
GitHub Username: redatawfik

Doctor Finder

Description

The "Doctor Finder" app help you locate the best doctor near you. You can view doctor profile which display all information about doctor including education, rating, location, and contact information,...etc.

Intended User

App is intended for users looking for nearest and best doctor in various specializations in USA.

Features

- Search for doctors by city,medical speciality, gender or doctor name.
- View details of the doctor's profile and qualifications.

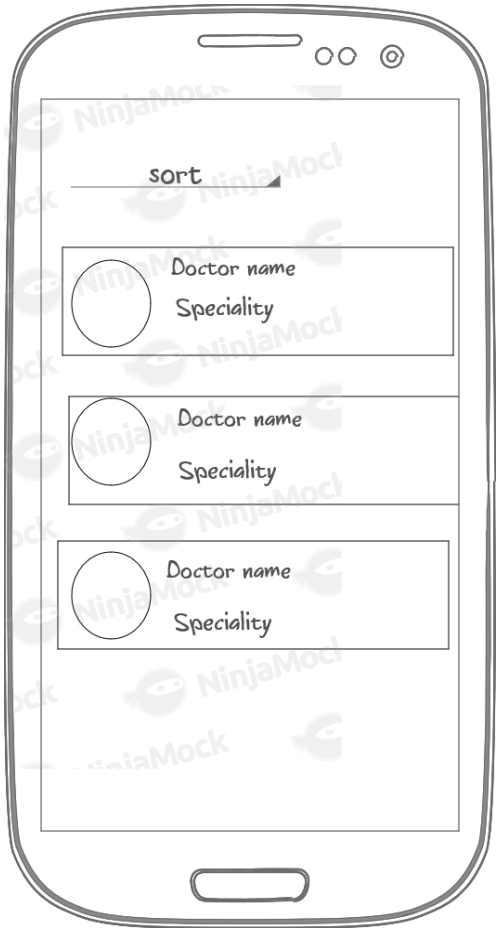
- View practical location information and clinic or hospital address.
- View phone numbers and email address.
- Navigate using the integrated Google Map and directions.

User Interface Mocks

Screen 1

Search Activity : Activity to setup request link and preferences including gender , location ,name , condition and speciality.

Screen 2



Search results activity: contain list of doctors card each card contain doctor name,image,and speciality
Change sort order according to name,best matches, rating and distance.
Each cars lead to doctor profile.

Screen 3

Doctor profile Activity.

Screen 4

App widget shows Doctor’s image, Name, Specialty, location and Landline Number

Key Considerations

App is written in the Java Programming Language

How will your app handle data persistence?

App uses Room Persistence Library and new Android Architecture :viewModel and Livedata to store favorite doctor profile in local SQLite database.

Describe any edge or corner cases in the UX.

App starts on user favorite doctor which loaded from local database and contain search button start search activity to setup search preferences . After searching list of doctor displayed when item clicked doctor profile Activity opened contain all doctor information.

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

- Android Studio version : 3.0.1
- Gradle version : 4.1
- Picasso v:2.5.2 : handle API request and caching of image.
- ButterKnife v:8.8.1 : find and automatically cast the corresponding view in your layout
- Retrofit v:2.4.0 : for handling HTTP API Connection to Better Doctor Database.
- Gson v:2.8.0 : convert a JSON string to an Java object.
- Room Persistence ,ViewModel and LiveData version 1.1.1 to save user favorite doctor for offline work.

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

- Google Location API: to pick user location . version :15.0.1
- Google places API : we implement that in search activity to let user select search area version :15.0.1
- Google Maps API: we implement that in profile Activity to show doctor location in the map. version :15.0.1

Next Steps: Required Tasks

Task 1: SearchActivity Setup:

Activity Elements:

1. EditeText -Query-: For Doctor's name or patient condition.
2. Button -Location-: when clicked opens Google places using places service to let user select search area. -> location saved in member variable.
3. Button -Specialty-: when clicked searchable spinner will be Displayed contain list of doctor's specialities which saved in xml file before parsing it to Arraylist of Specialty object which passed to Searchable spinner. ->specialty saved in member variable.
4. ButtonGroup -Gender- : to let user choose Doctor's gender (male,female,no preference)and save it im member variable.

5. Button -Search-: put Query, Location, Specialty and Gender in Intent Extra and start SearchResultsActivity.

Task 2: SearchResultsActivity Setup:

- Init class member variable(Query, SearchArea, Userlocation , Specialty and Gender from intent extra.
- Connect to HTTP API using Retrofit passing this member variables above as parameter.
- Request return json object parsing to list of doctors which passing to adapter of recyclerview in SearchResultsActivity .
- Setup models for json object.

Task 3: ProfileActivity Setup:

- Build UI for Profile Activity.
- Show doctor's location in MapView using Map Services.
- Buttons(zip code and fax number) show dialog message with the number.
- Button (call) send doctor's Landline number in implicit intent.
- Like button save doctor object in database using Room Library for offline work.

Task 4: App widget implementation:

Ui:

1. TextView for Doctor's name
2. TextView for Doctor's specialty
3. TextView for Doctor's Location
4. Button to call doctor

Widget is Updated with the last row in database which is the last liked doctor.

Task 5: Room Persistence Library implementation:

1- **Dao: Contains the methods used for accessing the database which are:**

1. **Method loading all rows in database to show them in SavedDoctorActivity.**

2. **Method insert doctor in database and called in ProfileActivirt when like button clicked**
3. **Method delete doctor from database when like button disabled**
4. **Method returning last row and pass it for Widget class.**

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**"