**TechSupport**

Michal Minár, Adrián ondov

Obsah

[1. Návrh aplikácie 3](#_Toc101182517)

[1.1. Chat 3](#_Toc101182518)

[1.2. Fórum 3](#_Toc101182519)

[1.3. Články 3](#_Toc101182520)

[2. Akceptačné testy 4](#_Toc101182521)

[2.1. Front end 4](#_Toc101182522)

[2.1.1. Napísanie správy inému používateľovi 4](#_Toc101182523)

[2.1.2. Pridať nový článok 5](#_Toc101182524)

[2.1.3. Položiť novú otázku do fóra 6](#_Toc101182525)

[2.1.4. Úprava komentáru neprihláseným používateľom 7](#_Toc101182526)

[2.1.5. Vymazať článok bez prihlásenia 8](#_Toc101182527)

[2.2. Back end 9](#_Toc101182528)

[2.2.1. Získanie všetkých otázok a následný výber konkrétnej s načítaním komentárov 9](#_Toc101182529)

[2.2.2. Získanie histórie správ a následné prečítanie posledných správ prihláseného a žiadaného používateľa 10](#_Toc101182530)

[2.2.3. Získanie článku a načítanie zodpovedajúcich obrázkov 11](#_Toc101182531)

[2.2.4. Získanie všetkých komentárov pre konkrétnu otázku s ich následným vymazaním neprihláseným používateľom 12](#_Toc101182532)

[2.2.5. Vymazanie obrázku so zadaním nesprávneho ID otázky 13](#_Toc101182533)

[3. Obrazovky 14](#_Toc101182534)

[3.1. Chat 14](#_Toc101182535)

[3.2. Fórum 15](#_Toc101182536)

[3.3. Články 15](#_Toc101182537)

[3.4. Ostatné 16](#_Toc101182538)

[4. Databáza 17](#_Toc101182539)

[5. Back-end 18](#_Toc101182540)

[5.1. Chat 18](#_Toc101182541)

[5.2. Fórum 18](#_Toc101182542)

[5.3. Články 19](#_Toc101182543)

[5.4. Ostatné 20](#_Toc101182544)

[5.5. Postman API dokumentácia 20](#_Toc101182545)

[6. Front end 21](#_Toc101182546)

[6.1. Komponenty 21](#_Toc101182547)

[6.2. Stránky 21](#_Toc101182548)

[6.2.1. Stránka ArticleAddScreen 22](#_Toc101182549)

[6.2.2. Stránka ArticleEditScreen 22](#_Toc101182550)

[6.2.3. Stránky ArticleIndexScreen a ArticleIndexAllScreen 23](#_Toc101182551)

[6.2.4. Stránka ArticleShowScreen 23](#_Toc101182552)

[6.2.5. Stránka ForumAddScreen 24](#_Toc101182553)

[6.2.6. Stránka ForumEditScreen 24](#_Toc101182554)

[6.2.7. Stránka ForumIndexScreen 25](#_Toc101182555)

[6.2.8. Stránka ForumShowScreen 25](#_Toc101182556)

[6.2.9. Stránka AboutUsScreen 26](#_Toc101182557)

[6.2.10. Stránka HomeScreen 26](#_Toc101182558)

[6.2.11. Stránka LoginScreen 27](#_Toc101182559)

[6.2.12. Stránka RegisterScreen 27](#_Toc101182560)

[6.2.13. Stránka SettingsScreen 28](#_Toc101182561)

[6.2.14. Stránka ChatIndexScreen 28](#_Toc101182562)

[6.2.15. Stránka ChatShowScreen 29](#_Toc101182563)

[6.2.16. Stránky CallScreen a JoinScreen 29](#_Toc101182564)

[6.3. Štýlovanie 30](#_Toc101182565)

[6.4. WebRTC 30](#_Toc101182566)

[Príloha A: Zadanie 31](#_Toc101182567)

# Návrh aplikácie

Požiadavky projektu vyplývajú zo zadania, ktoré je uvedené v prílohe za záver dokumentu. Všetky obrázky, návrhy a dokumenty sú dostupné aj mimo tohto dokumentu na nasledovnom [odkaze](https://github.com/Mino21M/fiit-mtaa-app-docu). Primárnym účelom aplikácie je zhromažďovať a prehľadne poskytovať informácie ohľadom technickej podpory pre používateľov mobilných telefónov s operačným systémom Android. Primárne časti aplikácie sú detailne zhrnuté nižšie spolu s vysvetlením a odôvodnením. Jednotlivé návrhy obrazoviek sú k nahliadnutiu v nasledujúcej podkapitole spolu s akceptačnými testami a štruktúrou databázy v podobe databázového modelu.

## Chat

Chat slúži na odosielanie a rýchly feedback medzi používateľmi, ktorý potrebujú rýchlo vyriešiť problém na svojom zariadení. Mimo textu môžu využiť aj možnosť video-hovoru pre zlepšenie a zrýchlenie hľadania riešenia. Podrobný návrh obrazoviek zahŕňa len dve obrazovky – prvá obsahuje prehľad histórie používateľov, s ktorými si bude prihlásený používateľ písať a druhá konkrétnu históriu správ medzi danými používateľmi.

## Fórum

Hlavný cieľ fóra je podobný ako pri chat-e, avšak tentokrát bude otázka / problém sprístupnený aj ostatným používateľom k nahliadnutiu, ktorý môžu prispieť svojim riešením. Následne môže používateľ, ktorý sa spýtal otázku označiť niektorú z komentárov za riešenie jeho problému, čo bude mať za následok rýchlejšie nájdenie odpovede pre budúcich používateľov s rovnakým problémom. Vzhľadom na komplexnejšiu štruktúru fóra obsahuje návrh obrazoviek 3 obrazovky – prvá slúžiaca ako prehľad otázok s jednoduchým filtrom, druhá obsahujúca prehľad konkrétnej otázky a posledná obsahuje formulár pre pridanie novej otázky používateľom.

## Články

Posledná časť aplikácie zahŕňa články, ktoré môžu pridať len špeciálny používatelia a ich primárnym účelom je zhrnúť najčastejšie tipy a triky alebo návody na časté chyby a problémy. Články, podobne ako fórum, môžu obsahovať prílohy, ktoré majú za účel uľahčiť čitateľovi možnosť pochopenie článku. Pre články existujú dve hlavné kategórie, ktoré sú zmienené aj vyššie, a to návody (pre riešenie problémov) a tipy a triky (pre uľahčenie používania zariadení). Počet aj význam obrazoviek je rovnaký ako pri fóre.

# Akceptačné testy

## Front end

### Napísanie správy inému používateľovi

|  |  |
| --- | --- |
| **Test 1: Napísanie správy inému používateľovi (+)** | |
| **Vstupné podmienky:** | Používateľ je prihlásený pod akýmkoľvek typom účtu v aplikácii a je pripojený na internet. |
| **Výstupné podmienky:** | V chat. histórii sa zobrazí nová správa a do databázy sa zapíše odoslaný obsah správy. |
| **Postup:** | 1. Používateľ sa pomocou hlavného MENU prepne na zobrazenie správ.  2. Po kliknutí sa zobrazí história používateľov s kým si prihlásený používateľ v minulosti písal.  3. Používateľ klikne na meno používateľa, ktorému chce napísať správu.  4. Zobrazí sa história správ medzi chceným používateľom a prihláseným používateľom.  5. Používateľ zadá želanú správu a klikne na tlačidlo odoslať.  6. Správa sa zobrazí v chatovacej histórii. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Pridať nový článok

|  |  |
| --- | --- |
| **Test 2: Pridať nový článok (+)** | |
| **Vstupné podmienky:** | Používateľ je prihlásený pod používateľom s admin právami, je pripojený na internet a názov článku je jedinečný. |
| **Výstupné podmienky:** | Článok je úspešne uložený do databázy, používateľ je upozornený o úspechu a zobrazí sa mu finálna verzia článku. |
| **Postup:** | 1. Používateľ sa pomocou hlavného MENU prepne na Best practices.  2. Zobrazí sa sumár aktuálnych článkov s tlačidlom "+".  3. Používateľ klikne na tlačidlo „+“.  4. Po kliknutí sa zobrazí formulár na vyplnenie nového článku.  5. Používateľ zadá požadované hodnoty a klikne na tlačidlo uloženia.  6. Používateľ je presmerovaný na novo vytvorený článok. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Položiť novú otázku do fóra

|  |  |
| --- | --- |
| **Test 3: Položiť novú otázku do fóra (+)** | |
| **Vstupné podmienky:** | Používateľ je pripojený do internetu a prihlásený. |
| **Výstupné podmienky:** | Používateľovi je zobrazená hláška o úspešnom pridaní otázky do databázy, používateľ je automaticky presmerovaný na novo vytvorenú otázku. |
| **Postup:** | 1. Používateľ klikne na MENU, vyberie možnosť fórum a zvolí možnosť pridať novú otázku.  2. Zobrazí sa mu formulár na vyplnenie požadovaných hodnôt s možnosťou pridať obrázok / video.  3. Po vyplnení formuláru a stlačení tlačidla uloženia je používateľ automaticky presmerovaný na novo vytvorenú otázku. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Úprava komentáru neprihláseným používateľom

|  |  |
| --- | --- |
| **Test 4: Úprava komentáru neprihláseným používateľom (-)** | |
| **Vstupné podmienky:** | Používateľ je pripojený do internetu, môže / nemusí byť prihlásený. |
| **Výstupné podmienky:** | Zobrazí sa hláška o nemožnosti editovať komentár cudzích používateľov. |
| **Postup:** | 1. Používateľ klikne na otázku na hlavnej obrazovke, na ktorej chce editovať komentár.  2. Zobrazí sa požadovaná otázka.  3. Používateľ klikne na komentár, ktorý chce upravovať.  4. Zobrazí sa hláška o nutnosti prihlásenia a nemožnosti editovať komentár anonymným / neprihláseným používateľom, ktorá sa po 10 sekundách automaticky skryje. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Vymazať článok bez prihlásenia

|  |  |
| --- | --- |
| **Test 5: Vymazať článok bez prihlásenia (-)** | |
| **Vstupné podmienky:** | Používateľ je pripojený do internetu, nemusí byť prihlásený. |
| **Výstupné podmienky:** | Možnosť vymazať článok nie je umožnená. |
| **Postup:** | 1. Používateľ klikne v MENU na „Best practices“.  2. Zobrazí sa prehľad článkov.  3. Používateľ klikne želaný článok.  4. Zobrazí sa požadovaný článok.  5. V hornej časti obrazovky je tlačidlo vymazania článku.  6. Po kliknutí na opísané tlačidlo je používateľ upozornený na nutnosť prihlásenia pod admin účtom. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

## Back end

### Získanie všetkých otázok a následný výber konkrétnej s načítaním komentárov

|  |  |
| --- | --- |
| **Test 1: Získanie všetkých otázok a následný výber konkrétnej s načítaním komentárov (+)** | |
| **Vstupné podmienky:** | Prístup na internet. |
| **Výstupné podmienky:** | Vrátenie otázky s komentármi. |
| **Postup:** | 1. Request GET - Požiadavka na načítanie všetkých otázok.  2. Reply [200] - Odpoveď so všetkými otázkami, bez komentárov a znenia samotnej otázky. Odpoveď obsahuje pole otázok primárne s hodnotami názov, dátum vytvorenia a meno tvorcu otázky.  3. Request GET - Požiadavka na vrátenie konkrétnej otázky.  4. Reply [200] - Odpoveď s konkrétnou otázkou, komentármi a jej znením. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Získanie histórie správ a následné prečítanie posledných správ prihláseného a žiadaného používateľa

|  |  |
| --- | --- |
| **Test 2: Získanie histórie správ a následné prečítanie posledných správ prihláseného a žiadaného používateľa (+)** | |
| **Vstupné podmienky:** | Autorizačný token, prístup na internet, ID druhého používateľa. |
| **Výstupné podmienky:** | Zníženie počtu neprečítaných správ medzi autentifikovaným používateľom a žiadaným používateľom na databáze. |
| **Postup:** | 1. Request GET - Požiadavka na načítanie neprečítaných správ medzi používateľmi.  2. Reply [200] - Vrátenie poslednej neprečítanej správy, celkového počtu neprečítaných správ, mena a ID používateľa.  3. Request PUT - Požiadavka na označenie správ za prečítané.  4. Reply [200] – Prázdna odpoveď s informovaním o úspechu. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Získanie článku a načítanie zodpovedajúcich obrázkov

|  |  |
| --- | --- |
| **Test 3: Získanie článku a načítanie zodpovedajúcich obrázkov (+)** | |
| **Vstupné podmienky:** | Prístup na internet. |
| **Výstupné podmienky:** | Vrátenie článku s obrázkami. |
| **Postup:** | 1. Request GET - Získanie všetkých aktuálnych článkov.  2. Reply [200] - Vrátenie všetkých článkov bez hlavnej časti článku (textu) a obrázkov, s hodnotami názov článku, kategórie a menom tvorcu.  3. Request GET - Načítanie konkrétneho článku na základe predchádzajúcej odpovede.  4. Reply [200] - Vrátenie celého článku vrátane textu a ID obrázkov, ktoré je možné načítať postupným dopytovaním sa na jednotlivé ID. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Získanie všetkých komentárov pre konkrétnu otázku s ich následným vymazaním neprihláseným používateľom

|  |  |
| --- | --- |
| **Test 4: Získanie všetkých komentárov pre konkrétnu otázku s ich následným vymazaním neprihláseným používateľom (-)** | |
| **Vstupné podmienky:** | Prístup na internet. |
| **Výstupné podmienky:** | Dáta zostanú na databáze nezmenené. |
| **Postup:** | 1. Request GET - Získanie všetkých otázok bez komentárov a znenia otázky.  2. Reply [200] - Odpoveď so všetkými otázkami, bez komentárov a znenia samotnej otázky, s hodnotami názov článku, kategórie a menom tvorcu.  3. Request GET - Získanie otázky so všetkými komentármi a jej znením.  4. Reply [200] - Odpoveď s konkrétnou otázkou, komentármi a jej znením.  5. Request DELETE - Požiadavka na vymazanie komentára  6. Reply [401] - Dáta zostanú na databáze nezmenené. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

### Vymazanie obrázku so zadaním nesprávneho ID otázky

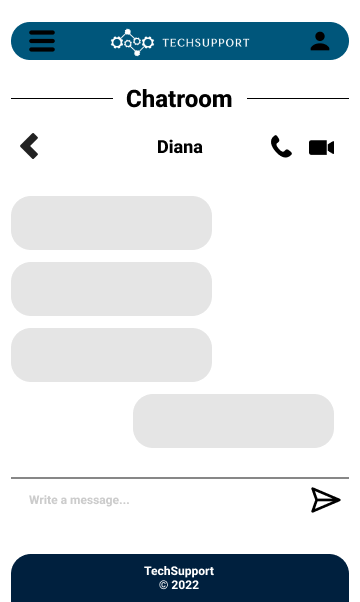
|  |  |
| --- | --- |
| **Test 5: Vymazanie obrázku so zadaním nesprávneho ID otázky (-)** | |
| **Vstupné podmienky:** | Prihlasovacie údaje privilegovaného používateľa, prístup na internet |
| **Výstupné podmienky:** | Dáta zostanú na databáze nezmenené. |
| **Postup:** | 1. Request GET - Získanie všetkých otázok bez komentárov a znenia otázky.  2. Reply [200] - Odpoveď so všetkými otázkami, bez komentárov a znenia samotnej otázky, s hodnotami názov článku, kategórie a menom tvorcu.  3. Request GET - Získanie otázky so všetkými komentármi, jej znením a ID príloh.  4. Reply [200] - Odpoveď s konkrétnou otázkou, komentármi a jej znením.  5. Request DELETE - Odoslanie požiadavky obsahujúcu ID prílohy a nesprávnu hodnotu otázky.  6. Reply [404] - Chybová hláška 404. |
| **Výsledok:** PASS **✓** ~~FAIL ✖~~ | |

# Obrazovky

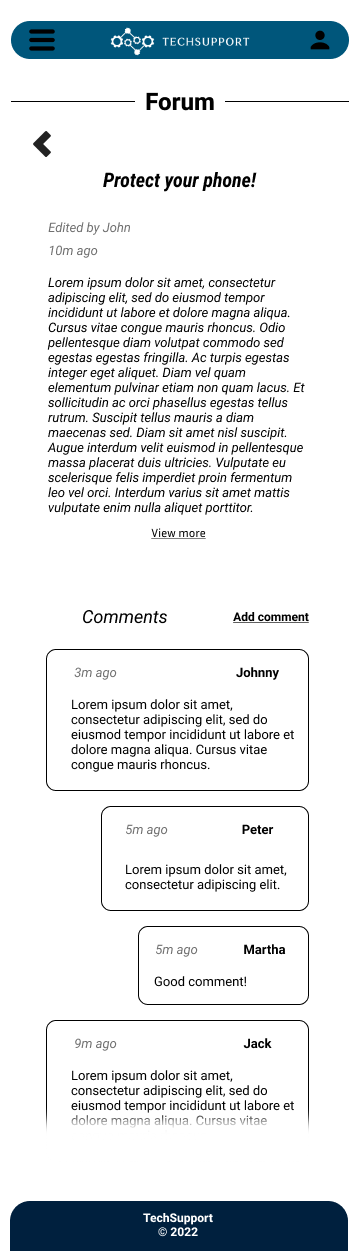
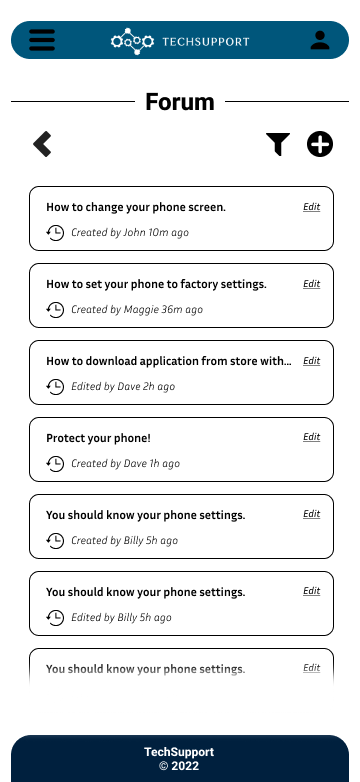
Jednotlivé návrhy obrazoviek sú uvedené nižšie podľa kategórii ktorú časť aplikácie zobrazujú. Časť s chat-om obsahuje dve obrazovky – prehľad histórie používateľov a prehľad histórie konverzácie medzi dvomi používateľmi. Fórum a články obsahujú po 3 obrazovky, presnejšie prehľad, zobrazenie konkrétneho článku / otázky a formulár na pridanie nového / úpravu článku / otázky. Táto kategória zobrazuje obrazovky v štádiu **wireframe-ov**. Ich aktuálnejší stav je možné prezrieť si v kapitole 6. Stránky boli pri vytváraní zmenené najmä z dôvodu, že niektoré návrhové vlastnosti boli implementačne náročné.

## Chat

Obrázok, na ktorom je text

Automaticky generovaný popis

## Fórum

Obrázok, na ktorom je text

Automaticky generovaný popis

## Články

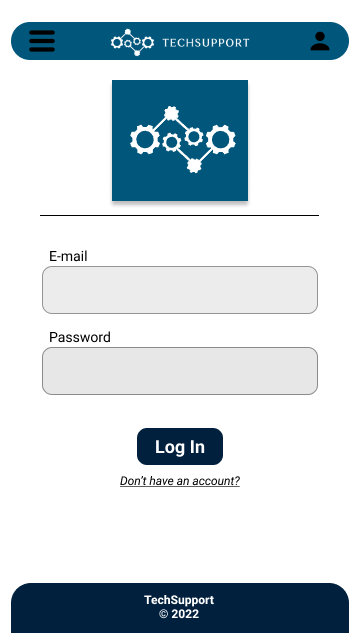
Obrázok, na ktorom je text

Automaticky generovaný popisObrázok, na ktorom je text

Automaticky generovaný popisObrázok, na ktorom je text

Automaticky generovaný popis

## Ostatné

 Obrázok, na ktorom je text

Automaticky generovaný popis

Obrázok, na ktorom je text, snímka obrazovky, znak

Automaticky generovaný popis Obrázok, na ktorom je text

Automaticky generovaný popis

# Databáza

Návrh databázy je detailne zobrazený na obrázku nižšie. Jednotlivé tabuľky sú vytvorené pomocou migrácii vo framework Laravel a sú dostupné spolu s celým zdrojovým kódom pre back-end aplikácie na tomto [odkaze](https://github.com/Mino21M/fiit-mtaa-app-back). Tabuľky *personal\_access\_tokens*, *migrations*, *failed\_jobs* a *password\_resets* sú vytvorené Laravel-om automaticky, a teda nezohľadujú pre aplikáciu žiadnu priamu interakciu. Tabuľka *users* je rovnako vytvorená automaticky, avšak sme ju modifikovali pridaním stĺpca *role*, pre určenie role práve prihláseného používateľa. Tabuľky končiace názvami *\_assets* slúžia na ukladanie binárnych štruktúr / súborov do databázy. Zvyšné názvy tabuliek priamo zodpovedajú aplikačnej funkcionalite.

Obrázok, na ktorom je stôl

Automaticky generovaný popis

# Back-end

Celý back-end je napísaný vo framework-u Laravel v9, fungujúcom na PHP v8 programovacom jazyku. Jednotlivé API volania sú zahrnuté v súbbore *routes/api.php*. Celý program je testovaný pomocou POSTMAN. Export volaní a odpovedí je možné nájsť na nasledovnom [odkaze](https://github.com/Mino21M/fiit-mtaa-app-docu) na ceste *REST API/json*, kde si je možné jednotlivé *json* súbory importovať do vlastnej inštancie POSTMAN. Krátky súhrn API endpoint-ov je dostupný nižšie. Autentifikácia používateľov voči backendu je vykonávaná na základe BEARER tokenov, ktoré sa generujú po zadaní správneho emailu a hesla používateľa. Po zadaní korektných prihlasovacích údajov je v odpovedi odoslaný token.

## Chat

*/api/chats*

*GET:* Načítanie histórie používateľov, s ktorými si autentifikovaný používateľ v minulosti písal.

*/api/chats/{user\_id}*

*GET:* Vrátenie histórie správ medzi používateľom *{user\_id}* a autentifikovaným používateľom.

*POST*: Odoslanie správy špecifikovanej v REQUEST BODY používateľovi *{user\_id}.*

/api/chats/*{user\_id}/read*

*PUT:* Označenie správ medzi používateľom *{user\_id}* za prečítané.

## Fórum

*/api/questions*

*GET:* Vrátenie všetkých otázok bez znenia textu otázky, komentárov a príloh.

*POST:* Pridanie novej otázky.

/api/*questions/latest*

*GET:* Načítanie 4 najaktuálnejších otázok.

/api/*questions/search*

*POST:* Filtrovanie otázok z fóra na základe niektorých atribútov.

*/api/questions/{question\_id}*

*GET:* Zobrazenie znenia otázky vrátane komentárov a príloh.

*PUT:* Opravenie znenia otázky, alebo jej názvu.

*DELETE:* Vymazanie otázky vrátane komentárov a príloh.

*/api/questions/{question\_id}/assets/{asset\_id}*

*GET:* Vrátenie binárnej hodnoty prílohy *{asset\_id}* pre otázku *{question\_id}*.

DELETE: Vymazanie prílohy *{asset\_id}* pre otázku *{question\_id}*.

*/api/questions/{question\_id}/assets*

*POST:* Pridanie novej prílohy *{asset\_id}* pre otázku *{question\_id}*.

*/api/questions/{question\_id}/comments/{comment\_id}/answer*

*POST:* Označenie otázky *{question\_id}* za zodpovedanú komentárom *{comment\_id}.*

*/api/questions/{question\_id}/comments*

*POST:* Pridaniekomentára *{comment\_id}* pre otázku *{question\_id}.*

*/api/questions/{question\_id}/comments/{comment\_id}*

*PUT:* Úpravakomentára *{comment\_id}* pre otázku *{question\_id}*.

*DELETE:* Vymazanie komentára *{comment\_id}* pre otázku *{question\_id}.*

## Články

*/api/articles*

*GET:* Načítanie všetkých článkov.

*POST*: Vytvorenie nového článku s dátami uloženými v REQUEST BODY.

*/api/articles/{article\_id}*

*GET*: Špecifický článok s ID *{article\_id}* načítaný spolu s komentármi a ID príloh.

*PUT*: Úprava špecifického článku s ID *{article\_id}*.

*DELETE*: Vymazanie článku s ID *{article\_id}*.

*/api/articles/{article\_id}/search*

*POST:* Prehľadávanie článkov podľa názvu špecifikovanom v REQUEST BODY.

*/api/articles/{article\_id}/assets*

*POST*: Pridanie prílohy pre článok *{article\_id}*.

*/api/articles/{article\_id}/assets/{asset\_id}*

*GET*: Načítanie konkrétnej prílohy *{asset\_id}* z článku *{article\_id}*.

*DELETE:* Odstránenie prílohy *{asset\_id}* z článku *{article\_id}*.

*/api/articles/latest/{category}*

*GET*: Načítanie najaktuálnejších článkov z kategórie *{category}*.

## Ostatné

*/api/test*

*GET*: Test pre overenie funkčnosti API endpoint.

*/api/token*

*POST:* vrátenie BEARER tokenu na základe emailu a hesla v REQUEST BODY.

*/api/register*

*POST:* vrátenie BEARER tokenu na základe emailu, mena a hesla v REQUEST BODY.

*/api/users*

*POST:* Zmena mena, hesla alebo emailu používateľa.

*/api/logout*

*POST:* odhlásenie (vymazanie tokenu) používateľa.

## Postman API dokumentácia

Pre kompletnú dokumentáciu API volaní a ich možných odpovedí sme vytvorili dokumentáciu priamo z aplikácie Postman. Každá veľká kategória v našej aplikácii má svoju vlastnú dokumentáciu. Linky sú uvedené nižšie.

**Questions**: <https://documenter.getpostman.com/view/15459180/Uyr5nyT6>

**Chat**: <https://documenter.getpostman.com/view/15459180/Uyr5nyT5>

**Other**: <https://documenter.getpostman.com/view/15459180/Uyr5nef2>

**Articles**: <https://documenter.getpostman.com/view/15459180/Uyr5neez>

# Front end

Front-end aplikácie je implementovaný vo framework-u React Native a primárne určený pre platformu Android. Zdrojový kód je dostupný na [odkaze](https://github.com/Mino21M/fiit-mtaa-app-front). Front-end bol rozdelený do viacerých kategórií, menovite **domovská stránka**, **fórum**, **články** a **chat**. Každá kategória obsahuje súbor ďalších podstránok, ktoré pomáhajú dosiahnuť primárnu funkcionalitu. Front-end plne pracuje s API volaniami, ktoré boli implementované do back-end časti, aby bolo možné zobrazovanie skutočných dát priamo v aplikácii. Front-end taktiež realizuje vopred zadané **akceptačné testy**, aby bola demonštrovaná jeho správna funkcionalita. V nasledujúcich častiach budú opísané niektoré z významných komponentov z hľadiska front-end časti, ale taktiež sa zameriame na vysvetlenie fungovania jednotlivých stránok a ich podstránok.

## Obrázok, na ktorom je text Automaticky generovaný popisKomponenty

Prvým významným komponentom aplikácie je **hlavička**. Hlavička v tomto prípade ostáva fixná a nemenná naprieč celou aplikáciou. Drží v sebe tlačidlá, ktoré pre používateľa sprístupňujú **hlavné menu** a **používateľské menu**. Oba typy menu sú zobrazené na obrázkoch napravo. Hlavička taktiež obsahuje hlavné logo našej fiktívnej spoločnosti, ktorá poskytuje technickú podporu.

Ďalším komponentom je **pätička**. Rovnako ako pri hlavičke, aj tento komponent nemení svoju pozíciu, tvar, ani viditeľnosť. Nachádza sa na spodnej strane aplikácie. Pätičku je možné vidieť v predošlých obrázkoch na spodnej strane.

Čo sa týka rozloženia stránky, hlavnou formou je nasledujúce usporiadanie: hlavička, obsah, pätička. Obsahová časť je celá zaobalená do prvku **ScrollView**, aby tento prvok nemusel byť vytvorený pre každú stránku osobitne.

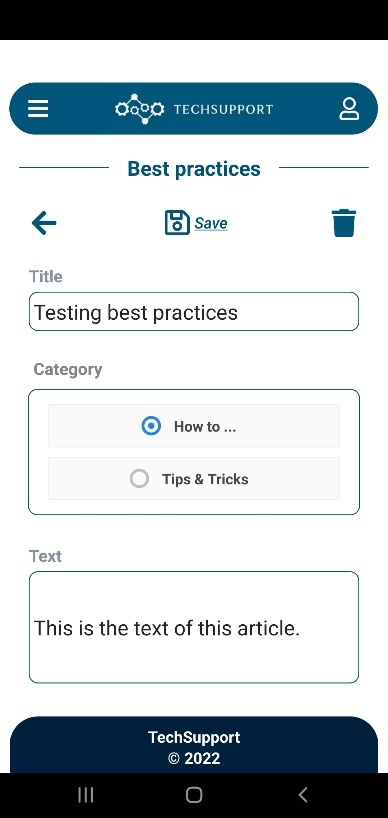
Jednou z veľkých pomocí je osobitný súbor **Navigator.js**, ktorý umožňuje prechod medzi jednotlivými stránkami veľmi jednoducho. Každá stránka má pridelené meno, pričom po zavolaní tohto mena nás Navigator presmeruje na požadovanú stránku.

## Stránky

V tejto časti si prejdeme všetky stránky, ktoré sú obsiahnuté v našej aplikácii. Jednotlivé stránky budú obsahovať stručný popis toho, na čo slúžia, a prípadne budú spomenuté ďalšie komponenty, ktoré sa na danú stránku viažu. Radi by sme poznamenali, že každá stránka má svoj osobitný JavaScript súbor, ktorý jej poskytuje **štýly**. Týmto sme sa chceli vyhnúť excesívnemu používaniu inline štýlovania.

## Stránka ArticleAddScreen

V tejto stránke sa rieši pridávanie článkov do aplikácie. Pridanie článku je možné vykonať iba z **administrátorského účtu**. Články sú rozdelené do dvoch kategórii, a to **How To...** a **Tips & Tricks**. Na stránke používateľ vypĺňa názov článku, jeho text a kategóriu, ale môže taktiež k článku pripojiť aj **prílohu**, ktorá môže predstavovať nejaký textový súbor, prípadne nejaký obrázok. Po vyplnení údajov je článok vytvorený a jeho dáta sú zapísané do databázy. Následne je používateľ presmerovaný priamo na stránku novo vytvoreného článku, ktorý si v prípade potreby môže ešte upraviť, alebo ho môže úplne odstrániť.

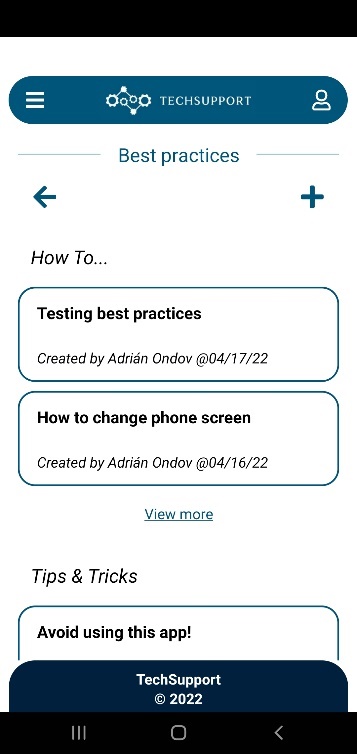


## Stránka ArticleEditScreen

Ako bolo spomenuté vyššie, používateľ s administrátorskými právami môže články aj upravovať. Práve na to je prispôsobená táto stránka. Jej rozloženie je totožné so stránkou ArticleAddScreen, jedinou zmenou sú dáta, ktoré sa z databázy získavajú pri načítavaní stránky. Všetky textové vstupy sú teda vopred vyplnené z databázy a používateľ má možnosť ich upraviť/vymazať.

## Stránky ArticleIndexScreen a ArticleIndexAllScreen

Obrázok, na ktorom je text

Automaticky generovaný popisTieto stránky zobrazuje kompletný zoznam článkov, ktoré sa v databáze momentálne nachádzajú. Kategorizácia je realizovaná iba na stránke ArticleIndexScreen, čo sa na druhej stránke nenachádza, avšak na stránke ArticleIndexAllScreen je umožnené **full-text** **vyhľadávanie na základe názvu článku**. Používateľ sa takto vie k článku dostať omnoho rýchlejšie, najmä ak by sa v databáze takých článkov nachádzalo veľa. Po kliknutí na konkrétny článok je potom používateľ presmerovaný na jeho osobitnú stránku.

Obrázok, na ktorom je text

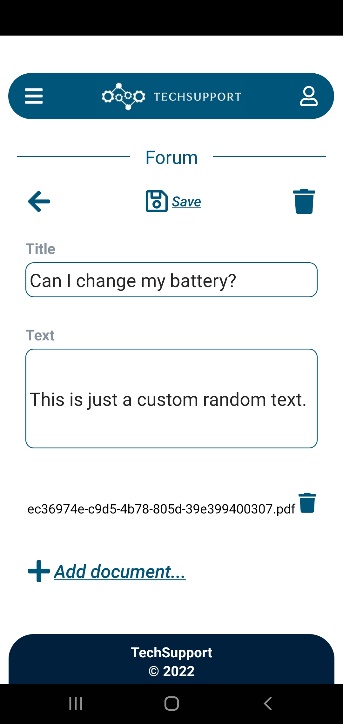
Automaticky generovaný popis

## Stránka ArticleShowScreen

Stránka ArticleShowScreen zobrazuje už konkrétny článok. Je tu možné nájsť názov článku, jeho autora a čas, kedy bol vytvorený, alebo taktiež aj jeho kategóriu. Na spodnej časti stránky sú zobrazené prílohy, ak sa k danému článku nejaké viažu.

## Stránka ForumAddScreen

Na tejto stránke sa prvýkrát dostávame k ďalšej časti, a to k **fóru**. Podobne ako pri článkoch, aj tu je ich vytváranie a editácia riešená podobne. Stránka ForumAddScreen má preto implementované vytváranie nových príspevkov do fóra. Pri tvorbe príspevku je možné vyplniť jeho názov, textový obsah, a prípadne je možné pridať prílohu. Po uložení príspevku je používateľ presmerovaný na jeho stránku, kde si môže príspevok skontrolovať a prezerať.



## Stránka ForumEditScreen

Editácia príspevkov vo fóre je realizovaná práve na tejto stránke. Štylizačne vyzerá totožne ako predošlá stránka, teda ForumAddScreen, ale s tou zmenou, že sú dáta vopred vyplnené z databázy. Používateľ môže akýkoľvek z dostupných atribútov (názov, text, prílohy) ľubovoľne meniť. Príspevok je možné z tejto stránky taktiež odstrániť, ale to môže urobiť iba administrátor, alebo **používateľ, ktorý príspevok vytvoril**.

## Stránka ForumIndexScreen

Pre zobrazenie celého zoznamu príspevkov, resp. opýtaných otázok vo fóre, bola vytvorená práve táto stránka. Jedná sa teda o zobrazenie zoznamu príspevkov, ktoré si používateľ môže prehliadať. Je tu implementované **full-text vyhľadávanie na základe názvu otázky**.

## Obrázok, na ktorom je text Automaticky generovaný popisObrázok, na ktorom je text, snímka obrazovky, vizitka Automaticky generovaný popisStránka ForumShowScreen

Na tejto stránke sa zobrazuje konkrétny príspevok fóra. Odtiaľ je možné prejsť do editácie príspevku, ale taktiež je tu možné **pridať komentár** k príspevku. Komentovať môže **iba prihlásený používateľ**, bez ohľadu na jeho rolu. Používateľ môže upraviť text svojho komentára, alebo ho môže vymazať. Ak sa jedná o výstižnú odpoveď na položenú otázku, autor otázky môže odpoveď označiť za **najlepšiu**, čím ju dá do pozornosti ostatných používateľov. Otázka je následne označená ako zodpovedaná aj na stránke ForumIndexScreen. Všetky spomenuté scenáre sú zobrazené v obrázkoch napravo.

Obrázok, na ktorom je text

Automaticky generovaný popis

## Stránka AboutUsScreen

Táto stránka slúži len na estetické účely. Dá sa k nej pristúpiť cez kontextové menu v hlavičke. Na nej používateľ nájde stručné informácie o sídle a kontaktných údajoch našej fiktívnej firmy. Stránka ponúka iba tlačidlo „späť“, aby bolo možné dostať sa na predošlú stránku rýchlo.

## Stránka HomeScreen

HomeScreen predstavuje hlavnú stránku našej aplikácie. Je na nej možné nájsť viacero prepojení na iné stránky, ako napríklad Kategórie v dolnej časti stránky, kde sú prepojenia na všetky typy stránok v aplikácii. Taktiež je na nej zobrazená tabuľa so 4 najnovšími príspevkami vo fóre, ktoré po kliknutí taktiež prepoja používateľa priamo na stránku daného príspevku. Ide o veľmi jednoduchú stránku, ktorá má mať skôr estetický charakter, než aby poskytovala veľa funkcionalít.

## Obrázok, na ktorom je text Automaticky generovaný popisStránka LoginScreen

Na prihlásenie používateľa slúži stránka LoginScreen. Používateľ sa môže po zadaní e-mailu a hesla prihlásiť, samozrejme ale iba ak má daný používateľ vytvorený účet. Ak nie, aplikácia ho vráti späť s chybou, ktorá príde od API volania. Navyše je na stránke vytvorený link, ktorý odkazuje používateľa na stránku registračného formulára. Na túto stránku je používateľ takmer vždy presmerovaný, ak ide v aplikácii vykonať niečo, čo si vyžaduje prihláseného používateľa.

## Obrázok, na ktorom je text Automaticky generovaný popisStránka RegisterScreen

Registrácia je zabezpečená pomocou registračného formulára, ktorý sa nachádza priamo na tejto stránke. Používateľ doňho zadáva svoje meno, prihlasovací e-mail a heslo, ktoré musí opakovaným napísaním potvrdiť. Ak je registrácia úspešná, používateľ je presmerovaný na domovskú stránku, pričom je automaticky prihlásený. V opačnom prípade mu vyskočí chybová hláška s chybami, ktoré vrátilo API volanie. Registračný formulár obsahuje link, ktorý prepojí používateľa priamo na stránku prihlásenia v prípade, že je daný používateľ už registrovaný a nepotrebuje vykonať registračný proces znova. Týmto sme chceli docieliť zrýchlenie pohybu v aplikácii.

## Obrázok, na ktorom je text Automaticky generovaný popisStránka SettingsScreen

Stránka SettingsScreen poskytuje používateľovi rôzne nastavenia profilu. Je možné zmeniť si meno, e-mail, alebo dokonca aj heslo, ktoré má databáza uložené. Pre autenticitu stránky bola táto funkcionalita spojazdnená, a to aj napriek tomu, že si ju zadanie, alebo akceptačné testy nevyžadujú.

## Obrázok, na ktorom je text Automaticky generovaný popisStránka ChatIndexScreen

Na zobrazenie všetkých konverzácii používame stránku ChatIndexScreen. Jej rozloženie sa podobá na rozloženie stránky so zoznamom príspevkov vo fóre. Konverzácie sú v jednoduchosti vypísané pod seba a na každú z nich je možné kliknúť. Následne je používateľ presmerovaný na konkrétny chat s iným používateľom.

## Stránka ChatShowScreen

Stránka ChatShowScreen zobrazuje už konkrétnu konverzáciu medzi dvoma používateľmi. Správy z druhej strany sú zarovnané doľava, zatiaľ čo naše správy sú zarovnané doprava, podobne ako pri bežných chat aplikáciách. Na spodnej časti stránky sa nachádza vstup, kde používateľ môže napísať svoju správu a následne ju odoslať druhému používateľovi. Po odoslaní správy sa jej text zobrazí aj v chat-e.

## Stránky CallScreen a JoinScreen

Tieto stránky sú zamerané na zahájenie hovoru za pomoci **WebRTC**. Používatelia, ktorí si chcú zavolať, sú pridaní do osobitnej miestnosti, kde môžu svoj hovor vykonať. Na jednej strane je potrebné, aby používateľ inicializoval hovor, zatiaľ čo na druhej strane ho používateľ potrebuje iba prijať, resp. sa doňho pripojiť. Používatelia pri hovore vidia svoj obraz z kamery, ale taktiež aj obraz z kamery na druhej strane. Používateľ ďalej môže z hovoru buď odísť, alebo si môže otočiť kameru, prípadne si vie vypnúť zvuk hovoru. Hovor teda je možné realizovať, pričom sa využíva požadované WebRTC.

## Obrázok, na ktorom je text, snímka obrazovky, elektronika, počítač Automaticky generovaný popisŠtýlovanie

Takmer každá stránka má svoju vlastnú dodatkovú stránku vytvorenú len pre jej **štýly**. Každý komponent, ktorý to potrebuje, má vytvorený svoj vlastný záznam v štýloch. Súbory, ktoré tieto štýly držia, sa nachádzajú v osobitnom priečinku, oddelene od stránok. Takto bolo možné dosiahnuť väčšiu prehľadnosť, aby sme sa vyhli priveľkému inline štýlovaniu. Súbory so štýlmi sú organizované do rovnakých kategórií a priečinkov, ako sú aj konkrétne stránky, aby bolo jednoduché prideliť jednej stránke práve jedno štýlovanie. Toto je však použité iba na bežné stránky v aplikácii, komponenty ako hlavička, alebo pätička, používajú inline štýlovanie. Ako príklad je vedľa uvedený jeden zo súborov, ktorý obsahuje štýly pre konkrétnu stránku.

## WebRTC

Implementáciu WebRTC videohovoru sme implamentovali za pomoci firestore, react-native-webrtc a repozitáru na githube ([link](https://github.com/DipanshKhandelwal/react-native-webrtc-firebase)), ktorý dané funkcionality už mal funkčné a implementované. Kód z daného repozitáru bolo potrebné trochu upraviť keďže posledný commit bol z pred dvoch rokov a nebol udržiavaný. Spomenutá úprava zahŕňala inštaláciu nových packages a drobnú úpravu zdrojového kódu *CallScreen* a *JoinScreen*. Po ich sfunkčnení sme obrazovky upravili do podoby v akej sa práve teraz nachádzajú z pohľadu UI a zjednodušili sme možnosť vytvoriť hovor a pripojiť sa k hovoru. V pôvodnom kóde bolo potrebné zadať ID „miestnosti“, v ktorej sa mal hovor konať (čo sa ďalej uložilo do databázy firestore). V našej aplikácii bolo možné využiť ID používateľov pre vytvorenie takejto miestnosti, kde dané ID sa určovalo z nasledovnej časti kódu: *route.params.my\_id > route.params.op\_id ? `${route.params.my\_id}\_${route.params.op\_id}` : `${route.params.op\_id}\_${route.params.my\_id}`*. Zvyšná časť kódu fungovala pomocou *firestore*, ktorý upozornil jedného z používateľov v prípade, ak sa daná miestnosť zmenila (pridala sa *sdp* požiadavka zo strany používateľa vytvárajúceho hovor). Následne druhý používateľ v *JoinScreen* na základe požiadavky vytvoril odpoveď a rovnako ju uložil na firestore, pomocou ktorého sa o odpovedi dozvedel aj prvý používateľ. Do obrazovky *CallScreen* (vytvárajúcej hovor) sa dá pripojiť stlačením tlačidla so symbolom telefónu. Druhý používateľ sa pripojí do tohto hovoru stlačením tlačidla so symbolom plus a do pár sekúnd sa celý hovor aj s obrazom a zvukom spustí.

# Príloha A: Zadanie

Zadanie projektu – MTAA 2022

Vytvorte vo dvojiciach mobilnú aplikáciu (iOS / Android) podľa Vášho zadania, ktoré si zvolíte na prvom cvičení. Zadanie bude preberané a hodnotené formou viacerých kontrolných bodov v priebehu semestra.

Front-end

Každá aplikácia musí mať aspoň 5 obrazoviek, kde každá obrazovka pokrýva nejaký use case. Do počtu 5 obrazoviek sa nepočíta splash screen (úvodná obrazovka počas načítavania aplikácie) ani login screen (prihlásenie používateľa).

Každá aplikácia musí pokryť aspoň jeden scenár vytvorenia dát, zobrazenia dát, úpravy dát a mazania dát (CRUD). Každá aplikácia musí komunikovať s vlastným back-endom pomocou HTTP volaní založených na princípoch REST API.

Aplikácia môže byť naprogramovaná v ľubovoľnej technológií, ktorej výstupom je natívny kód, tzn. nie sú povolené technológie, ktoré renderujú HTML (web view).

Povolené technológie na front-end aplikácie:

* SWIFT (iOS)
* Java (Android)
* Kotlin (Android)
* React Native (cross-platform)
* Xamarin (cross-platform)
* Flutter (cross-platform)
* Corona (cross-platform) – nie je úplne odporúčané na tomto predmete
* Unity

Nepovolené technológie:

* Ionic
* 7Framework
* Apache Cordova
* Angular.js
* jQuery Mobile
* Adobe PhoneGap
* Electron

Akákoľvek iná technológia podlieha schváleniu cvičiacim

Back-end

Back-end aplikácie realizujte formou vlastného kódu a vlastnej inštalácie databázy (technológia je na vás), ako aj samotné prostredie (XAMPP priamo na vašom notebooku, Virtuálny server cez VirtualBox). Ako databázu si môžete zvoliť MariaDB, MySQL, PostgreSQL, MongoDB. Ak máte záujem o inú databázu, podlieha schváleniu cvičiacim.

Aplikácia bude s back-endom komunikovať prostredníctvom API endpointov (konkrétnych URL ktoré bude aplikácia volať prostredníctvom HTTP volaní). Backend musí mať implementovaných aspoň 5 volaní, pričom musí mať minimálne jedno volanie GET, PUT, POST a DELETE. Minimálne jedným volaním musí byť aplikácia schopná zapísať do databázy binárny obsah (obrázok, video, PDF, ...) a jedným volaním musí byť schopná aplikácia takýto obsah načítať.

Videokonferencia

Aplikácia musí podporovať možnosť pre používateľov nadviazať video-konferenčný hovor, kde sa účastníci vidia a počujú. Videokonferencia bude technologicky založená na protokole WebRTC, pričom na zabezpečenie tejto funkcie môžete použiť ľubovoľnú knižnicu, ktorú nájdete na internete. (Je potrebné v dokumentácií uviesť ktorú a ako bola použitá).

Dokumentácia k projektu

Súčasťou projektu je vypracovanie dokumentácie, ktorá definuje aký je účel aplikácie, akú funkcionality aplikácia pokrýva, prostredníctvom akých volaní komunikuje s back-endom, aké dáta sa prenášajú.

Dokumentácia API endpointov môže byť realizovaná formou ľubovoľného nástroja na správu API dokumentácie (HTML export, ....)

Ďalej sú súčasťou dokumentácie akceptačné testy pre front-end aj back-end.

Ukážky API dokumentácií:

* https://openweathermap.org/current
* https://billdu.docs.apiary.io/#reference/documents/documents/list-all-documents
* https://www.zabbix.com/documentation/4.0/manual/api

Zdrojové kódy a rozdelenie práce

Každá dvojica si vytvorí dva GitHub repozitáre, jeden pre front-end, jeden pre back-end. Každá dvojica si musí rovnomerne rozdeliť prácu aj na front-ende aj na back-ende (rozdeliť si implementáciu endpointov, rozdeliť si implementáciu obrazoviek) a toto rozdelenie musí byť jasné z commitov v GitHub repozitári.

Akceptačné testy

Každá dvojica navrhne 10 akceptačných testov, pričom budú rozdelené nasledovne:

* 5 akceptačných testov aplikácie
  + Z toho 3 kladné (správne používanie), 2 záporné (nesprávne používanie)
  + Každý akceptačný test frontendu musí pozostávať aspoň z 3 krokov, ktoré na seba nadväzujú a tvoriť logický celok (dávať zmysel, testujeme funkčnosť niečoho)
* 5 akceptačných testov pre back-end
  + z toho 3 kladné (správne používanie), 2 záporné (nesprávne používanie)
  + Každý test back-endu musí pozostávať aspoň z dvoch dvojíc requestresponse, ktoré na seba nadväzujú.

**Ukážka akceptačného testu front-end:**

|  |  |
| --- | --- |
| **Test 1: Vytvorenie nového článku** | |
| **Vstupné podmienky:** | Používateľ ktorý má rolu administrátora je prihlásený v aplikácií, aplikácia má pripojenie na Internet |
| **Výstupné podmienky:** | V aplikácií pribudne článok, ktorý sa uloží do databázy a bude možné ho zobraziť. |
| **Postup:** | 1. Používateľ stlačí tlačidlo MENU, ktoré sa vyroluje zľava 2. Používateľ stlačí tlačidlo NOVÝ ČLÁNOK 3. Zobrazí sa formulár na pridanie článku 4. Používateľ vyplní nadpis, text článku, a vyberie obrázok 5. Používateľ klikne na tlačidlo ULOŽIŤ 6. Zobrazí sa informácia o vytvorení článku 7. Po 3 sekundách je používateľ v aplikácií presmerovaný na nový článok |
| **Výsledok:** PASS / FAIL | |

Obrázok, na ktorom je stôl

Automaticky generovaný popis