

Detailed System Structure Models

Class Diagrams :

1. Under SVC Central Server

Figure 1, shows the classes that are part of the central server for saving the group information in the online mode of the Under android application. To provide security in terms of information sharing, we are keeping the information only on the group. The following are details of the packages and their underlying classes.

- **Package Name : api**
 - **Class : GroupResponse** – This is a “Plain Old Java Object” (POJO) that represents how a Group is serialized to/from JSON. It only contains members that are serializable to/from JSON-native data types. They are returned by resource handlers to clients such as GroupsResource.
- **Package Name: client**
 - **Class : UnderHTTPClient and UnderHTTPClientException** – UnderHTTPClient objects are HTTP clients that are employed to talk to the undersvc. Every new instantiation allows specifying an arbitrary URL for accessing data; by default this is set to <https://localhost:8080>. It also consists of methods that allow the CRED (Create, Read/Get, Edit/Update, Delete) operations on the groups. It is also used in an integration test which performs actual CRED operations against an in-memory database. UnderHTTPClientException consists of the handling of exceptions during any of these operations.
- **Package Name: core**
 - **Class: Group** - Group is an entity class that represents how a group is stored in a relational database. This is intentionally distinct from the {@link GroupResponse} class which exists only to be serialized as JSON. This is part of the hibernate configuration that is used for storing the information on the database.
 - **Class: GroupMember** – This is an entity class representing a member of a group, as stored in a backing database.
 - **Class: GroupRestriction** – This is an entity class for representing how a group restriction is stored in a backing database.
 - **Class: UserRating** - User Rating is an entity class for representing all the ratings that a user got, in the particular time frame.

- **Package Name: db**
 - **Class: GroupDAO** - GroupDAO is the data access layer for Groups. This class is responsible for performing database operations on Groups.
 - **Class: UserDAO** – This facilitates the data access for the group member of the groups.
- **Package Name: resources**
 - **Class: GroupResources** – Is a resource for handling GET (get group), POST (update group), and DELETE (delete group) requests to `/groups/{groupId}`. Other HTTP requests are not supported.
 - **Class: GroupsResource** - Is a resource for handling GET (list groups), and POST (create group) requests to `/groups`. Other HTTP methods are not supported.
 - **Class: UserRatingResource** - Is a resource for GET and POST to `/ratings/{memberId}`.
- **Package Name: util**
 - **Class: GeometryUtils** – This class holds static utility methods for geometry operations. Currently, the only method here is a static method to convert to/from a class from a 3rd-party library.
- **Class: undersvcApplication** – This is the base application for undersvc. It handles bootstrapping the application configuration from environment variables and command-line arguments, and initializing the service.
- **Class: underSvcConfiguration** – This class only exists to hold runtime configuration variables for undersvc.

Better quality image (Figure 1)- https://drive.google.com/file/d/1G4-mKUmaS08XuOP9C8LJ0-_S5gJn_fw8/view?usp=sharing



Figure 1 Class diagram for Under SVC - Central Server

2. Under Application:

Figure 2 shows the classes that are part of the main android application for ride-sharing functionality. The application is structured from the User Interface part and all the functionality of the corresponding view is called in the backend in separate classes. The following are details of the packages and their underlying classes.

Package Name: Appview

- **Class: Login** – This class has members and functions to login into the application. It allows the user to enter the user information and validate the same. If the user is verified correctly, the user is logged in to the application. This class also provides a gateway for new user to register with the application.
- **Class: Input Validation** – This class is responsible for validating the user input during login. It validates the user input with the stored database value and allows the user to login based on it.
- **Class: Signup-** This class is used to sign up users into the application which has various fields in it which needs to be entered by the user to make an account with the application. It has functions to save all the fields in the server and a local database which then allows the login class to validate and allow access to the signed-up users.
- **Class : DatabaseHelper** – This class is helped to enter the user details to the database and then used to store it in it and this is mainly called from signup page where all the user information is entered.
- **Class: UserHome** – This class is responsible for displaying the main page of the application. Once a user logs into the application, the user is displayed with the Map view which allows the user to select where the user to select the destination. This class calls all corresponding objects in separate threads to make sure the application is running as expected. Once the user selects a destination, this class is also responsible for creating or joining a group making the

user to share the ride. It also has functions and members to invoke the google nearby services to find the nearby peer to peer communication devices which constitutes the whole functionality of the application.

- **Class: GroupListFragment** – This class has functions to list all the available groups in order to make a common template for the view. This class is further used in both join and create group lists.
- **Class: GroupCreateFragement** – This class is responsible for creating a group with the input provided by the user. The inputs include the maximum capacity of the ride, preferences of the user while sharing a ride, Id of the user, destination of the ride and mode of transport etc. All this information is stored in the Central server as well to make sure the information is shared when no nearby customers are found. This information is then broadcasted via a google nearby service which allows the other user to know about a group heading towards a desired destination.
- **Class : JoinGroupList** – This class is responsible for displaying the list of groups for users to join when the join group button is pressed.
- **Class: GroupDetailFragement** – This class is responsible to list the details of the group when a user selects a join group button. This makes sure that all the information related to the group is listed in detail for a user to make his/her selection to join a group. It has members and corresponding functions to call the database and list the same.

Package Name: GroupManager

- **Class : User** - This class is responsible for holding all the members of the user details including the user location, name, gender and preferred details so as to process it as objects as part of the groups.
- **Class : groupData** – This class holds the value of the group including the start location, destination and can be used to fetch these details from the database and used in the files wherever possible.



Better quality image (Figure 2)- https://drive.google.com/file/d/1YMfmWQgTsT3zOqYou7WPaiA_LK_gBhoC/view?usp=sharing

Figure 3 represents the class diagram of the various main functionalities of the under Application. These classes are called from the android app view classes and processed in the backend to make the application function as expected. The following are the details of the package and their classes.

Package: Storage

- **Class: APIClient** – This class calls the external central server API from the application. The server instance is created and made to run which further has functions to create groups and store it.
- **Class: UserEntity** – This class holds the details of the user including his personal information to be stored in the central server.
- **Class: Store** – This class stores the user entity and the ratings details from the user into the local database.
- **Class – Cloud StorageActivity** - This class stores the user entity and the ratings details from the user into the central server.

Package : WIFI P2P

- **Class: Nearbyer** – This class is responsible for maintaining the peer to peer communication between the devices and establishing the protocol to send and receive messages. This is part of Google's nearby service open sourced

framework. It has various methods and members to make sure the communication between two devices is maintained via WIFI network in order to make the application available.

- **Class: Nearbymsg** – This is a simple class to transmit the network state message to be shared between the devices based on the communication status.

Package: Group Manager

- **Interface: GroupIO** - It is an abstraction layer over Group-related operations over Peer to Peer networks and the central server. It maintains a cache of available Groups. It also handles requests to join or leave groups in google nearby services.
- **Class: GroupIOImpl** - It wraps both APIClient class and Nearbyer class to provide a unified and consistent interface to group operations both globally and locally. Groups are fetched in a background thread which is started by calling GroupIOImplstartListening() function. It also has functions to advertise the groups through nearby service.
- **Class: Group** – This is the main group class which holds all the members of the group including the start location, destination location, list of all the users in the group and various other properties of the group like preferences of the user etc. Creating an object of this class provides all the necessary information of under group, which was used by many other classes.
- **Class: JoinRequest** - Join Request represents a request to join a group. It consists of a joiner and a group owner ID. It joins a user to group based on the preferences and the group owner's consent.

- **Class: JoinLeaveResponse** - It is responsible for handling the consent of the group owner in terms of response to join or leave a group.
- **Class: LeaveRequest** – It represents a request to leave a group. It consists of a leaver and a group owner ID.

Package: Util

- **Class:GeometryUtil** - GeometryUtil contains helper methods for reading and writing Well-Known-Text (WKT) representations of MULTIPOINT geometry strings. It converts the latitude and longitude information into strings which is used in application for processing.

Figure 3 Class diagram for Under - System and backend classes

Better quality image (Figure 3)- <https://drive.google.com/file/d/1esBy23YXMko6rWVivCeeDWxG0Kde9a5I/view?usp=sharing>