

SLOVENSKÁ TECHNICKÁ UNIVERZITA V BRATISLAVE
FAKULTA INFORMATIKY A INFORMAČNÝCH
TECHNOLÓGIÍ

Denis Štrbák, Tomáš Hanzel

AKTIVITY V OKOLÍ

MOBILNÉ TECHNOLOGIE A APLIKÁCIE

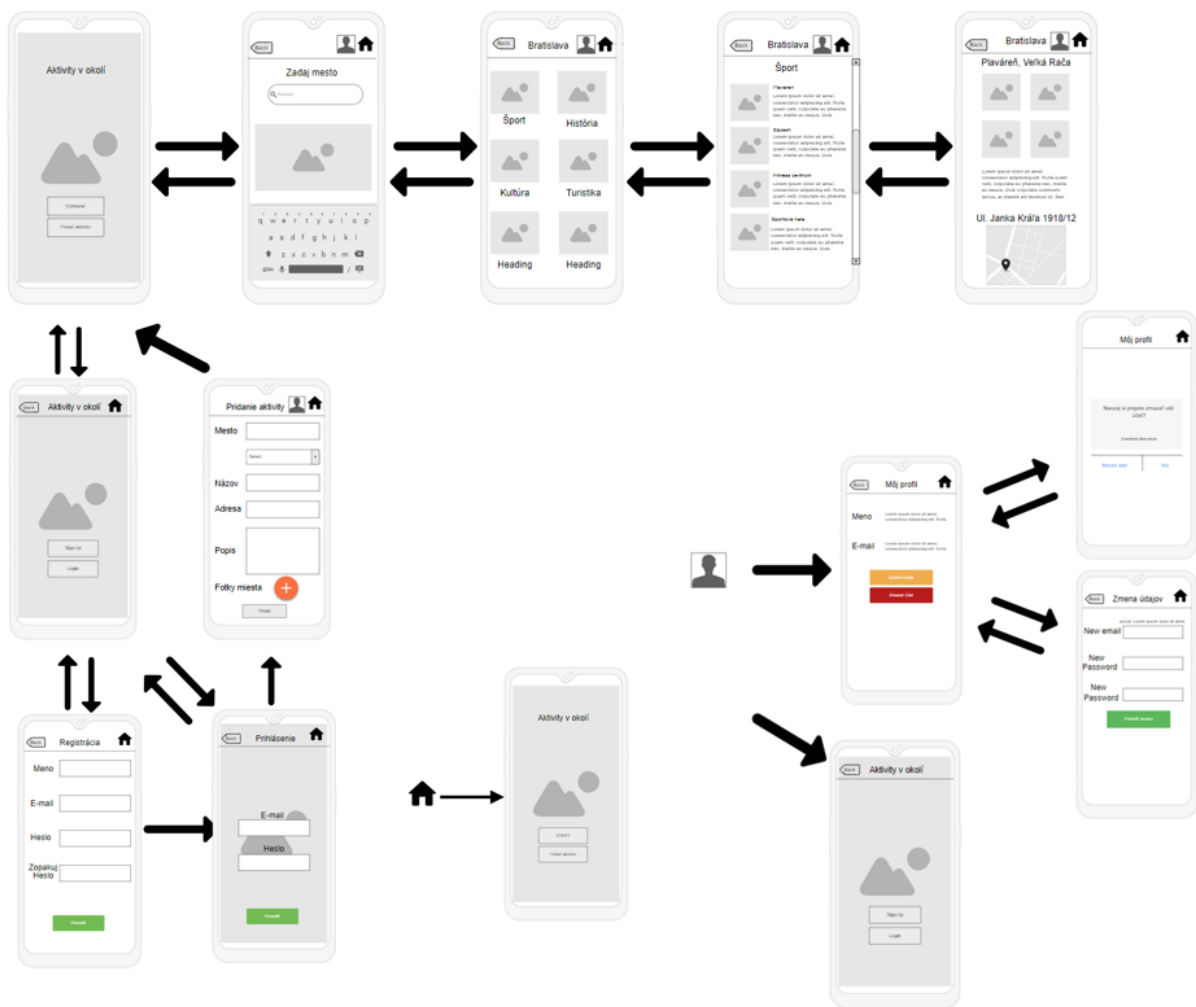
Študijný program: B-INFO 4

Cvičiaci: Ing. Viktor Šulák

Apríl 2021

Wireframes, REST API endpointy, návrh DB

Celá schéma wireframes:

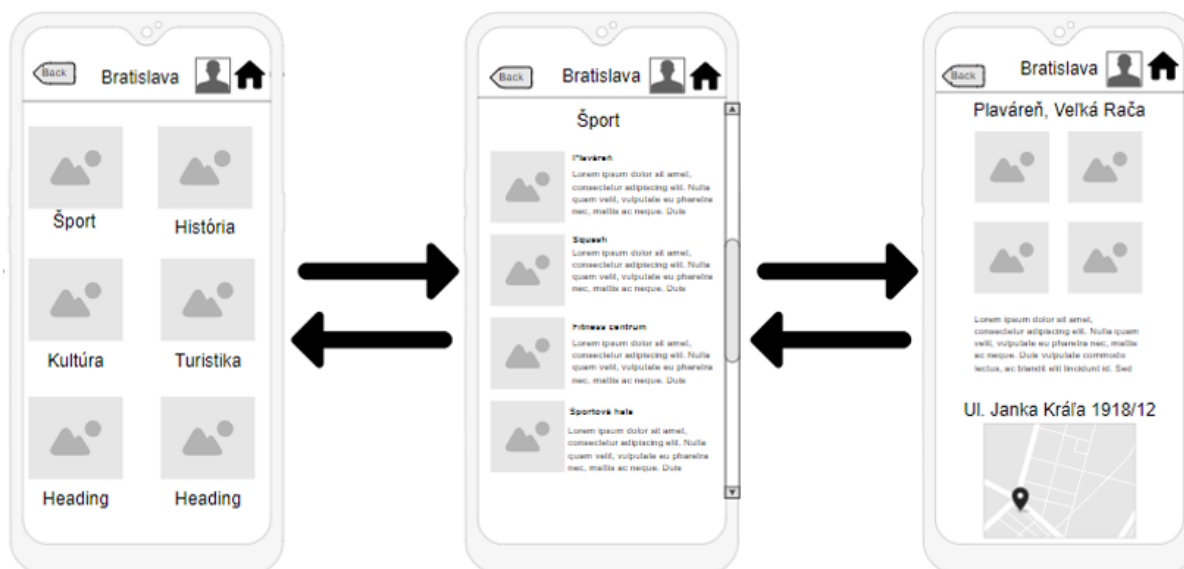


Postupne si prejdeme jednotlivé wireframes.



Na začiatku bude úvodná obrazovka, kde si môžeme vybrať či chceme aktivitu pridať alebo vyhľadať.

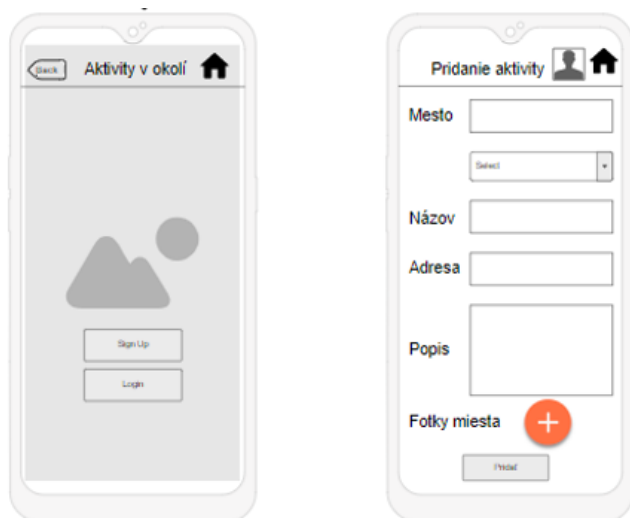
Keď klikneme na vyhľadať, tak budeme musieť zadať dané mesto, v ktorom chceme nájsť aktivitu.



Napíšeme napríklad Bratislava a nájde nám rôzne druhy aktivít, ktoré sa v danom meste dajú robiť. Budeme si môcť vybrať okruh aktivít, ako napríklad: Šport, história, kultúra, turistika...

Keď si vyberieme kategóriu, tak nám vyskočia už konkrétne veci k danému okruhu, napríklad po rozkliknutí Šport nám postupne vyskočí: Plaváreň, Squash, fitness centrum, športová hala..., tak ako je vidieť na strednom wireframe.

Keď už sa rozhodneme rozkliknúť napríklad aktivitu Plaváreň, tak nám vyskočí galéria danej aktivity, krátky popis o danej aktivite, adresa a mapa, kde sa nachádza, tak ako to môžeme vidieť na pravom wireframe.



Keď sa rozhodneme na začiatku pridať aktivitu, tak sa musíme prihlásiť alebo zaregistrovať. Pridávať aktivity sa bude dať len prihláseným užívateľom. Prehliadať

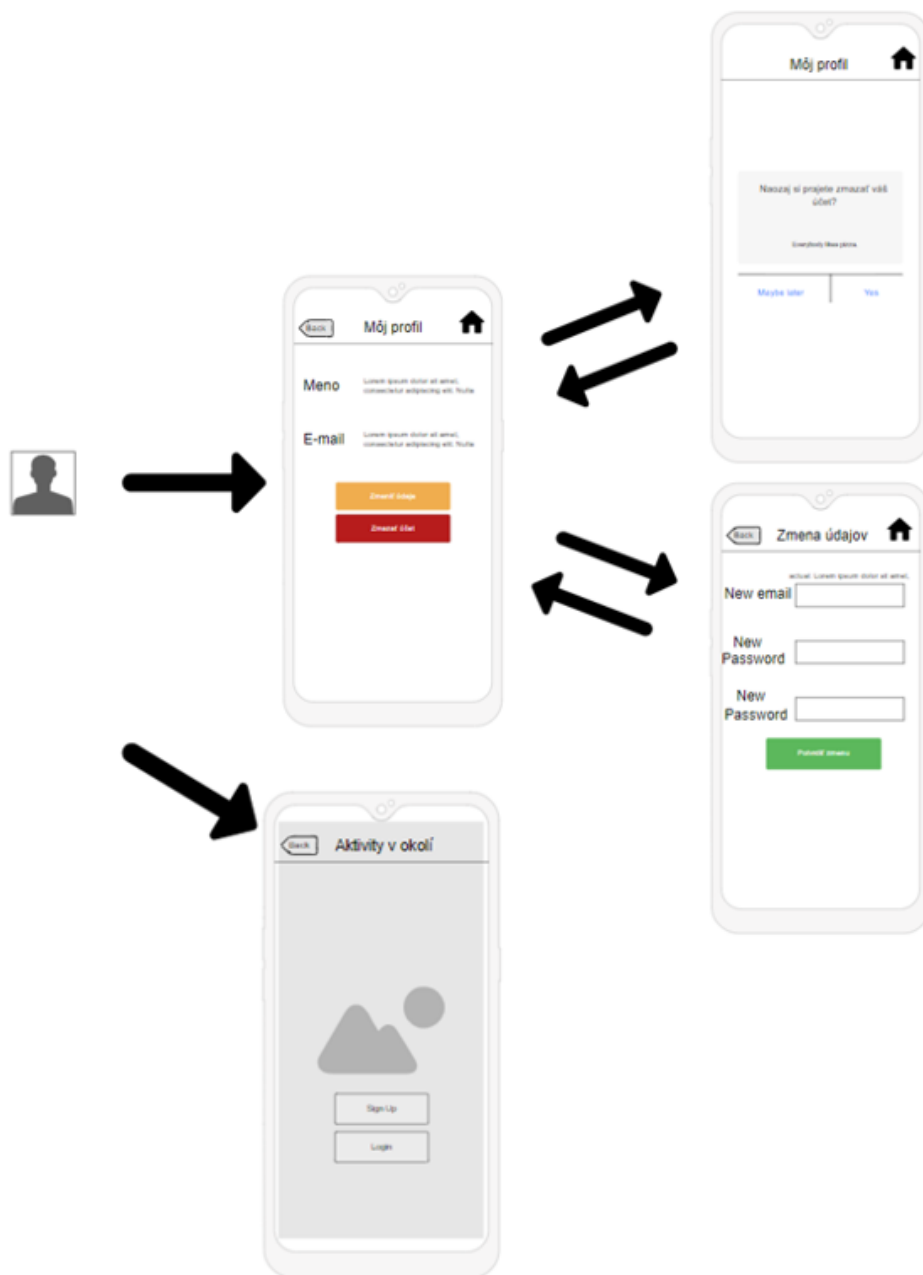
aktivity si bude môcť hocikto. Keď sa prihlásený používateľ rozhodne pridať nejakú aktivitu, tak bude musieť zadať mesto, vybrať druh aktivity, názov aktivity, adresu, popis a pridať aspoň 1 fotku, tak ako to ukazuje wireframe napravo.



Ak sa neregistrovaný používateľ rozhodne zaregistrovať do našej aplikácie, tak mu vyskočí wireframe vľavo, kde bude musieť zadať meno, e-mail a heslo. Keď sa zaregistrujeme, tak sa bude treba prihlásiť, prípadne aj bez registrácie sa dá prihlásiť a vyskočí wireframe ako je napravo. Ak sa zadajú zlé údaje (e-mail alebo heslo), tak to vyskočí ako je na wireframe vpravo a vypíše to: "Nesprávne používateľské údaje. Skús to ešte raz.".



Po kliknutí na hociktorom wireframe na domček, ktorý sa bude zobrazovať poväčšine v pravom hornom rohu, tak nás to hodí na úvodnú obrazovku ako je vyššie Wireframe, kde sa bude dať začať zase od začiatku, alebo pridať aktivitu.



Po kliknutí na hociktorom wireframe na avatar, ktorý sa bude nachádzať poväčšine tiež v pravom hornom rohu nás to môže presmerovať na 2 rozdielne wireframes.

1. Ak nie sme prihlásený
2. Ak sme prihlásený

Ak nebudeme prihlásený, tak nás to prehodí na wireframe, ktorý nám zobrazí Prihlásenie alebo registráciu účtu.

Ak budeme prihlásený, tak nás to hodí na náš účet, kde budú informácie o ňom. Meno a email, ktorý sme zadávali na začiatku pri registrácii účtu.

Bude sa dať takisto účet vymazať alebo Zmeniť údaje v ňom(email a password).

REST API ENDPOINTY:

Budeme potrebovať:

GET	/activities/activity	Returns one activity	↕
GET	/users/login	returns information about login	↕
PUT	/users/update	updates account	↕
POST	/activities/add	creates new activity	↕
DELETE	/users/delete	deletes user account	↕
GET	/cities	returns cities	↕
POST	/users/add	creates new user	↕
GET	/activities/	Returns list of activities	↕

GET

/activities/activity Returns one activity

↶

Returns activity of given type in a given city

Parameters

Try it out

Name	Description
id	returns activity by id
string	
(query)	<div>id - returns activity by id</div>

Responses

Code	Description	Links
200	activity	No links
	<div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "item1": { "city": "Bratislava", "name": "Bratislavský hrad", "address": "Ul. Janka Kráľa 1918/12", "description": "Lorem ipsum dolor", "photos": "string" } }]</pre></div>	
400	invalid parameter	No links

GET**/users/login** returns information about login

Returns data about login

Parameters[Try it out](#)**Name****Description****e-mail**

information about e-mail

string*(query)*

e-mail - information about e-mail

password

information about password

string*(query)*

password - information about password

Responses**Code****Description****Links**

200

Ok

No links

404

Bad password or e-mail

No links

PUT**/users/update** updates account

update user information

Parameters[Try it out](#)**Name****Description****id**

update e-mail or password in account

string*(query)***Request body**

application/json

**Account details**[Example Value](#) | [Schema](#)

```
{
  "e-mail": "hiworld@hiworld.com",
  "password": "1A2B3C4D"
}
```

Responses**Code****Description****Links**

200

Updated

No links

404

Not found id

No links

POST**/activities/add** creates new activity

adds new activity to the system

Parameters

Try it out

No parameters

Request body

application/json



Activity details

Example Value | Schema

```
{
  "city": "Bratislava",
  "name": "Bratislavský hrad",
  "address": "Ul. Janka Kráľa 1918/12",
  "description": "Lorem ipsum dolor",
  "photos": "string"
}
```

Responses

Code	Description	Links
201	item added	No links
400	invalid input, object invalid	No links
409	an existing item already exists	No links

DELETE**/users/delete** deletes user account

Deletes user account

Parameters[Try it out](#)

Name	Description
------	-------------

id	Delete user by this id
----	------------------------

string	
--------	--

(query)	
---------	--

Responses

Code	Description	Links
204	Deleted	No links
404	Id not found.	No links

GET

/cities returns cities

↩

Returns list of all cities

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>list of all cities</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "item1": { "id": 3524, "name": "Bratislava" }, "item2": { "id": 6565, "name": "Kosice" } }]</pre></div>	No links
400	Failed to retrieve data	No links

POST

/users/add creates new user

adds new user to the system

Parameters

Try it out

No parameters

Request body

application/json

User data to add

Example Value

Schema

```
{
  "email_address": "john.doe@gmail.com",
  "password": "sadfokoeklewfkldfb14"
}
```

Responses

Code	Description	Links
201	item created	No links
400	invalid input, object invalid	No links
409	an existing item already exists	No links

POST

/users/add creates new user



adds new user to the system

Parameters

Try it out

No parameters

Request body

application/json



User data to add

Example Value | Schema

```
{
  "email_address": "john.doe@gmail.com",
  "password": "sadfokoeklewfkldfb14"
}
```

Responses

Code	Description	Links
201	item created	No links
400	invalid input, object invalid	No links
409	an existing item already exists	No links

GET

/activities/ Returns list of activities

Returns list of activities of given type in a given city

Parameters

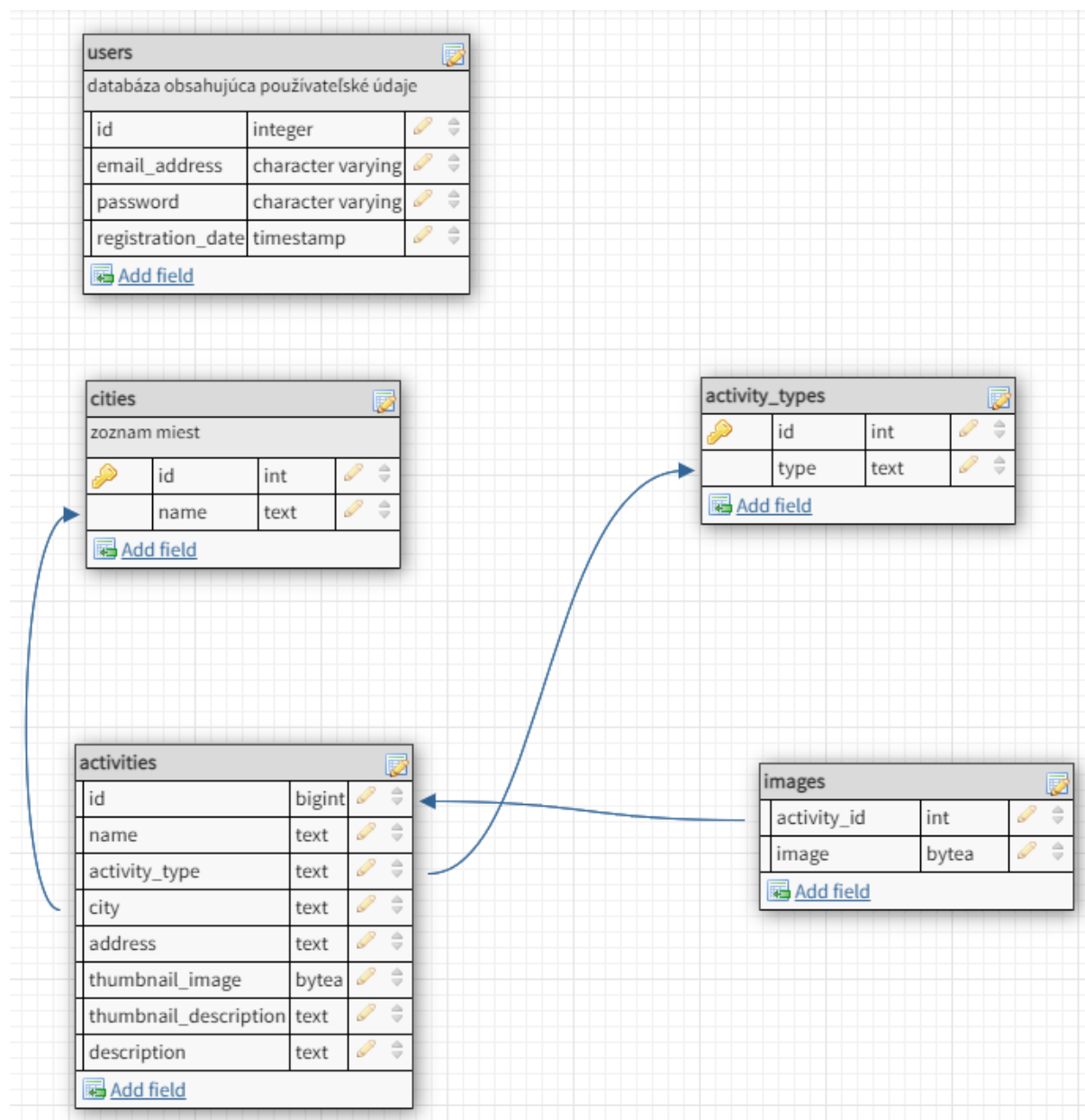
Try it out

Name	Description
city	Collect results from this city
string (query)	<div>city - Collect results from this city</div>
activity_type	Collect activities of this type
string (query)	<div>activity_type - Collect activities of this type</div>

Responses

Code	Description	Links
200	<div>list of activities</div> <div>Media type</div> <div>application/json</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "item1": { "id": "1234", "name": "Bratislavský hrad", "thumbnail_description": "Najväčší hrad v strednej európe", "thumbnail_image": "string" }, "item2": { "id": "1235", "name": "Dom Sv. Tomáša", "thumbnail_description": "Najstarší kostol v meste", "thumbnail_image": "string" } }]</pre></div>	No links
400	<div>invalid parameter</div>	No links

Návrh databázy:



Akceptačné testy

Akceptačné testy aplikácie:

Test 1: Vyhľadanie aktivity	
Vstupné podmienky:	Používateľ nemusí byť prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	V aplikácii zobrazí konkrétnu aktivitu pre používateľa, kde je vidieť jej fotky, adresu, popis, názov
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo VYHLÁDAŤ.2. Používateľ zadá MESTO, v ktorom chce danú aktivitu vyhľadať.3. Používateľ si vyberie druh aktivity(Šport, história...)4. Používateľ si vyberie konkrétnu aktivitu.5. V aplikácii sa zobrazí konkrétna aktivita s údajmi
Výsledok:	PASS

Test 2: Registrácia nového používateľa	
Vstupné podmienky:	Používateľ nie je prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	V aplikácii je založený nový účet, do ktorého sa používateľ môže prihlásiť a pridať aktivitu, príp. si meniť údaje vo svojom profile na svojom účte
Postup:	<ol style="list-style-type: none">1. Používateľ stlačí tlačidlo AVATARU na hociktovej stránke alebo na úvodnej obrazovke stlačí PRIDAŤ AKTIVITU.2. Používateľ zadá údaje, ktorými sa chce zaregistrovať: Meno, E-mail, Heslo.

	3. Používateľ sa môže prihlásiť do svojho účtu.
Výsledok:	PASS

Test 3: Zmena údajov v účte	
Vstupné podmienky:	Používateľ môže, ale aj nemusí byť prihlásený, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľ vidí svoje zmenené údaje vo svojom účte
Postup:	<ol style="list-style-type: none"> 1. Používateľ sa prihlási do svojho účtu 2. Používateľ klikne na svojho avatara, po ktorom sa mu zobrazí jeho profil a údaje o jeho účte. 3. Používateľ klikne na tlačidlo Zmeniť Údaje. 4. Používateľ si zadá svoj nový e-mail alebo heslo, ktoré chce zmeniť. 5. Používateľ klikne na tlačidlo Potvrdiť. 6. Používateľ vidí vo svojom účte zmenené údaje.
Výsledok:	PASS

Test 4: Prihlásenie do účtu	
Vstupné podmienky:	Používateľ nie je prihlásený do aplikácie, aplikácia potrebuje pripojenie na internet

Výstupné podmienky:	Používateľovi vyskočí, že zadal zlé meno alebo heslo, resp. že sa spolu tieto údaje nezhodujú
Postup:	<ol style="list-style-type: none"> 1. Používateľ klikne na AVATAR hore v aplikácii 2. Používateľ klikne na Prihlásenie 3. Používateľ zadá e-mail a heslo na prihlásenie 4. Používateľovi vyskočí, že zadal zlé údaje
Výsledok:	FAIL

Test 5: Pridanie aktivity	
Vstupné podmienky:	Používateľ musí byť prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľovi vyskočí, že nezadal všetky potrebné údaje nato, aby mohla byť aktivita pridaná
Postup:	<ol style="list-style-type: none"> 1. Používateľ sa musí prihlásiť do účtu 2. Používateľ klikne na tlačidlo PRIDAŤ AKTIVITU 3. Používateľ zadá MESTO, DRUH AKTIVITY 4. Používateľ zadá NÁZOV, ADRESU 5. Používateľ zadá POPIS a nahrá FOTO 6. Používateľovi vyskočí, že nezadal všetky potrebné údaje
Výsledok:	FAIL

Akceptačné testy pre back-end:

Test 1: Zobrazenie aktivít určitého typu	
Vstupné podmienky:	Používateľ nemusí byť prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľovi sa zobrazia ukážky aktivít určitého typu
Postup:	<p>1. Používateľ najprv potrebuje zoznam miest z ktorých si môže vybrať</p> <p>Request - GET {url}/cities</p> <p>Response - 200 application/json</p> <pre>[{ "item1": { "id": 3524, "name": "Bratislava" }, "item2": { "id": 6565, "name": "Kosice" } }]</pre> <p>2. Následne si používateľ vyberie typ aktivity</p> <p>Request - Get</p> <p>{url}/activities/?city=Bratislava&activity_type=history</p> <p>Response - 200 application/json</p> <pre>[{ "item1": { "id": "1234",</pre>

	<pre> "name": "Bratislavský hrad", "thumbnail_description": "Najväčší hrad v strednej európe", "thumbnail_image": "string" }, "item2": { "id": "1235", "name": "Dom Sv. Tomáša", "thumbnail_description": "Najstarší kostol v meste", "thumbnail_image": "string" } }]</pre>
Výsledok:	PASS

Test 2: Pridanie aktivity	
Vstupné podmienky:	Používateľ musí byť prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľ pridá aktivitu do systému

Postup:	<p>1. Používateľ sa najskôr musí prihlásiť do aplikácie</p> <p>Request - GET {url}/users/login/?email=john.doe@gmail.com&password=qwerty123</p> <p>Response - 200 OK</p> <pre>{ "id": "1235" }</pre> <p>2. Následne vyplní potrebné polia, vyberie obrázok a klikne tlačidlo potvrdiť.</p> <p>Request - POST {url}/activites/add</p> <p>Request body - {</p> <pre> "city": "Bratislava", "name": "Bratislavský hrad", "address": "Ul. Janka Kráľa 1918/12", "description": "Lorem ipsum dolor", "photos": "string" }</pre> <p>Response - 201 item added</p>
Výsledok:	PASS

Test 3: Aktualizácia profilu	
Vstupné podmienky:	Používateľ nie je prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľ si zmení osobné údaje v účte
Postup:	<p>1. Používateľ sa najskôr musí prihlásiť do aplikácie</p> <p>Request - GET {url}/users/login/?email=john.doe@gmail.com&password=qwerty123</p> <p>Response - 200 OK</p>

	<pre>{ "id": "1235" }</pre> <p>2. Následne vyplní zmenené údaje</p> <p>Request - PUT</p> <p>{url}/user/update/?id=1235</p> <p>Request body - {</p> <pre> "e-mail": "hiworld@hiworld.com", "password": "1A2B3C4D" }</pre> <p>Response - 200 Updated</p>
Výsledok:	PASS

Test 4: Registrácia a prihlásenie používateľa	
Vstupné podmienky:	Používateľ nie je prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľovi sa nepodarí prihlásiť do aplikácie
Postup:	<p>1. Používateľ vyplní registračné údaje</p> <p>Request - POST</p> <p>{url}/users/add</p> <p>Request body -</p> <pre>{ "email_address": "john.doe@gmail.com", "password": "sadfokoeklewfkldfb14" }</pre> <p>Response - 201 Item created</p> <p>2. Používateľ sa následne pokúša prihlásiť do aplikácie so zlým heslom</p>

	Request - GET {url}/users/login/?email= john.doe@gmail.com &password=qwerty123 Response - 404 Bad password or email
Výsledok:	FAIL

Test 5: Zmazanie účtu	
Vstupné podmienky:	Používateľ nie je prihlásený v aplikácii, aplikácia potrebuje pripojenie na internet
Výstupné podmienky:	Používateľovi sa nepodarí vymazať účet
Postup:	<ol style="list-style-type: none"> 1. Používateľ sa najskôr musí prihlásiť do aplikácie Request - GET {url}/users/login/?email=john.doe@gmail.com&password=qwerty123 Response - 200 OK <pre>{ "id": "1235" }</pre> 2. Používateľ sa následne snaží vymazať svoj účet vo svojom profile, ale prvý request mu vrátil zlé ID a nejde mu to, pretože také ID neexistuje Request - DELETE {url}/users/delete/1235 Response - 404 Not Found <pre>{ "id": "1235" }</pre>
Výsledok:	FAIL

Backend

API VOLANIA

V našom backende sme si urobili 8 API volaní, ktoré budeme potrebovať na celý chod nášho systému.

1. USER_REGISTRATION
2. USER_DELETE
3. USER_UPDATE
4. USER_LOGIN
5. GET_CITIES
6. GET_ACTIVITIES
7. GET_ACTIVITY
8. ADD_ACTIVITY

1. USER_REGISTRATION - POST

Toto API volanie budeme potrebovať na registráciu používateľa do systému.

Skontroluje sa, že či sú v aplikácii vyplnené všetky údaje pri registrácii a či je e-mail vyplnený správne, teda obsahuje dobrý formát, kde sa nachádza @ atď.

Následne sa skontroluje, že či sa v databáze už nenachádza tento e-mail, resp. či už tam nie je zaregistrovaný účet s týmto e-mailom.

Pokiaľ Áno, tak to vyhodí error 409 Conflict. Ak sa nenachádza daný e-mail v systéme, tak ho pridá do databázy, vytvorí mu token a registrácia je hotová. Vyhodí to 201 Created.

2. USER_DELETE - DELETE

Toto API volanie sa bude využívať na zmazanie účtu z databázy, teda z aplikácie. Na základe tokenu sa bude porovnávať, či už sa nachádza v databáze. Ak sa daný token nachádza v databáze, tak to účet z databázy vymaže a vyhodí status 204.

Pokiaľ sa, ale takýto token nenachádza v databáze, tak to účet samozrejme nemá zmazať, ktorý, tak to vyhodí status 404 Not Found.

3. USER_UPDATE - PUT

Ako už znie aj z názvu, tak toto API volanie sa bude používať na update usera. Teda, keď si bude chcieť používateľ zmeniť svoje osobné údaje v účte. Najskôr sa samozrejme zisti pomocou tokenu, že či je daný používateľ prihlásený v nejakom účte. Ak nie, tak vyhodí status 400 Bad Request. Ďalej sa bude kontrolovať, že či používateľ nenechal prázdne údaje a teda vyplnil zmenu e-mailu alebo hesla(zadať musí oboje). Následne sa skontroluje, či sa e-mail nachádza v dobrom formáte. Ak všetko prebehne úspešne, tak sa ešte skontroluje, že či sa už daný účet nenachádza v databáze zaregistrovaných používateľov. Ak sa nachádza, tak vyhodí status 409 Conflict. Pokiaľ sa nenachádza, tak sa updatnu používateľské údaje a všetko prebehne tak ako má. Vráti to status code 200 OK.

4. USER_LOGIN - POST

Toto API volanie sa bude používať pri prihlásení používateľa do aplikácie. Z body sa načíta e-mail a heslo, ktorým sa používateľ prihlási. Skontroluje sa, či sa daný účet nachádza v databáze. Ak sa nenachádza, tak to vráti status code 404 Not Found. Ak sa tam nachádza, tak to vráti token používateľa, teda sa tým potvrdí, že takýto účet spolu s heslom sa nachádzajú v databáze. Vráti to status code 200 OK.

```
1  {
2    "token": "Nq9pyYPMTPBsRHmW"
3  }
```

5. GET_CITIES - GET

Toto API volanie bude vracať názov mesta a id daného mesta z databázy. Vráti to všetky mestá, ktoré sa nachádzajú v databáze. Bude sa používať, keď si používateľ bude vyberať danú aktivitu a na začiatku si určuje mesto, z ktorého daná aktivita bude. Vráti dané mestá a takisto aj status code 200 OK.

```
{
  {
    "id": 1,
    "name": "Bratislava"
  },
  {
    "id": 2,
    "name": "Tinaava"
  },
  {
    "id": 3,
    "name": "Piešťany"
  }
}
```

Toto API volanie nám slúži na pridanie aktivity do databázy. Najskôr sa skontroluje, či daný používateľ je prihlásený a teda má svoj token, pretože aktivitu môžu pridávať iba prihlásení používatelia. Následne sa kontroluje, či sa dané mesto, ktoré používateľ pridal nachádza v databáze s mestami, ak nie, tak sa mesto pridá do databázy cities. Potom sa kontroluje, či sa už daná aktivita nenachádza v databáze.

Ak sa nachádza, tak vrátime status code 409 Conflict, pretože nie je možné pridať takú istú aktivitu. Keď sa nenachádza, tak ju tam vložíme a vrátime status code 201 Created.

ZMENY OPROTI NÁVRHU

1. Rozhodli sme sa zmeniť LOGIN API volanie z GET na POST.
2. Do databázy images sme vložili stĺpec ID.
3. Pri registrácii sa každému používateľovi vytvorí unikátny TOKEN.
4. Na prihlasovanie sme sa rozhodli použiť TOKEN namiesto ID používateľa.
5. DELETE používateľa robíme na základe TOKENU a nie ID používateľa.

CESTA K REPOZITÁRU K BACKENDU

<https://github.com/strbakdenis/mtaa-strbakd-hanzelt>

Frontend

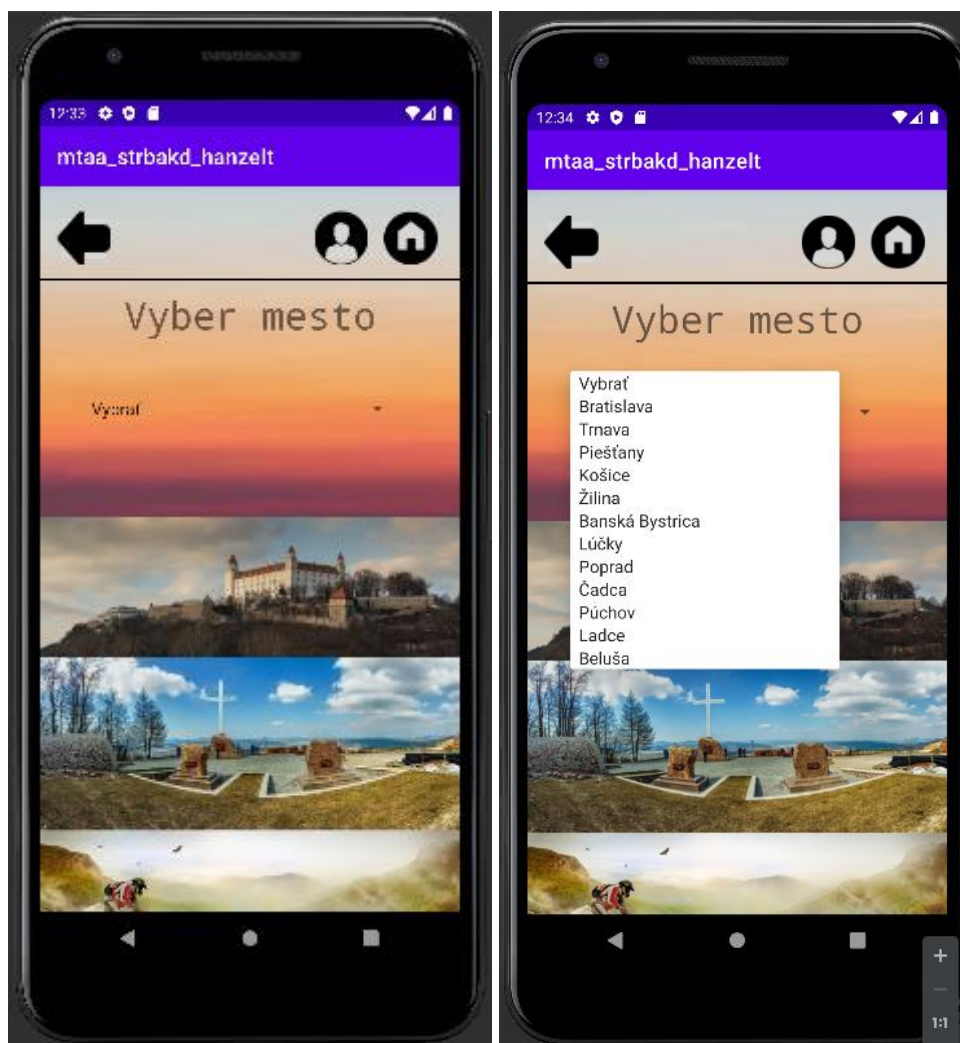
Po zapnutí aplikácie sa nám zobrazí úvodná obrazovka, na ktorej je vidieť úvodný nadpis aplikácie „Aktivity v okolí“.

Máme na výber 2 možnosti, z ktorých si môžeme vybrať ako budeme ďalej pokračovať.

1. Stlačíme tlačidlo Vyhľadať a ideme si nájsť nejakú konkrétnu aktivitu.
2. Keď stlačíme tlačidlo Pridať aktivitu, tak sa môžu udiť 2 rôzne situácie.
 - Nie sme prihlásený a hodí nás to na prihlásenie/registráciu.
 - Sme prihlásený a hodí nás to na pridanie aktivity, kde sa zadávajú údaje o danej aktivite.



Ak si vyberieme 1. možnosť a teda ideme si vyhľadať nejakú konkrétnu aktivitu, tak nám vyskočí obrazovka, na ktorej si môžeme vybrať konkrétne mesto, ktoré sa nachádza v databáze.



Po výbere mesta nás to automaticky prehodí do ďalšej obrazovky, v ktorej sa zobrazí výber typu aktivity v danom konkrétnom meste.

Ak by sme náhodou zabudli aké mesto sme vybrali, tak nám to v strede obrazovky vypisuje konkrétne mesto, na ktoré sme klikli.

Prípadne, ak sme klikli nechtiac na iné mesto ako sme chceli, alebo sme si to rozmysleli a chceme si vybrať iné mesto, tak stačí kliknúť na tlačidlo šípky v ľavo hore, ktorá nás vráti o jeden krok späť a môžeme si znova vybrať. (Toto platí vo všetkých obrazovkách, že po kliknutí tlačidla späť sa vrátíme späť o jeden krok.)

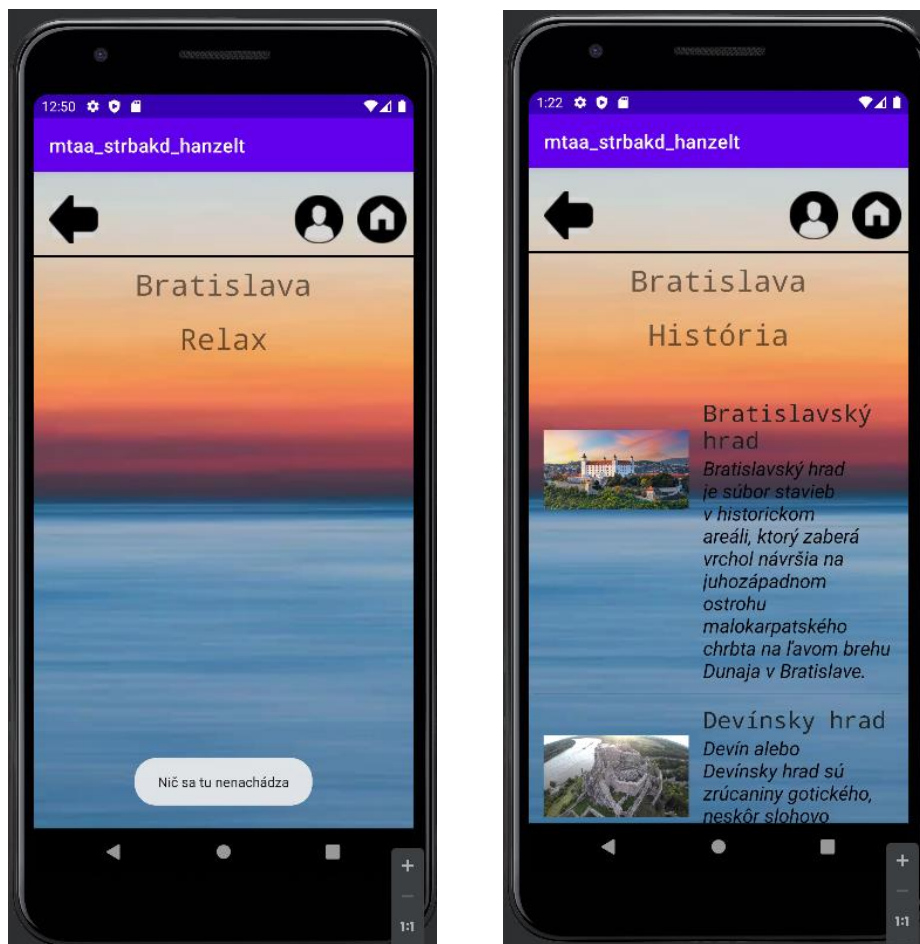
Typ aktivity si môžeme vybrať zo 6 možností: Šport, História, Turistika, Zábava, Relax a Kultúra.



Po kliknutí na hociktorú možnosť nás to prehodí do ďalšej obrazovky, kde sa zobrazia už konkrétne aktivity v danom meste, ktoré sú typu aký si klikneme.

Ak sa náhodou žiadna aktivita v danom meste s konkrétnym typom aktivity nenachádza, tak to vyhodí dole okienko, kde vypíše, že sa tam nič nenachádza.

Pokiaľ sa tam, ale aktivity nachádzajú, tak ich konkrétne vypíše a budú klikateľné, teda môžeme si na hociktorú z nich kliknúť, aby sme sa dozvedeli viac o tejto aktivite. Takisto tam bude vidieť nejaký krátky popis a obrázok konkrétnej aktivity.



Obrazovka je samozrejme posúvacia a teda môže sa posúvať dole, keď tam je viacero aktivít. Môžeme vidieť, že po každom kliknutí sa zaznamenáva aj mesto aj typ aktivity, ktorý sme si vybrali a teda, ak sme náhodou zle klikli, tak vidíme čo sme si vybrali a že sa môžeme vrátiť späť a vybrať si niečo iné.

Po rozkliknutí konkrétnej aktivity, teda napríklad ako je vyššie spomínaný Bratislavský hrad sa nám otvoria informácie o konkrétnej aktivite. Bude sa tam nachádzať popis aktivity, adresa a tlačidlo GALÉRIA.

Týmto tlačidlom ako už je z názvu jasné sa budeme vedieť prekliknúť do galérie a zobrazíť si obrázky.



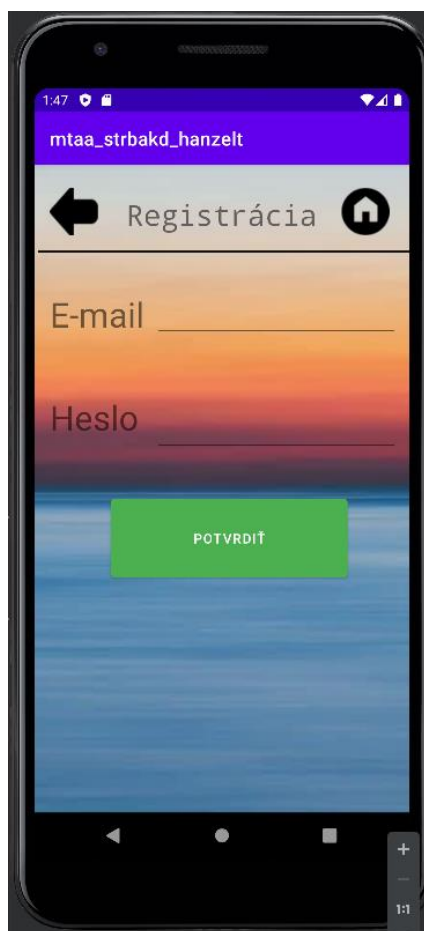
Takisto ako sme mohli vidieť, tak na každej obrazovke máme ikonku DOMOV a ikonku PROFIL. Po kliknutí na DOMOV, ktorá sa nachádza vpravo hore prvá z kraja, tak nás to vždy hodí na úvodnú obrazovku aplikácie a teda na miesto akoby sme akurát zapli aplikáciu.

Po kliknutí ikonky PROFIL sú 2 možnosti na výber, ktoré sa môžu stať.

- Nie sme prihlásený a hodí nás to na prihlásenie/registráciu.
- Sme prihlásený a hodí nás to do nášho účtu, teda zobrazí nám informácie o našom účte.



Ak nemáme vytvorený účet v aplikácii, tak klikneme na ZAREGISTROVAŤ a vytvoríme si účet zadáním základných údajov a to je e-mail a heslo.

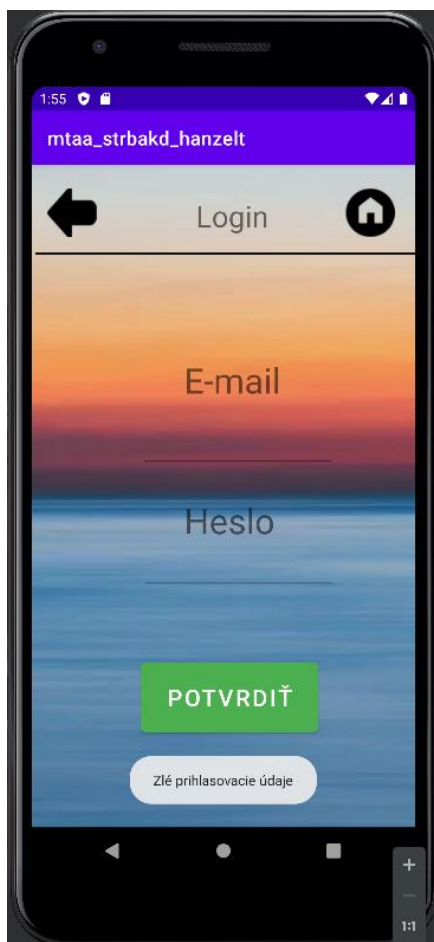
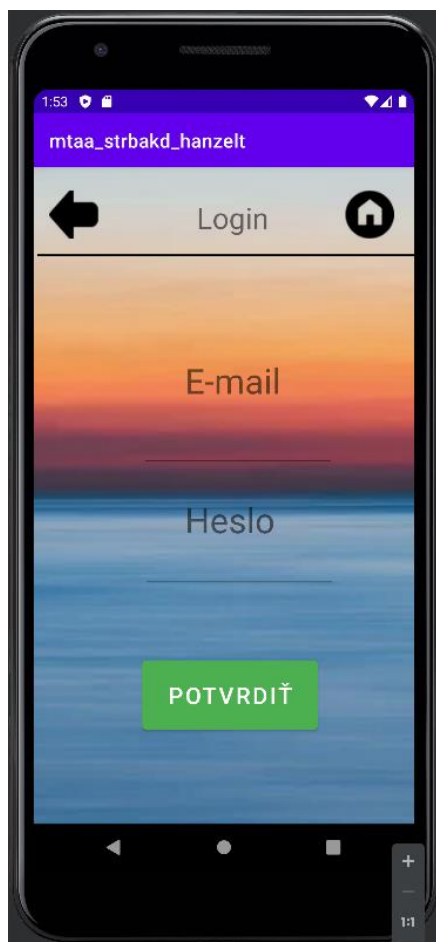


Po kliknutí tlačidla POTVRDIŤ sa znova môžu stať 2 rôzne scenáre.

- Zaregistruje nás to a údaje pridá do databázy.
- Nezaregistruje nás to a vypíše dôvod, teda e-mail sa už používa alebo zle zadané údaje(napr. v emaily sa nenachádza @).

Po úspešnej registrácii nás to posunie na prihlásenie, kde treba zadať e-mail a heslo na prihlásenie do databázy. Znova sa môžu stať 2 rôzne scenáre.

- Podarí sa nám prihlásiť a všetko zbehne, vypíše okienko Úspešné prihlásenie.
- Nepodarí sa nám prihlásiť lebo sa takéto údaje nenašli v databáze, teda takýto používateľ s týmito údajmi tam nie je zaregistrovaný.



Po úspešnej registrácii alebo prihlásení si dokážeme otvoriť náš účet, v ktorom sa nám zobrazia údaje o našej obrazovke. Takisto sa dajú na tejto obrazovke urobiť 3 rôzne aktivity.

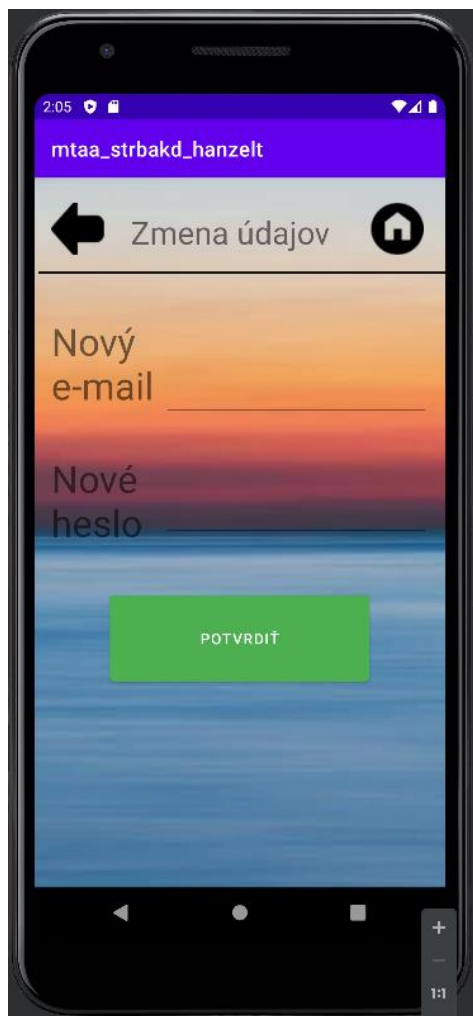
- Odhlásiť sa z účtu
- Zmeniť údaje účtu(E-mail, heslo)
- Zmazať účet



Po kliknutí na tlačidlo Odhlásiť nás to odhlási z účtu a hodí na úvodnú obrazovku aplikácie a teda na miesto akoby sme akurát zapli aplikáciu.

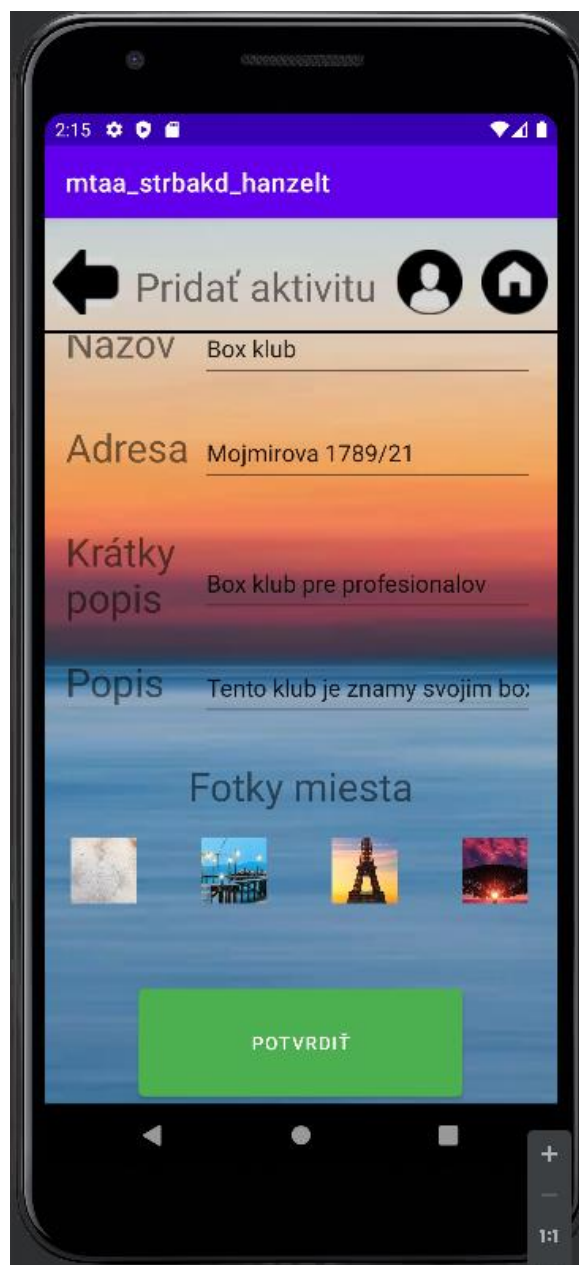
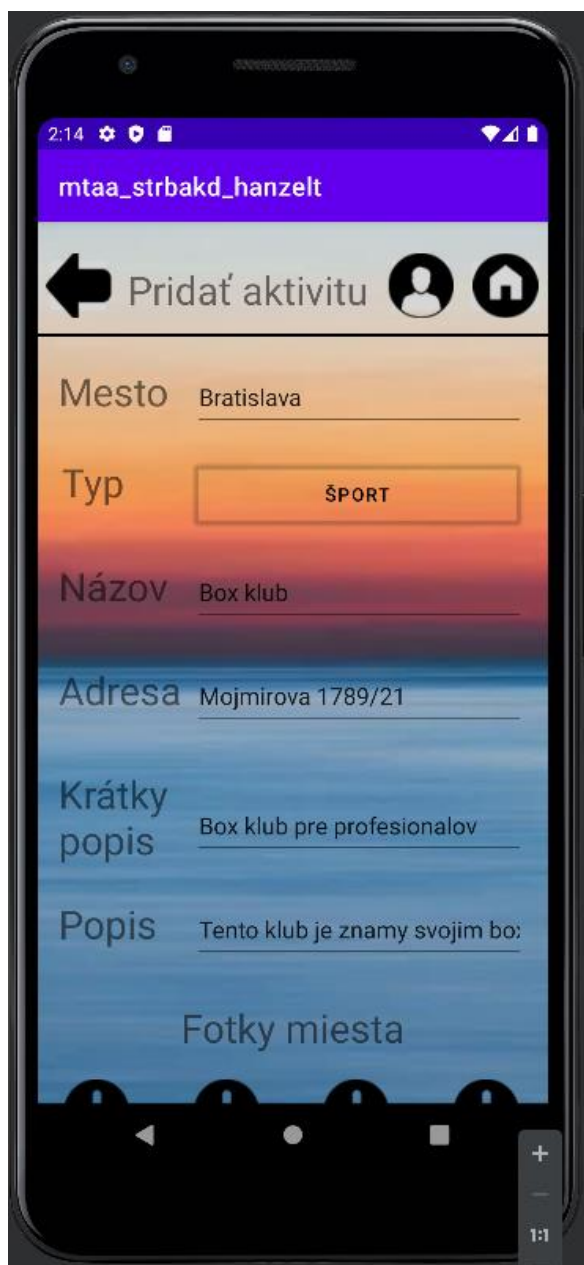
Po kliknutí tlačidla Zmeniť údaje nás to hodí do ďalšej obrazovky o zmene údajov, ktoré treba vyplniť. (E-mail a heslo).

Po kliknutí tlačidla Zmazať účet nám vyhodí modal window, kde sa nás to spýta či si naozaj prajeme zmazať účet. Ak klikneme na áno, tak sa zmaže účet a hodí nás to na úvodnú obrazovku aplikácie a teda na miesto akoby sme akurát zapli aplikáciu.



Posledná vec, ktorá sa dá urobiť v našej aplikácii je už konkrétne pridanie aktivity do databázy a teda aj do aplikácie. Na úvodnej obrazovke pokiaľ sme prihlásený klikneme na tlačidlo PRIDAŤ AKTIVITU.

Zobrazí sa nám obrazovka, kde treba vyplniť všetky údaje ináč sa nám nepodarí pridať aktivitu. Takisto v aplikácii máme na výber koľko obrázkov tam chceme pridať, musíme pridať minimálne 1, maximálne 4.



Stlačením tlačidla POTVRDIŤ sa daná aktivita pridá do databázy a takisto do aplikácie, kde si ju budeme môcť potom vyhľadať.

ZMENY OPROTI NÁVRHU

1. Potrebovali sme trošku zmeniť backend. Zmenili sme v ňom jednu vec, konkrétne v pridání aktivity, kde sme zabudli nastaviť, keď pridávame aktivitu, aby sa prvý obrázok automaticky nastavil ako thumbnail picture.
2. Trošku sme upravili zobrazenie konkrétnej aktivity. Pôvodne tam mali byť obrázky, text, adresa a mapa. Upravili sme to tak, že je tam text(popis danej aktivity), adresa a tlačidlo galéria. Pomocou toho sa vieme dostať už ku konkrétnym obrázkom.
3. Pridali sme ďalšiu vec, na ktorú sme zabudli pri návrhu a tou bolo Odhlásenie používateľa z účtu.

CESTA K REPOZITÁRU K FRONTENDU

<https://github.com/strbakdenis/mtaa-strbakd-hanzelt-frontend>

ZÁVER

Myslíme si, že sme našu aplikáciu urobili tak ako sme si ju navrhli s minimálnymi zmenami, ktoré sme si na začiatku neuvedomili alebo na ne zabudli a sú potrebné na plynulý chod aplikácie. Tento projekt nás veľa naučil a sme radi, že sme ho mohli robiť. Naučili sme sa lepšie pracovať s pythonom a backendom, ktorý sme robili v django. Takisto sme sa prvý krát stretli s programom android studio a kotlinom, v ktorom sme sa naučili pohybovať a vytvárať ako taký frontend. Späťne hodnotíme túto skúsenosť s projektom aplikácie ako pozitívnu.