Fakulta riadenia a informatiky

Žilinská univerzita v Žiline

**Vývoj aplikácií pre mobilné zariadenia**

Semestrálna práca

Patrik Trstenský

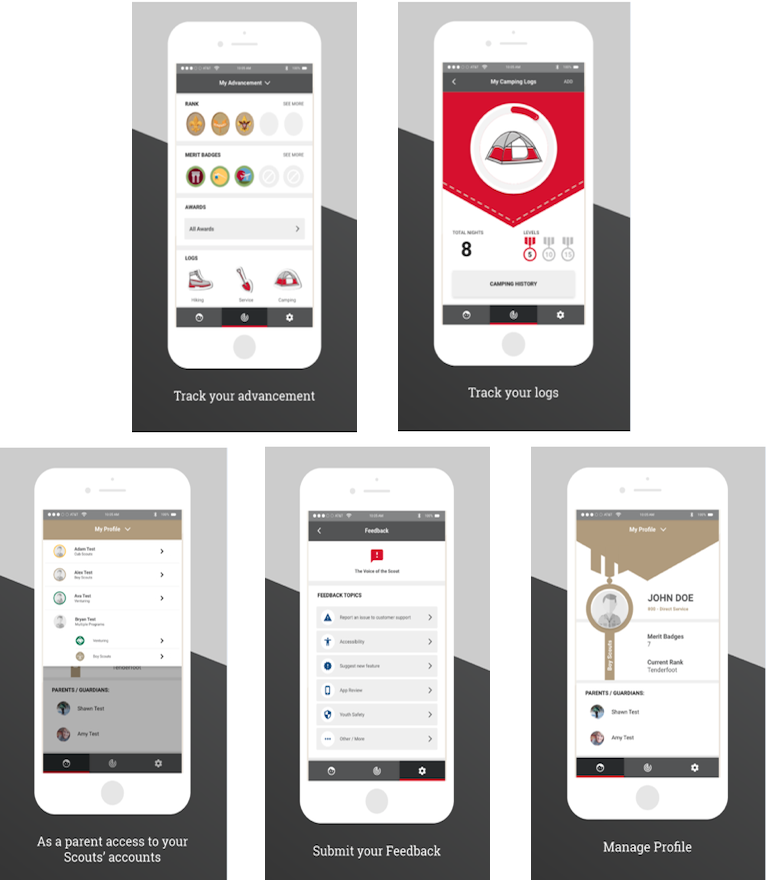
2019/2020 5ZY034

Zadanie semestrálnej práce

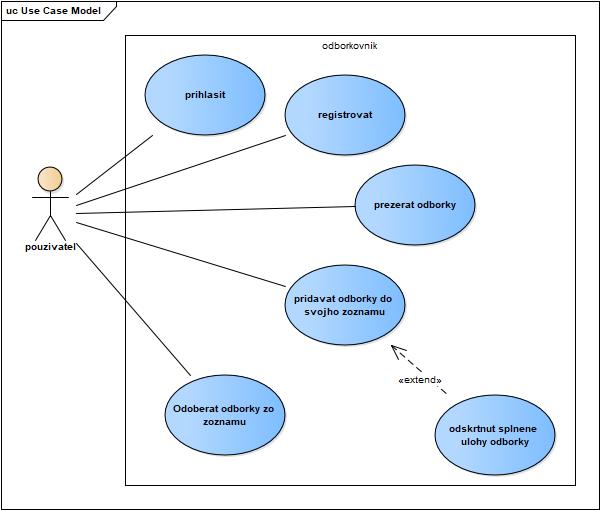
Zadaním mojej práce bolo navrhnúť mobilnú aplikácie pre skautskú komunitu. Keďže ako iste viete a stretli ste sa či už vo filme alebo naozaj zo skautmi a je myslím pre nich celkom silno charakteristické nosenie šerp, také tie dlhé látky na ktorých majú pripnuté odznaky ktoré získali ak splnili nejaké požadované úlohy. Našou úlohou bolo teda navrhnúť aplikáciu kde by si mohli používatelia prezerať tieto aplikácie a pozerať si aké musia splniť úlohy. A postupne si ich odškrtávať.

Podobné aplikácie

Podarilo sa nám nájsť podobnú aplikáciu ktorá akoby tiež zaznamenáva priebeh a získavanie odboriek a ďalších vyznamenaní. Musím povedať že v porovnaní s ňašou aplikaciu má širší záber a pekný nápad zo získavaním levlu za splnené odborky na ukážku posielam nejaké screeny z aplikácie Scouting.



Analýza

Na základe vyššie uvedej požiadavky sme definovali tieto základné funkcie pre použivateľa ktoré sa nám zdali najdôležitejšie a tiež pre to že sme sa nechceli púšťať do komplexnejšieho vypracovania, sme identifikovali tieto požiadavky ktoré sme si znazornili v diagrame pripadou použitia.

Návrh

Aplikácia bude mať základnú jednotku ktorým bude teda používateľ ktorý bude mať okrem svojich základných atribútom aj nejaké doplňujúce ako napríklad skautský zbor a pod. Bude mať možnosť zaregistrovať kde budem používať email, a samozrejme heslo. Predstava by bola že by sa po prihlásený zobrazil zoznam jeho odbočiek ktoré plní. Pri nich by si mohol prezerať aké obsahujú úlohy čiže nejaký detailnejší pohľad. To iste by platí aj pre ešte nepriradené aplikácie, u ktorých by platilo niečo podobné akurát by tam nebola možnosť zaznamenávania progresu, počet splnených úloh.

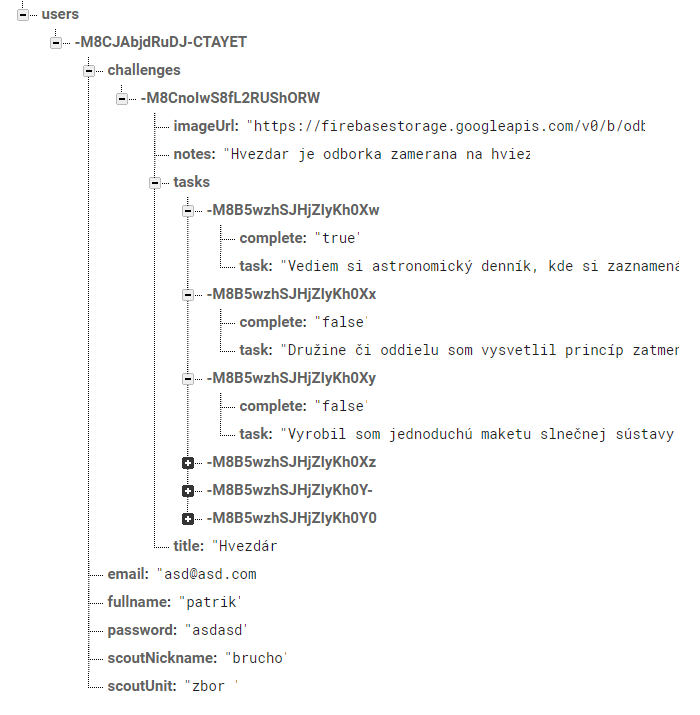
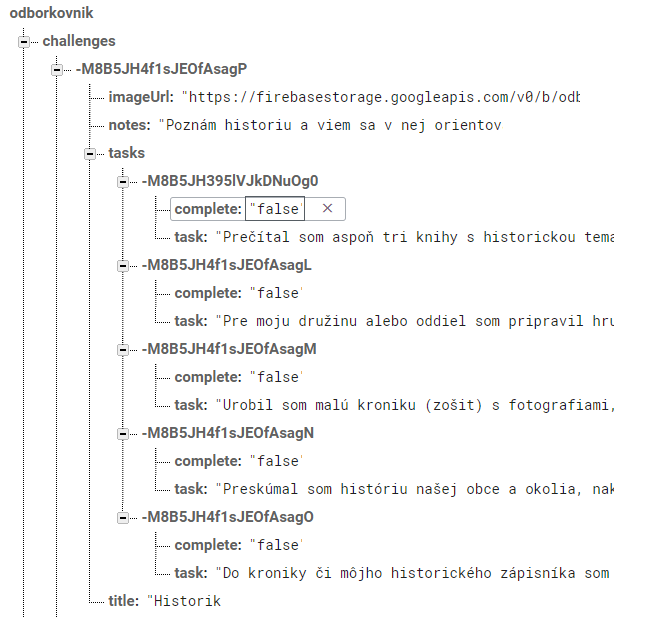
Implementácia

Keďže sme sa rozhodli aby sa naši používatelia prihlasovali do našej aplikácie a mali svoje konto budeme používať dátový server kde budeme tieto informácie ukladať. Po pátraní na internete sme sa rozhodli pre firebase, je to niečo nové a zaujímavé. Obrovskou výhodou bolo že mala už predefinované niektoré veci takže napríklad pri notifikácií používateľa nám stačilo iba jednoducho odchytiť odpoveď od servera samozrejme ak bola chybová a vypísať ju.

private void loginUser() {  
 String nickname = editTextEmail.getText().toString().trim();  
 String password = editTextPassword.getText().toString().trim();  
  
 if (TextUtils.*isEmpty*(nickname)) {  
 Toast.*makeText*(this,"Please enter email", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 if (TextUtils.*isEmpty*(password)) {  
 Toast.*makeText*(this,"Please enter password", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 progressDialog.setMessage("Login User...");  
 progressDialog.show();  
  
 fireBaseAuth.signInWithEmailAndPassword(nickname, password)  
 .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {  
 @Override  
 public void onComplete(@NonNull Task<AuthResult> task) {  
 if (task.isSuccessful()) {  
 FirebaseUser user = fireBaseAuth.getCurrentUser();  
 updateUI(user);  
 } else {  
 String msg = task.getException().getMessage();  
 msg = msg.split("\\.", 2)[0];  
 if ( task.getException() instanceof FirebaseAuthInvalidUserException ) {  
 Toast.*makeText*(MainActivity.this, "Email not in use", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(MainActivity.this, msg,Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
 progressDialog.cancel();  
 }  
 });

v podstate nám stačí zavolať iba jednu metódu a o všetko je postarané. Jedine čo je potrebné správne nastaviť dependenci a nastavenie ktoré potrebujeme do Grade priečinka.

Ďalšou veľmi veľkou zaujímavosťou od firebase je že je to real-time databáza, myslím že pre našú aplikáciu táto informácie moc nie je dôležitá ale mňa to zaujalo a tiež že je typu noSQL databáza bolo to niečo nové a troška trvalo kým sme si na to zvykli a vytvorili sme



Firebase neponúka iba ukladanie dát do databázy ale tiež aj spravu súborov kde sme si ukladali obrázky daných odbočiek a následne načítavali.

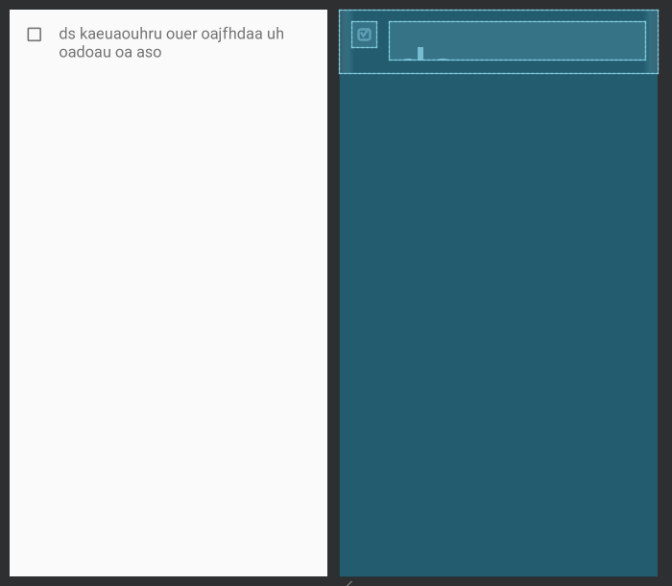
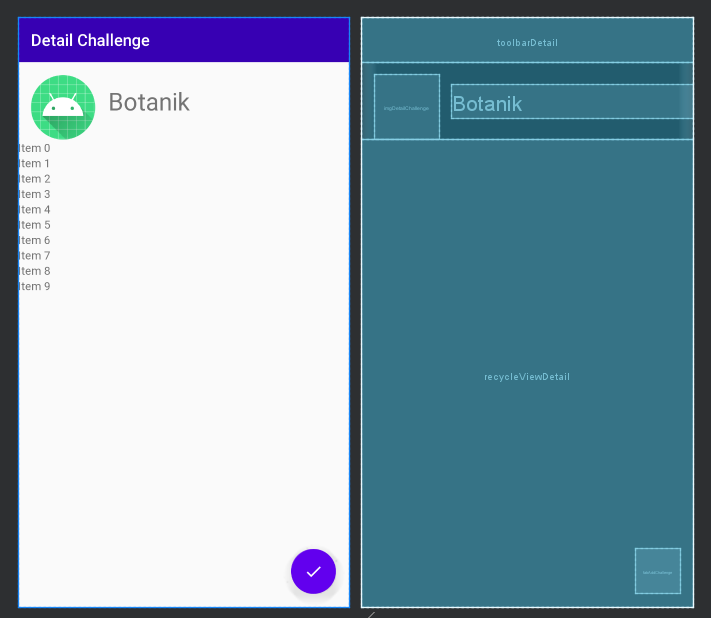
Ďalšou veľmi dôležitou a uľahčujúcou vecou v našej aplikácií bolo definovanie a mapovanie objektov v našej práci sme definovali tri objekty Používateľ, Odborka, a Uloha ktoré su aj pekne hierarchický usporiadané tak ako som napísal. Keď využívame ORM, môžeme jednoducho zapisovať a pracovať z databázou pomocou vytvorených objektov.

private void loadChallenges() {  
 refDatabase.addValueEventListener(new ValueEventListener() {  
 @Override  
 public void onDataChange(@NonNull DataSnapshot dataSnapshot) {  
 for (DataSnapshot ds : dataSnapshot.getChildren()) {  
 Challenge challenge = ds.getValue(Challenge.class);

potom môžeme jednoducho pracovať z objektmi z atribútmi ktoré sme dostali z databázy.

ďalej sme v mojej práci používali widgety na zobrazenie zoznamu ktoré majú rovnakú formu. Napríklad náš zoznám pre jednotlivé odborky sme si definovali, vytvorili vlastný pohľad a implementovali používame tri druhy, a to pre zoznam úloh ktoré vypisujeme a tatíčko či ju už splnil. Pre malý pohľad odboriek nadpis, koľko má úloh a obrázok.

Na ľavo môžeme vidieť aktivitu pre zobrazenie detailu odbory kde bude obrázok nadpis, a na pravo môžeme vidieť jednu úlohu z definovanýn nezmyselným textom len pre ukážku no v skutočnosti je prázdny. Takto sme to použili aj pri prehľadávaní odboriek len sme schovali checkbox.



Ďalej sme implementovali ďalšie zaujímavosti ako DiffUnitCallBack triedu ktorá slúži hlavne na to aby sme neprekresľovali celý zoznam, vymazanie a znova načítanie, ktorú potom pouívame v Adaptéroch pre naše zoznamy, ako kontrolu zmeny. Kde porovnávame atribúty. Ja konkrétne porovnávam z celého pohľadu iba jeden a to ten ktorý viem že sa bude meniť.