Zadanie

Mobilné technológie a aplikácie

Vytvorte v dvojiciach mobilnú aplikáciu (iOS/Android) podľa Vášho zadania, ktoré si zvolíte na prvom cvičení. Zadanie bude odovzdané v nasledujúcich kontrolných bodoch:

Odovzdanie c.1 (3. tyzden)

Navrhnite dátový model mobilnej aplikácie, ktorý bude reprezentovať všetky potrebné dáta špecifikované pre vašu skupinu. Na základe neho definujte zoznam volaní na server potrebných na zabezpečenie vytvárania, mazania, úpravy a zobrazenia dát (CRUD metódy). Taktiež definujte odpovede serveru a obsah volaní. Pri návrhu sa riaďte princípmi REST API.

Navrhnite jednotlivé obrazovky mobilnej aplikácie s dôrazom na rozloženie prvkov – vytvorte tzv. „low fidelity mockups“. Navrhnuté obrazovky by mali umožňovať mazanie, aktualizáciu a zobrazenie dát.

Hodnotenie:
Zadanie sa odovzdáva v 3. týždni a je hodnotené 5 bodmi, z čoho je na získanie zápočtu nutné získať minimálne 1 bod.

Odovzdanie c.2 (4. týždeň)

Vytvorte akceptačné a funkcionálne testy pokrývajúce funkcionalitu aplikácie v rozsahu odovzdania prvého zadania v 7. týždni. Tieto testy musia byť obsiahnuté aj v dokumentácii k prvému zadaniu.
Úloha sa prezentuje na cvičení v 6. týždni semestra na a je hodnotená 5 bodmi, z čoho je na získanie zápočtu nutné získať minimálne 1 bod.

Odovzdanie c.3 serverová časť (6. tyzden)

Vytvorte serverovú stranu aplikácie ľubovoľným spôsobom, ktorá bude schopná poskytovať zoznam všetkých dát na serveri a operácie typu zobrazenie, úpravu a mazanie nad jednotlivými záznamami.

Odovzdanie c.4 verzia app 1.0 (7. tyzden)
Odovzdanie c.5 verzia final (9. tyzden)

Naprogramujte prvú verziu mobilnej aplikácie pre Android (po dohode pre inú platformu), ktorá bude schopná komunikovať so serverom pomocou protokolu HTTP a obsluhovať všetky API volania serverovej strany aplikácie. Všetky tieto operácie by sa mali vykonávať vo vlastných obrazovkách aplikácie podľa návrhov z predchádzajúceho zadania.

K zadaniu vytvorte dokumentáciu reprezentujúcu súčasný stav projektu a opisujúci riešenie serverovej aj mobilnej strany aplikácie.

Prvá verzia zadania sa odovzdáva v 7. týždni a je hodnotená 5 bodmi, z čoho je na získanie zápočtu nutné získať minimálne 1 bod.

Prezentujte vývojovú verziu mobilnej aplikácie, ktorá musí mať implementované aspoň dve obrazovky a musí byť schopná komunikovať so serverovou stranou aspoň jedným volaním.

Finálna verzia zadania sa odovzdáva v 7. týždni a je hodnotená 15 bodmi, z čoho je na získanie zápočtu nutné získať minimálne 7 bodov.

Hodnotenie:

Základné úlohy - 5 + 5 + 5 + 5 + 15 = 35 bodov (min. 15 bodov)
Doplnkové úlohy - 10 + 10 = 20 bodov (nie je minimum)

Úloha za doplnkové body (9. tyzden):

Naprogramujte aplikáciu tak, aby bola schopná zobrazovať, ukladať, upravovať a mazať, dáta prevzaté zo serveru aj po strate internetového pripojenia, t.j. dáta po komunikácii so serverom udržiavajte synchronizované prostredníctvom lokálnej databázy mobilnej aplikácie.

Hodnotené 10 bodmi, pri nesplnení študent nezíska body ale nie je postihovaný.

Odovzdanie c.6 verzia app websocket (10. tyzden) - voliteľná úloha

Komunikáciu so serverom prerobte tak, aby sa využívala technológia WebSocket namiesto HTTP prenosu dát medzi mobilnou aplikáciou a serverovou stranou.

K zadaniu vytvorte dokumentáciu reprezentujúcu súčasný stav projektu a opisujúci riešenie serverovej aj mobilnej strany aplikácie.

Hodnotenie:

Zadanie sa odovzdáva v 10. týždni semestra a je hodnotené 10 bodmi. Úloha je doplnková, pri jej nesplnení študent nezíska body ale nie je nijak postihovaný.

40b základ
25b bonusy
MAXIMUM 65b (zisk 60b z cvičení je dostatočný na získanie známke bez absolvovania skúšky)
MINIMUM 30b

Keywords

- Data provider
- Service provider
- UUID
- HTTP / websocket komunikacia
- RestAPI / RestFull Api
- HTTP status codes (IANA)
- Gitignore

Requirements

- Reachibility
- Caste commitovanie + komentovanie commitov (budeme kontrolovat kontributorstvo)

Harmonogram cviceni

1. **Cvičenie nebude**
2. **Oboznámenie so zadaním, Navrh API, Mockups aplikacie, Struktura dat (navrh databazy)**
   uvod do studia + zadanie + ulohy do dalsieho cvika (odovzdat 1 den pred 3. cvikom)
   a. vyber platformy (iOS/Android)
   b. rozbehanie vyvojoveho prostredia
   c. synchronizovanie projektov s remote repozitarom (bitbucket.org)
      i. .gitignore file => well known
   d. ulohy, ktore maju byt do 3. cvika spravene
      i. navrh API
      ii. Mockups aplikacie
      iii. Struktura dat (navrh databazy)
3. **Kontrola 1. odovzdania + zaciatok prác na zadaní**
   a. kontrola navrh API, mockups, struktura dat
   b. zaciatok developmentu, konzultacie
4. **Práca na Zadaní**
   a. vytvorenie si API cez parse.com
5. **Ukážka aplikácie schopnej komunikovať na cvičení + akceptačné a funkčné testy**
   a. beta verzia aplikacie
   b. vytvorenie akceptacnych testov a zapisanie do dokumentacie
6. **Odovzdanie návrhu akceptačných a funkčných testov**
   a. Akceptacne testy musia byt zapisane v dokumentacii
   b. development + konzultacie
7. **2. odovzdanie Verzia aplikácie 1.0**
   a. na cviceni sa moze developovat + konzultovat
   b. do konca tyzdna odovzdat zadanie
8. **Preberanie Zadania. / Práca za verzii 2.0**
   a. rozdanie API na websocketovy server + example ako posielat data
   b. preberanie 2. odovzdania
9. **Preberanie Zadania. / Práca za verzii 2.0**
10. **Verzia aplikácie 2.0**
    a. na cviceni sa moze developovat + konzultovat
    b. do konca tyzdna odovzdat zadanie
11. **Preberanie zadania**
12. **Zapocet**

Dokumentacia musi obsahovat

Zadanie
Analyza vyvojoveho prostredia
Analyza vybraneho zamerania zadania
Vybrane nastroje
Mockups
Struktura dat
RestAPI
Akceptacne testy (aspon 20 az 30 riadkova tabulka “akcia”, “reakcia”, “stav”)
Pouzite Libky
Rozpis prac, kto co robil (Plan prac)

Navrh API

musi:

GET	/entita	get list
GET	/entita/:id	get one detail
PUT	/entita/:id	update
POST	/entita	create
DELETE	/entita/:id	delete

optional:

GET	/entita/:id/photo	get photo link for an entity
PUT	/entita/:id/photo	set photo link for an entity
DELETE	/entita/:id/photo	clear photo link for an entity

Application TYPES!

- `application/json`
- `application/x-www-form-urlencoded`
- `multipart/form-data`
- `text/html`

Example zapis API:

Get contact list	`GET /contacts`
Get contact details	`GET /contacts/:userId`
Update contact details	`PUT /contacts/:userId`
Add contact	`POST /contacts`
Remove contact	`DELETE /contacts/:userId`

Request

```
GET /contacts
application/json
```

Request headers

@param string **application-id**
@param string **secret-key**

Request parameters

-

Response Type

```
application/json
```

Response parameters

@param array **users**
*int **user** - remote user ID*
*string **email** - full email*
*bool **localUserIsConfirmed***
*bool **remoteUserIsConfirmed***
*int **contactType***

```
    ContactTypeOutgoing = 1,
    ContactTypeIncoming = 2,
    ContactTypeFriend = 3,
```

HTTP response codes

200 OK
Request successful
400 Bad Request
Input parameters incorrect

Examples


```
[
 {
  user: 1,
  email: "user1@example.com",
  localUserIsConfirmed: true,
  remoteUserIsConfirmed: true,
  contactType: 1
 },
 {
  user: 2,
  email: "user2@example.com",
  localUserIsConfirmed: true,
  remoteUserIsConfirmed: true,
  contactType: 2
 }
]
```

Aplikacia musi obsahovat:

- Staticky Login screen s validaciami (email + password) => napevno meno/heslo
- Zoznam danej entity (list / dlazdice) (Android) meno + pripadne poradove cislo (pripadne avatar)
  - tu musi byt moznost refreshu danych dat
  - mazanie jednotlivych zaznamov
  - vytvaranie
- Detail enity
  - editacia
  - mazanie
- Screen na vytvaranie

\+ vsetky mozne vyskakovacie okna, ktore tam mozu nastat pri roznych kontrolach a validaciach (pripjenie na internet, zle priajate data, validacia prihlasovacich udajov, validacia pri vytvarani obsahu, screen s “loaderom” => ked caka pouzivatel na dokoncenie dopytu na server)

Entita nad ktorou sa pracuje

- musi obsahovat aspon jeden multimedialny subor (napr.: obrazok) s moznostou jeho detailneho nahladu

Priklad:

Request
GET /cafe
Request parameters
id=object_id

Request example
/cafe?props=name%2CobjectId%2Cphoto&where=category=2