

Snap Car

Technical Documentation

Project Description:

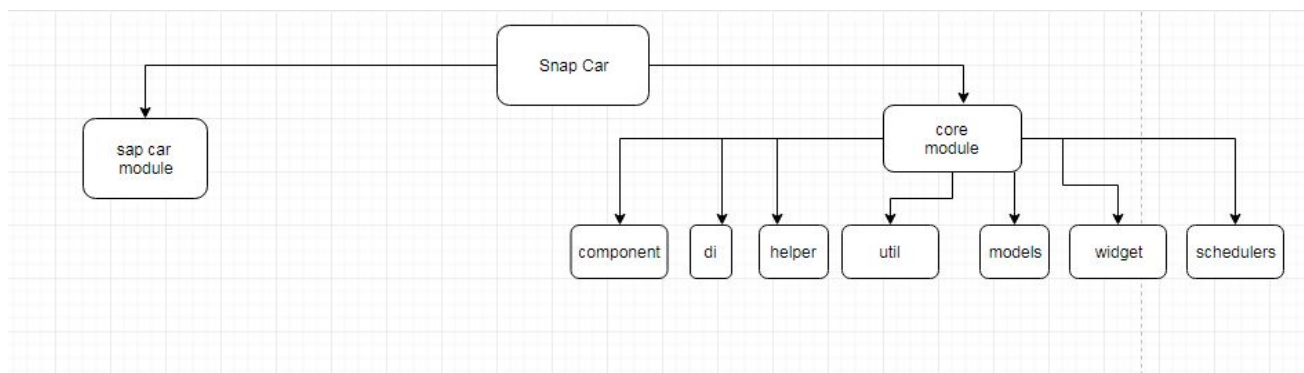
- Snap car project has two types of users “showroom ” & “individual user”.
- Individual and Showroom users can add cars and make at least one car tracked.
- Showrooms can also add more details about their location and cars.
- Also all users can edit , delete and mark their car as sold.
- Also they can edit and view their profile.
- Individual and Showroom users can add their interests and update it to receive notifications when matched car publish.
- Showrooms can also request to be vip or hot zone.
- Users can see blocked list, payment history, discount and promotions, Also send feedback.

- Setting menu containing change password , about us, contact us and change language.
- Notifications setting for chat, matching cars and offer .

Used Libraries:

- [AppCompat, Design support libraries \(Default android components\)](#)
- [Retrofit \(Networking\)](#)
- [Glide \(Image Processing\)](#)
- [Google Play Services \(Location services\)](#)
- [Firebase \(Push Notifications,Realtime database,Crashlytics \)](#)
- [ButterKnife \(Binding Views\)](#)
- [Rxjava & Rxandroid \(Reactive Extensions\)](#)
- [Dagger2 \(dependency injection\).](#)

Project Structure :



1 - Core module :

1.1 : Component : - **Activity** : contains the base component like activity, presenter and screen that is used to handle the common functions between activities.

1.2 : Di :

- **Activity** : contains the activity component and activity module used for dependency injection.

- **Qualifier** : is used for distinguishing between similar objects during a dependency injection.

- **Scope**: annotations identifies the scope level to be used for during a dependency injection.

1.3 : helper :

- **WebSocket** : contains classes that is help to deal with socket .
- Also contains enhanced built in functions and classes.

1.4 : Models : contains the reusable data classes like chat response , model and viewModel.

1.5 : Schedules : contains classes that can deals with android threads like main and worker threads.

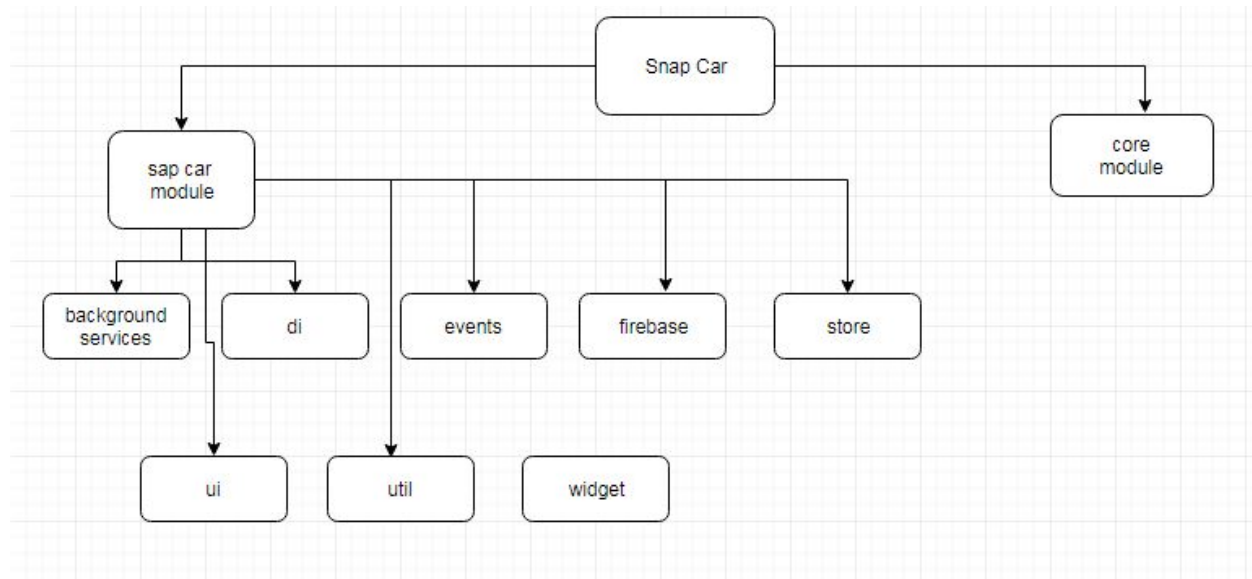
- **Qualifiers** : is used for distinguishing between differents threads during a dependency injection.

1.6 : Social Login Helper : contains all reusable social media that used for login and register.

1.7 : util :contains all helper classes to deal with different resources like shared preference , text , map .. ,etc.

1.8 : Widget : contains all reusable custom views like dialogs ,editText and bottom sheet.

1.9 : Chat : contains all configurations that used to create chat component in project.



2 - Snap car module :

2.1 : Background services : contains classes that used for create job dispatcher services.

- **Di** : contains the service component and module that used during dependency injection.

2.1 : Di:

- **Activity** : contains activity components and modules.
- **Application** : contains components and modules that used for application dependency injections.
- **Fragment** : contains fragment components and modules.
- **Room** : contains the room db initialization.
- **service** : contains components and module for service and service qualifier.

2.2 : Events : contains all interfaces that used to fire events.

2.3 : Firebase : contains classes that used for push notifications.

2.4 : Store :

- **Api** : contains retrofit classes like reusable endpoints, services, and api utils that used for create retrofit and contain the base url.
- **Model** : contains all data classes like response and model and view model and mapper class that is used for change class like from model to viewModel.
- Also contains repos classes that control data source if caching is exist.
- Also contains web service class that contains all project services .

2.5 : Ui : contains all project activities and fragments divided into packages each package contains

- 1 - **activity** that is extend the core activity and controls activity view ,
- 2 - **presenter** that extend the core presenter and used for manage data from and to views , deals with networks calls and validate inputs data.
- 3 - **screen** that extend the core screen and used to fire events in views.
- 4 - **adapters** classes that is used for inflate data in recyclerview.
- 5 - **activity Args** a class that is used to navigate between activity to another activity.

2.6 : Util : contains project utils that used to deals with different resources like views ,files, and text.

2.7 : widget : contains all custom views classes.

Project activities :

1 : splash activity :

- Description : the first screen and contains the application logo.
- Layout : activity_splash.xml.
- Java Package : splash.

2 : Tutorials activity :

- Description : the second screen that helps user to understand the application and user can see it only the first time when application is installed.
- Layout : activity_tutorial.xml.
- Java Package : tutorial.
-

3 : Login activity :

- Description : This screen allows the user to login with his account.
- Layout : activity_login.xml.
- Java Package : login.
- API : “user/auth/signin” , type -> POST, params , params -> [name, password, os, mobile_token]

4 : Forgot password activity :

- Description : This screen allows the user to get reset his password .
- Layout : activity_forget_password.xml.
- Java Package : forgetpassword.
- API : “user/auth/password/email” , type -> POST, params -> [name, reset_method]

5 : Activation activity :

- Description : This screen allows to activate his account or reset password by checking the validation code if correct .
- Layout : activity_activation.xml.
- Java Package : activation.
- API : 1- resend code "user/auth/valid-phone" , type -> POST, params -> [phone, reset_method]

6 : Reset Password activity :

- Description : This screen allows to type and confirm his new password .
- Layout : activity_reset_password.xml.
- Java Package : resetpassword.
- API : 1- resend code "user/auth/active-account" , type -> POST, params -> [phone, reset_password_code, password, reset_method, os, mobile_token]

7 : Registration step one activity :

- Description : This screen allows to type his phone .
- Layout : activity_phase_one_registration.xml.
- Java Package : phaseoneregistration.
- API : 1- resend code "user/auth/valid-phonet" , type -> POST, params -> [phone]
2 - country list "country_list " , type -> GET

8 : Registration step two activity :

- Description : This screen allows to complete his profile.
- Layout : activity_phase_two_registration.xml.
- Java Package : phasetworegistration.

- API : 1- resend code “user/auth/signup” , type -> POST, params -> [name, area, type, email, phone, firebase_uid, os, country_id, city_id, password, mobile_token]
- 2 - country list “country_list ” , type -> GET

9 : Nearby fragment :

- Description : This screen to display nearby cars , showrooms and hot zones on map .
- Layout : fragment_explorer.xml.
- Java Package : explorer.
- API : 1- filter “filter-map” , type -> POST, params -> [api_token, car_model_id , brand_id, category_id, show_all, men_car, woman_car, damaged_car, vip_showrom, showroom, vip_hotzone, hotzone, longitude, latitude]
- 2 - make favorite “favorite ” , type -> POST params - > [api_token, car_id, user_id]

10 : Explore fragment :

- Description : This screen to explore cars and showrooms that not on map and ads.
- Layout : fragment_nearby.xml.
- Java Package : nearby.
- API : 1- filter “filter-explore” , type -> POST, params -> [api_token, car_model_id , brand_id, category_id, show_all, men_car, woman_car, damaged_car, vip_showrom, showroom, vip_hotzone, hotzone, longitude, latitude]
- 2 - make favorite “favorite ” , type -> POST params - > [api_token, car_id, user_id]
- 3 - brands “interest/create ” , type ->GET params - > [api_token]

11 : profile fragment :

- Description : This screen allows user to view his profile and make requests like be vip or join hotzone.
- Layout : fragment_profile.xml.
- Java Package : profile.
- API : 1- profile "get-profile" , type -> GET, params -> [api_token]
2 - request vip "request/vip " , type -> POST params - > [api_token]
3 - request hotzone "request/hotzone " , type ->GET params - > [api_token]
4 - request ads "request/ads " , type ->GET params - > [api_token]
5 - logout "logout " , type ->POST params - > [api_token]

12 : contact us activity :

- Description : This screen allows user to contact us.
- Layout : activity_contact_us.xml.
- Java Package : contactus.
- API : 1- settings "settings" , type -> GET, params -> []
2 - contact us "contact-us " , type -> POST params - > [name, email, message]

13 : Feedback activity :

- Description : This screen allows user to send his feedback.
- Layout : activity_feedback.xml.
- Java Package : feedback.
- API : 1- settings "settings" , type -> GET, params -> []
2 - subjects "feedback/create" , type -> GET params - > [api_token] ,
3 - post feedback "feedback" , type -> PARTparams - > [api_token,feedback_category_id, details, attachment, subject]

14 : Notification Settings activity :

- Description : This screen allows user to control notification settings.
- Layout : activity_notification_setting.xml.
- Java Package : notificationsetting.
- API : 1- notification “users/setting” , type -> POST, params -> [api_token, notification_setting]

15 : discount and promotions activity :

- Description : This screen allows user to explore discounts and promotions.
- Layout : activity_coupon.xml.
- Java Package : coupon
- API : 1- coupons“coupons” , type -> GET, params -> [api_token]

16 : discount fragment :

- Description : This screen allows user to explore discounts.
- Layout : fragmet_discount.xml.
- Java Package : discount

17 : hotzone fragment :

- Description : This screen allows user to explore hotzone.
- Layout : fragmet_hotzone.xml.
- Java Package : hotzone

18 : Payment history activity :

- Description : This screen allows user to check his payment history.

- Layout : activity_payment_history.xml.
- Java Package : paymenthistory
- API : 1- history “car/payment-history” , type -> POST, params -> [api_tokeng]

19 : My interest activity :

- Description : This screen allows user to see his intertest.
- Layout : activity_my_interestes.xml.
- Java Package : myinterests
- API : 1- interest “interest” , type -> GET, params -> [api_tokeng, page]
2- remove interest “interest/{interest_id}” , type -> DELETE, params -> [api_token, interest_id]

20 : Add interest activity :

- Description : This screen allows user to add new interest.
- Layout : activity_add_interest.xml.
- Java Package : addinterest
- API : 1- interest “interest/create” , type -> GET, params -> [api_tokeng] 2- add interest “interest” , type -> POST, params -> [api_token, gear_type, specification_id, method_of_sale_id, car_color_id, mvpi, brand_id, car_model_id, price_from, price_to, year_from, year_to, car_condition, seller_type, owner_type, nearby, city_id, country_id, kilometer_from, kilometer_to, category_id, vehicle_registration, big_sale_id]

21 : Edit interest activity :

- Description : This screen allows user to edit his interest.
- Layout : activity_edit_interest.xml.
- Java Package : editinterest

- API : 1- interest “interest/{interest_id}” , type -> PATCH, params -> [api_tokeng] 2- add interest “interest” , type -> POST, params -> [api_token, gear_type, specification_id, method_of_sale_id, car_color_id, mvpi, brand_id, car_model_id, price_from, price_to, year_from, year_to, car_condition, seller_type, owner_type, nearby, city_id, country_id, kilometer_from, kilometer_to, category_id, vehicle_registration, big_sale_id]

22 : My cars activity :

- Description : This screen allows user see his cars.
- Layout : activity_my_cars.xml.
- Java Package : mycars
- API : 1- car “car” , type -> GET, params -> [api_tokeng, page] 2- seavr data “settings” , type -> GET, params -> [] 3- remove car “car/{id}” , type -> GET, params -> [id, delete_reason, api_token]
-

23 : Edit profile activity :

- Description : This screen allows user edit his profile.
- Layout : activity_edit_profile.xml.
- Java Package : editprofile
- API : 1- update “update-profile” , type-> PATCH, params -> [api_token, name,email, country_id, city_id, area, image]

24 : Edit phone activity :

- Description : This screen allows user edit his phone.
- Layout : activity_edit_phone.xml.
- Java Package : editphone
- API : 1- car “request-update-phone” , type-> POST, params -> [api_token, phone]

25 : Edit showroom phone activity :

- Description : This screen allows user edit his phone.
- Layout : activity_edit_showroom_phone.xml.
- Java Package : editshowroomphone
- API : 1- update“request-update-phone” , type-> POST, params -> [api_token, phone, phones]

26 : Edit showroom location activity :

- Description : This screen allows user edit his location.
- Layout : activity_edit_showroom_location.xml.
- Java Package : editshowroomphonelocation
- API : 1- update“update-location” ,type-> PATCH, params -> [api_token, country_id, city_id, area, longitude, latitude]

27 : Favorites fragment :

- Description : This screen allows user see his favorites.
- Layout : fragment_favorites.xml.
- Java Package : favorites
- API : 1- favorites “favorites” ,type-> GET, params -> [api_token, page]
- 2- delete or add car “favorites” ,type-> POST, params -> [api_token, car_id]
- 2- delete or add user “favorites” ,type-> POST, params -> [api_token, user_id]

28 : Add car activity:

- Description : This screen allows user to add new car.
- Layout : activity_add_car.xml.
- Java Package : addcar
- API : 1- requirement “car/create” ,type-> GET, params -> [api_token] ,
- 2- seavr data “settings” , type -> GET, params -> []

, 3 - add "car" , type -> POST, params -> [api_token, contact_option, post_type, is_tracked, car-status, manufacturing_year, car_specification_id, transmission, car_color_id, kilometer, under_warranty, warranty, mvpi, notes, car_model_id, price_before, price_after, engine_capacity_cc, images, installment_price_from, installment_price_to, exchange, longitude, latitude, examination_image, category_id, brand_id, price, vehicle_registration, price_type]

29 : Edit car activity:

- Description : This screen allows user to edit his car.
- Layout : activity_edit_car.xml.
- Java Package : editcar
- API : 1- requirement "car/create" ,type-> GET, params -> [api_token] ,
2- seavr data "settings" , type -> GET, params -> []
3 - update "car/{id}" , type -> PATCH, params -> [api_token, contact_option, post_type, is_tracked, car-status, manufacturing_year, car_specification_id, transmission, car_color_id, kilometer, under_warranty, warranty, mvpi, notes, car_model_id, price_before, price_after, engine_capacity_cc, images, installment_price_from, installment_price_to, exchange, longitude, latitude, examination_image, category_id, brand_id, price, vehicle_registration, price_type]

30 : View car activity:

- Description : This screen allows user to view his car.
- Layout : activity_view_car.xml.
- Java Package : viewcar
- API : 1- car "car-details" ,type-> GET, params -> [api_token,car_id]

31 : Notification fragment :

- Description : This screen allows user to see his notifications.
- Layout : fragment_notification.xml.
- Java Package : notification
- API : 1- notification "notification" ,type-> GET, params -> [api_token]

32 : Survey Activity :

- Description : This screen allows user to answers survey.
- Layout : activity_survey.xml.
- Java Package : survey
- API : 1- survey “survey” ,type-> GET, params -> [api_token], 2- post survey “survey” ,type-> POST, params -> [api_token, survey_id, answer]

34 : Visitor profile Activity :

- Description : This screen allows user to see other users profiles.
- Layout : activity_visitor_profile.xml.
- Java Package : visitorprofile
- API :1 - add interest data“interest/create ” , type ->GET params - > [api_token], 2 - user“user/{user_id} ” , type ->GET params - > [api_token], 3 - rate “users/rate/{user_id} ” , type ->POST params - > [api_token, rate, note] 4 - block “block-people” , type ->POST params - > [api_token, user_id, reason] 5 - user cars “search-user” , type ->GET params - > [api_token, user_id, brand_id, model_id, category_id, year_from, year_to, price_to, price_from, longitude, latitude], 6 - chat “chat/chat/{user_id}” , type ->GET params - > [api_token, user_id]
-

35 : Verification letter Activity :

- Description : This screen allows user to upload his documentation.

- Layout : acticity_verification_letter.xml.
- Java Package : verificationletter
- API : 1- profile“get-profile” , type -> GET, params -> [api_token], 2- letter“send/verification-letter” , type -> PART , params -> [api_token, part]
-

36 : Verification letter Activity :

- Description : This screen allows user to upload his documentation.
- Layout : acticity_verification_letter.xml.
- Java Package : verificationletter
- API : 1- profile“get-profile” , type -> GET, params -> [api_token], 2- letter“send/verification-letter” , type -> PART , params -> [api_token, part]

37 : Update password Activity :

- Description : This screen allows user to update his password.
- Layout : acticity_update_password.xml.
- Java Package : updatepassword
- API : 1- update “update-password” , type -> GET, params -> [api_token, old_password, new_password]

38 : Settings Activity :

- Description : This screen allows user to browse available settings
- Layout : acticity_settings.xml.
- Java Package : settings
- API : 1- change language “language” , type -> PATCH, params -> [api_token, language]

39 : Search Activity :

- Description : This screen allows user to browse his search result

- Layout : activity_search.xml.
- Java Package : search
- API : 1- basic search “search” , type -> GET, params -> [api_token, brand_id, model_id, category_id, gender, year_from, year_to, price_from, price_to, longitude, latitude, page], 2- advanced search “search” , type -> GET, params -> [api_token, brand_id, model_id, category_id, gender, year_from, year_to, price_from, price_to, longitude, latitude,page, under_warranty, engine_capacity_cc, color_id, kilometer_from, kilometer_to, transmission, car_status, tracked, post_type, is_installment, is_big_sale], 3- make favorite “favorite ” , type -> POST params - > [api_token, car_id, user_id]

40 : Privacy Activity :

- Description : This screen allows user to accept privacy and terms conditions.
- Layout : activity_privacy.xml.
- Java Package : privacyactivity
- API : 1- settings “settings” , type -> GET, params -> [], 2- accept “users/terms” , type -> POST, params -> [api_token, terms],3 - logout “logout ” , type ->POST params - > [api_token] ,4- profile“get-profile” , type -> GET, params -> [api_token]
-
-

41 : Confirm commission Activity :

- Description : This screen allows user to confirm snap commission.
- Layout : activity_confirm_comission.xml.
- Java Package : confirmcomission
- API : 1- pay later “car/actions/skip-payment/{car_id}” , type -> PATCH, params -> [api_token,car_id]

42 : Chat thread Activity :

- Description : This screen allows user to chatting with another user.
- Layout : activity_chat_thread.xml.
- Java Package : chatthread

43 : Chat search Activity :

- Description : This screen allows user to search in his inbox.
- Layout : activity_chat_search.xml.
- Java Package : chatsearch
- API : 1- chat list "chat/chat" , type -> GET, params -> [api_token]

44 : Chat fragment :

- Description : This screen allows user to see his inbox.
- Layout : fragment_chat.xml.
- Java Package : chat
- API : 1- chat list "chat/chat" , type -> GET, params -> [api_token]

45 : Block activity :

- Description : This screen allows user to see his blocked users.
- Layout : activity_block_list.xml.
- Java Package : blocklist
- API : 1- block list "block-people" , type -> GET, params -> [api_token,page], 2- unblock "block-people/{user_id}" , type -> GET, params -> [api_token]
-

46 : Advanced search activity :

- Description : This screen allows user to search using advanced feature.
- Layout : activity_advanced_search.xml.
- Java Package : advancedsearch

- API : 1 - data for add interest "interest/create" , type -> GET params -> [api_token]

47 : About us activity :

- Description : This screen allows user to read about us.
- Layout : activity_about_us.xml.
- Java Package : about us
- API : 1- settings "settings" , type -> GET, params -> []

Base url -> <http://18.224.193.88/backend/public/>

Socket url -> <ws://18.224.193.88:8080/chat>

User url -> <api/v1/user/auth/>

App url -> <api/v1/user/app/>

Image upload url -> <api/v1/user/app/file/upload> (multipart)

Application design -> <https://sketch.cloud/s/Ragdy/>

Firebase : email -> snapcar.ksa@gmail.com

Password -> [Apb@amotorz1t](#)

Database : <http://18.224.193.88/phpmyadmin/index.php>

Username : root

Password : 1z+&KI%cA_,E

Collection : <https://www.getpostman.com/collections/d65491c2a46dbed3b2ae>