GitHub Username: hanansabry

Meet Your Doctor

Description

Meet your Doctor is an app for anyone who searches for best doctor around him in any specialist (in the selected city) as it includes large database of all doctors in most common specialists and their details like fees, clinic times, clinic address and useful information about the doctor.
It also provides you with real reviews from other patients to help the user find the best doctor that he looking for.

Intended User

The app is for all patients (in the selected city) who looking for best doctor in any specialist

Features

Main features of the app:

- Search and find doctors around you based on doctor specialist and area.
- Save specific doctor as favourite to easily find him next time.
- Write reviews about doctors to help other patients find the best doctor for them.
- Call the doctor's clinic directly from the app to book appointment.
- Support Arabic and English Languages.
- Push notifications every time new doctor is added to the database or when any update of doctor's information occured.

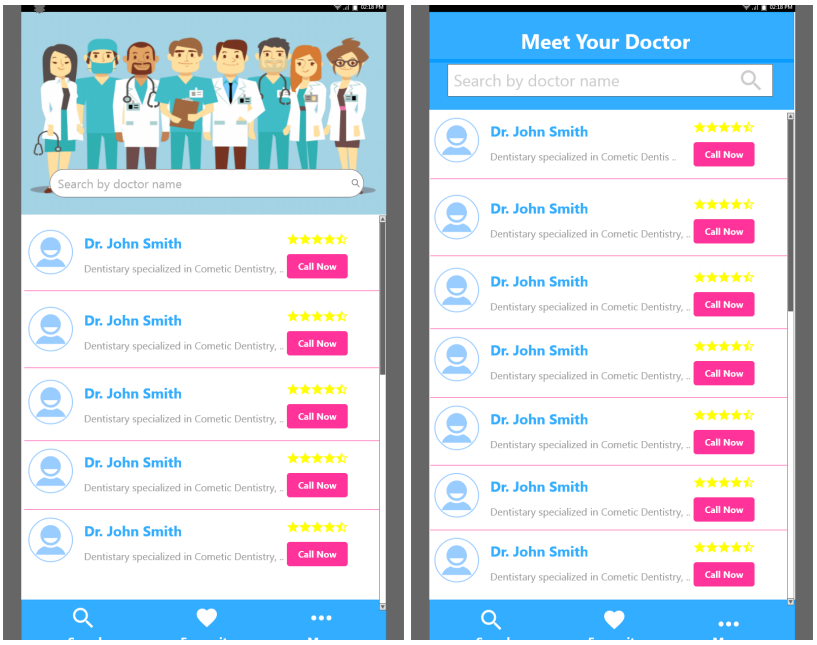
User Interface Mocks

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.

Home-Search Screen

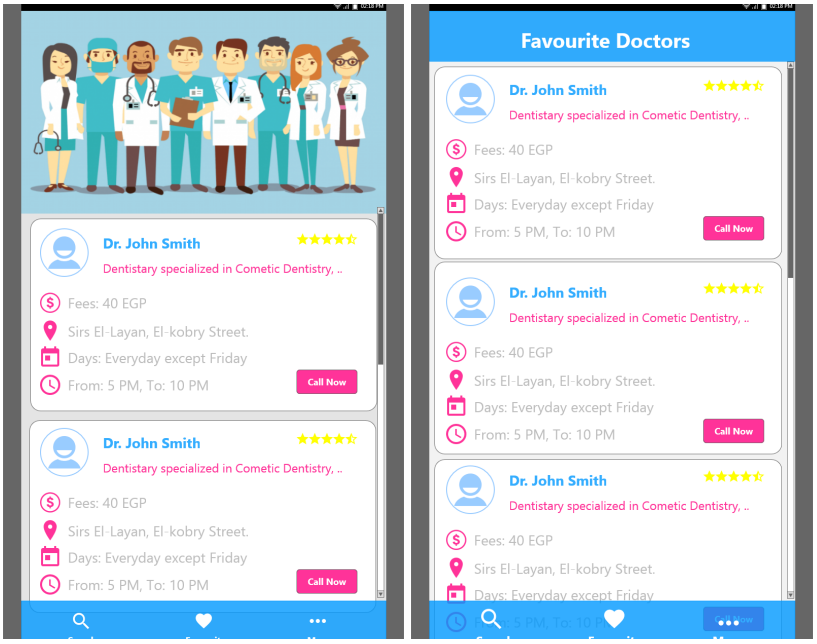
- The main screen of the app "Search Screen" (Expanded and Collapsed) that enables the user to search for doctor by name in any specialist using "Search bar". And bottom navigation to navigate between main screens "Search, Favourite Doctors, and More options"
- The main content contains the most common specialists that organized in recyclerview.

Doctors List Screen



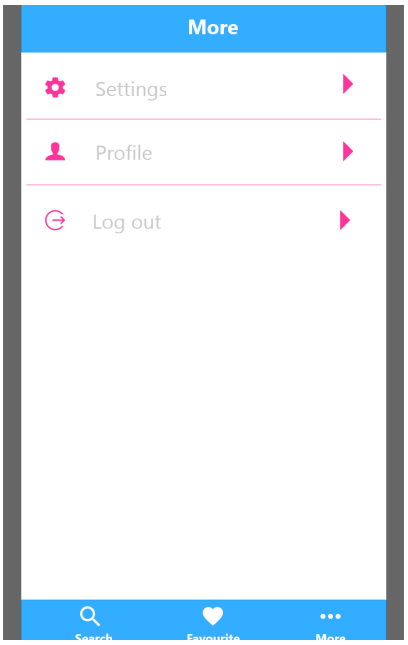
- When the use selected any specialist from the previous screen, the app shows all doctors in this specialist by their ratings and the ability to call clinic directly to book an appointment. Also search by doctor name in this list of doctors.
- The screen is shown in expanded and collapsed mode.

Favourites Screen



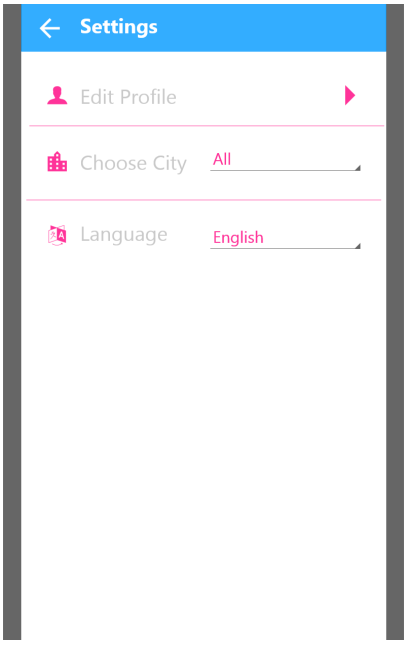
- Favourite doctors of the user and each item contains details about every doctor as shown in the image.
- The screen is shown in expanded and collapsed mode.

More Screen



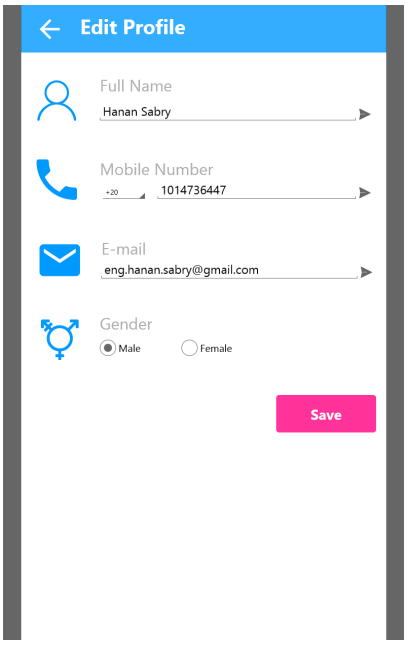
- More Screen contains more options to the user (Settings, show user profile, and logout)

Settings Screen



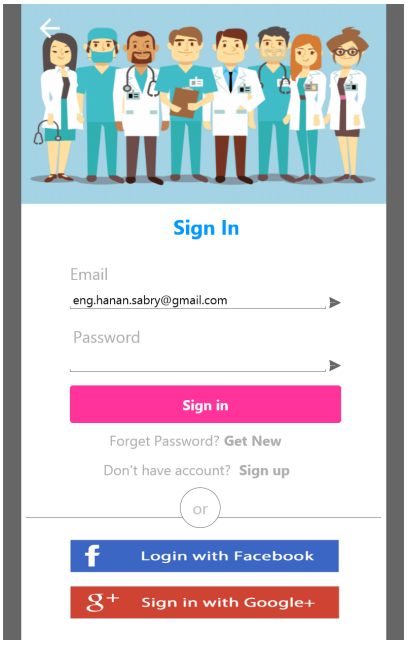
- Settings screen enables the user to edit his profile, change the city where he wants to search for doctors and choose the application language.

Edit Profile Screen



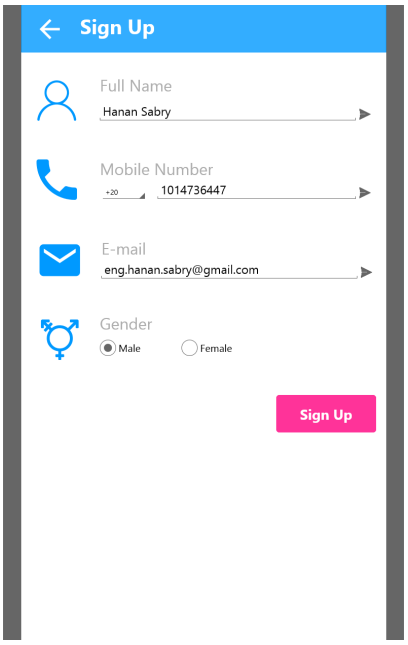
- Edit Profile Screen enables the user to edit his data.

Login Screen



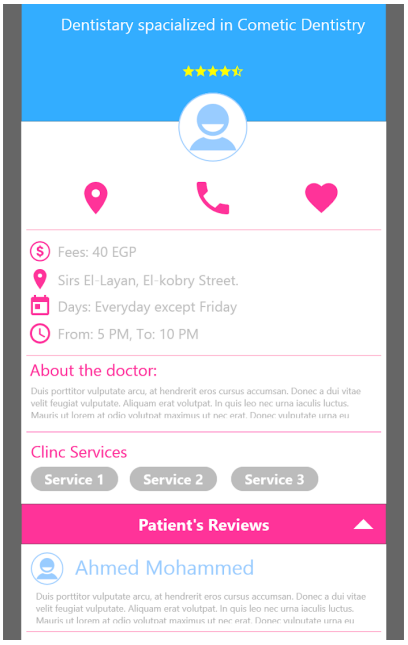
- User can make profile for the app by signing up with email or login with facebook or Google account.

Sign Up Screen



- Signup screen using email.

Doctor Profile Screen



- Profile of the doctor that contains all details about the doctor and the clinic and the patients' reviews and fab that enables the user to write review of the doctor.

Widget

- Widget of the app will contains most common specialists and by selecting any of them, it moves the user to the Doctors List screen for this specialist in the app.

Key Considerations

How will your app handle data persistence?

- Using Firebase Realtime Database to store all data of the doctors.
- Using SQlite database with content provider to store favourites doctors.

Describe any edge or corner cases in the UX.

- App handle temporary network interruptions by enable disk persistence in the Firebase API. So cached data is available while offline and resends any write network connections is restored.

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

- Glide: to handle the loading and caching of images.
- Moshi: for json parsing.
- Butter Knife: for boilerplate code reducing.
- Retrofit: for network requests.
- Android Support Library.
- Android Design Library.
- Google Services.

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

- Firebase Realtime Database to handle data persistence
- Firebase Authentication to handle both for username/email/password and for social authentication
- Firebase Cloud Messaging for push notification to the user every time any updated happened to the database.
- Google maps to view doctor's clinic address on Google Maps.

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

- Create new project in Android Studio.
- Configure all libraries by adding all necessary dependencies.

Task 2: Collect and Design required icons and Images that will be used in the application with all needed resolutions.

Task 2: Implement UI for Each Activity and Fragment

- Implement the bottom navigation which be part of the three main screens "Search, Favourite, More".
- Implement UI for Main screens which will contains of App bar, Content Fragment and bottom navigation.
- Implement two fragments layout to represent the search screen. One for most common specialists and the other for list of the doctors of selected specialist.
    - Design specilaist_item layout for specialists Recyclerview.
    - Design doctor_item layout for doctors list Recyclerview.
- Implement doctor_favourite_item layout for "favourites doctors" Recyclerview which will be the content of the fragment that represents the Favorites Screen.
- Implement settings fragment layout.
- Implement UI for "Doctor Profile Screen", "Login Screen", "Signup Screen", "Edit Profile Screen".
- Implement UI for the widget.

Task 3: Data model classes

- Create data classes which handle all data of the application

Task 4: Using Dummy data

- Start fill the layout with dummy data to ensure that everything is working okay before using real data.

Task 5: Setup Firebase

- Create project using firebase console.
- Link firebase project with the android studio.
    - In the project gradle file, add the required Google Services plugin.
    - In the app gradle file, apply Google Services plugin.
    - Add the dependencies of required features of Firebase (FCM, Realtime database, Authentication).

Task 6: Architect the backend of Firebase realtime database

- List out the data that firebase will store and deliver to the app.

Task 7: Sign up for an API key to use Google Play Services.

Task 8: Populate application with real data

- Configure firebase realtime database to populate layout with real data and replace old dummy data

Task 9: Configure content provider for store favourite list of doctors.