GitHub Username: anitagold

Restaurant Guide

Description

When traveling abroad or to an unusual place, it is difficult to decide where and what to eat. Which is a good restaurant, which one is not? Which one is too expensive? What kind of restaurants are nearby? Can I find italian, japanese or vegan restaurant nearby? Which one is a tourist trap? Where the local people go to eat?
My app will help decide.

Intended User

Travelers or anybody who is in an unusual place.

Features

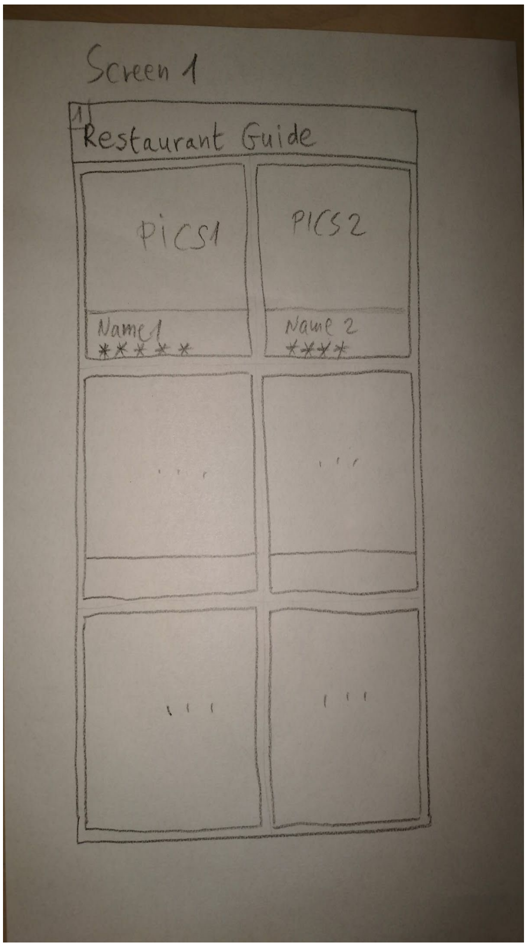
Main features of my app:

-

- Lists the restaurants and other places nearby, where you can find something to eat or drink.
- Order the list by review score to make the decision easier
- You can set the distance where you want to receive restaurants
- Show detailed data for each place (pictures, name, overall review score, address, phone, open hours, and reviews with users who gave it and why
- FAB button on detail page to share it
- widget for the list of contact info the nearby restaurant
- App is written solely in the Java Programming Language
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- App includes content descriptions
- App uses Google Maps
- App uses Google Adwords
- App uses IntentService to fetch data

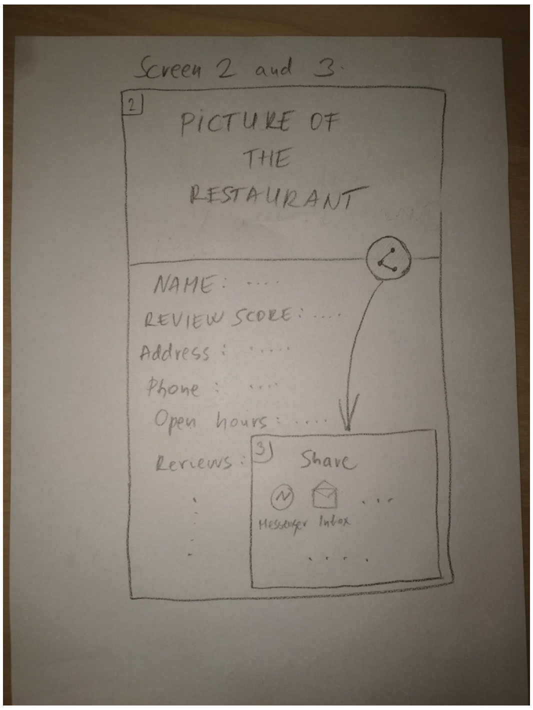
User Interface Mocks

Screen 1



List of the places. Activity with scrollable layout and cards to show the nearby restaurants.

Screen 2

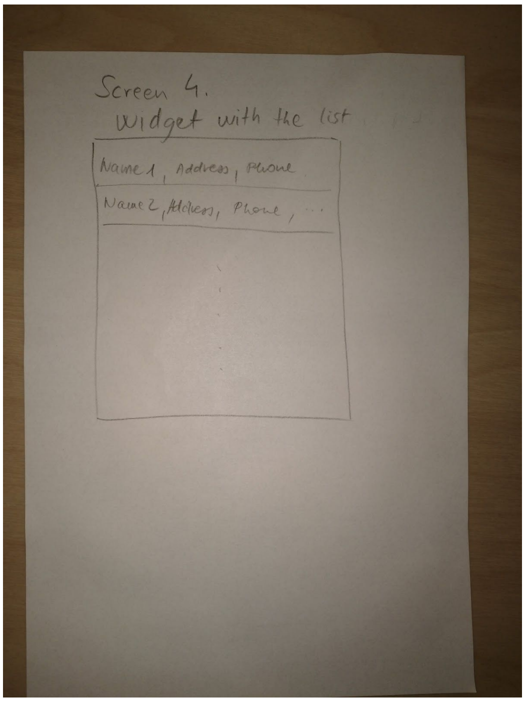


Detail page for detailed information about the selected restaurant, like address, review score, phone, opening hours and reviews. There is a FAB for share the info of this restaurant,

Screen 3

Share screen. See it above together with screen 2. It lists the possible apps you can share the info with.

Screen 4



Widget with the list of the best restaurants nearby.
It contains the contact info: Name, address, phone

Key Considerations

How will your app handle data persistence?

I build a Content Provider and store the data in SQLite database because the volume of the data is not too big. If you choose a restaurant don't want to see more than 100 result and browse terabytes of data. SQLite is good for this small and easy task, doesn't need to analyse the data in the server. It is fast and free. The data of the restaurants is quite static, doesn't change much. The data is more or less well organized so relational database suit well. I think this is better for this task than Firebase Realtime database.

Describe any edge or corner cases in the UX.

Use cards to make the list of restaurants beautiful and uniform appearance.
Prepare more restaurant so use scrollable views.
At the detail page prepare if not all data was given.
The detail page should be scrollable too.
Use material design because it is modern looking and well designed.

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

- Support libraries like AppcompatActivity to support multiple API versions
- Google maps APIs (Places and Location) to get the nearby restaurants
- Glide to handle the loading and caching of images.
- SQLite to store the data locally
- Google Play Services for LocationServices and API availability
- Retrofit to get the data (REST client)
- Butterknife for the code looks better and easier to read

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

I will use LocationServices from Google Play Services. If the connection to the Google Play services is successful we can get the current location of the user (latitude and longitude GPS coordinates). We also need to check the appropriate permission was granted to check the location.

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

- Plan the app
- Figure out the necessary components (libraries, activities, etc)
- Configure libraries
- figure out the database
- make sure every component would work together properly
- get the necessary API key

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity
- Build UI for DetailActivity

Task 3: Implement Google Play Services

- use LocationServices from Google Play Services to get the location.
- check whether the connection to the Google Play services is successful
- check if the appropriate permission was granted

Task 4: Create Retrofit REST client

- Define the Retrofit interface
- Configure Retrofit converter and adapters
- Test the result

Task 5: Create SQLite database

- Create the contracts of the SQLite
- Create the DbHelper and Provider

Task 6: Create the widget

- Create a simple widget for the app

Task 7: Create tests

- Create the usual tests whether it works properly

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**"