­­­Programmer’s Guide

Table of Contents

Intro **2**

Server 3-5

Mobile………………………………………………………………………..6

Intro

In this guide will we explain the main parts of application, and how the code is divided and where it’s written.

The application itself is an Android application, the mobile code is written in java using Android studio environment.

In addition, we used AWS services to implement the server code that contains the main logic of the application, so the game logic, poker tables and users data are saved on the cloud,

The client (mobile user) just have the layout code, and whenever they want to play they connect using their unique id.

The controller of the game is implemented in the server code, it runs the games and send/receives update to/from users. And the users always get the updated game status from the server,

And display it on the layout, the send messages on their turn.

Server

We used the following AWS services :

1. API Gateway.
2. Lambda.
3. DynamoDB.
4. CloudWatch.

1) API Gateway:

This services enables you to create your own API, we choose the Web Socket API, so after deploying our API, there is a gateway on the entrance, and every client that wants to invoke a procedure from the API, have to create a web-socket connection with the API.

The API contains the procedures that we provided, and one required procedure that has to be implemented, called “onConnect”, and it’s invoked the beginning of the connection.

The API is http based, and the data sent is in Json format.

In every valid message sent to the gateway, there is a field called “action”, and it contains the name of procedure that we want to call.

For more clearance, there are no game or program running on the server, but the game played by creating web –socket connections then waiting/send on/to it.

To implement the procedures of the API, we use the LAMBDA service.

2) Lambda

Using Lambda we can create and implement our procedures.

Whenever a lambda function is called, a new instance of the function is created and starts running on the server.

Lambda functions can invoke other lambda functions, and execute operations on the database, so every lambda function has it’s own “role permissions” (for example: reading from database, or invoking other function).

3) DynamoDB

This is the database used to save the game’s data.

There is a table in the DB called “users”, it’s uses to save the users’ info like balance, name, facebook id.

In addition, for every poker table ( table that players join in the app), there is a matching table in the database, to save the table status like: small and big blinds, center cards, players’ cards … .

Operations can be done on the database using the boto3

Python library.

4) CloudWatch:

This service is used for debugging, so it displays the log files created while running the lambda functions.

To use this service import logging python library.

Mobile:

Like it’s said above, the mobile code is written in java using Android studio.

The mobile code implements the layout logic of the user and not the game logic.

In order to start the game, the user has to authenticate himself using his facebook account, so we integrated the “Facebook SDK”, that contains the login procedure as a black box. We used Facebook’s API in order to get all information needed for the player in the game.

In order to join a poker table, the user creates a web-socket connection with the API Gateway, we used okhttp3 library for the web-socket which we imported using the gradle file.