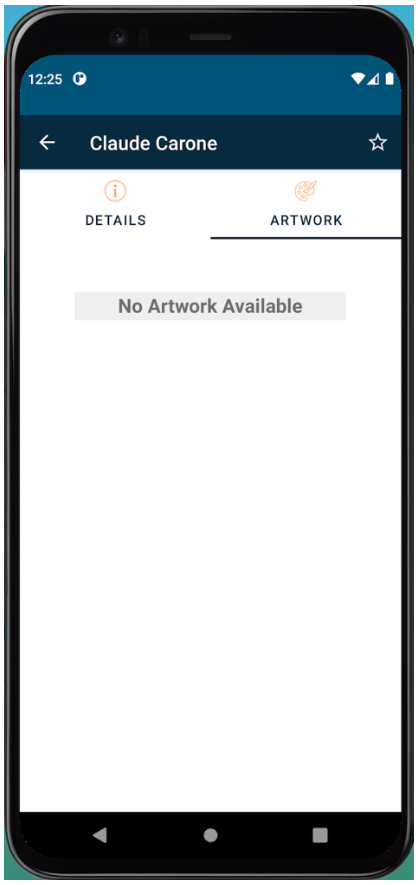
1860–1969

Description

All art, design, decorative art, and architecture produced from roughly 1860 to 1970.

Figure 5.6

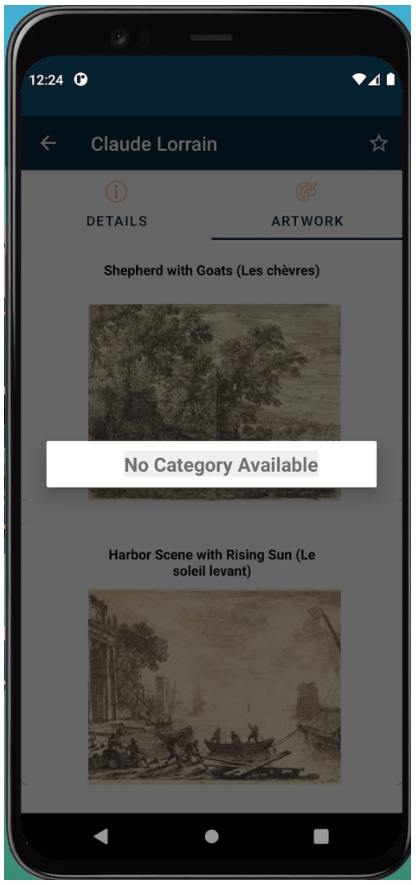
Category Dialog Box



No Artwork Available

Figure 5.7

No Artwork


12:24
Claude Lorrain
DETAILS
ARTWORK
Shepherd with Goats (Les chèvres)
Harbor Scene with Rising Sun (Le soleil levant)

No Category Available

Figure 5.8

No Category

5.6 Local Storage

You should save the list of favorite artists to local storage. When the app is opened, you should load them, then fetch data from your backend.

We will be using Shared Preferences for storing the app data for this assignment. See hints for more details on the implementation of the Shared Preferences.

5.7 Progress bar

Every time the user has to wait before they can see the data, you must display a progress bar as shown in Figure 4. The progress bar is to be present across the **Home screen, Search Result screen and Detailed Artist Information screen**. One idea would be to display the progress bar by default (where needed) and then hide it as soon as the data is received/view is prepared. See hints for progress bar related ideas and styling.

Note: Based on your implementation, you might need to put progress bars in different places.

5.8 Summary of detailing and error handling

1. Make sure there are no conditions under which the app crashes

2. Make sure all icons and texts are correctly positioned as in the video/screenshots

3. Make sure the screens and toasts are correctly displayed.

4. Make sure there is a progress bar on initial loading of the home screen, and while loading the search result screen, detailed artist information screen.

5. Make sure the styling for different features matches the video/screenshots.

5.9 Additional Info

For things not specified in the document, grading guidelines, the video, or in Piazza, you can make your own decisions. But keep in mind about the following points:

- Always display a proper message and do not crash if an error happens.
- All HTTP requests should be asynchronous and should not block the main UI thread. You can use third party libraries like Volley to achieve this in a simple manner.

6. IMPLEMENTATION HINTS

6.1 Icons

The images used in this homework are available in the zip file mentioned in the Piazza post for Homework #9.

Furthermore, please refer to the following websites and search for additional icons.

1. https://materialdesignicons.com
    a.

Icon Description	**Icon name to search for**
Right arrow button inside Favorites items	Chevron-right
Search Icon	Magnify
Favorite Icons	Star, Star-outline
Details Icon	Alert/Error
Artwork Icon	Paint

You can choose to work with xml/png/svg/jpg versions. We recommend using xml as it is easy to modify colors by setting the Fill Colors.

6.2 Third party libraries

Sometimes using 3rd party libraries can make your implementation much easier and quicker. Some libraries you may have to use are:

6.2.1 Volley HTTP requests

Volley can be helpful with asynchronous http request to load data. You can also use Volley network ImageView to load photos in Google tab. You can learn more about them here:

https://developer.android.com/training/volley/index.html

6.2.2 Picasso

Picasso is a powerful image downloading and caching library for Android.

http://square.github.io/picasso/

6.2.3 Glide

Glide is also a powerful image downloading and caching library for Android. It is like Picasso. You can also use Glide to load images.

https://bumptech.github.io/glide/

6.3 Working with action bars and menus

https://developer.android.com/training/appbar/setting-up

https://stackoverflow.com/questions/38195522/what-is-oncreateoptionsmenumenu-menu

6.4 Displaying Progress Bars

https://stackoverflow.com/a/28561589

https://stackoverflow.com/questions/5337613/how-to-change-color-in-circular-progress-bar

6.5 Implementing Splash Screen

There are many ways to implement a splash screen. This blog highlights almost all of them with examples:

https://android.jlelse.eu/the-complete-android-splash-screen-guide-c7db82bce565

6.6 Adding the App Icon

https://dev.to/sfarias051/how-to-create-adaptive-icons-for-android-using-android-studio-459h

6.7 Adding ellipsis to long strings

https://stackoverflow.com/questions/6393487/how-can-i-show-ellipses-on-my-textview-if-it-is-greater-than-the-1-line

6.8 Adding a button to ActionBar

https://developer.android.com/training/appbar/actions

https://stackoverflow.com/questions/12070744/add-back-button-to-action-bar

https://stackoverflow.com/questions/34110565/how-to-add-back-button-on-actionbar-in-android-studio

6.9 Implementing a Recycler view in android

https://developer.android.com/guide/topics/ui/layout/recyclerview

https://stackoverflow.com/a/40217754

https://abhiandroid.com/materialdesign/recyclerview-gridview.html

6.10 Adding Toasts

https://stackoverflow.com/questions/3500197/how-to-display-toast-in-android

https://developer.android.com/guide/topics/ui/notifiers/toasts

6.11 Passing variables to intent

https://stackoverflow.com/questions/2405120/how-to-start-an-intent-by-passing-some-parameters-to-it

6.12 Formatting Text using HTML in TextView

https://stackoverflow.com/a/27462961

6.13 Open Link in browser

https://www.tutorialkart.com/kotlin-android/android-open-url-in-browser-activity/

6.14 Back press behavior on Back button

https://stackoverflow.com/a/27807976

6.15 Search Bar

To implement the search functionality, these pages will help:

https://www.youtube.com/watch?v=9OWmnYPX1uc

https://developer.android.com/guide/topics/search/search-dialog

6.16 To implement favorites sections

https://developer.android.com/reference/android/content/SharedPreferences

https://www.journaldev.com/9412/android-shared-preferences-example-tutorial

6.17 Implementing Dialogs

https://mkyong.com/android/android-custom-dialog-example/

https://androidexample.com/Custom_Dialog_-
_Android_Example/index.php?view=article_discription&aid=88&aaid=111

https://medium.com/@suragch/creating-a-custom-alertdialog-bae919d2e

6.18 Sectioned RecyclerView Adapter

https://github.com/luizgrp/SectionedRecyclerViewAdapter/tree/master/app/src/main/java/io/github/luizgrp/sectionedrecyclerviewadapter/demo/example1

https://github.com/luizgrp/SectionedRecyclerViewAdapter/blob/master/app/src/main/res/layout/section_ex1_header.xml

https://github.com/luizgrp/SectionedRecyclerViewAdapter/blob/master/app/src/main/res/layout/section_ex1_item.xml

https://github.com/luizgrp/SectionedRecyclerViewAdapter/blob/master/app/src/main/res/layout/fragment_ex1.xml

6.19 Using Recyclerview inside ScrollView

https://stackoverflow.com/questions/27083091/recyclerview-inside-scrollview-is-not-working

7. FILES TO SUBMIT

You should also ZIP all your source code (without image files and third-party modules) and submit the resulting ZIP file by the end of the demo day.

You will have to submit a video of your assignment demo. Details for that will be sent separately.

****IMPORTANT****

All videos are part of the homework description. All discussions and explanations on Piazza related to this homework are part of the homework description and will be accounted into grading. So please review all Piazza threads before finishing the assignment.