**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)](https://www.google.com.eg/search?q=support+libraries+documentation&oq=support+libraries+documentation&aqs=chrome..69i57j0l5.13423j0j7&sourceid=chrome&ie=UTF-8)

● [Retrofit (Networking)](https://square.github.io/retrofit/)

● [Glide (Image Processing)](https://github.com/bumptech/glide)

● [Google Play Services (Location services)](http://www.androiddocs.com/google/play-services/setup.html)

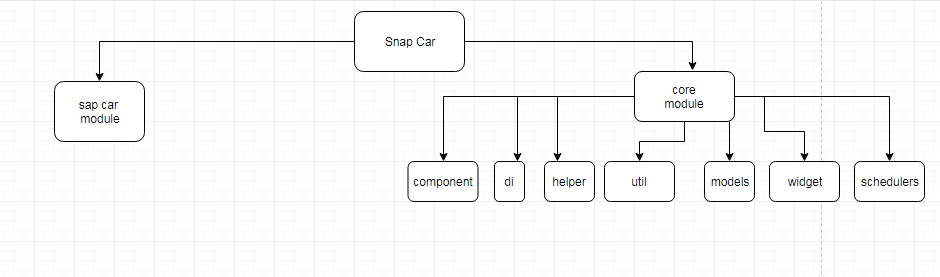
● [Firebase (Push Notifications,Realtime database,Crashlytics )](https://firebase.google.com/)

● [ButterKnife (Binding Views)](http://jakewharton.github.io/butterknife/)

● [Rxjava & Rxandroid (Reactive Extensions)](https://github.com/ReactiveX/RxJava)

● [Dagger2 (dependency injection).](https://github.com/google/dagger)

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.

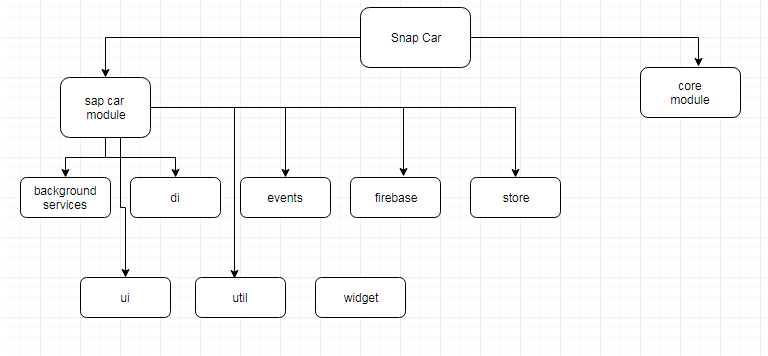
* **Qualifiers** : is used for distinguishing between differents threads during a dependency injection.

**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 reps 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 differents 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- seawr 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- seawr 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, proice\_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- seawr 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, proice\_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 : acticity\_survey.xml.
* Java Package : survey
* API : 1- survey “survey” ,type-> GET, params -> [api\_token], 2- post survey “survey” ,type-> POST, params -> [api\_token, servey\_id, answer]

**34** : **Visitor profile Activity** :

* Description : This screen allows user to see other users profiles.
* Layout : acticity\_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 : acticity\_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 : acticity\_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 commision.
* Layout : acticity\_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 : acticity\_chat\_thread.xml.
* Java Package : chatthread

**43** : **Chat search Activity** :

* Description : This screen allows user to search in his inbox.
* Layout : acticity\_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**](http://18.224.193.88/phpmyadmin/index.php)

**Username : root**

**Password : 1z+&Kl%cA\_,E**

**Collection : https://www.getpostman.com/collections/d65491c2a46dbed3b2ae**