How to Use this Template

1. Create a new document, and copy and paste the text from this template into your new document [Select All → Copy → Paste into new document]
2. Name your document file: "**Capstone_Stage1**"
3. Replace the text in green

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 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.

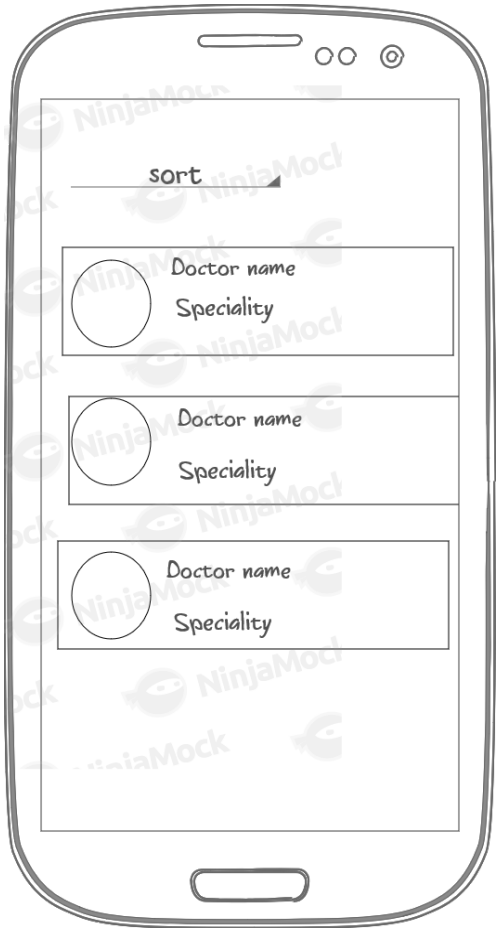
User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Google Drawings, www.ninjamock.com, Paper by 53, Photoshop or Balsamiq.

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.

Doctor profile Activity.

Add as many screens as you need to portray your app's UI flow.

Key Considerations

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.

Picasso: handle API request and caching of image.
Room Persistence Library ,ViewModel and LiveData for Local database.

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

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

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

Setup Search Activity: in this activity i will setup request link which return list of doctor objects according to user preferences in this Activity.
I will use retrofit for handling API request.

Subtasks:

- Build UI for Search Activity.
- Pick search area location from google service location
- Setup doctor's speciality to let user choose from.
- Make a request and return list of doctors object

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Search Result Activity.
- Create card for each result
- Implement Radio button collection to choose sort order.
- Each card lead to doctor's profile .

Task 3: Your Next Task

- Build UI for Profile Activity.
- Implement Google Places Services to show doctor's location on the map

Task 4: Your Next Task

- Implement Room Library to store favorite doctor in local database

Add as many tasks as you need to complete your app.

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**"