|  |  |
| --- | --- |
| **Topic** | Android Project for Training |
| **Document Name** | Solution of “Kalan Kullanımlarım” Application |
|  |  |

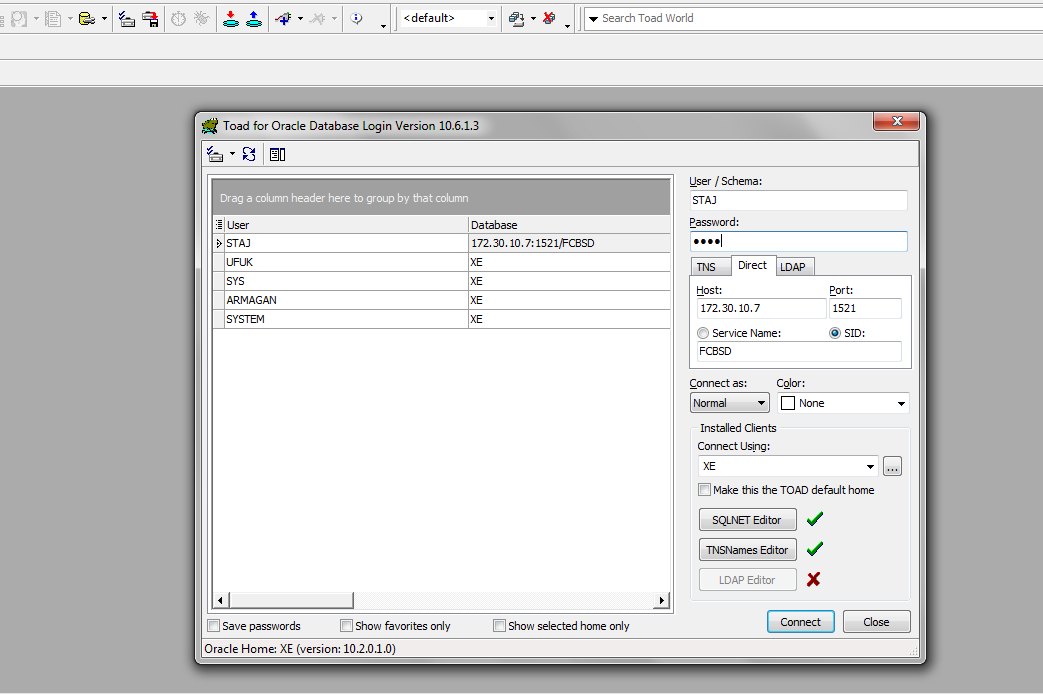
|  |  |  |  |
| --- | --- | --- | --- |
| **Document Difficulty Level** | | | |
| **Beginner** | **Junior** | **Senior** | **Expert** |
| □ | ■ | □ | □ |

# Document History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Ver | Comments |
| 27.06.2017 | Ufuk Armağan Özgür | 1.0 | Initial Draft |

## Creating a Database

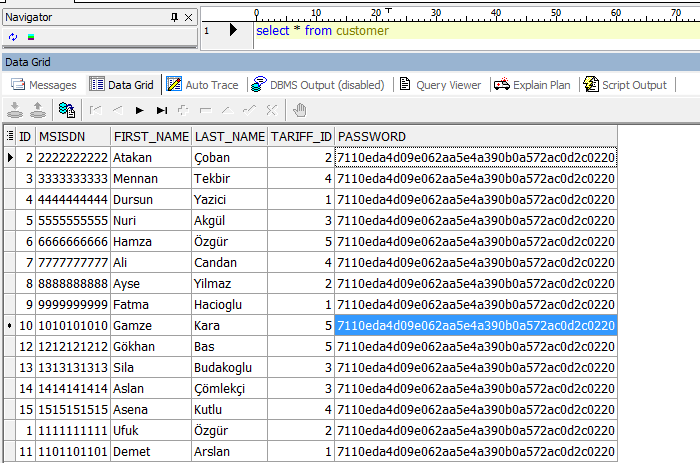
Database was created with given user account that names “STAJ” on i2i System’s server.



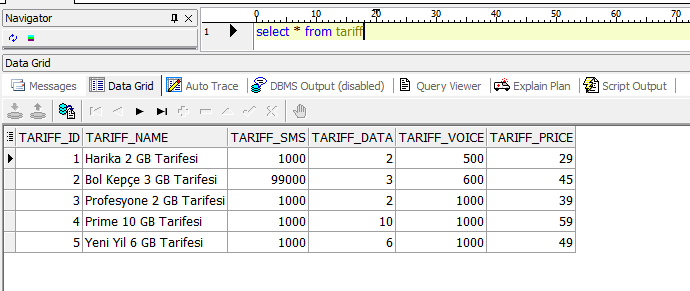
Three table was created . CUSTOMER , TARIFF , CUSTOMER\_USAGE.

* CUSTOMER table consist of personal informations about customer. There is relation with tariff table. Each customer must have one tariff , one tariff may belong to one or more customer. Additional this table consist of customer passwords for log in our application.
* TARIFF table consist of information about tariffs. Like usage rights , tariff price. This table doesnt have any relationship with other tables.
* CUSTOMER\_USAGE table consist of customer’s all usage amounts. If programmer or someone wants see customer’s usage history , these informations records in read\_date column.

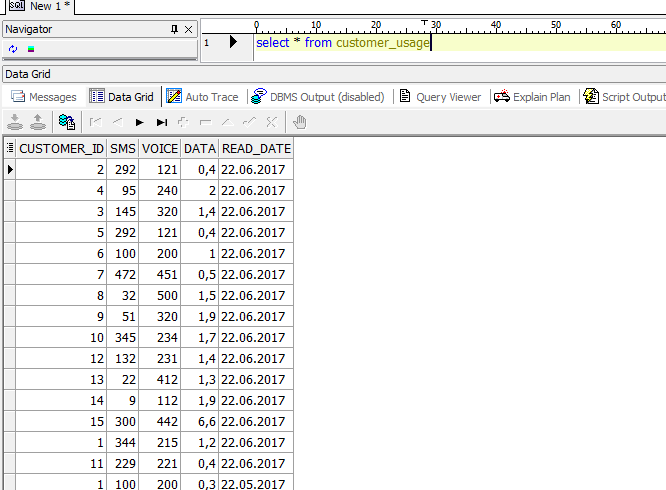
CUSTOMER table ;



TARIFF table



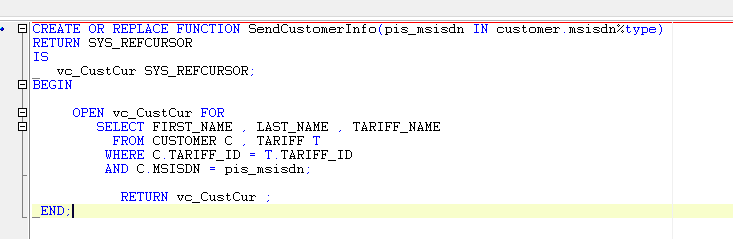
CUSTOMER\_USAGE table;



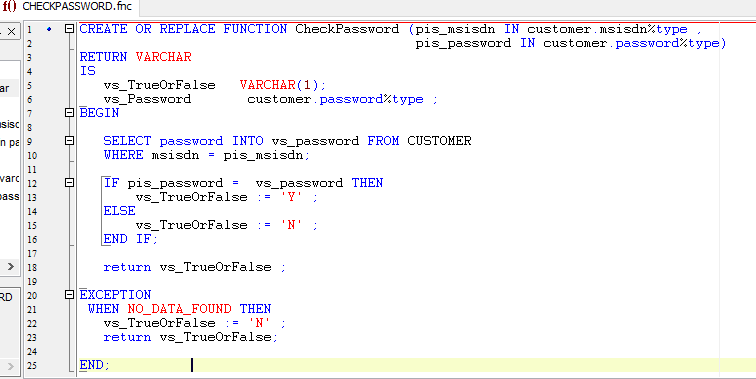
After creating tables and relations between tables , PL/SQL scripts was created.

* SENDCUSTOMERINFO function return a cursor that consist of first name , last name and tariff name of customer who has coming msisdn(phone number) parameter.
* CHECKPASSWORD fuction return just one character : Y or N. It take two parameter msisdn and password , if these parameters exist in table then return Y , if not return N.
* SENDREMAINUSAGE function return a cursor that remain usages of customer who has coming msisdn parameter. CUSTOMER\_USAGE table consist customer usage that different months. This function return most recent months’ remain usages.

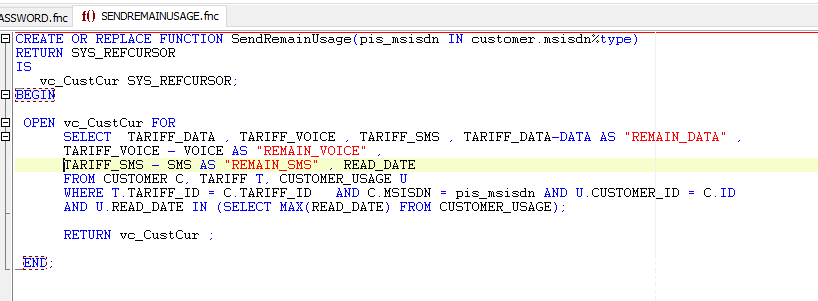
SENDCUSTOMERINFO function;



CHECKPASSWORD fuction;

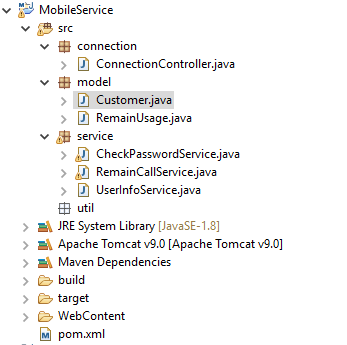


SENDREMAINUSAGE function;

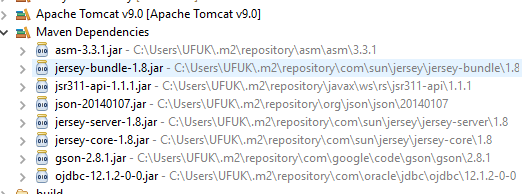


After that, written web services that run background (server side) on Java. Purpose of this services ;

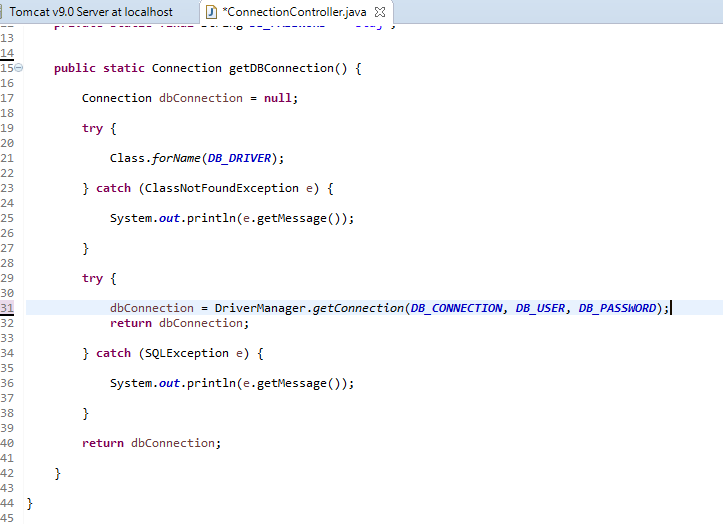
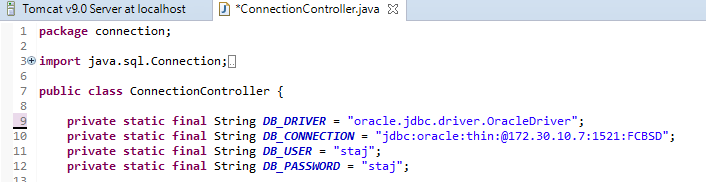
* Reach stored procedures or functions on Database and get them.
* Parse this result to json format.
* Send this json file as a response to a application (clients).



Firstly Tomcat Server intalled. Then required jar files was added to Maven Dependencies.



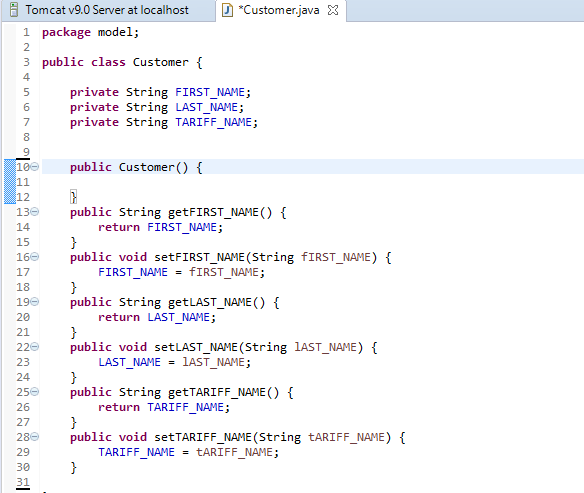
ConnectionController.java class was written. This class include connection informations. This class have a method that name is “getDBConnection()” . This method provide connection from database with connection informations. Service classes will call this function for get dbConnection.



Created models for represent values that return DB scripts. Cursors were parsed and assign these models.

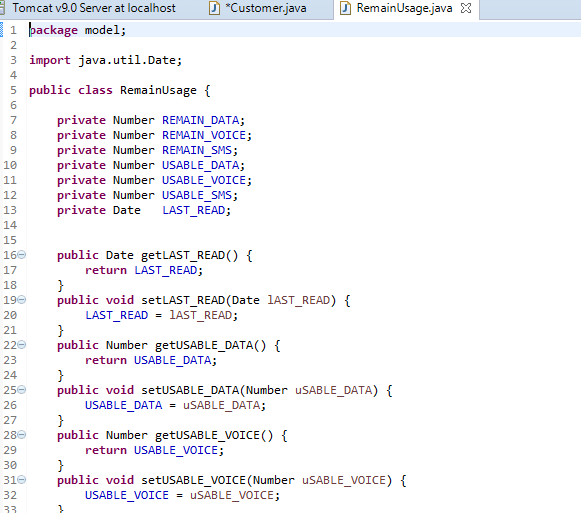
“Customer” model represent “SendCustomerInfo” PL-SQL script’s return cursor. Each column that cursor assign this model’s concern variable.

Assign processes will write in “UserInfoService” class between this model’s variable and coming cursor’s columns. That is just model.



“RemainUsage” model represent “SendRemainUsage” PL-SQL script’s return cursor. Each column that cursor assign this model’s concern variable.

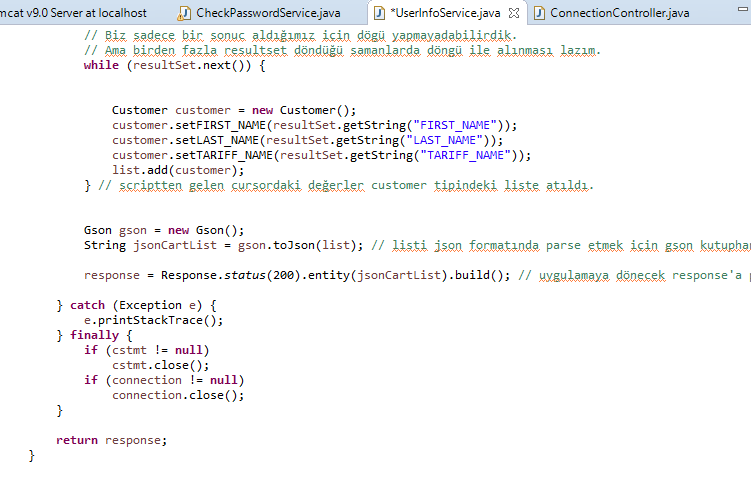
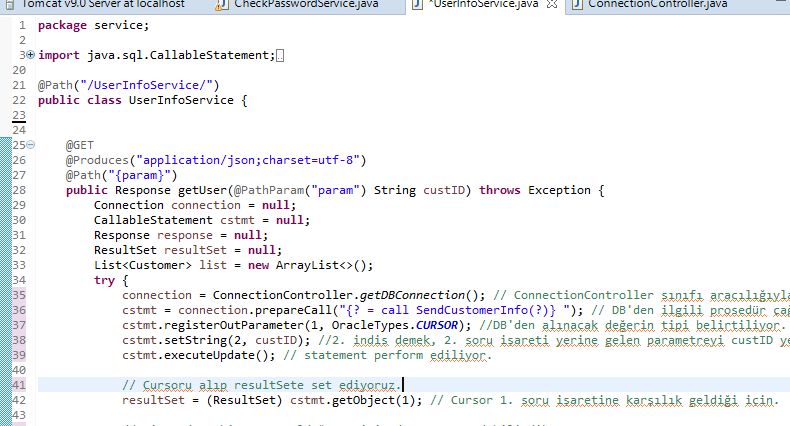
Assign processes will write in “RemainCallService” class between this model’s variable and coming cursor’s columns. That is just model.



There are three script that coming from DB. There is “CheckPassword” script also. But just two model created. There is no need for CheckPassword script. Because this script returns just one character : ‘Y’ or ‘N’. So a creating model fort his script unnecessary.

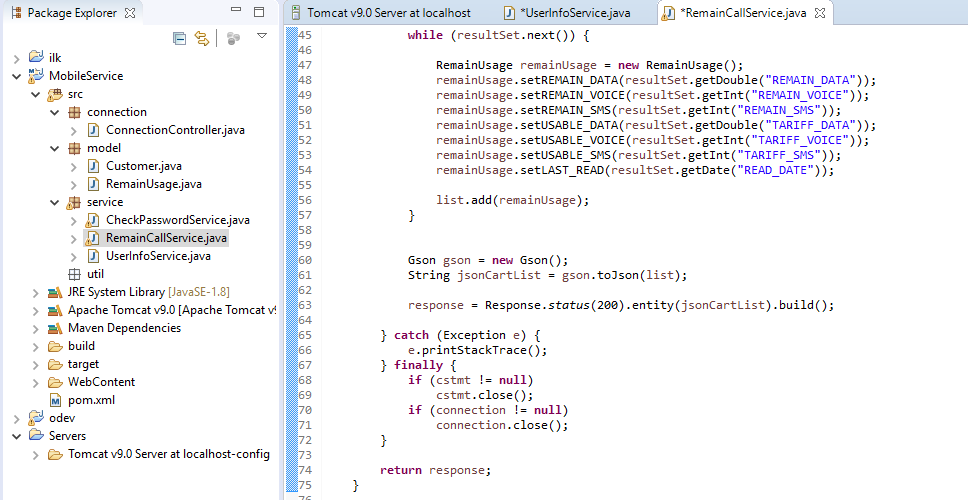
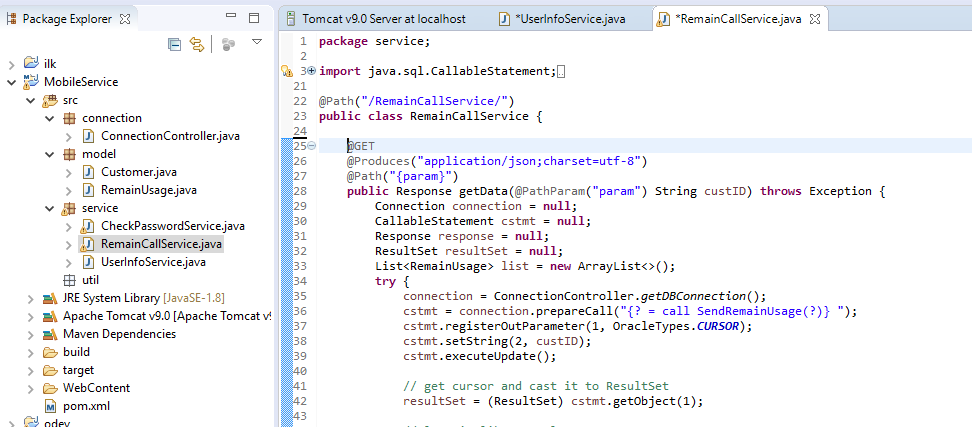
“UserInfoService” class was created. This class’s mission is get cursor that return from “SendCustomerInfo” DB function. Get cursor’s columns and convert it json format and return as response when it call.

* Firstly connection was opened using “ConnnectionController” class’s “getDBConnection” method.
* SendCustomerInfo script called within CallableStatement class.
* Cursor’s columns put inside to list have Customer model’s variable.
* This list converted to json format. This converting process did using “Gson” library’s “ToJson” method.
* Finally this json file returned as response.



“RemainCallService” class was created. This class’s mission is get cursor that return from “SendRemainUsage” DB function. Get cursor’s columns and convert it json format and return as response when it call.

The same processes was done. Just model and coming function was changed.

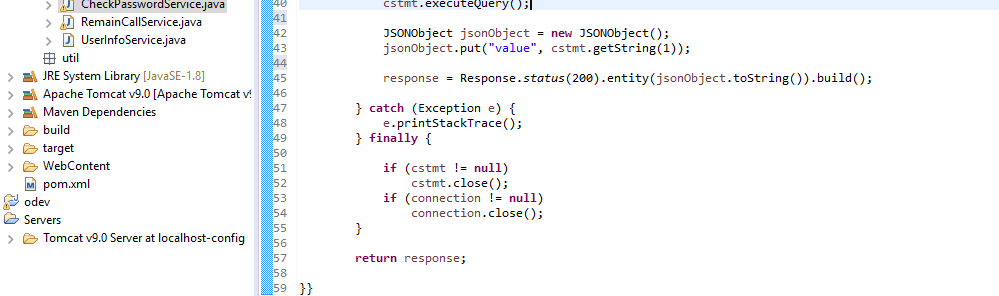
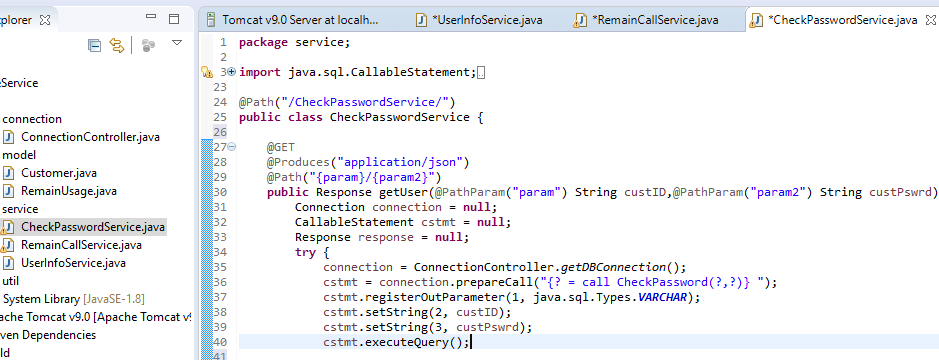


“CheckPasswordService” was created. This class’s mission is get string that return from ”CheckPassword” DB function. Then coming character (Y or N) converted json and return as response when it call.

* There is some differents between other service classes and “CheckPasswordService” class.
* This service take two parameters because CheckPassword PL/SQL function on DB takes two parameter also: Msisdn and password. First two services use just msisdn.
* This service take result of PL/SQL script as varchar, not cursor like first two services.
* ResultSet did not use here. Actually it did not used in first two services. If more than one value return from DB , then ResultSet is necessity. But our services took one value from scripts each step. But in terms of compability , ResultSet used. Even so it is not necessity.
* There was no need to use Gson library here. One character value send as JSONObject

Without Gson’s toJson method.

* This service more simple than first two services. One character Json value return as response when it call.



Services was written. Then Apache Tomcat Server run.

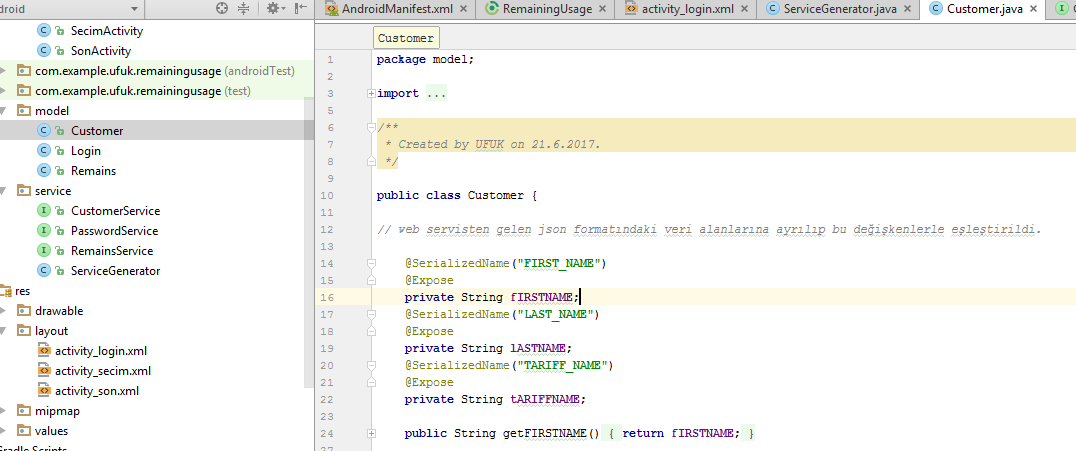
Server side coding was completed. After that Android side coding started.



These web services was written for using “Kalan Kullanımlarım” Android Project. Services return results as json format. These results need parsed for using on Android Studio.

Three models created. Each model corresponds each web services.

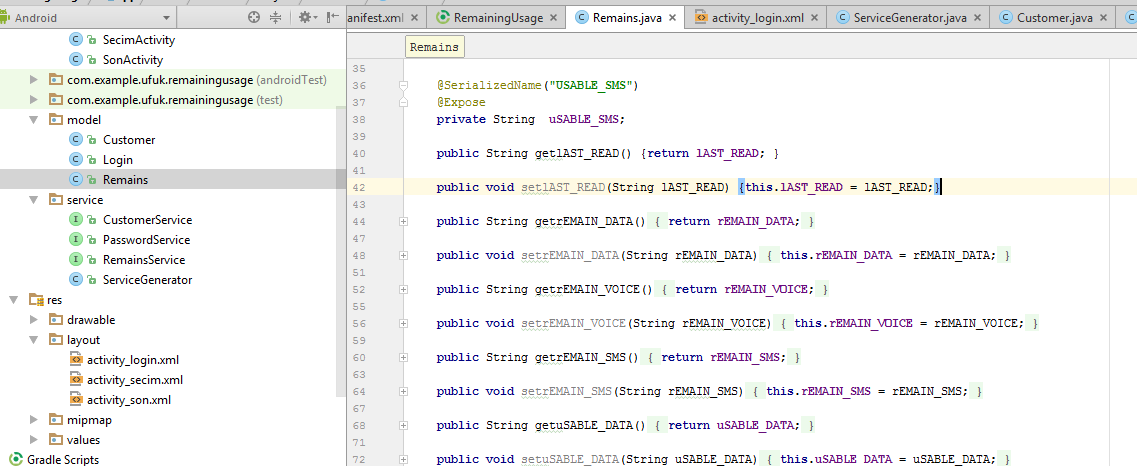
Following one is “Customer” model for correspond values that return from “UserInfoService.java”.



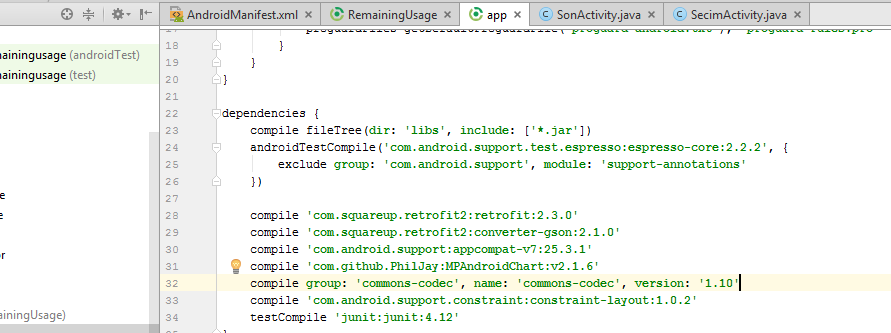
This is “Login” model for correspond values that return from “CheckPasswordService.java”.



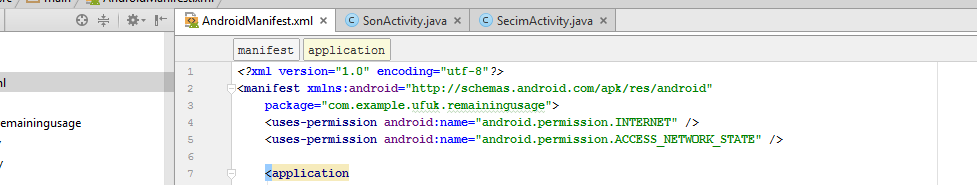
This is “Remains” model for correspond values that return from “RemainCallService.java”.



“RETROFIT 2” library was used for get services result from server. This library’s dependencies added to build.gradle file.

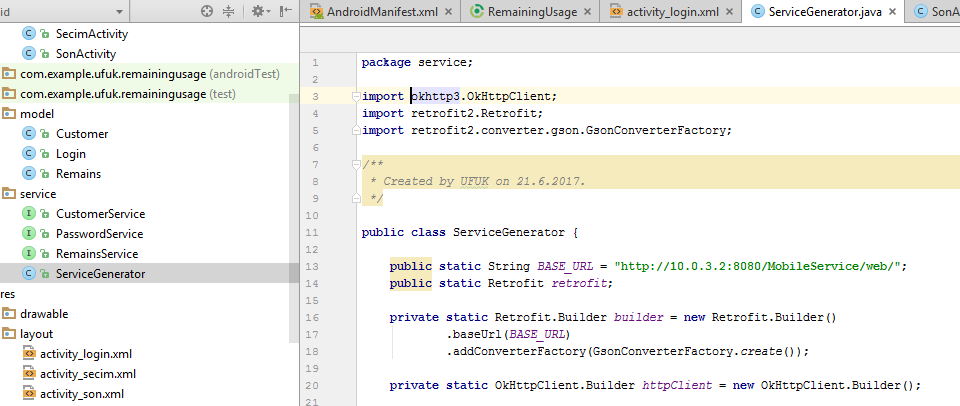


Permissions about internet and network added to AndroidManifest.xml file.



Coding was started. Firstly “ServiceGenerator” class (service) was created.

* This class written for get values from “MobileService” service using Retrofit’s functions.
* ServiceGenerator’s createService method take service class as parameter. And create retrofit connection for coming service class.

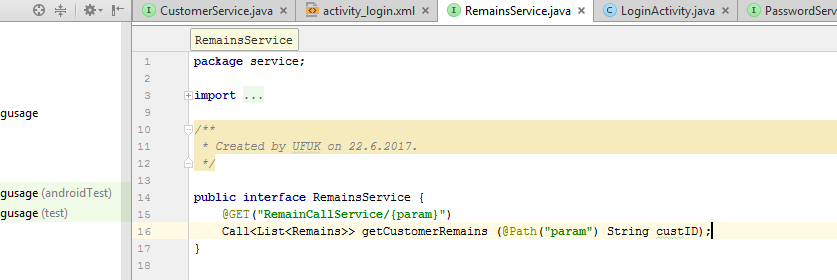


CustomerService , PasswordService , RemainService was created. These service classes will use in ServiceGenerator class’s createService method as parameter to build retrofit connection.

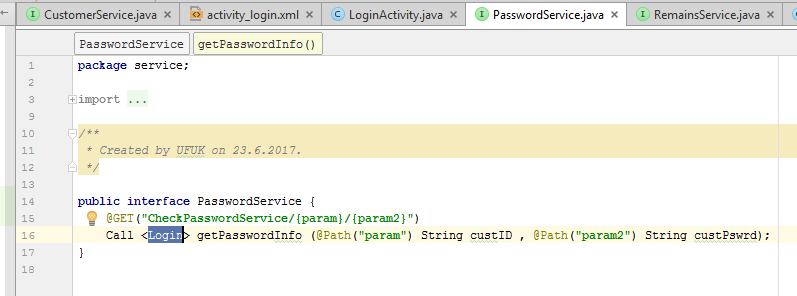
CustomerService;



RemainsService ;



PasswordService has a some differents. It perform according to two parameter. But same as logically.



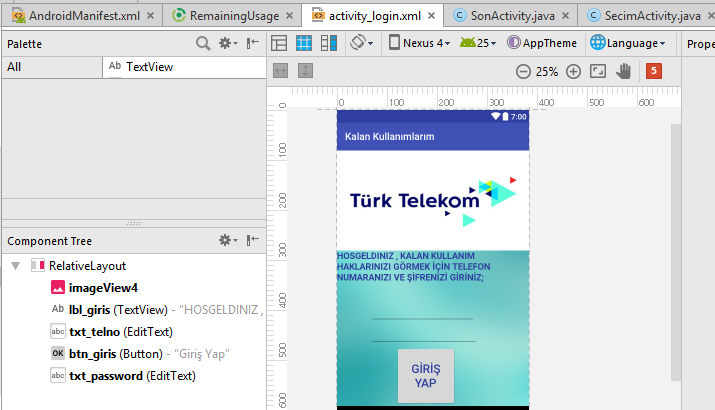
These parameters in this service classes(param, param2) will take from textboxes or etc. And send ServiceGenerator’s createService method. Then this method create retrofit connection to go WebServices. After went WebServices for example UserInfoService, UserInfoService take data’s about customers from DB with SendCustomerInfo PL/SQL script according the msisdn and password. And UserInfoService convert this script’s result to json format. Response taken with Retrofit functions and assign our Customer type variables.

After written this service classes , activities created.

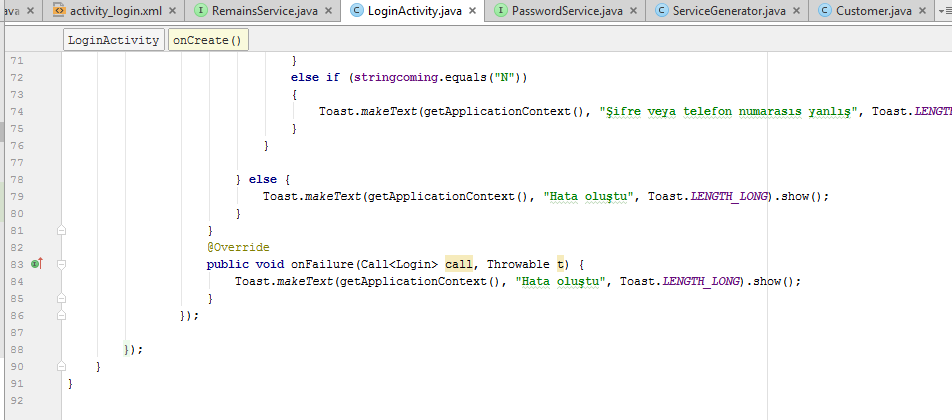
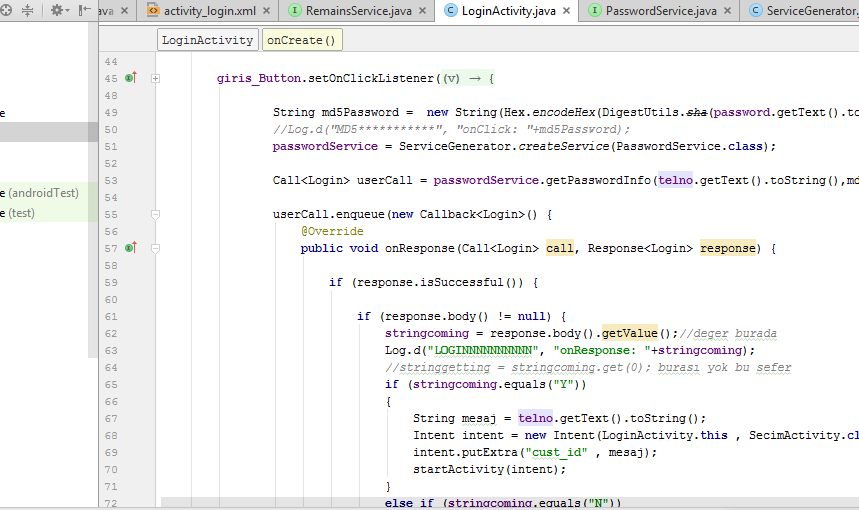
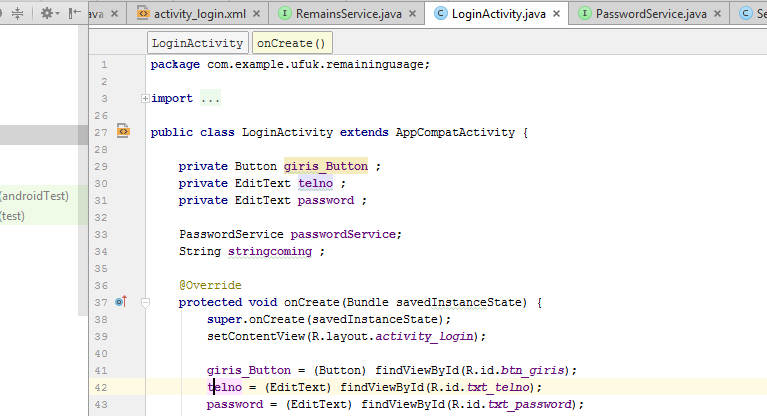
First activity is LoginActivity. This activity’s mission ;

* Take phone number and password from user.
* When press a Giris Yap button; sending them to web service in android as parameter, then sending to java service.
* Check phone number and password exist or not in database.
* Customer’s password encrypted with “md5”.
* If customer exist , SecimActvity activity started.
* Customer’s phone number send to SecimActivity activity as “cust\_id” variable. Because SecimActivity need phone number for using as parameter to send services.

This is design of LoginActivity;



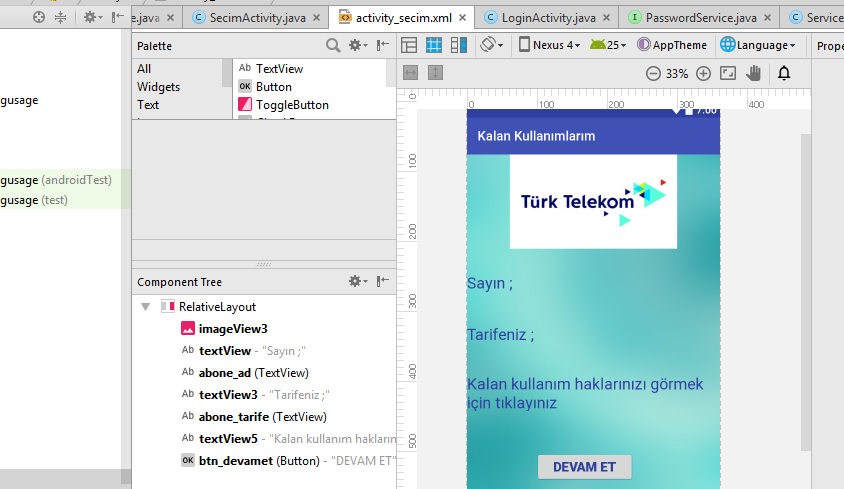
This is source code of LoginActivity;



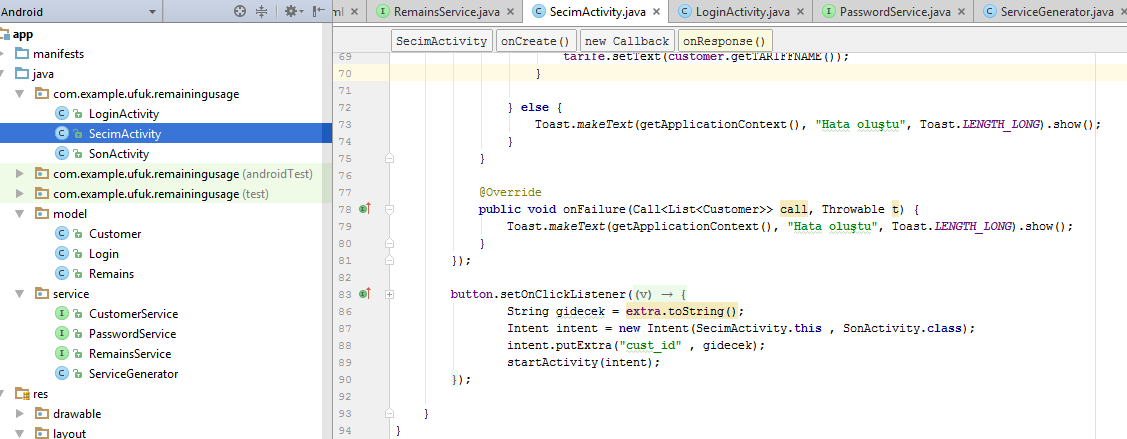
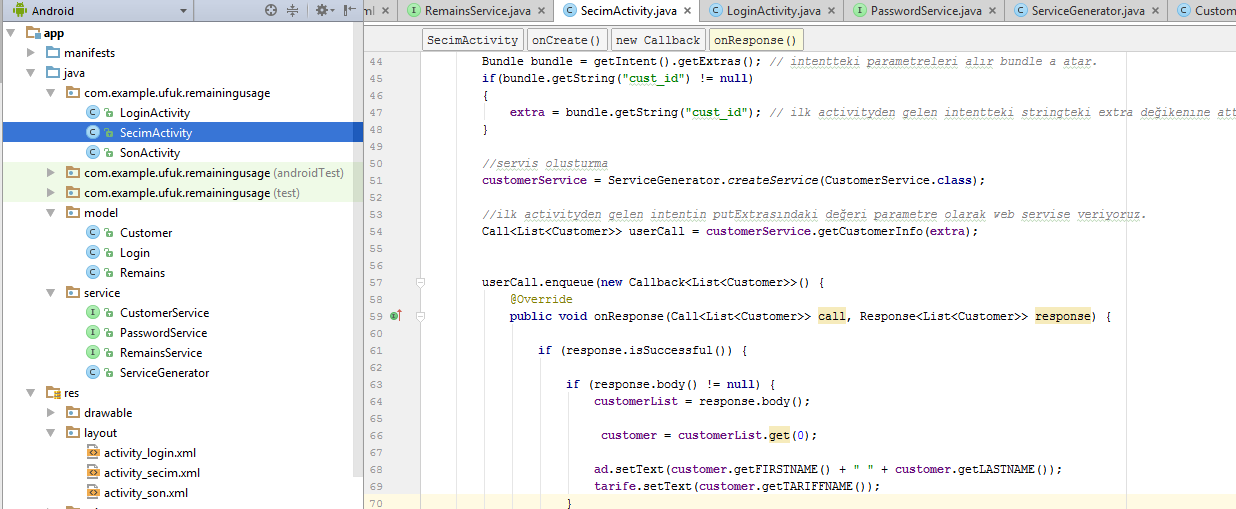
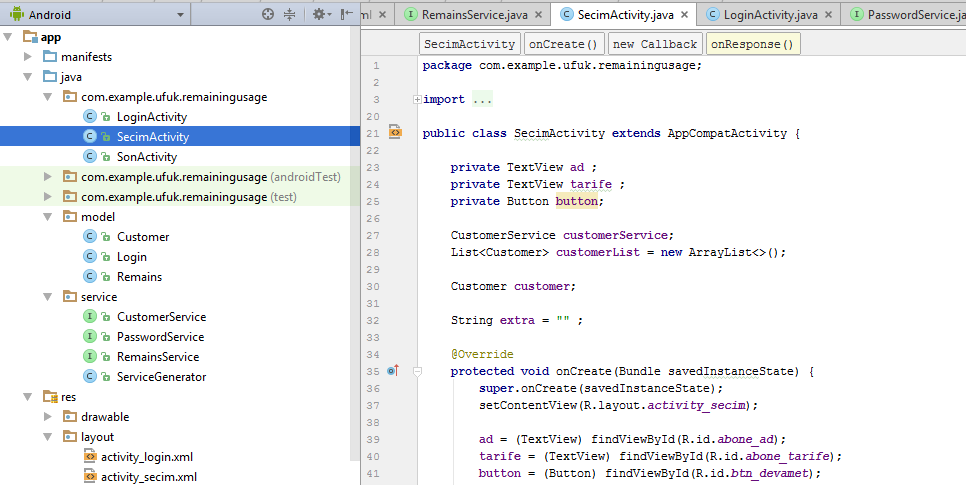
Customer’s login process done with LoginActivity activity. Then SecimActivity activity created. “SecimActivity” class missons;

* Take customer’s phone number from LoginActivity as coming “cust\_id” variable. Phone number necessity for get customer’s informations. It will send as parameter to server’s UserInfoService via android’s CustomerService.
* CustomerService called and started within ServiceGenerator’s createService method. Phone number given to CustomerService as parameter.
* CustomerService send phone number to server’s UserInfoService as parameter , then UserInfoService call SendCustomerInfo PL/SQL script and take result. Converted this results to json format and return us as response.
* This response taken and assign our variables.
* Finally customer’s name , surname and tariff name print to on screen.
* SonActivity called when press Devam Et button. Phone number of customer send to SonActivity as “cust\_id” variable. Because SonActivity need phone number for using as parameter to send services.

This is design of SecimActivity;



These are source codes of SecimActivity;



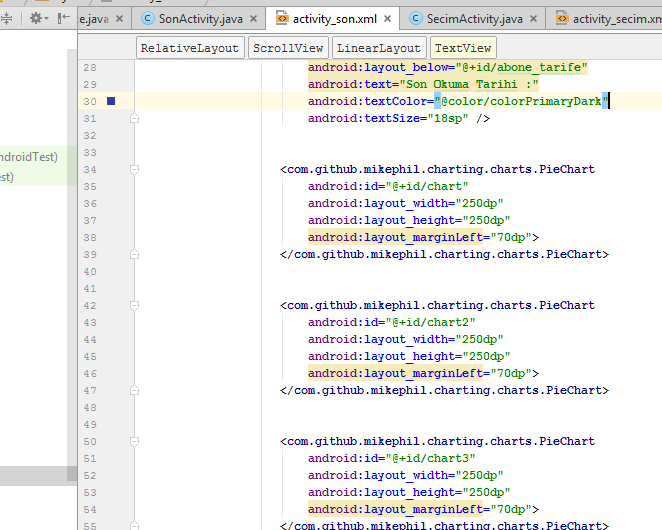
SonActivity was created. This is most important activity in application. Missions of this activity;

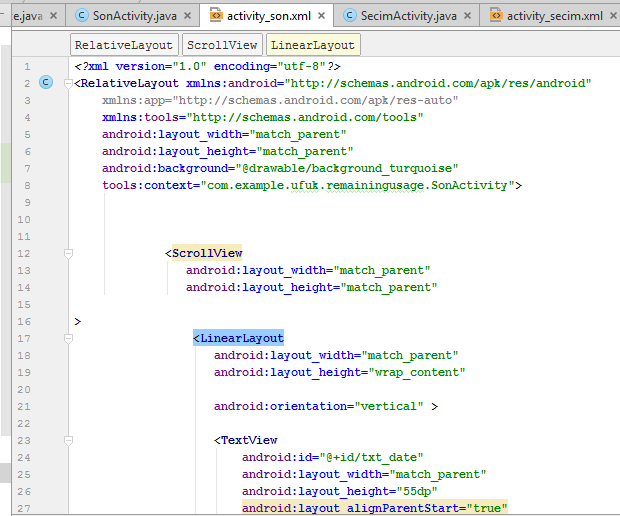
* General purpose is display customer’s remain usage rights.(Voice,sms,data)
* Used Piechart animation for display these datas.
* PieChart is library. This library’s dependencies added to build.gradle file. And import in top of activity.



* Customer’s phone number taken from SecimActivity as coming “cust\_id” variable. Phone number necessity for get customer’s remain usages. It will send as parameter to server’s RemainCallService via android’s RemainsService.
* RemainsService called and started within ServiceGenerator’s createService method. Phone number given to RemainsService as parameter.
* RemainsService send phone number to server’s RemainCallService as parameter , then RemainCallService call SendRemainUsage PL/SQL script and take result. Converted this results to json format and return us as response.
* This response taken and assign our variables.
* Customer’s usage amounts , remain amounts and last read date informations taken with this response.
* Then this datas diplay in PieChart within loadPieChart() method. For each usages(sms,data,voice) one piechart was created.

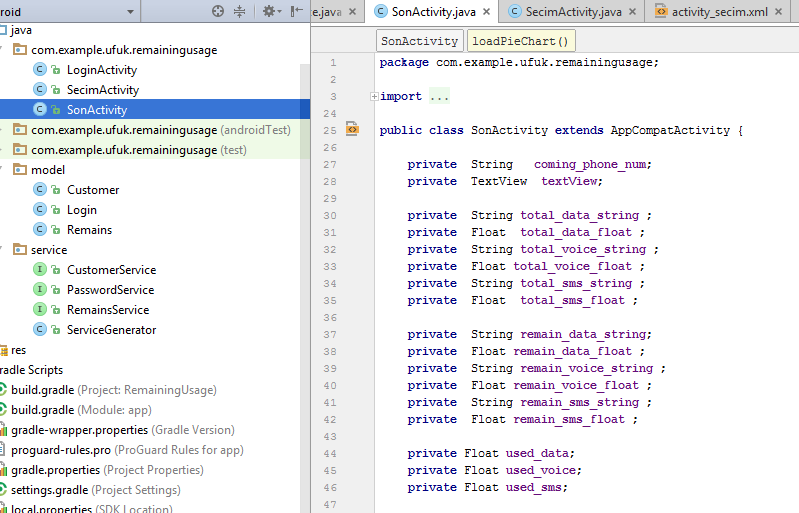
This is design of this activity as xml text. Because piecharts doesn’t in design. It must added as xml code.



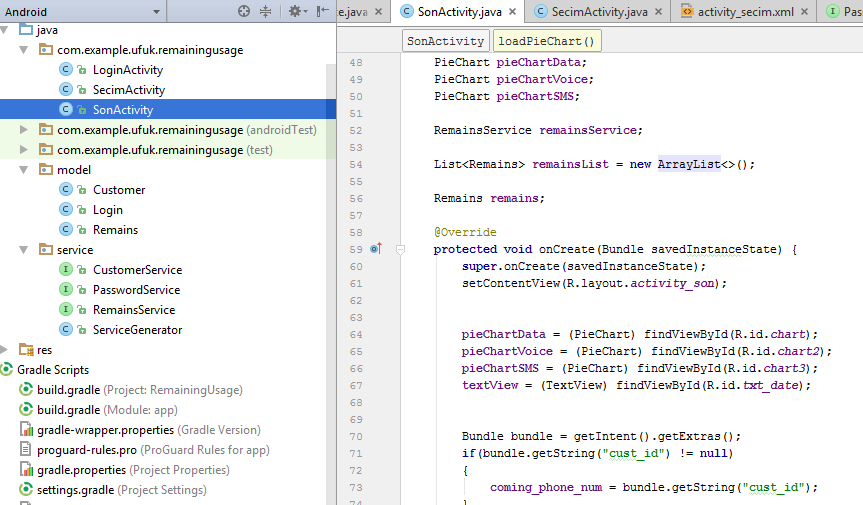


These are source codes of SonActivity;

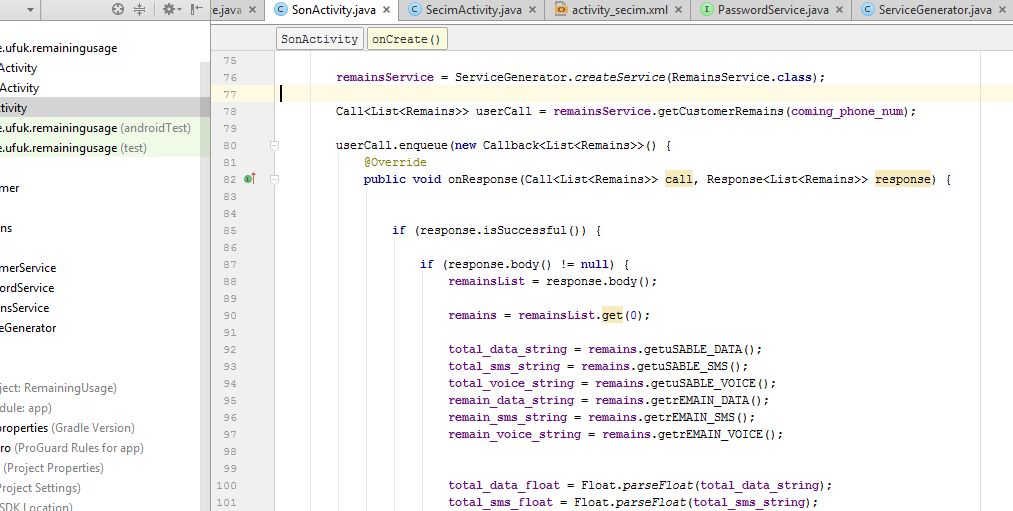
Defining variables;



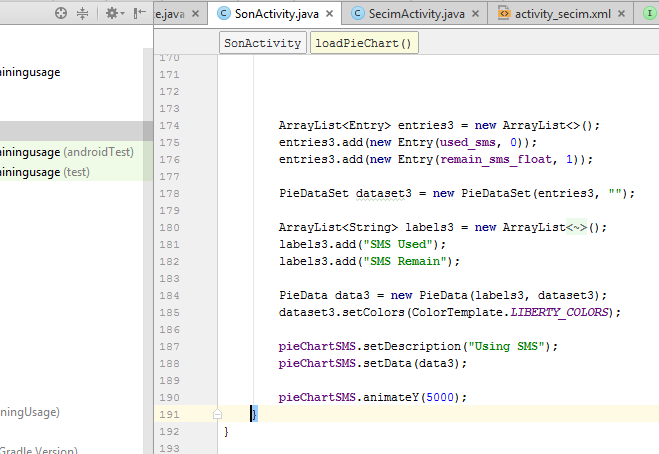
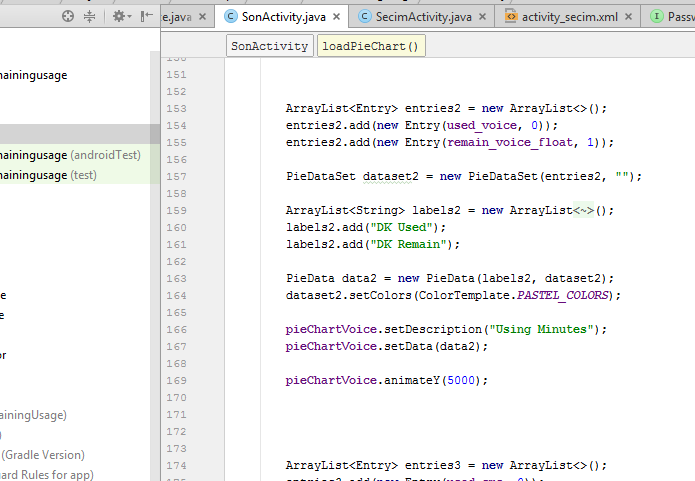
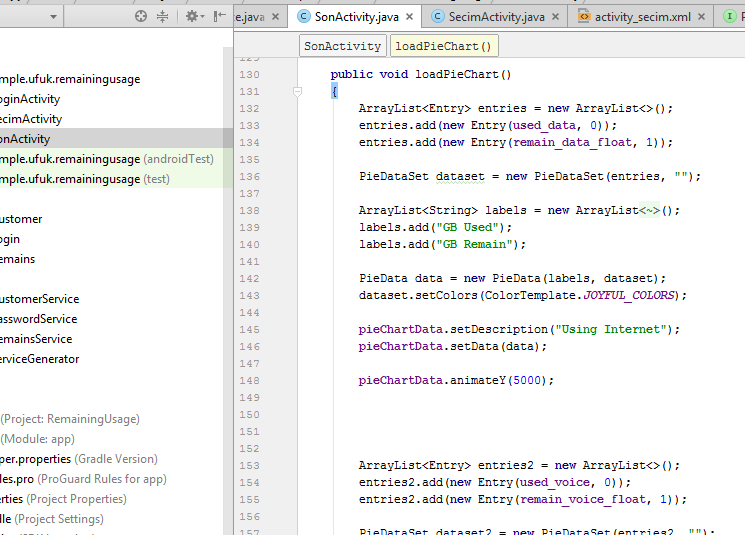
Getting phone number ;



Get connection between services to get values. After get response , response’s values assign local variables.



Creating loadPieChart fonction. Usage datas load to Piechart. There are 3 PieDataSet because we have 3 kind of data to show. Data , sms , voice. Each one needs new PieDataSet.



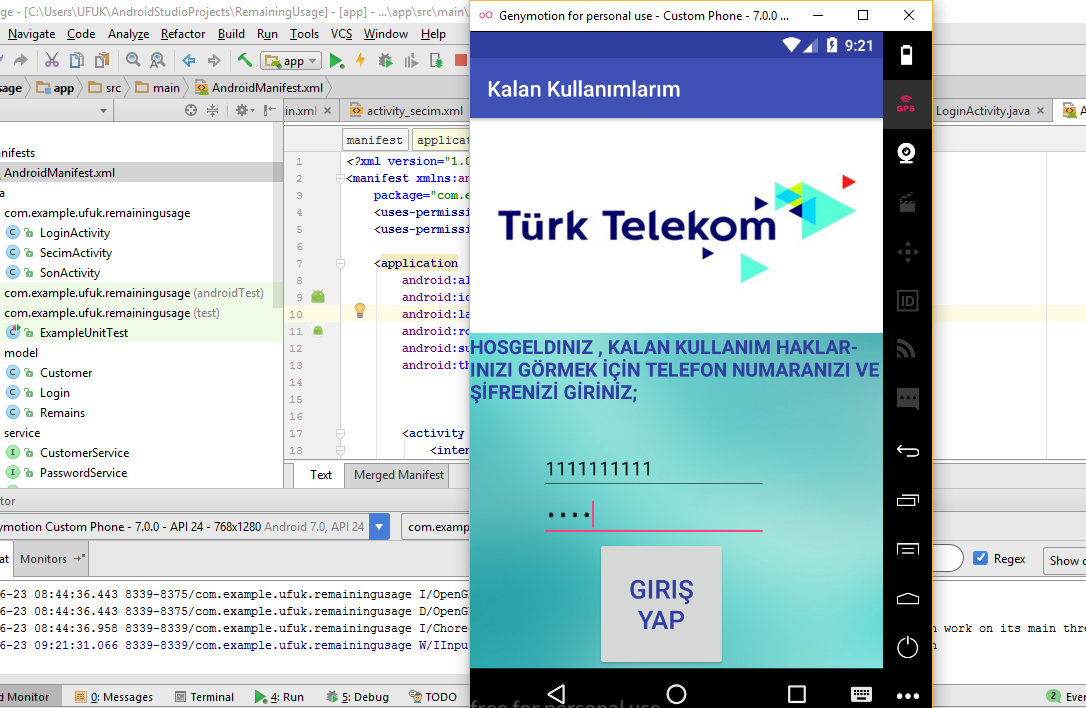
“Kalan Kullanımlarım” Android Project finished. Summary ;

* Database created in Oracle.
* PL/SQL scripts written for each process.
* Web services created in Java, for get PL/SQL script result and send to Android Project. Tomcat Server builded. These services run on Tomcat Server.
* Results take from Android Project. Retrofit library used for get datas from server.
* Models and services created for correspond coming datas.
* Datas get from server according to phone number and password that user enter at login activity.
* This datas display screen on labels and piecharts.

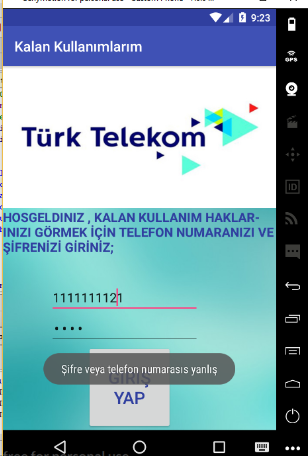
Firstly , tomcat server must run.



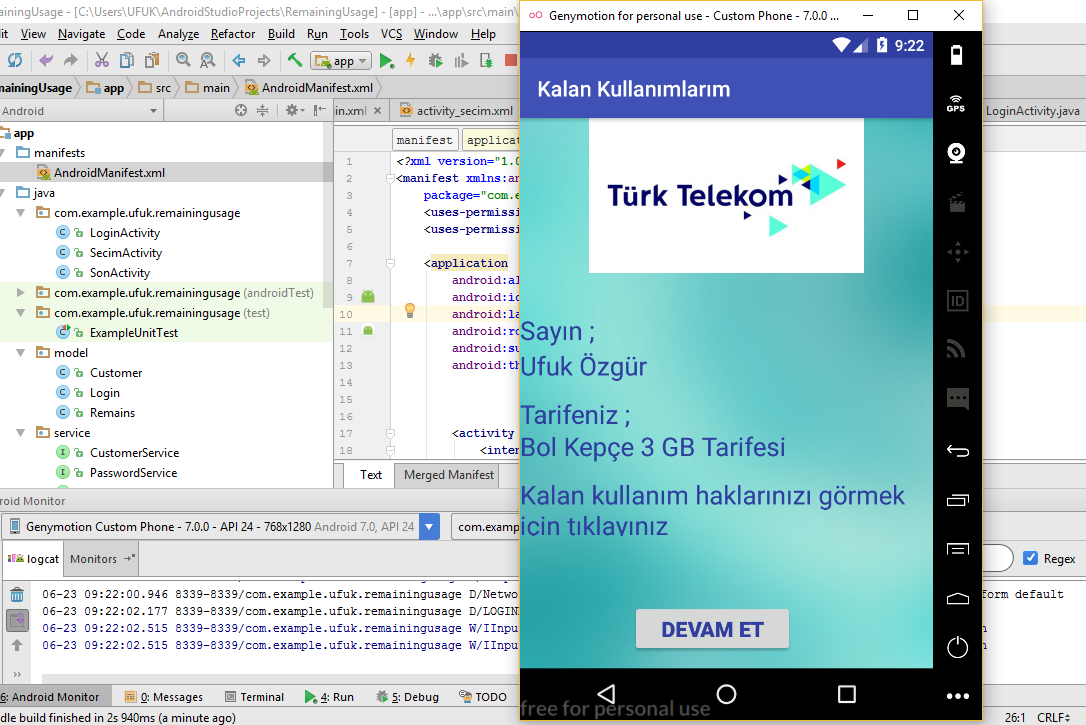
This is login screen. We login with phone number and password. And press Giris Yap button.



If phone number or password is incorrect , then error toast appear.



After we login with correct phone number and password , second screen coming. Customer’s name and tariff name display. Then being pressed devam et button.



After press devam et button , customer uses and remaining uses screen on piecharts. Additionaly last read date display.

