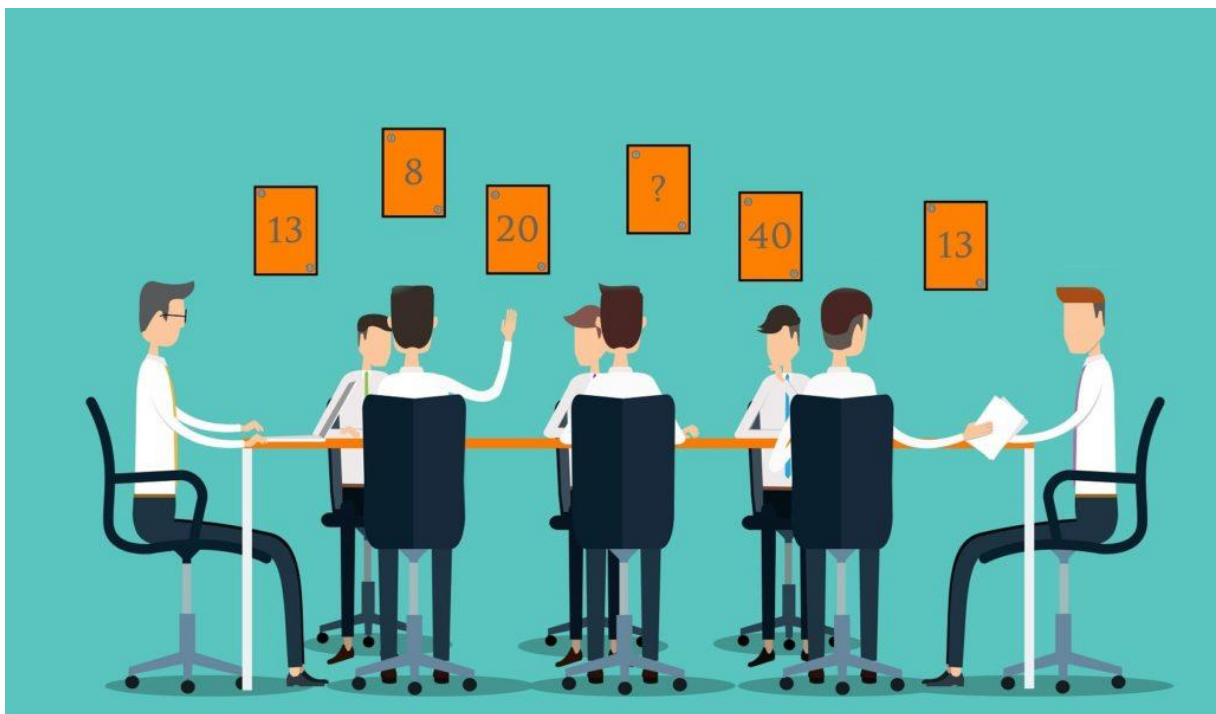


Planning Poker Application



~ Android programozás ~

Benkő Edina

Informatika III.

Tartalom

1. Alkalmazás felépítése	3
2. Alkalmazás áttekintése programozói szempontból	3
2.1 Admin rész	3
2.1.1 Áttekintés	3
2.1.2 Kódrészletek	5
2.2 Felhasználói rész	7
2.2.1 Áttekintés és kódrészlet	7
3. Alkalmazás áttekintése felhasználói szempontból:	9
3.1 Admin rész	9
3.1.1 Bejelentkezés	9
3.1.2 Menü	9
3.1.3 Válaszok megtekintése	10
3.1.4 Új csoport hozzáadása	11
3.2 Felhasználó rész	11
3.2.1 Bejelentkezés	11
3.2.2 Kérdésekre való válaszadás	12
3.2.3 Összes csoporthoz tartozó kérdés listázása és válaszok megtekintése	12

1. Alkalmazás felépítése

A Planning Poker alkalmazás témája az, hogy a felhasználóknak olyan kérdés sorozatokra kell választ adnia, amelyet az admin egy kérdéscsoportba válogatott össze és abban fogalmazta meg kérdéseit.

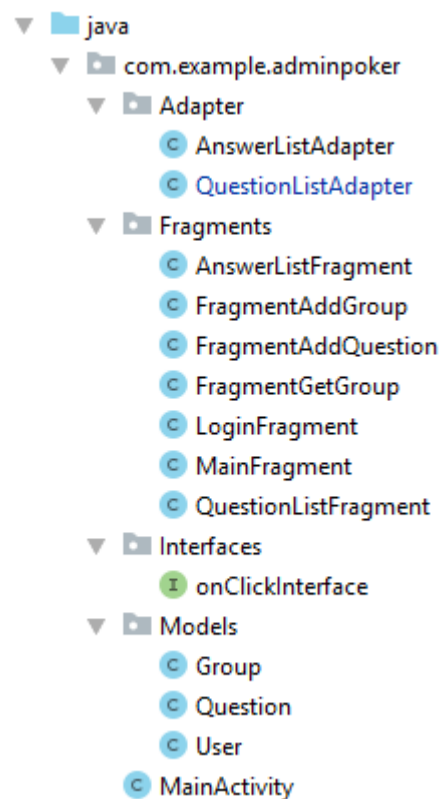
Az alkalmazást két nagy részre lehet osztani: Felhasználói rész és Admin rész.

2. Alkalmazás áttekintése programozói szempontból

2.1 Admin rész

2.1.1 Áttekintés

Az alkalmazás megvalósítása a következő csomagokba szerveződött:



Egy Activity alkalmazása, és az összes többi “képernyő” az Fragmentre alapszik.

Adatbázisként **Firestore** volt használva, ugyanazt a Firestore adatbázist használja a User is és az Admin is.

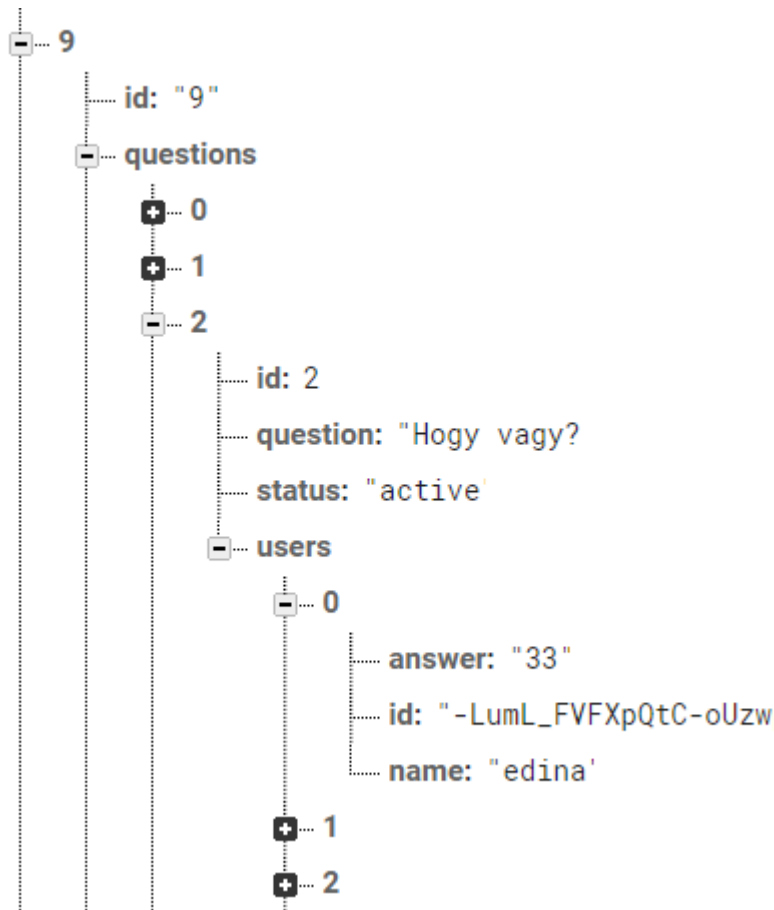
Három **modelként** szolgáló osztályt használtam:

- **Group osztály:** amely tartalmaz többek között egy Question típusú tömböt is, így jelölöm az adott csoporthoz tartozó kérdéseket.

- **Question osztály:** amely tartalmaz többek között egy User típusú tömböt is, így jelölöm az adott kérdésre érkezett összes választ.
- **User osztály**

Firestore-ben az adatbázis struktúra:

(Ugyanaz az admin és a felhasználói rész esetében is)



RecyclerView Adaptereket a grouphoz tartozó kérdések kilistázásánál használtam, valamint a kérdéshez tartozó összes válasz kilistázásánál.

Fragmentek által megvalósítottam a következőket:

- LoginFragment
 - ez az admin bejelentkezését szolgálja, ő regisztrálni nem tud, csak bejelentkezni, mivel a bejelentkezési adatok adottak, ezért csak egy admin létezik.
 - login adatok: benkoedina (12345)

- MainFragment: ez egy aprócska menüként szolgál az admin-nak, hogy mit is szeretne tenni.
- FragmentAddGroup, FragmentAddQuestion: Ha új csoportot szeretne hozzá adni új kérdésekkel akkor ez jön használatba
- FragmentGetGroup, QuestionListFragment, AnswerListFragment:
Hogyha meg szeretné tekinteni az adott csoporthoz tartozó kérdéseket, majd a kiválasztott kérdésre a válaszokat.

2.1.2 Kódrészletek

Megadott csoporthoz tartozó kérdések lekérése az adatbázisból egy q1 listába (QuestionListFragment):

```
firebaseDatabase = firebaseDatabase.getInstance();
databaseReference = firebaseDatabase.getReference();
final View view = inflater.inflate(R.layout.fragment_questionlist, container, attachToRoot: false);
Bundle bundle = this.getArguments();
final String groupid = bundle.getString( key: "groupid"); //adott group_id lekérése

databaseReference.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        Group g = dataSnapshot.child(groupid).getValue(Group.class);
        //adott group_id -hoz tartozó kérdések lekérése
        final ArrayList<Question> q1 = g.getQuestions();
    }
});
```

q1 lista átadása a recyclerView adapternek:

```
//adapter megvalositas es recycler view
recyclerView = (RecyclerView)view.findViewById(R.id.questionrecyclerView);
recyclerView.addItemDecoration(new DividerItemDecoration(getContext(), DividerItemDecoration.VERTICAL));
adapter = new QuestionListAdapter(q1, getContext(), onclickInterface);
recyclerView.setHasFixedSize(true);
recyclerView.setLayoutManager(new LinearLayoutManager(getContext()));
recyclerView.setAdapter(adapter);
```

Ha kattintott bizonyos kérdésre, akkor annak a válaszait feldolgozzuk egy tömbbe, és átadjuk az AnswerListFragment-nek, amit majd ő is hasonlóan fog beállítani és kilistázni mint a QuestionListFragment:

```

onClickInterface = (onClickInterface) (pos) -> {

    AnswerListFragment Afragment = new AnswerListFragment();

    final Bundle bundle = new Bundle();

    if (ql.get(pos).getUsers() != null) {

        ArrayList<User> users = ql.get(pos).getUsers();
        ArrayList<String> userAnswers = new ArrayList<>();
        //lekerjuk az adott kerdesre a valaszokat es azt adjuk tovabb

        for (User u : users)
        {
            String value = u.getName() + ": " + u.getAnswer();
            userAnswers.add(value);
        }
        bundle.putStringArrayList("answers", userAnswers);

    }
    else {
        // ha nincs a kerdesre semmi valaszt akkor ures listat ad tovabb a masik fragmentnek
        ArrayList<String> userAnswers = new ArrayList<>();
        bundle.putStringArrayList("answers", userAnswers);
    }
}
Afragment.setArguments(bundle);
as

FragmentManager fr = getFragmentManager().beginTransaction();
fr.replace(R.id.container, Afragment);
fr.addToBackStack(null);
fr.commit();

```

FragmentAddGroup: bekérünk egy szöveget és értéket radiobutton-ból, amely igazolja, hogy aktív vagy nem aktív a csoport.

FragmentAddQuestion: bekérünk öt kérdést és öt aktív/nem aktív státuszt, maximálisan csak ennyi kérdést lehet hozzá adni a csoporthoz, majd azt hozzá adjuk az adatbázishoz az objektumok létrehozása után.

```

// adatok bekereese es validalasa
Button bt_add = view.findViewById(R.id.bt_add_question);
final EditText et_q1 = view.findViewById(R.id.et_question1);
final EditText et_q2 = view.findViewById(R.id.et_question2);
final EditText et_q3 = view.findViewById(R.id.et_question3);
final EditText et_q4 = view.findViewById(R.id.et_question4);

//Radio Button-ok ellenorzes
RadioGroup radioGroup1 = view.findViewById(R.id.groupradio1);
if (radioGroup1 != null) {
    radioGroup1.setOnCheckedChangeListener((group, checkedId) -> {

        status1 = (R.id.activ1 == checkedId) ? "active" : "inactive";
        Toast.makeText(getContext(), status1, Toast.LENGTH_SHORT).show();

    });
}

```

```

//kivesszük a másik fragment által küldött group adatokat
Bundle bundle = this.getArguments();
final String group_id = bundle.getString( key: "group_id");
final Boolean group_status = bundle.getBoolean( key: "status");
bt_add.setOnClickListener((view) -> {

    String q1 = et_q1.getText().toString();
    String q2 = et_q2.getText().toString();
    String q3 = et_q3.getText().toString();
    String q4 = et_q4.getText().toString();

    //Question tomb létrehozása
    ArrayList<Question> questions = new ArrayList<>();

    // question objektumok létrehozása
    Question question1 = new Question( id: 0,q1,status1);
    Question question2 = new Question( id: 1,q2,status2);
    Question question3 = new Question( id: 2,q3,status3);
    Question question4 = new Question( id: 3,q4,status4);

    //hozza adjuk a tombhoz
    questions.add(question1);
    questions.add(question2);
    questions.add(question3);
    questions.add(question4);

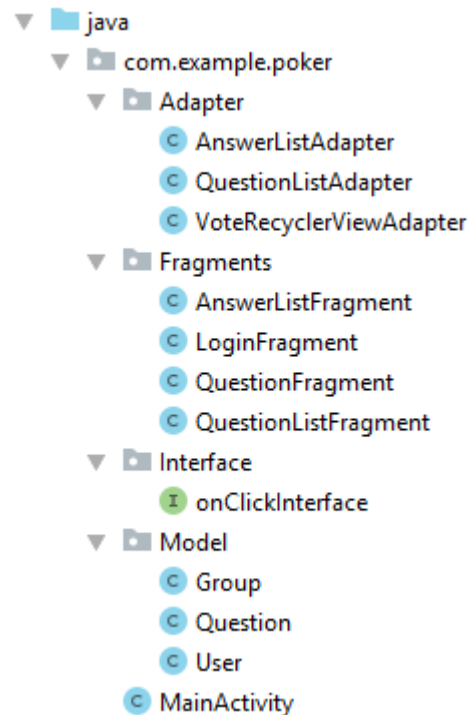
    // group objektum létrehozása
    Group group = new Group(group_id,group_status,questions);
    // hozzá adjuk az adatbázishoz
    databaseReference.child(group.getId()).setValue(group);
    Toast.makeText(getContext(), text: "Added Successfully",Toast.LENGTH_LONG).show();
    Log.d( tag: "GROUP", group.toString());
});

```

2.2 Felhasználói rész

2.2.1 Áttekintés és kódrészlet

A felhasználói program a következő csomagokba szerveződött:



Egy Activity alkalmazása, és az összes többi “képernyő” az Fragmentre alapszik.

Adatbázisként **Firestore** volt használva, ugyanazt a Firestore adatbázist használja a User is és az Admin is.

RecyclerView Adaptereket a grouphoz tartozó kérdések kilistázásánál használtam, valamint a kérdéshez tartozó összes válasz kilistázásánál, valamint a válaszlehetőségek is RecyclerView-al vannak megvalósítva.

Fragmentek által megvalósítottam a következőket:

- LoginFragment
 - ez a felhasználó bejelentkezését szolgálja, kötelező módon meg kell adnia egy érvényes csoport id-t és egy nevet, ezt követően tud tovább lépni a válaszadó részre.

```
bt_login.setOnClickListener((view) -> {  
  
    //login ellenorzes  
    if (et_name.getText().toString().isEmpty() || (et_groupid.toString().isEmpty())) {  
  
        Toast.makeText(getApplicationContext(), text: "Fields cannot be empty!", Toast.LENGTH_SHORT).show();  
    } else {  
  
        final String groupId = et_groupid.getText().toString();  
        final String name = et_name.getText().toString();  
  
        //megnezi hogy aktiv-e vagy nem aktiv a group ahova csatlakozni szeretne a user  
        checkGroupStatus(groupId, name);  
  
    }  
  
});
```

- QuestionFragment
 - ez szolgál arra, hogy minden kérdést a válaszlehetőségekkel együtt megjelenítsen
 - kiolvassa az adatbázisból az összes kérdést, leellenőrzi őket, hogy aktívak-e és beállítja minden egyes kérdésre az adott választ.

- válaszok hozzáadása a kérdéshez

```
//letrehozzuk a kulcsot  
String key = databaseReference.child(groupid).child("questions").child(String.valueOf(position)).child("users").push().getKey();  
  
//user objektum  
//adapterbol kinhyert valasz, amire klikkelt  
User user = new User(key, name, adapter.getItem(adapter.getPos()));  
  
//a mar meglevo user tombhoz, amit az elejen kertunk le, a mar eddig meglevo valaszokkal, ahhoz hozza adjuk  
users.add(user);  
//a fent lekert question tombhoz hozza adjuk a user tombot, ami all az uj+a regi valaszokbol  
question.setUsers(users);  
  
//felulirjuk az eddigi group id adott pozicion levo kerdest  
databaseReference.child(groupid).child("questions").child(String.valueOf(position)).setValue(question);
```

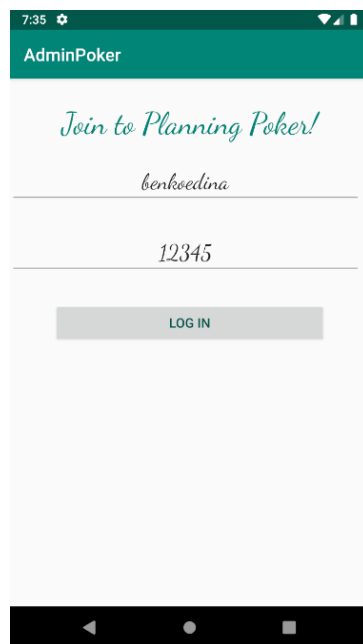

- QuestionListFragment, AnswerListFragment: ugyanaz, mint az adminnál.

VoteRecyclerViewAdapter arra szolgál, hogy a válaszokat megjelenítse minden kérdés alá

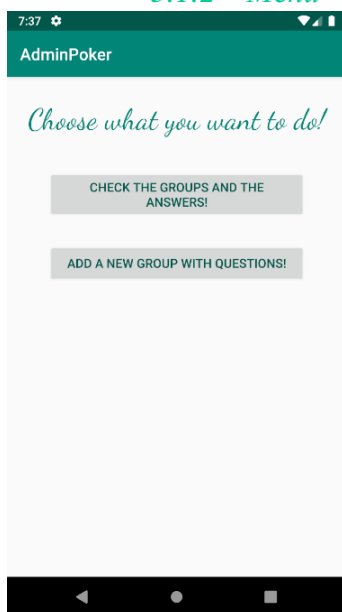
3. Alkalmazás áttekintése felhasználói szempontból:

3.1 Admin rész

3.1.1 Bejelentkezés

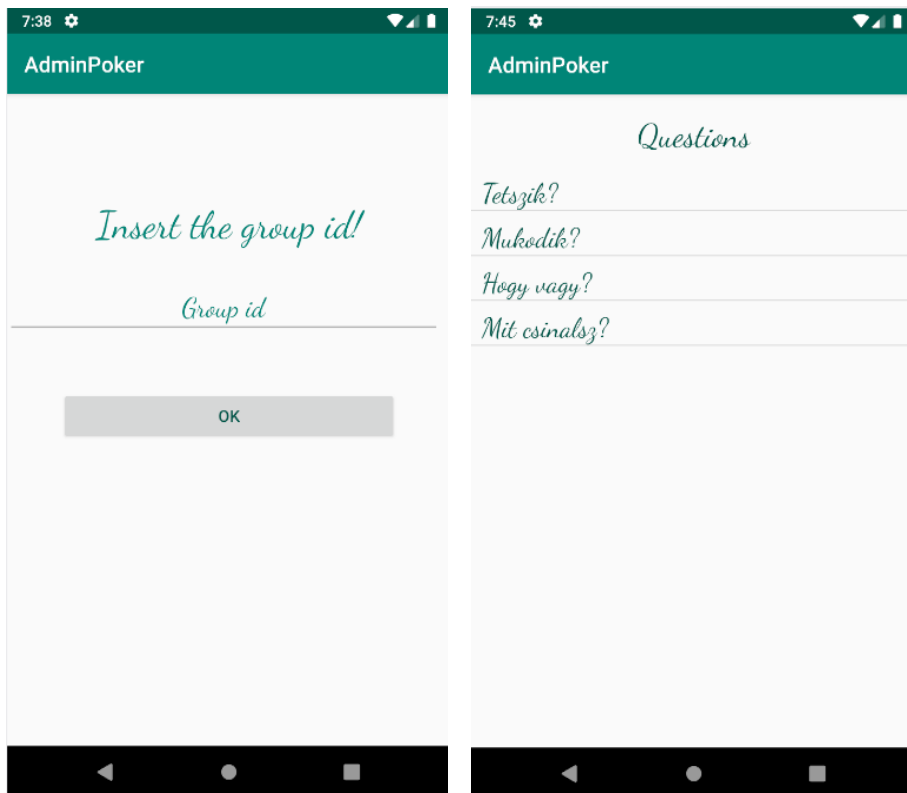


3.1.2 Menü

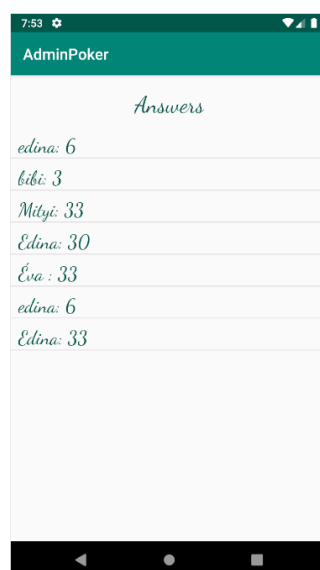


Lehetősége van az adminnak kiválasztani, hogy új csoportot hoz létre új kérdésekkel, vagy megtekint egy eddigi csoportot a kérdésekkel és válaszokkal együtt

3.1.3 Válaszok megtekintése



Megkell adnia, hogy melyik eddig létrehozott csoportot szeretné megtekinteni, majd azután az ok gombra kattintva átjut egy gördülő listára, ahol megjelennek az aktív és nem aktív kérdések is. Majd ha valamelyik kérdésre rákattint, akkor egy új listában megjelennek az arra adott felhasználók válasza és nevei.



3.1.4 Új csoport hozzáadása

The image displays two screenshots of the AdminPoker application. The left screenshot, titled 'AdminPoker', shows the 'Add a New Group!' screen. It features a text input field for 'Group id', two radio buttons labeled 'active' and 'inactive', and a button labeled 'ADD NEW GROUP'. The right screenshot, also titled 'AdminPoker', shows the 'Add a New Question!' screen. It contains four text input fields for questions, each followed by two radio buttons labeled 'active' and 'inactive'.

Új csoport létrehozásánál be kell írja a csoport id-t, és annak a státuszát, majd tovább lépéssel 4 kérdés és annak státuszának hozzáadására van lehetősége.

3.2 Felhasználó rész

3.2.1 Bejelentkezés

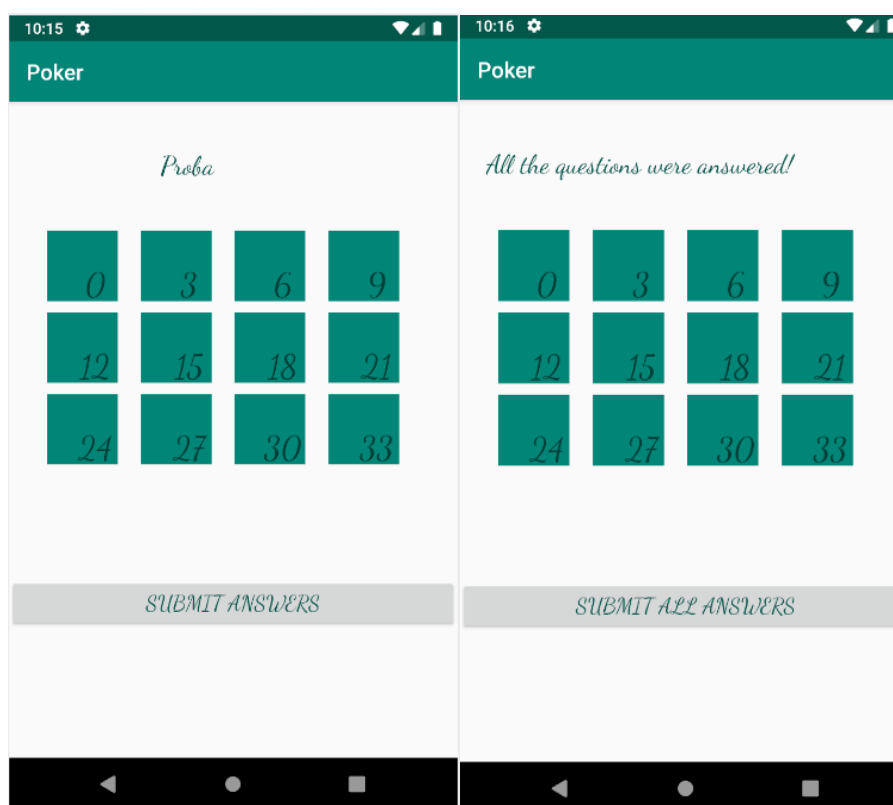
Bejelentkezésnél a felhasználónak meg kell adnia egy érvényes csoport id-t és a saját nevét. Hogyha ezek az adatok rendben vannak, akkor mehet tovább a csoporthoz tartozó kérdések megválaszolásához.

The image shows a screenshot of the Poker application. The screen is titled 'Join to Planning Poker!'. It has two text input fields, one for 'Group id' and one for 'Your Name'. Below these fields is a button labeled 'LOG IN'.

3.2.2 Kérdésekre való válaszadás

Amint átkerült ide a felhasználó, akkor sorban választ tud adni a kérdésekre, a kérdés alatt megjelenő kártyák közül egyikre kattintva. Ahányas a szám rajta annyi az ő válasza. Fontos, hogy csak a csoporton belüli aktív kérdésekre tud választ adni.

Abban a pillanatban ahogy elfogytak a kérdések, akkor feljön egy olyan szöveg, miszerint már mindenre válaszolt, ezért újra meg kell nyomnia a *submit all answers* gombot.



3.2.3 Összes csoporthoz tartozó kérdés listázása és válaszok megtekintése

Amint megnyomta a felhasználó a *submit all answers* gombot tovább jut egy gördülő listához, ahol a csatlakozott csoporthoz tartozó összes kérdést megtekintheti. Ha egyre rákattint, akkor megtekintheti az adott kérdésre adott összes választ.

10:18	11:15
Poker	Poker
<i>Questions</i>	<i>Answers</i>
<i>Also kerdes</i>	<i>edina: 6</i>
<i>Proba</i>	<i>bibi: 3</i>
<i>Proba</i>	<i>Mityi: 33</i>
<i>Proba</i>	<i>Edina: 30</i>
	<i>Eva : 33</i>
	<i>edina: 6</i>
	<i>Edina: 33</i>
	<i>Palma: 0</i>