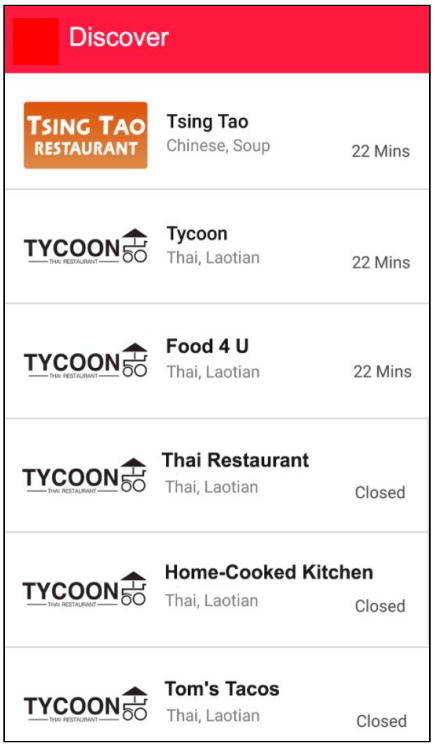
Android Take-Home Project V2.0

Background

At DoorDash, we allow customers to order delivery from their favorite local restaurants. For this exercise, you will be building a ‘lite’ version of DoorDash. Users will be able to view a list of stores with basic store details.

We expect this project to take between 2-4 hours depending on your skill level. Once complete please upload the project to a github repository and provide a link to the repository to the recruiter you’re working with.

Requirements



1. Users should be able to browse a list of restaurants near DoorDash HQ (37.422740, -122.139956)
2. What you create should resemble, if not exactly match, the mock to the right.
3. The list, images and text should display and work properly on a variety of screen sizes.
4. There should be unit tests for the core functionality you create.
5. The code should be well-architected and modular, you will be making additions to this project throughout the interview process so make sure you have something you can work well with.

Other Guidance/Suggestions

- Check the API Spec below for necessary endpoint information.
- Feel free to use open source libraries (Retrofit, ButterKnife, RxJava, etc.) as you see fit.
- Using Java is fine but Kotlin is highly encouraged as well.
- Don’t worry about supporting older Android versions. All our testing will be performed using API level 26 emulators or higher.
- Make sure you provide a reasonable level of layering/architecture for core logic and moving data around the app. We frown on having lots of logic/operations in Activity or Fragment classes.
- If you have strong skills or expertise with a given pattern, library, or approach, we want to see it. Feel free to use this as an opportunity to show off your skills.

How we'll grade your project

Grading your project will come down to 3 core areas.

1. **Functionality**
   We'll test that the app loads quickly and displays the list of restaurants properly per the mock above. This will include scrolling up and down quickly and loading the app on a variety of emulators. Make sure your app renders the images and controls properly on a variety of screens.
2. **Architecture**
   We'll review the approach you used for layering/structuring your code and how modular/reusable the components were. Any common patterns like MVP, MVVM, or something similar will be more than sufficient. Overloading view-level classes with logic or operations will be viewed in a negative way.
3. **Testing**
   We'd like to see a small example demonstrating that you know how to write unit tests and unit-testable code. 5-8+ unit tests that verify some of the functionality you created should be more than sufficient.

You should treat what you create as if it's a production app and provide that level of quality for the above areas.

Extra Credit (100% not required)

While absolutely not required you should feel free to enhance or add additional features to the project to show off your skills or show what you're capable of. **This is 100% not required** but if you feel compelled to add something extra we'd love to see it.

When You're Done

Upload the project to a shared github repository and provide a link for it to the recruiter. If you're unable to use github, zip up the entire project directory and provide that to the recruiter.

Thank you in advance for your hard work!

API Spec

Restaurant list view:
/v2/restaurant/?lat=<LAT>&lng=<LNG>&offset=0&limit=50

- GET returns a list of ***Restaurant*** objects
- Required params
  - lat
  - lng

    - offset
    - limit
- ***Restaurant*** object key fields:
    - id
    - name
    - description - *type of food*
    - cover_img_url - *restaurant thumbnail url*
    - status - *string representing status of the restaurant ("30 mins", "closed", etc.)*
    - delivery_fee - *in cents*
- Example: https://api.doordash.com/v2/restaurant/?lat=37.422740&lng=-122.139956

Restaurant detail view:
Note: We only add this here because a lot of candidates ended up figuring out this endpoint anyway and added a details page as extra credit. Using this endpoint is not required to complete the core assignment.
/v2/restaurant/<restaurant_id>/

- GET returns a single ***Restaurant*** object
- Example: https://api.doordash.com/v2/restaurant/30/