
- Description
- Intended User
- Features
- User Interface Mocks
  - Screen 1
  - Screen 2
- Key Considerations
    - How will your app handle data persistence?
    - Describe any corner cases in the UX.
    - Describe any libraries you'll be using and share your reasoning for including them.
    - Describe how you will implement Google Play Services.
- Next Steps: Required Tasks
  - Task 1: Project Setup
  - Task 2: Implement UI for Each Activity and Fragment
  - Task 3: Search
  - Task 4: Admob
  - Task 5: Widget
  - Task 6: Analytics

GitHub Username: mpao

Florence's Architectures

Description

Explore the most famous '900 architectures of the city of Florence through the Open Data provided by the Municipality of Florence. Read their story to get ready to see them live

Intended User

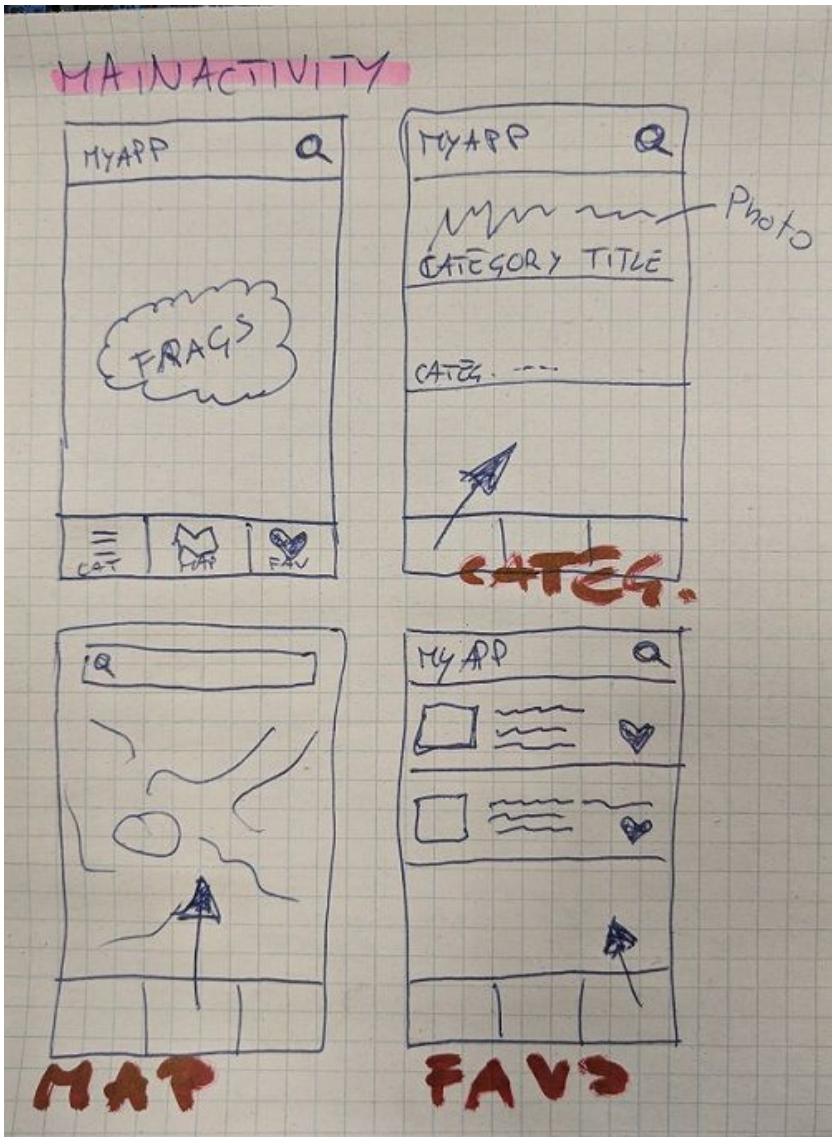
Resident citizens and tourists on holiday.

Features

- App is written in Java (required)
- List of architectures categories
- Map with placeholder and your position
- Add your favorites buildings
- Search a building in the list
- Read the information about a building, see some photo about it
- Widget w/ a random building information
- All images will have a content description
- Texts support RTL/RTR
- Application strings are stored in string.xml for ENG/ITA language support
- **Buildings Data are only in italian (the API endpoint is only in italian)**
- App theme extends AppCompat
- App support vectorDrawables
- App implements an IntentService for pull data from the API when first installed and on “swipe to refresh” gesture

User Interface Mocks

Screen 1

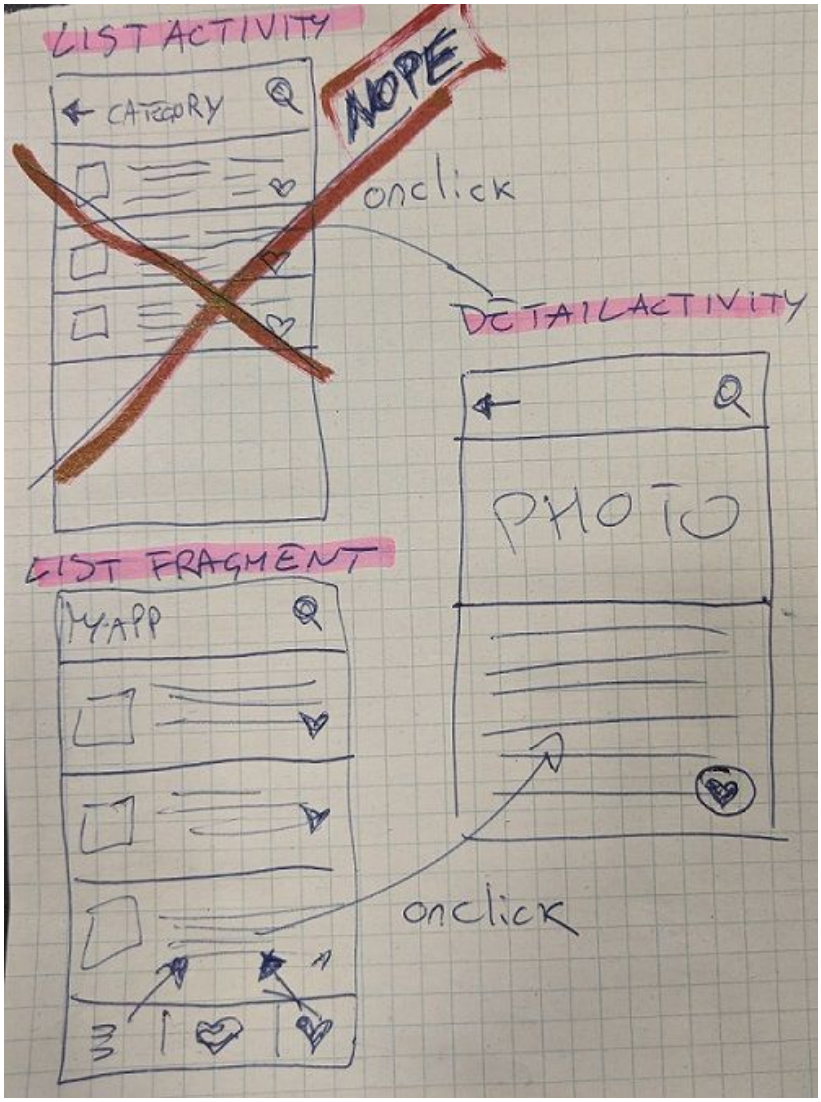


MainActivity is composed by a bottom navigation bar for three different fragments:

- Categories
- Map
- Favorites

The Categories fragment is the “Home” one

Screen 2



Clicking on a building element will open the DetailActivity with the building’s information.
If there are more than one, i.e. from favs or a category, a list fragment will be proposed

Screen 3

Building's name
Lorem ipsum dolor sit amet, et pri prima consetetur theophrastus, eu brute discere veritus vel...

5x2 Widget with a random building: name and a brief description will be shown
When the widget is clicked, the tull detail activity for that building will be shown

Key Considerations

How will your app handle data persistence?

I will use sqlite w/ a content provider.
The content provider provides:

- A list of categories of buildings
- A list of buildings from a category
- A list of all the buildings
- A list of favorites buildings
- A single building with his attributes

Describe any edge or corner cases in the UX.

I will use a bottom navigation bar w/ 3 fragments:

- One for a categories list
- Map
- Favorites list

NB: as said in https://stackoverflow.com/a/44190200/1588252 (yes it's an answer of mine) I'll prefer that, on the back press, the app returns to the home view (categories fragment) before exiting.

When the user click on a element representing a building, either from the list, from the map or from the favorites, a detail activity will open with the building information and the opportunity to add that element to the favorites list.
Clicking on a category, will open a fragment with the list of building in the MainActivity. This view will be reused for the favs fragment

NAVIGATION:
MainActivity(Categories) -> [LIST] -> DetailActivity
MainActivity(Maps) -> DetailActivity
MainActivity(Favs) -> [LIST] -> DetailActivity
Search -> [LIST] -> DetailActivity

Using a fragment for the list I can declare DetailActivity as child of MainActivity

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

I will use

- minSdk 21, targetSDK 27
- Android Studio 3.1.2+
- Support Libs 27.1.0
- Gradle 4.4+
- Picasso 2.71828+
- Dagger2 2.14.1+
- Retrofit 2.3.0+

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

I use the Map service and the location of the user, these are the main and featured services.
The location will serve to see the user's position relative to the elements and doesn't have any other feature.
So, as second service, I will use Admob.

If I don’t find it difficult I would like to try analytics, to see which buildings are the most clicked

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and break them down into tangible technical tasks that you can complete one at a time until you have a finished app.

Task 1: Project Setup

Prepare the system architecture w/ AAC
Create the service for pulling data and save them in the database
Prepare the database and the content provider

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity w/ bottom navigation
- Build UI for Categories fragment w/ recyclerview
- Build UI for a List Fragment w/ recyclerview
- Build UI for Map fragment
- Build UI for Favorites fragment
- Build UI for the Detail activity

Task 3: Search

- Create a view to search a building

Task 4: Admob

- Implement Admob service

Task 5: Widget

- Implement widget

Task 6: Analytics

- Implement Analytics service

Submission Instructions

- After you've completed all the sections, download this document as a PDF [File → Download as PDF]
    - Make sure the PDF is named "**Capstone_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"