Fakulta informatiky a informačných technológií STU v Bratislave

Mobilné technológie a aplikácie

**My Important Day**

Tamás Szakál

Zsolt Kiss

Piatok 8:00-9:50  
Ing. Viktor Šulák

2020/2021

Obsah

[Zadanie 1 – Návrh projektu 3](#_Toc68206867)

[Cieľ aplikácie 3](#_Toc68206868)

[Funkcie aplikácie 3](#_Toc68206869)

[API 5](#_Toc68206870)

[Databáza 5](#_Toc68206871)

[Wireframe 6](#_Toc68206872)

[Zadanie 2 – Akceptačné testy 7](#_Toc68206873)

[Testy pre back-end 7](#_Toc68206874)

[Test 1: Registrácia nového konta 7](#_Toc68206875)

[Test 2: Registrácia existujúceho konta 8](#_Toc68206876)

[Test 3: Prihlásenie sa s existujúcim kontom 9](#_Toc68206877)

[Test 4: Prihlásenie sa s neexistujúcim kontom 10](#_Toc68206878)

[Test 5: Prihlásenie sa s neplatným heslom 11](#_Toc68206879)

[Testy pre front-end 12](#_Toc68206880)

[Test 1: Registrácia nového konta, nevhodný formát e-mailu 12](#_Toc68206881)

[Test 2: Registrácia nového konta, prázdna kolonka hesla 13](#_Toc68206882)

[Test 3: Zmena mena používateľa 14](#_Toc68206883)

[Test 4: Vytvorenie novej udalosti, udalosť je vložená do kalendára 15](#_Toc68206884)

[Test 5: Vytvorenie udalosti, nevyplnené povinné pole priorita 16](#_Toc68206885)

[Zadanie 3 – Back end 17](#_Toc68206886)

[POST /register/ 17](#_Toc68206887)

[POST /login/ 18](#_Toc68206888)

[POST /logout/ 18](#_Toc68206889)

[GET /account/ 19](#_Toc68206890)

[PUT /account/password/ 19](#_Toc68206891)

[PUT /account/properties/ 20](#_Toc68206892)

[POST /events/ 21](#_Toc68206893)

[GET/events/all/ 22](#_Toc68206894)

[GET /events/id/ 23](#_Toc68206895)

[PUT /events/id/update/ 23](#_Toc68206896)

[DEL /events/id/delete/ 24](#_Toc68206897)

[Zmeny oproti návrhu 24](#_Toc68206898)

# Zadanie 1 – Návrh projektu

## Cieľ aplikácie

V dnešnom uponáhľanom svete nie je nezvyčajné, ak zabúdame na veci, ktoré sú pre nás inak dôležité. Iní ľudia majú tendenciu otáľať, ak im nič nemá pripomínať, aby si plnili svoje povinnosti. Ako študenti univerzity, pre nás je potrebné, aby sme náš čas využívali čo najefektívnejšie .

Preto sme sa rozhodli vytvoriť aplikáciu, ktorú môžeme používať v dnešnom živote, aby nám pomáhala organizovať náš deň. My Important Day je mobilná aplikácia, ktorá bude rozvíjaná na platformu Android.

## Funkcie aplikácie

Aplikácia vyžaduje registráciu používateľa, aby ju bolo možné používať. Pri registrácií je potrebné zadať meno, e-mailovú adresu a heslo. Po úspešnom registrácií používateľ sa môže prihlásiť, a využívať funkcie aplikácií:

1. Vytvorenie novej udalosti

* Používateľ musí zadefinovať predmet, dátum, miesto, prioritu aby bolo možné vytvoriť novú udalosť.
* Opcionálne používateľ môže zadať iné informácie o udalosti v sekcií Advanced, napr.: s kým treba kontaktovať pred udalosťou, čo má so sebou brať atď ... Tiež opcionálne môže zvoliť foto zo svojej galérie, čo bude zobrazené ako dostane notifikácie o udalosti
* Po vyplnení týchto údajov používateľ môže vytvoriť udalosť kliknutím na tlačítko Create. Udalosť bude vytvorená, vložená do kalendára, a používateľ bude presmerovaný na domovskú stránku.
* V ktoromkoľvek kroku môže používateľ stornovať vytvorenie udalosti, ak je v menu Event

1. Kalendár

* Navigovaním do kalendára, a následne vybraním dňa aplikácia zobrazí udalosť alebo udalosti, ktoré sú naplánované na tento deň
* Kliknutím na danú udalosť môže používateľ zmeniť údaje o udalosti. Obrazovka vyzerá tak isto, ako pri vytvorení udalosti, avšak teraz namiesto tlačítka Create je tam tlačítko Edit.

1. Nastavenie

* Kliknutím na ikonu v ľavom hornom rohu budeme presmerovaný do nastavení, kde môžeme zmeniť meno, heslo, alebo zadefinovať text, čo bude zobrazený pri vytvorení a zmenení udalosti, v podmenu Advanced.

1. Domovská stránka

* Kliknutím na ikonu, v dolnej časti v strede budeme presmerovaný na domovskú stránku, kde nám aplikácia zobrazí udalosti s informáciami na dnešný deň.

## API

Link na dokumentáciu API:

[Link](https://documenter.getpostman.com/view/14825605/Tz5p6dfF?fbclid=IwAR1x0W65PPKvWj5asCZJzwPnQ953yHkBU46G1MwCcwMP7l-AZ4QSHSinolY)

## Databáza

Každý užívateľ môže mať viac eventov, ale jeden event môže mat len jedného vlastníka/užívateľa.

1

∞

|  |  |  |
| --- | --- | --- |
| Key | Event | type |
| PK | ID | Integer |
| FK | userID | Integer |
|  | Subject | String |
|  | Date | ISO 8601 |
|  | Place | String |
|  | Priority | Integer |
|  | Advanced | String |
|  | Photo | Binary |

|  |  |  |
| --- | --- | --- |
| Key | User | type |
| PK | ID | Integer |
|  | Name | String |
|  | E-mail | String |
|  | Password | String |

## Wireframe

Obrázok, na ktorom je text, parkovanie, niekoľko, zoradené

Automaticky generovaný popis

# Zadanie 2 – Akceptačné testy

## Testy pre back-end

### Test 1: Registrácia nového konta

|  |  |
| --- | --- |
| **Test 1: Registrácia nového konta** | |
| **Vstupné podmienky:** | Používateľ, ktorý (ešte) nemá registrovaný profil (jeho e-mailová adresa v databáze nie je prítomná).  Aplikácia má pripojenie na internet |
| **Výstupné podmienky:** | Údaje o používateľa sa uložia do databázy, používateľ následne sa môže prihlásiť s týmito údajmi (e-mail a heslo) |
| **Postup:** | 1. Používateľ vyplní pole Name, E-mail, Password  2. Používateľ stlačí tlačítko Register.  3. Zobrazí sa notifikácia o úspešnej registrácií  4. Používateľ stlačí tlačítko OK  5. Používateľ je presmerovaný do prihlasovacieho menu |
| **Výsledok:** PASS / FAIL | |

### Test 2: Registrácia existujúceho konta

|  |  |
| --- | --- |
| **Test 2: Registrácia existujúceho konta** | |
| **Vstupné podmienky:** | Používateľ, ktorý už má registrovaný profil (jeho e-mailová adresa v databáze je prítomná).  Aplikácia má pripojenie na internet |
| **Výstupné podmienky:** | Údaje o existujúcom používateľovi sa neuložia do databázy |
| **Postup:** | 1. Používateľ vyplní pole Name, E-mail, Password  2. Používateľ stlačí tlačítko Register.  3. Zobrazí sa notifikácia o neúspešnej registrácií  4. Používateľ stlačí tlačítko OK  5. Používateľ je presmerovaný do registration menu |
| **Výsledok:** PASS / FAIL | |

### Test 3: Prihlásenie sa s existujúcim kontom

|  |  |
| --- | --- |
| **Test 3: Prihlásenie sa s existujúcim kontom** | |
| **Vstupné podmienky:** | Používateľ, ktorý už má registrovaný profil (jeho e-mailová adresa v databáze je prítomná)  Aplikácia má pripojenie na internet |
| **Výstupné podmienky:** | Používateľ bude prihlásený do svojho konta, kde má uložené všetky udalosti a nastavenia  Používateľ bude môcť využívať všetky používateľské funkcie aplikácie |
| **Postup:** | 1. Používateľ vyplní pole E-mail a Password  2. Používateľ stlačí tlačítko Log in.  3. Zobrazí sa notifikácia o úspešnej prihlásenie sa  4. Používateľ stlačí tlačítko OK |
| **Výsledok:** PASS / FAIL | |

### Test 4: Prihlásenie sa s neexistujúcim kontom

|  |  |
| --- | --- |
| **Test 4: Prihlásenie sa s neexistujúcim kontom** | |
| **Vstupné podmienky:** | Používateľ, ktorý (ešte) nemá registrovaný profil (jeho e-mailová adresa v databáze nie je prítomná)  Aplikácia má pripojenie na internet |
| **Výstupné podmienky:** | Používateľ dostane upozornenie, že konto v systéme neexistuje |
| **Postup:** | 1. Používateľ vyplní pole E-mail a Password  2. Používateľ stlačí tlačítko Log in.  3. Zobrazí sa notifikácia o neúspešnej prihlásenia |
| **Výsledok:** PASS / FAIL | |

### Test 5: Prihlásenie sa s neplatným heslom

|  |  |
| --- | --- |
| **Test 5: Prihlásenie sa s neplatným heslom** | |
| **Vstupné podmienky:** | Používateľ, ktorý má registrovaný profil (jeho e-mailová adresa v databáze je prítomná)  Aplikácia má pripojenie na internet |
| **Výstupné podmienky:** | Používateľ dostane upozornenie, že zadané heslo nie je správne |
| **Postup:** | 1. Používateľ vyplní pole E-mail a Password  2. Používateľ stlačí tlačítko Log in.  3. Zobrazí sa notifikácia o nesprávne zadaného hesla |
| **Výsledok:** PASS / FAIL | |

## Testy pre front-end

### Test 1: Registrácia nového konta, nevhodný formát e-mailu

|  |  |
| --- | --- |
| **Test 1: Registrácia nového konta, nevhodný formát e-mailu** | |
| **Vstupné podmienky:** | Používateľ, ktorý ešte nemá registrovaný profil (jeho e-mailová adresa v databáze nie je prítomný)  Aplikácia nemusí mať pripojenie na internet |
| **Výstupné podmienky:** | V aplikácií sa vyznačí kolonka e-mailu s červením ofarberním a vyskočí informácia, ktorý upozorní používateľa na nevhodný formát e-mailu. |
| **Postup:** | 1. Používateľ vyplní Name, E-mail, Password 2. Používateľ stlačí tlačítko Register 3. Vyznačí sa kolonka E-mail s červením ofarbením 4. Zobrazí sa informácia o nevhodnom formáte emailu 5. Po 3 sekundách informácia zmizne |
| **Výsledok:** PASS / FAIL | |

### Test 2: Registrácia nového konta, prázdna kolonka hesla

|  |  |
| --- | --- |
| **Test 2: Registrácia nového konta, prázdna kolonka hesla** | |
| **Vstupné podmienky:** | Používateľ, ktorý ešte nemá registrovaný profil (jeho e-mailová adresa v databáze nie je prítomná)  Aplikácia nemusí mať pripojenie na internet |
| **Výstupné podmienky:** | V aplikácií sa vyznačí kolonka hesla s červením ofarberním a vyskočí informácia, ktorý upozorní používateľa na nevyplnenú kolonku hesla. |
| **Postup:** | 1. Používateľ vyplní Name, E-mail 2. Používateľ stlačí tlačítkoRegister 3. Vyznačí sa kolonka Password s červením o farbením 4. Zobrazí sa informácia o neúspešnom vytvorení konta kvôli nevyplnenej kolonky password 5. Po 3 sekundách informácia zmizne |
| **Výsledok:** PASS / FAIL | |

### Test 3: Zmena mena používateľa

|  |  |
| --- | --- |
| **Test 3: Zmena mena používateľa** | |
| **Vstupné podmienky:** | Používateľ, ktorý má registrovaný profil (jeho e-mailová adresa v databáze je prítomná).  Používateľ je prihlásený.  Aplikácia má pripojenie na internet. |
| **Výstupné podmienky:** | V aplikácii sa zmení meno používateľa, ktorý sa uloží do databázy a bude možné ho zobraziť. |
| **Postup:** | 1. Používateľ stlačí tlačítko Settings 2. Zobrazí sa menu Settings 3. Používateľ stlačí tlačítko Change name 4. Zobrazí sa formulár na Change name 5. Používateľ vyplní pole Name 6. Používateľ stlačí tlačítko Save 7. Zobrazí sa informácia o úspešnom zmene používateľského mena 8. Používateľ je presmerovaný na stránke Settings |
| **Výsledok:** PASS / FAIL | |

### Test 4: Vytvorenie novej udalosti, udalosť je vložená do kalendára

|  |  |
| --- | --- |
| **Test 4: Vytvorenie novej udalosti, udalosť je vložená do kalendára** | |
| **Vstupné podmienky:** | Používateľ, ktorý má registrovaný profil (jeho e-mailová adresa v databáze je prítomná).  Používateľ je prihlásený.  Aplikácia má pripojenie na internet. |
| **Výstupné podmienky:** | Udalosť je zobrazená v kalendári (ak udalosť sa vzťahuje na dnešný deň, bude zobrazená aj na domovskej stránke) |
| **Postup:** | 1. Používateľ vyplní všetky povinné pole o udalosti 2. Používateľ stlačí tlačítko Create 3. Zobrazí sa upozornenie o úspešnom vytvorení udalosti 4. Kliknutím na ikon Calendar v ľavom dolnom rohu bude používateľ presmerovaný do kalendára 5. Používateľ si vyberie deň, na ktorý vytvoril udalosť 6. Udalosť je zobrazená |
| **Výsledok:** PASS / FAIL | |

### Test 5: Vytvorenie udalosti, nevyplnené povinné pole priorita

|  |  |
| --- | --- |
| **Test 5: Vytvorenie udalosti, nevyplnené povinné pole priorita** | |
| **Vstupné podmienky:** | Používateľ, ktorý má registrovaný profil (jeho e-mailová adresa v databáze je prítomná).  Používateľ je prihlásený.  Aplikácia nemusí mať pripojenie na internet. |
| **Výstupné podmienky:** | V aplikácií sa vyznačí kolonka priorita s červením ofarberním a vyskočí informácia, ktorý upozorní používateľa o nevyplnenom kolonky Priority. |
| **Postup:** | 1. Používateľ stlačí tlačítko + pre vytvorenie udalosti 2. Zobrazí sa formulár na pridanie udalosti 3. Používateľ vyplní Name, Subject, Date, Place 4. Používateľ stlačí tlačítko Create 5. Zobrazí sa informácia o neúspešnom vytvorení udalosti kvôli nevyplnenej kolonky Priority 6. Po 3 sekundách informácia zmizne |
| **Výsledok:** PASS / FAIL | |

# Zadanie 3 – Back end

## POST /register/

Tento API endpoint slúži na registráciu používateľa. Server odpovedá na POST žiadosť status kódom 200 pri úspešnej registrácií, a len vtedy dostane používateľ odpoveď obsahujúci Token, ktorý ďalej bude poskytovať práva pre používateľa, aby mohol využiť funkcie aplikácie, ktoré to vyžadujú. Informácie o používateľa budú uložené do databázy, tak isto aj Token.

|  |  |
| --- | --- |
| Predpoklady úspešnej registrácie | Vyplnené všetky povinné pole: username, email, password (bude to riešiť front end)  Ani jedno pole nemôže byť blank (bude to riešiť front end)  Email a username ešte v databáze neexistuje  Email má správny formát (bude to riešiť front end) |
| HTTP Status kódy | 200 OK  400 Bad Request |

{

    "response": "Successful registration!",

    "email": "test@gmail.com",

    "username": "Test user",

    "token": "d3fce63d9fb6609251eebe249003996f5ed5ae4d"

}

Príklad úspešnej registrácie

## POST /login/

Tento API endpoint slúži na prihlásenie sa používateľa do aplikácie, ale aj pre žiadanie nového Tokena pre používateľa, ak token bol vymazaný z databázy (ak používateľ sa odhlásil z aplikácie).

|  |  |
| --- | --- |
| Predpoklady úspešnej prihlásenie sa: | Používateľ je zaregistrovaný  Vyplnené všetky povinné pole: username, password (bude to riešiť front end)  Username a password zhodujú s údajmi v databáze  Ani jedno pole nemôže byť prázdne blank (bude to riešiť front end) |
| HTTP Status kódy | 200 OK  400 Bad Request |

{

    "token": "d3fce63d9fb6609251eebe249003996f5ed5ae4d"

}

Príklad úspešnej prihlásenie sa

## POST /logout/

Tento API endpoint slúži na odhlásenie sa používateľa z aplikácie. Jednoducho vymažeme asociovaný Token z databázy, teda používateľ stratí práva na používanie aplikácie. Aby mohol využívať aplikáciu v budúcnosti, musí sa prihlásiť znova, aby dostal nový token.

|  |  |
| --- | --- |
| Predpoklady úspešnej odhlásenie sa | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze |
| HTTP Status kódy | 200 OK  401 Unauthorized |

{

    "response": "Successfully logged out."

}

Príklad úspešnej odhlásenie sa

## GET /account/

Tento API endpoint slúži na zobrazenie informácií o aktuálne prihláseného používateľa.

|  |  |
| --- | --- |
| Predpoklady | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze |
| HTTP Status kódy | 200 OK  401 Unauthorized |

{

    "username": "Test user",

    "email": "test@gmail.com"

}

Príklad zobrazenie informácií o prihláseného používateľa

## PUT /account/password/

Tento API endpoint slúži na zmeny hesla pre aktuálne prihláseného používateľa..

|  |  |
| --- | --- |
| Predpoklady úspešnej zmeny hesla | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze  Vyplnené všetky povinné pole: old\_password, new\_password (bude to riešiť front end)  Ani jedno pole nemôže byť blank (bude to riešiť front end)  Pole old\_password zhoduje s heslom asociovaný s prihláseným používateľom v databáze |
| HTTP Status kódy | 200 OK  400 Bad Request  401 Unauthorized |

{

    "response": "The password was updated."

}

Príklad úspešnej zmeny hesla

## PUT /account/properties/

Tento API endpoint slúži na zmeny používateľského mena a emailovej adresy aktuálne prihláseného používateľa.

|  |  |
| --- | --- |
| Predpoklady | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze  Vyplnené všetky povinné pole: username, email (bude to riešiť front end)  Ani jedno pole nemôže byť blank (bude to riešiť front end) |
| HTTP Status kódy | 200 OK  400 Bad Request  401 Unauthorized |

{

    "username": "Test user",

    "email": "test@gmail.com"

}

Príklad úspešnej zmeny údajov

## POST /events/

Tento API endpoint slúži na vytvorenie novej udalosti. Povinné pole sú subject, date, place a priority. Ak používateľ nevyplní pole advanced, tak sa odosiela prázdne pole, alebo ak používateľ nenahrá fotku, tak pole pic bude inicializované na null. Udalosť obsahuje cudzí klúč, ktorý smeruje na id usera, ktorý jej vytvoril.

|  |  |
| --- | --- |
| Predpoklady úspešnej vytvorenie udalosti | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze  Vyplnené všetky povinné pole: subject, date, place, priority (bude to riešiť front end)  Polia subject a place nemôžu byť blank (bude to riešiť front end)  Date musí byť v správnom formáte YY-MM-DD hh:mm (bude to riešiť front end)  Priority musí byť z intervalu <1; 3> (bude to riešiť front end) |
| HTTP Status kódy | 200 OK  400 Bad Request  401 Unauthorized |

{

    "id": 1,

    "subject": "Test subject",

    "date": "2021-04-01T18:00:00Z",

    "place": "Test place",

    "priority": "3",

    "advanced": "Test advanced",

    "pic": "/media/images/pic.png",

    "user": 1

}

Príklad úspešnej vytvorenia novej udalosti

## GET/events/all/

Tento API endpoint slúži na zobrazenie všetkých vytvorených udalostí pre daný prihlásený používateľ.

|  |  |
| --- | --- |
| Predpoklady | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze |
| HTTP Status kódy | 200 OK  401 Unauthorized |

[

    {

        "id": 2,

        "subject": "Test subject 2",

        "date": "2021-04-01T19:00:00Z",

        "place": "Test place 2",

        "priority": "1",

        "advanced": "Test advanced 2",

        "pic": null,

        "user": 1

    },

    {

        "id": 1,

        "subject": "Test subject",

        "date": "2021-04-01T18:00:00Z",

        "place": "Test place",

        "priority": "3",

        "advanced": "Test advanced",

        "pic": "/media/images/pic.png",

        "user": 1

    }

]

Príklad zobrazenia všetkých udalostí

## GET /events/id/

Tento API endpoint slúži na zobrazenie jednej konkrétnej udalosti pre daný prihlásený používateľ.

|  |  |
| --- | --- |
| Predpoklady | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze  Aktuálne prihlásený používateľ vytvoril udalosť (nie niekto iný) |
| HTTP Status kódy | 200 OK  401 Unauthorized  404 Not Found |

{

    "id": 2,

    "subject": "Test subject 2",

    "date": "2021-04-01T19:00:00Z",

    "place": "Test place 2",

    "priority": "1",

    "advanced": "Test advanced 2",

    "pic": null,

    "user": 1

}

Príklad zobrazenia konkrétnej udalosti podľa id

## PUT /events/id/update/

Tento API endpoint slúži na zmenu už vytvorenej udalosti. Funguje podobne ako vytvorenie udalosti, avšak podmienkou je, že udalosť existuje.

|  |  |
| --- | --- |
| Predpoklady úspešnej zmeny udalosti | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze  Udalosť existuje  Aktuálne prihlásený používateľ vytvoril udalosť (nie niekto iný) |
| HTTP Status kódy | 200 OK  400 Bad Request  401 Unauthorize  404 Not Found |

{

    "id": 1,

    "subject": "Test subject UPDATED",

    "date": "2020-04-01T20:00:00Z",

    "place": "Test place UPDATED",

    "priority": "2",

    "advanced": "Test advanced UPDATED",

    "pic": null,

    "user": 1

}

Príklad úspešnej zmeny udalosti

## DEL /events/id/delete/

Tento API endpoint slúži na vymazanie už vytvorenej udalosti.

|  |  |
| --- | --- |
| Predpoklady úspešnej vymazanie udalosti | Header obsahuje Key „Authorization“, Value „Token xyz“, kde xyz je aktuálny token asociovaný s používateľom, existujúci v databáze  Udalosť existuje  Aktuálne prihlásený používateľ vytvoril udalosť (nie niekto iný) |
| HTTP Status kódy | 200 OK  401 Unauthorized  404 Not Found |

{

    "response": "The event was deleted."

}

Príklad úspešnej vymazanie udalosti

## Zmeny oproti návrhu

Back end našej aplikácie je navrhnuté tak, ako sme to predstavili v rámci prvého zadania. Pár vecí sa zmenilo, take sú napríklad mená API endpointov. Časom sme došli na to, že súčasná implementácia dáva viac zmysel pre používateľa aplikácie.

Github repozitár link:

[Zsolti0819/MTAA\_MyImportantDay\_xkiss\_xszakal (github.com)](https://github.com/Zsolti0819/MTAA_MyImportantDay_xkiss_xszakal)