Project 4 Task 2 – Artwork Search App

By

Chenxu Wang (AndrewID: chenxuw),

Ruidi Chang (AndrewID: ruidic)

Description:

My application takes a search string from the user, and uses it to fetch and display information of specific artwork from the Art Institute of Chicago.

Here is how my application meets the task requirements.

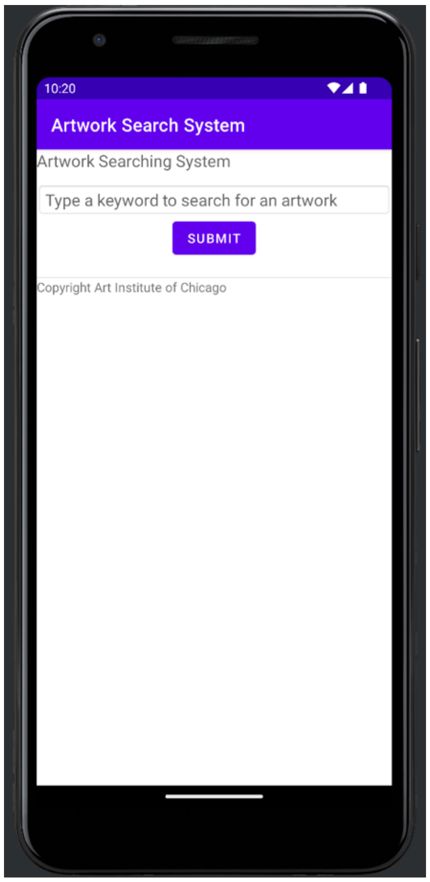
1. Implement a native Android application

The name of my native Android application project in Android Studio is: Artwork Search System

a. Has at least three different kinds of views in your Layout (TextView, EditText, ImageView, etc.)

My application uses TextView, EditText, Button, ListView, View and ImageView. See artwork.xml, search.xml and search_result.xml for details of how they are incorporated into the LinearLayout.

Here is a screenshot of the layout before the picture has been fetched.


10:20
Artwork Search System
Artwork Searching System
Type a keyword to search for an artwork
SUBMIT
Copyright Art Institute of Chicago

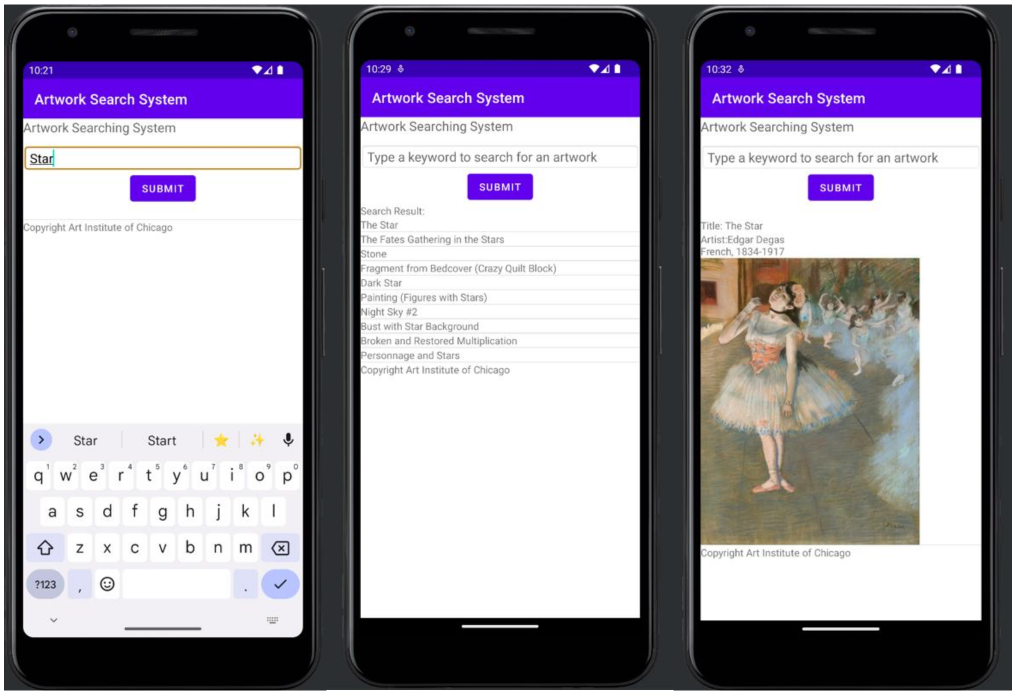
b. Requires input from the user

Here is a screenshot of the user searching for an artwork which contains a key word of “star”:

First, enter a search term (example: star) and it will show all related artworks, see figure 1 below.

Second, choose your desired artwork in the Search Result by click one result (example: The Star), see figure 2 below.

The App will then display the information about the artwork and the artist, together with the picture of the artwork, see figure 3 below.



Search Result:
The Star
The Fates Gathering in the Stars
Stone
Fragment from Bedcover (Crazy Quilt Block)
Dark Star
Painting (Figures with Stars)
Night Sky #2
Bust with Star Background
Broken and Restored Multiplication
Personnage and Stars
Copyright Art Institute of Chicago

Title: The Star
Artist:Edgar Degas
French, 1834-1917

Copyright Art Institute of Chicago

c. Makes an HTTP request (using an appropriate HTTP method) to your web service

My application does HTTP POST requests in InfoActivity.java, MainActivity.java and SearchActivity.java to the web service:

MainActivity.java and SearchActivity.java:

For search

https://polar-anchorage-77656.herokuapp.com/search

These two class makes the request for artwork search, it will send the parameters from the HTTP request to the servlet, the servlet will get the result from the API, parse to Json and select essential information. Finally, servlet will do the response to send the useful information back to the Android.

InfoActivity.java:

For content:

https://polar-anchorage-77656.herokuapp.com/content

For new search:

https://polar-anchorage-77656.herokuapp.com/search

For image:

https://www.artic.edu/iiif/2/"+image_id[0]+"/full/843,/0/default.jpg

InfoActivity class makes the request for artwork content. It will send the selected artwork id to the servlet and the servlet get the result and response the content to Android. The image of the artwork can be directly accessed from the API url. InforActivity class can also makes the request for new search, which is the same one as the precious classes.

d. Receives and parses an XML or JSON formatted reply from the web service

An example of the JSON formatted reply from the web service of searching is:

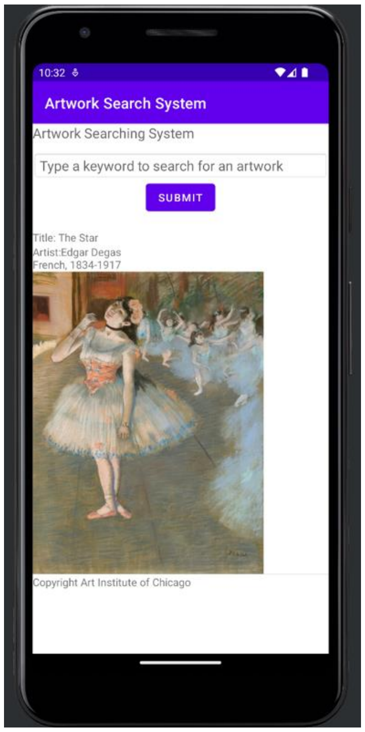
{"device":"sdk_gphone64_arm64","searchTerm":"star","thread_id":"978623"}

An example of the JSON reply formatted reply from the web service of one artwork is:

{"device":"sdk_gphone64_arm64","id":"57996","thread_id":"169070"}

e. Displays new information to the user

Here is the screen shot after the content and picture of an artwork has been returned.



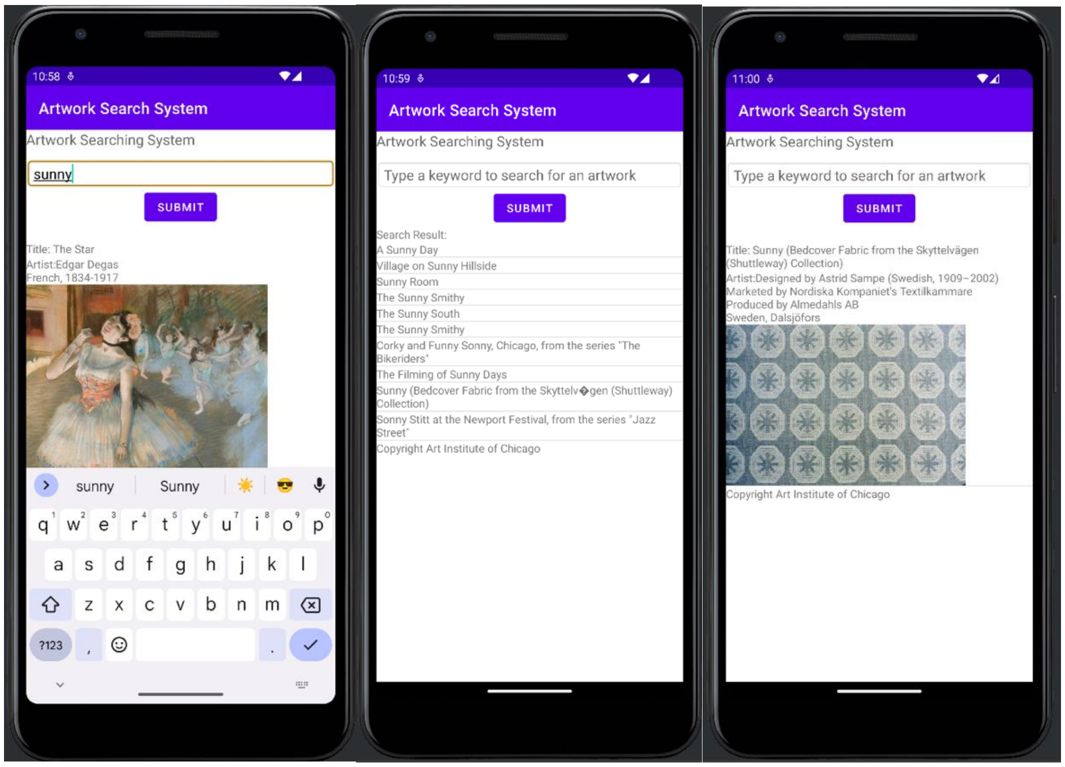
Artwork Searching System

Title: The Star
Artist:Edgar Degas
French, 1834-1917

Copyright Art Institute of Chicago

f. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

The user can type in another search term and hit Submit. Here is an example of having typed in "sunny".



Title: The Star
Artist:Edgar Degas
French, 1834-1917

Search Result:
A Sunny Day
Village on Sunny Hillside
Sunny Room
The Sunny Smithy
The Sunny South
The Sunny Smithy
Corky and Funny Sonny, Chicago, from the series "The Bikeriders"
The Filming of Sunny Days
Sunny (Bedcover Fabric from the Skyttelv�gen (Shuttleway) Collection)
Sonny Stitt at the Newport Festival, from the series "Jazz Street"
Copyright Art Institute of Chicago

Title: Sunny (Bedcover Fabric from the Skyttelvägen (Shuttleway) Collection)
Artist:Designed by Astrid Sampe (Swedish, 1909–2002)
Marketed by Nordiska Kompaniet's Textilkammare
Produced by Almedahls AB
Sweden, Dalsjöfors
Copyright Art Institute of Chicago

2. Implement a web application

a. Using an HttpServlet to implement a simple (can be a single path) API

In my web app project, the API I uses is Art Institute of Chicago.

Model: APIModel.java

Controller: SearchServlet.java and ContentServlet.java

b. Receives an HTTP request from the native Android application

The controller receives the HTTP Post request with several parameters and passes to model to get the response from API.

c. Executes business logic appropriate to your application

After user enter some words,

SearchServlet.java makes HTTP request to "https://api.artic.edu/api/v1/artworks/search?q="+input

It then passes the XML response and extracts the useful information and respond to Android.

After the user select the artwork,

ContentServlet.java makes HTTP request to

"https://api.artic.edu/api/v1/artworks/"+artworkID

It then passes the XML response and extracts the useful information and respond to Android.

d. Replies to the Android application with an XML or JSON formatted response.

An example of the JSON reply to the Android application of search results is:

```
[ { "id":60656,
      "title":"The Star"},
   {"id":134050,
      "title":"Broken and Restored Multiplication"},
   { "id":74967,
      "title":"The Fates Gathering in the Stars"},
   {  "id":182391,
      "title":"Dark Star"},
   …
   { "id":11791,
      "title":"Fragment from Bedcover (Crazy Quilt Block)"}]
```

An example of the JSON reply to the Android application of one artwork is:

```
{ "id":60656,
   "title":"The Star",
   "artist":"Hilaire Germain Edgar Degas",
   "image_id":"0850f3ab-a29d-acc7-0d3b-0551ceeea5ed",
   "place":"France",
   "category":"null",
   "artist_display":"Edgar Degas\nFrench, 1834-1917"}
```

3. Handle error conditions - Does not need to be documented.

4. Log useful information - Itemize what information you log and why you chose it.

SearchLog collection record the log during the searching phase. It records: device (user’s device), search_term (user entered search word), req (request time), res (response time), thread_id (an unique id to mark one activity), num (the amount of found search results). Since the API response result containing too many lines of artwork, I did not record it to the log.

SelectLog collection records the log that user select specific artwork. It records: device (user's device), req (request time), res (response time), thread_id (an unique id to mark one activity) from the Android request; and artwork: artist (artist name), id (artwork id), title (artwork name), image_id (id of artwork image), place (artwork origin), category (artwork category) from API response.

From these two logs, I analyzed the average latency in both request, number of visits in both request, the top 3 artwork category that user frequently select.

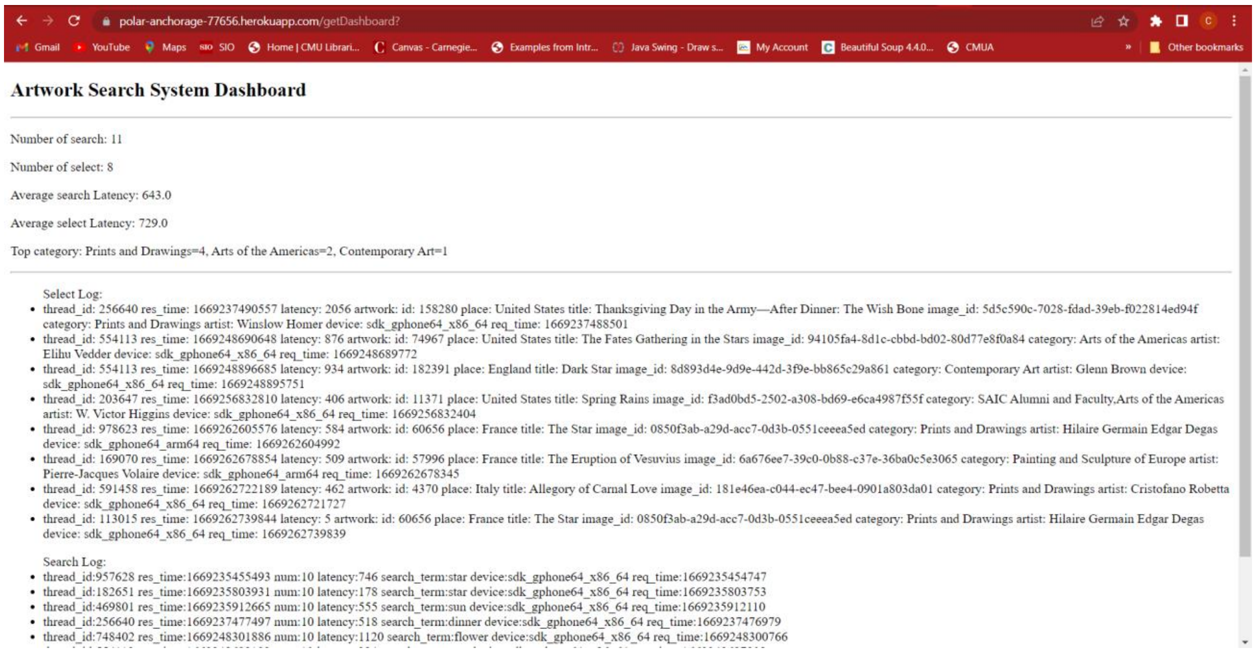
5. Store the log information in a database - Give your Atlas connection string with the three shards

"mongodb+srv://chenxuw:BSeE0xZelWVQpm6Q@project4.9ifyzrv.mongodb.net/?retryWrites=true&w=majority"

The database model is stored in DBModel.java, which contains insert and select action.

6. Display operations analytics and full logs on a web-based dashboard - Provide a screen shot.

The controller of the web is DashboardServlet.java, the model is DBModel.java, the view is dashboard.jsp.



Artwork Search System Dashboard

Number of search: 11

Number of select: 8

Average search Latency: 643.0

Average select Latency: 729.0

Top category: Prints and Drawings=4, Arts of the Americas=2, Contemporary Art=1

```
Select Log:
• thread_id: 256640 res_time: 1669237490557 latency: 2056 artwork: id: 158280 place: United States title: Thanksgiving Day in the Army—After Dinner: The Wish Bone image_id: 5d5c590c-7028-fdad-39eb-f022814ed94f
  category: Prints and Drawings artist: Winslow Homer device: sdk_gphone64_x86_64 req_time: 1669237488501
• thread_id: 554113 res_time: 1669248690648 latency: 876 artwork: id: 74967 place: United States title: The Fates Gathering in the Stars image_id: 94105fa4-8d1c-cbbd-bd02-80d77e8f0a84 category: Arts of the Americas artist:
  Elihu Vedder device: sdk_gphone64_x86_64 req_time: 1669248689772
• thread_id: 554113 res_time: 1669248896685 latency: 934 artwork: id: 182391 place: England title: Dark Star image_id: 8d893d4e-9d9e-442d-3f9e-bb865c29a861 category: Contemporary Art artist: Glenn Brown device:
  sdk_gphone64_x86_64 req_time: 1669248895751
• thread_id: 203647 res_time: 1669256832810 latency: 406 artwork: id: 11371 place: United States title: Spring Rains image_id: f3ad0bd5-2502-a308-bd69-e6ca4987f55f category: SAIC Alumni and Faculty,Arts of the Americas
  artist: W. Victor Higgins device: sdk_gphone64_x86_64 req_time: 1669256832404
• thread_id: 978623 res_time: 1669262605576 latency: 584 artwork: id: 60656 place: France title: The Star image_id: 0850f3ab-a29d-acc7-0d3b-0551ceeea5ed category: Prints and Drawings artist: Hilaire Germain Edgar Degas
  device: sdk_gphone64_arm64 req_time: 1669262604992
• thread_id: 169070 res_time: 1669262678854 latency: 509 artwork: id: 57996 place: France title: The Eruption of Vesuvius image_id: 6a676ee7-39c0-0b88-c37e-36ba0c5e3065 category: Painting and Sculpture of Europe artist:
  Pierre-Jacques Volaire device: sdk_gphone64_arm64 req_time: 1669262678345
• thread_id: 591458 res_time: 1669262722189 latency: 462 artwork: id: 4370 place: Italy title: Allegory of Carnal Love image_id: 181e46ea-c044-ec47-bee4-0901a803da01 category: Prints and Drawings artist: Cristofano Robetta
  device: sdk_gphone64_x86_64 req_time: 1669262721727
• thread_id: 113015 res_time: 1669262739844 latency: 5 artwork: id: 60656 place: France title: The Star image_id: 0850f3ab-a29d-acc7-0d3b-0551ceeea5ed category: Prints and Drawings artist: Hilaire Germain Edgar Degas
  device: sdk_gphone64_x86_64 req_time: 1669262739839

Search Log:
• thread_id:957628 res_time:1669235455493 num:10 latency:746 search_term:star device:sdk_gphone64_x86_64 req_time:1669235454747
• thread_id:182651 res_time:1669235803931 num:10 latency:178 search_term:star device:sdk_gphone64_x86_64 req_time:1669235803753
• thread_id:469801 res_time:1669235912665 num:10 latency:555 search_term:sun device:sdk_gphone64_x86_64 req_time:1669235912110
• thread_id:256640 res_time:1669237477497 num:10 latency:518 search_term:dinner device:sdk_gphone64_x86_64 req_time:1669237476979
• thread_id:748402 res_time:1669248301886 num:10 latency:1120 search_term:flower device:sdk_gphone64_x86_64 req_time:1669248300766
```

7. Deploy the web service to Heroku

The URL of my web service deployed to Heroku is:
https://polar-anchorage-77656.herokuapp.com/