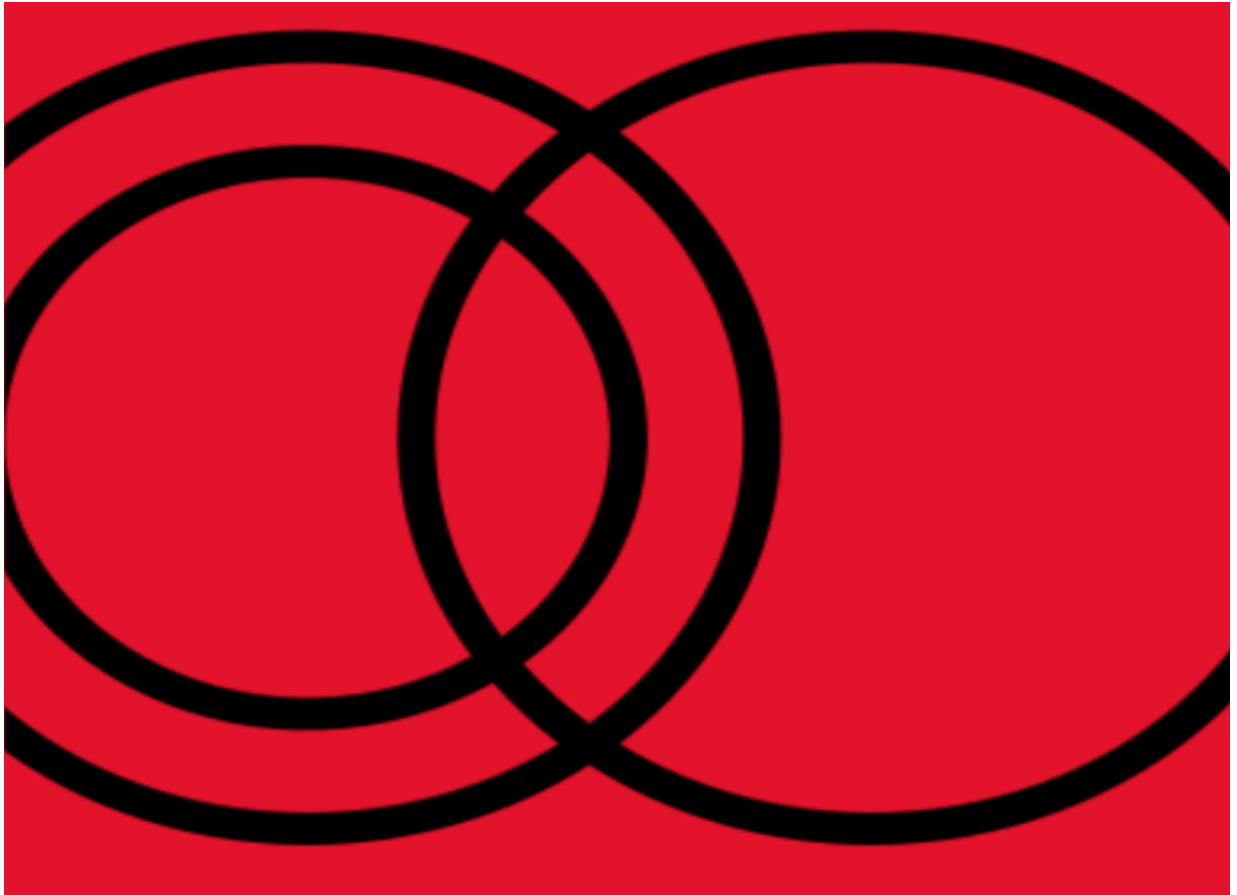


# PLANNING POKER



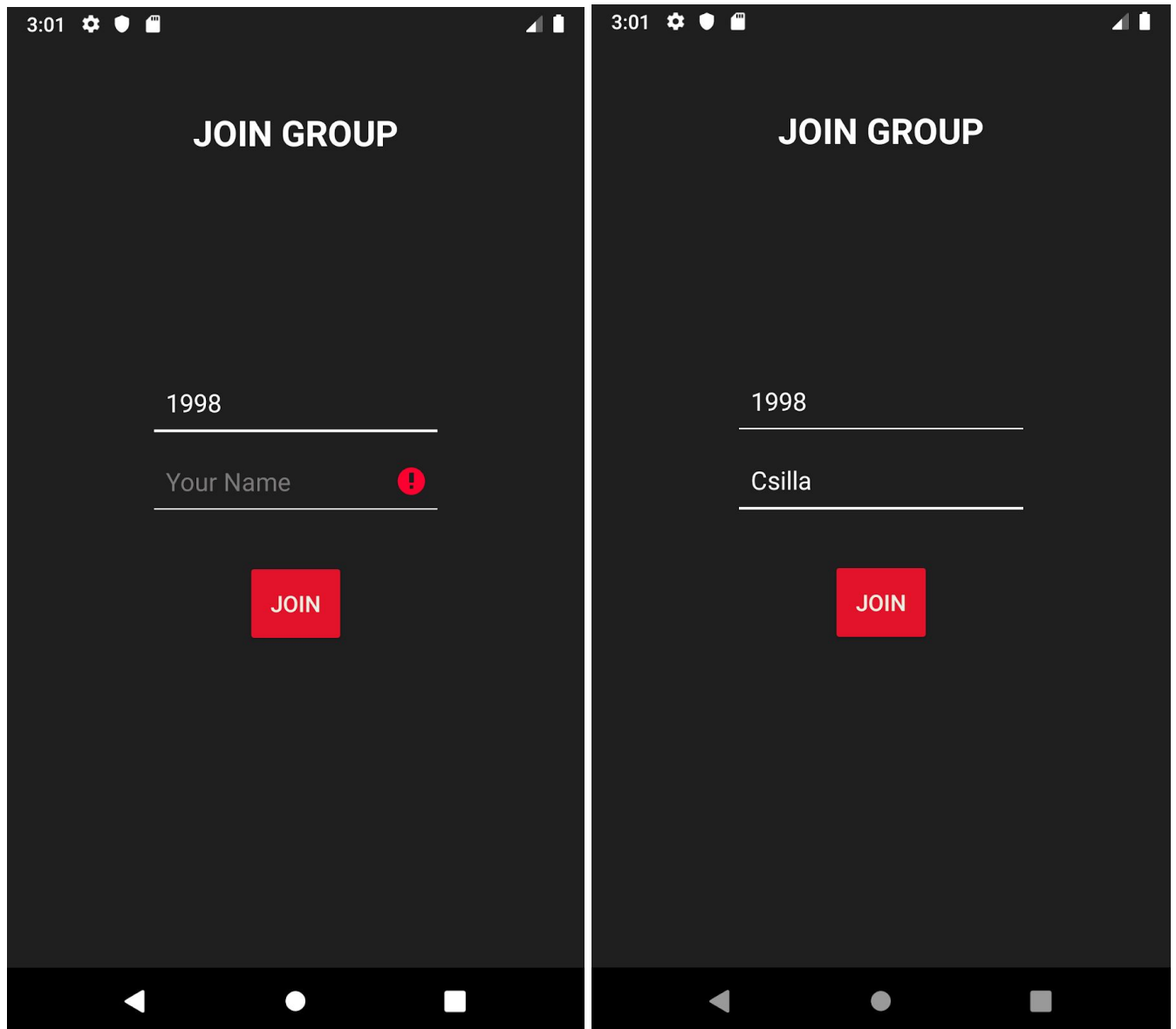
**Bende Csilla Boglárka**

2019 November

Informatika

## User

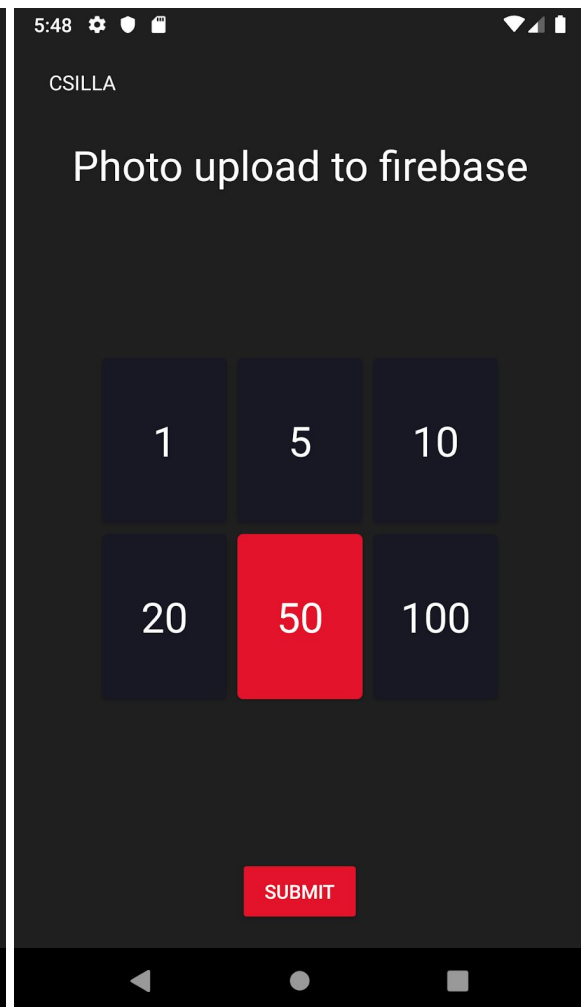
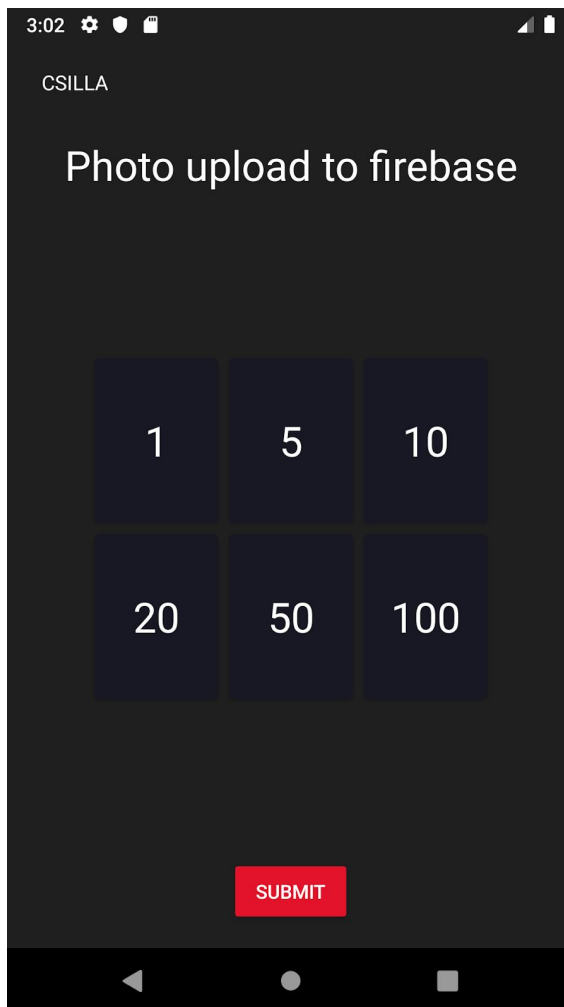
A User alkalmazás főoldalán csatlakozhat egy meglévő csoporthoz mindössze egy csoport ID-val és egy névvel. Az alábbi képen is látható.



A Join gombra kattintva bejelentkezik a csoportba ha sikeresen bejelentkezett megjelenik a User neve fent a bal sarokban, alatta pedig a kérdés és alattuk a kártyák melyek közül kiválaszthat egyet majd elküldheti választát. Ezután megjelenik hogy sikeres volt a választadás plusz egy gomb melyre rákattintva az

eredményeket tudjuk megnézni. A kódban így valósul meg.

```
private void connectSession(final String id, final String myName) {  
  
    conSession = false;  
    adminReference.addValueEventListener(new ValueEventListener() {  
        @Override  
        public void onDataChange(DataSnapshot dataSnapshot) {  
            conSession = false;  
            //itt nézi meg hogy valóban létezik olyan csoport es csak utána enged be  
            for (DataSnapshot item : dataSnapshot.getChildren()) {  
                if (item.child("groupId").getValue().equals(id)) {  
  
                    conSession = true;  
                    FragmentTransaction fr = getFragmentManager().beginTransaction();  
                    Fragment f = new FragmentShow();  
                    fr.addToBackStack(null);  
                    fr.replace(R.id.fragment_container, f);  
                    Bundle args = new Bundle();  
                    args.putString("groupId", id);  
                    args.putString("Name", myName);  
                }  
            }  
        }  
    });  
}
```

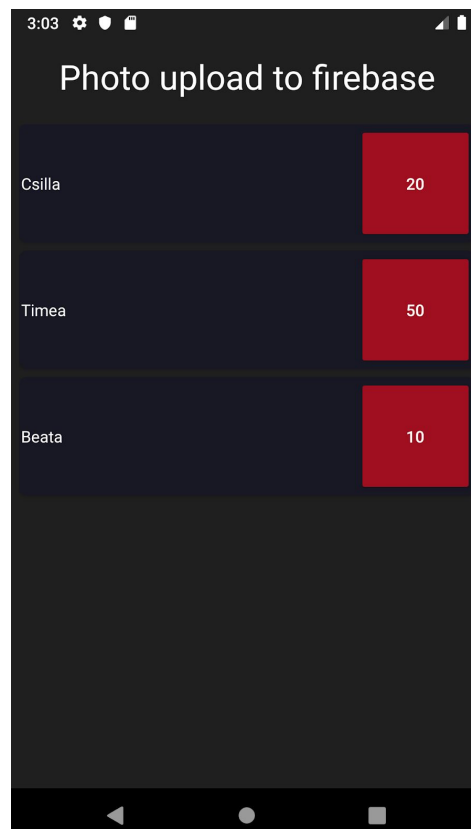
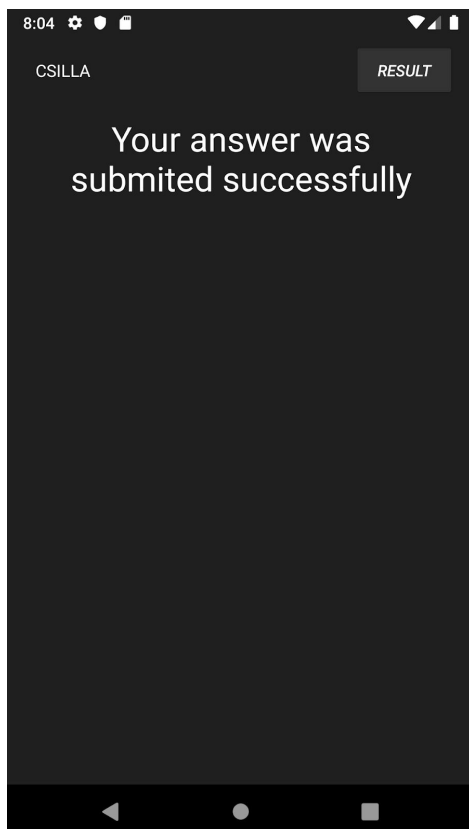


```

@Override
void initializeOnClickListeners() {
    answButton.setOnClickListener((v) -> {
        exists=false;
        llProgressBar.setVisibility(View.VISIBLE);
        answerReference.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for (DataSnapshot item: dataSnapshot.getChildren()) {
                    if(item.child("name").getValue().toString().equals(uName) &&
                        item.child("question").getValue().toString().equals(question) &&
                        item.child("groupId").getValue().toString().equals(groupId)) {
                        exists=true;
                    }
                }
                if(!exists) {
                    AnswerItem my = new AnswerItem(uName, answer, groupId, question);
                    String newKey = answerReference.push().getKey();
                    if (newKey != null) {
                        answerReference.child(newKey).setValue(my);
                        questionTextView.setText("Your answer was submitted successfully");
                        cards.setVisibility(View.GONE);
                        answButton.setVisibility(View.GONE);
                        llProgressBar.setVisibility(View.GONE);
                        showAnswers.setVisibility(View.VISIBLE);
                    }
                }
                else {
                    llProgressBar.setVisibility(View.GONE);
                    Toast.makeText(getContext(), text: "Mar volt valaszolva", Toast.LENGTH_SHORT).show();
                }
            }
        });
    });
}

```

Azt ellenőrzi, hogy az adott kérdésre már azonos névvel válaszoltak. A többi user válaszát is megtaláljuk erre a kérdésre az eredmények oldalán.



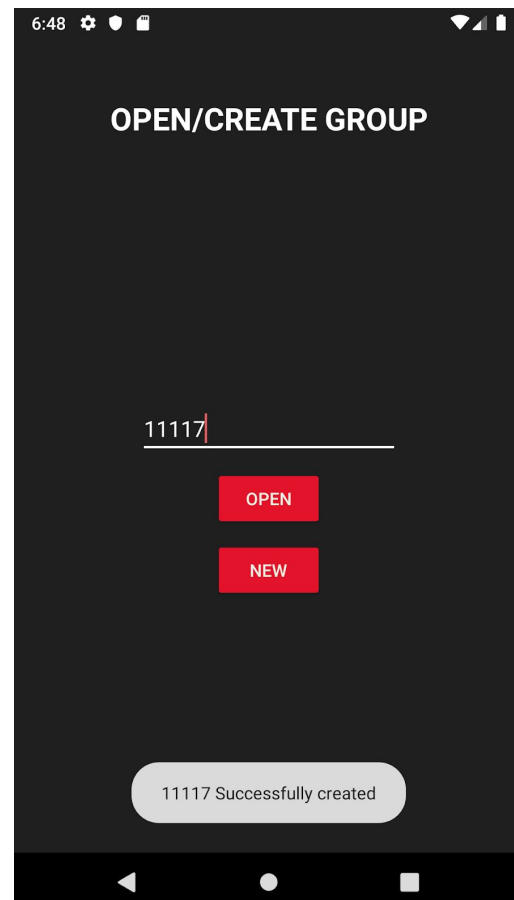
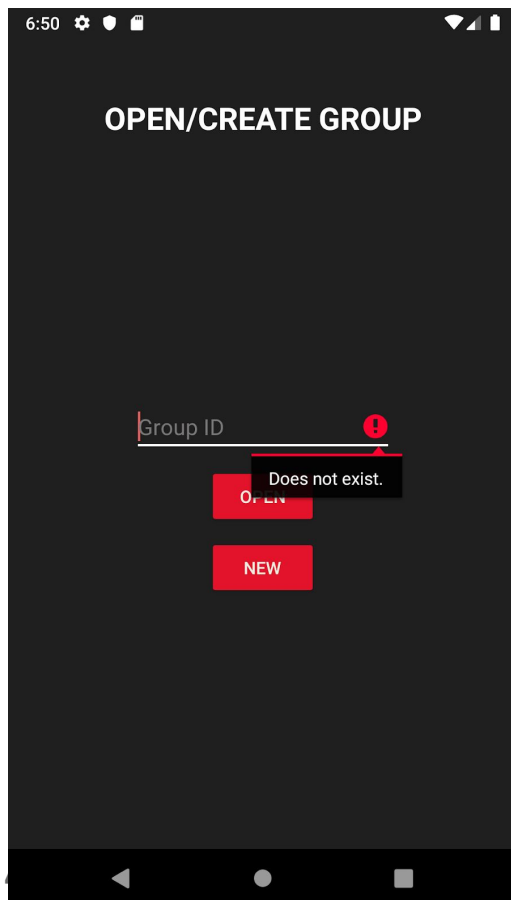
```

mAnswers = new ArrayList<>();
// az adatbázisból az összes adott választ
// kilistazza ugyanabból a csoportból a válaszolt kérdésre majd feltölti recycleviewt
answerRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        for (DataSnapshot item : dataSnapshot.getChildren()) {
            String id = item.child("groupId").getValue().toString();
            String squestion = item.child("question").getValue().toString();
            if (id.equals(groupId) && squestion.equals(question) ) {
                String name = item.child("name").getValue().toString();
                String answer = item.child("answer").getValue().toString();
                AnswerItem q1 = new AnswerItem(name,answer);
                mAnswers.add(q1);
            }
        }
    }
}

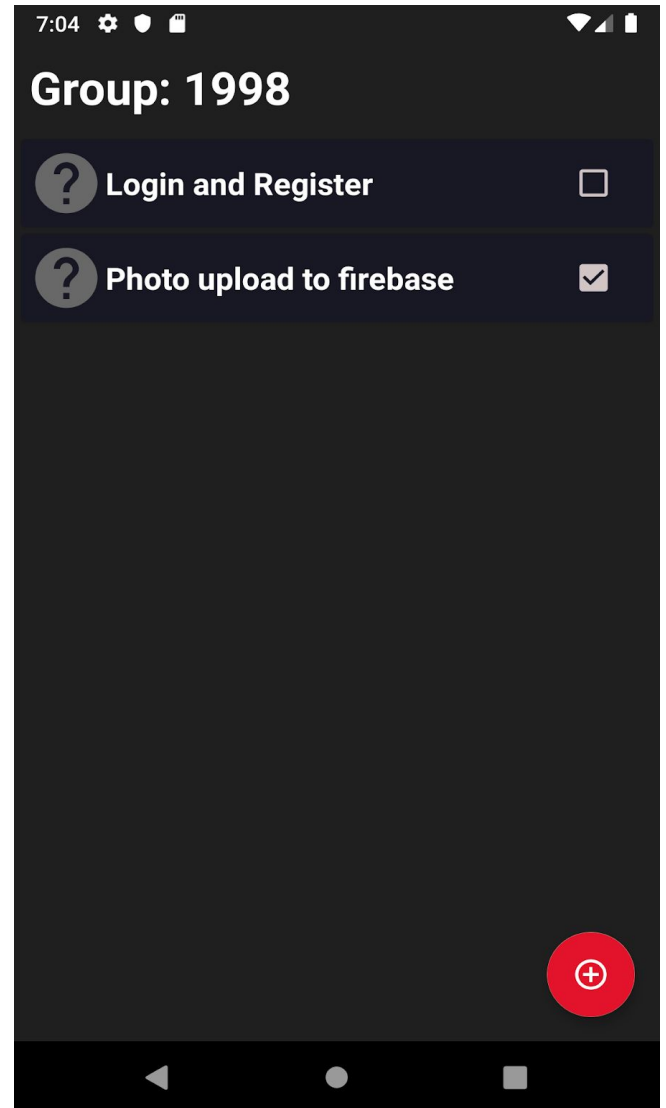
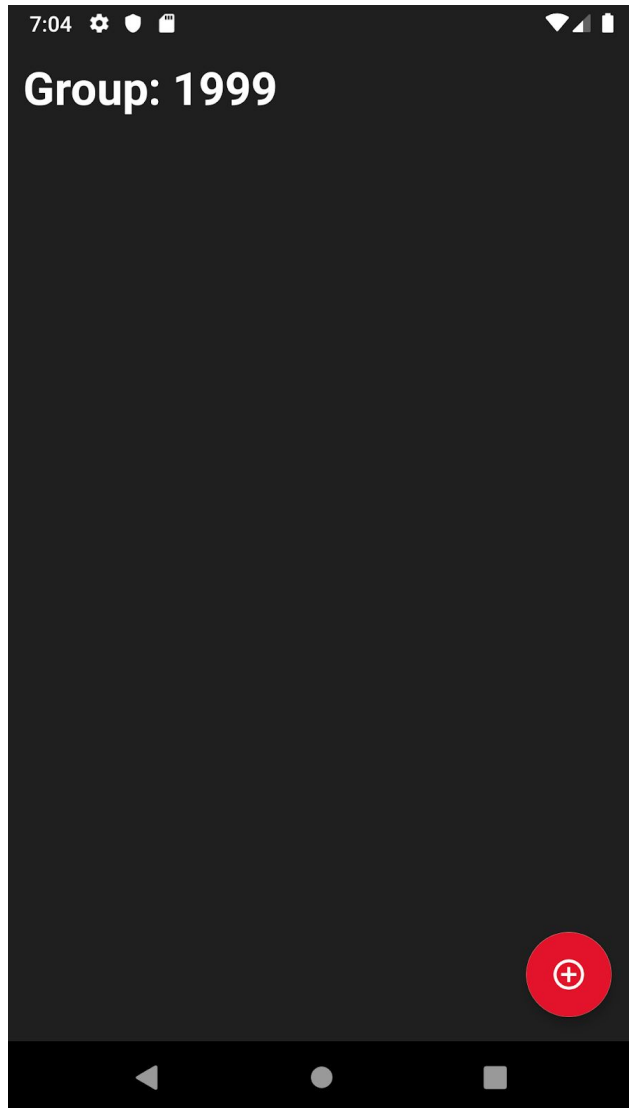
```

## Admin

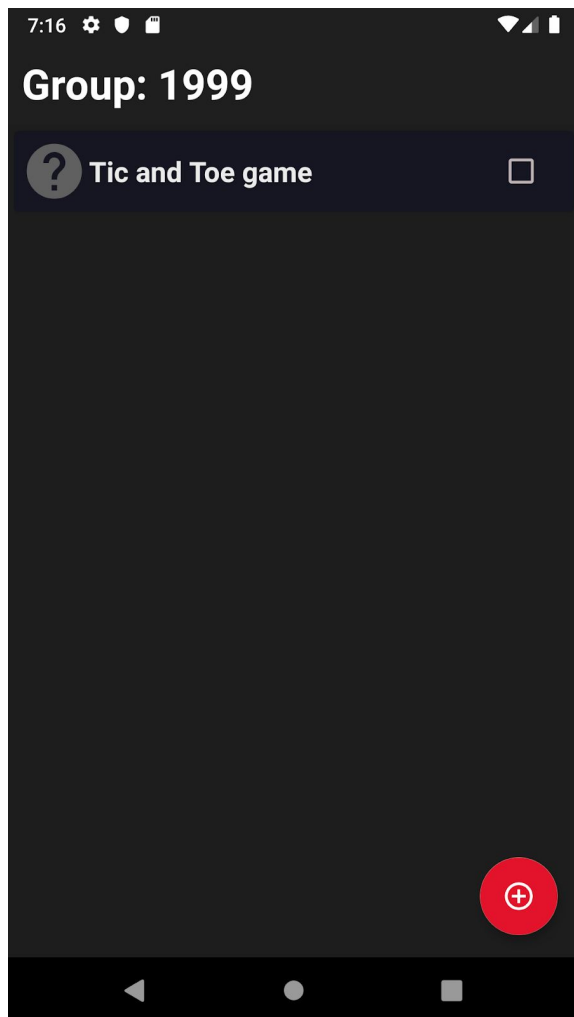
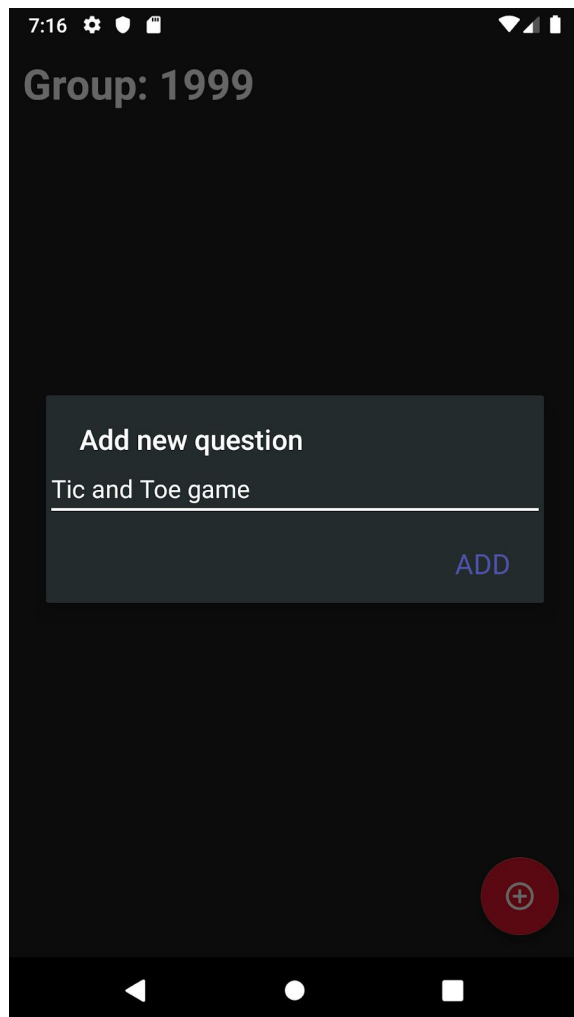
Az Admin alkalmazás főoldalán beléphetünk egy meglévő csoportba vagy létrehozhatunk újat. Új csoportot egy egész számokból álló ID-val lehet és csak akkor hozza létre ha még nem létezett előtte.



Ha új csoportba lépünk be csak csoport ID-ja és egy “+” gomb található amivel új kérdéseket tudunk a csoporthoz hozzáadni. Ha már egy kérdéssel létező csoportba lépünk be akkor megjelennek az összes addig hozzáadott kérdéseink.

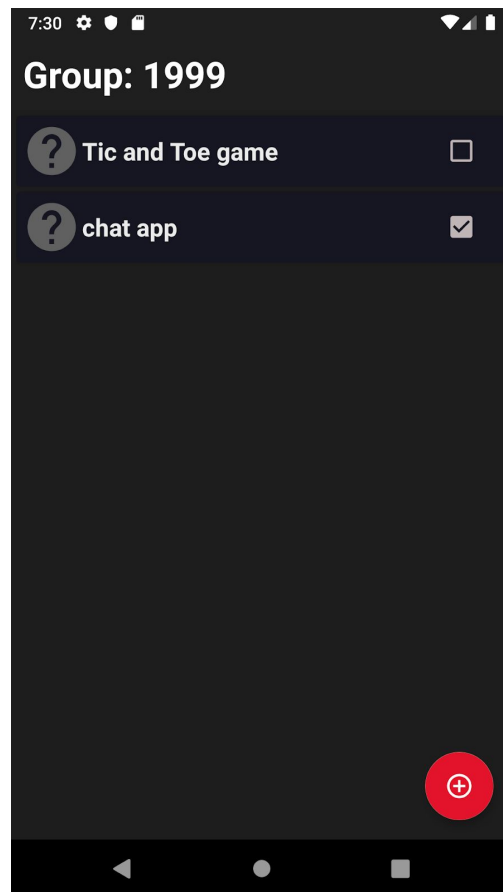
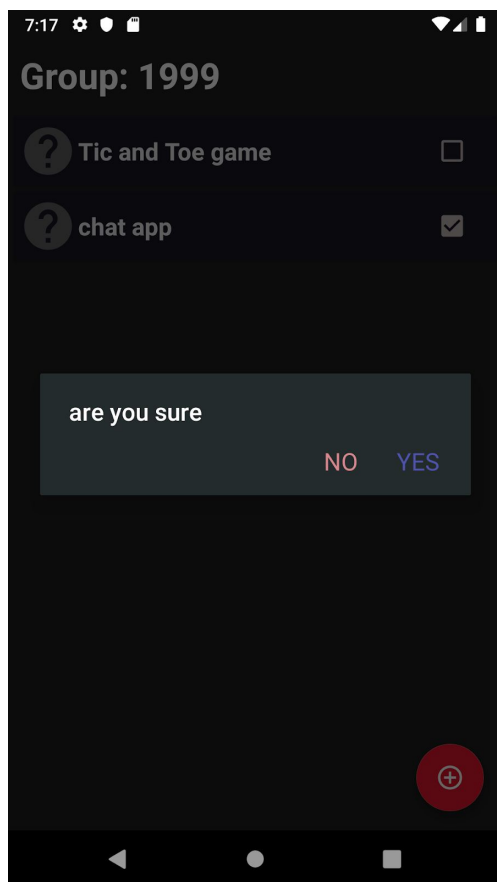


Ha új kérdést adunk hozzá és üresen hagyva akarjuk hozzáadni nem adja hozzá. Ugyan akkor az új hozzáadott kérdésünk megjelenik a többi között vagy mint első kérdés ha előtte nem volt több kérdés a csoportba.



```
AlertDialog.Builder builder = new AlertDialog.Builder(v.getContext(), R.style.MyDialogTheme);
builder.setTitle("Add new question");
final EditText question = new EditText(getContext());
question.setTextColor(Color.parseColor("colorString: "#FFFFFF"));
question.setBackgroundTintList(ColorStateList.valueOf(Color.parseColor("colorString: "#FFFFFF")));
LinearLayout layout = new LinearLayout(getContext());
layout.setOrientation(LinearLayout.VERTICAL);
layout.addView(question);
builder.setView(layout);
builder.setPositiveButton("add", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        question.setError(null);
        String mQuestion = question.getText().toString();
        if(mQuestion.isEmpty()) {
            question.setError("Please Enter question.");
        }
        else {
            applyQuestion(mQuestion);
            QuestionItem questions = new QuestionItem(mQuestion, active: false);
            FirebaseDatabaseHelper.Instance.InsertQuestion(questions, groupId);
        }
    }
});
builder.show();
```

Ha aktiválni szeretnénk egy kérdést a jelölő négyzetre kattintva megkérdi, hogy biztosan szeretnénk vagyis van lehetőségünk meggondolni.



```
int clickedPos = ((Integer)cb.getTag()).intValue();

if(cb.isChecked())
{
    if(lastChecked != null)
    {
        lastChecked.setChecked(false);
        mQuestionList.get(lastCheckedPos).setActive(false);
        final String q =mQuestionList.get(lastCheckedPos).question;
        questionsReference.addListenerForSingleValueEvent(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
                for(DataSnapshot item : dataSnapshot.getChildren()) {

                    if (item.child("question").getValue().toString().equals(q)) {
                        String key = item.getKey();
                        questionsReference.child(key).child("active").setValue("false");
                    }
                }
            }
        })
    }
}
```



Ha már volt előtte volt aktíválva valamelyik kérdés akkor azt állítja inaktívrá. Ha az eredményeket akarjuk megnézni akkor arra a kérdésre kattintva amelyiknek a válasza vagyunk kíváncsiak ezzel megtekinthető.

