# TechSupport

MICHAL MINÁR, ADRIÁN ONDOV

# Obsah

# 1. 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. 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.

## 1.1. 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.

## 1.2. 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.

## 1.3. 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.

# 2. Akceptačné testy

## 2.1. Front end

### 2.1.1. 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.<br>2. Po kliknutí sa zobrazí história používateľov s kým si prihlásený používateľ v minulosti písal.<br>3. Používateľ klikne na meno používateľa, ktorému chce napísať správu.<br>4. Zobrazí sa história správ medzi chceným používateľom a prihláseným používateľom.<br>5. Používateľ zadá želanú správu a klikne na tlačidlo odoslať.<br>6. Správa sa zobrazí v chatovacej histórii. |
| **Výsledok:** PASS / FAIL | |

### 2.1.2. 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.<br>2. Zobrazí sa sumár aktuálnych článkov s tlačidlom "+".<br>3. Používateľ klikne na tlačidlo „+“.<br>4. Po kliknutí sa zobrazí formulár na vyplnenie nového článku.<br>5. Používateľ zadá požadované hodnoty a klikne na tlačidlo uloženia.<br>6. Používateľ je presmerovaný na novo vytvorený článok. |
| **Výsledok:** PASS / FAIL | |

### 2.1.3. 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.<br>2. Zobrazí sa mu formulár na vyplnenie požadovaných hodnôt s možnosťou pridať obrázok / video.<br>3. Po vyplnení formuláru a stlačení tlačidla uloženia je používateľ automaticky presmerovaný na novo vytvorenú otázku. |
| **Výsledok:** PASS / FAIL | |

### 2.1.4. 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.<br>2. Zobrazí sa požadovaná otázka.<br>3. Používateľ klikne na komentár, ktorý chce upravovať.<br>4. Zobrazí sa hláška o nutnosti prihlásenia a nemožnosti editovať komentár anonymným / neprihláseným používateľom, ktorá sa po 10 sekundách automaticky skryje. |
| **Výsledok:** PASS / FAIL | |

### 2.1.5. 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“.<br>2. Zobrazí sa prehľad článkov.<br>3. Používateľ klikne želaný článok.<br>4. Zobrazí sa požadovaný článok.<br>5. V hornej časti obrazovky je tlačidlo vymazania článku.<br>6. Po kliknutí na opísané tlačidlo je používateľ upozornený na nutnosť prihlásenia pod admin účtom. |
| **Výsledok:** PASS / FAIL | |

## 2.2. Back end

### 2.2.1. 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.<br>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.<br>3. Request GET - Požiadavka na vrátenie konkrétnej otázky.<br>4. Reply [200] - Odpoveď s konkrétnou otázkou, komentármi a jej znením. |
| **Výsledok:** PASS ✓ ~~FAIL ✗~~ | |

### 2.2.2. 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.<br>2. Reply [200] - Vrátenie poslednej neprečítanej správy, celkového počtu neprečítaných správ, mena a ID používateľa.<br>3. Request PUT - Požiadavka na označenie správ za prečítané.<br>4. Reply [200] – Prázdna odpoveď s informovaním o úspechu. |
| **Výsledok:** PASS ✓ ~~FAIL ✗~~ | |

### 2.2.3. 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.<br>2. Reply [200] - Vrátenie všetkých článkov bez hlavnej časti článku (textu) a obrázkov, s hodnotami názov článku, kategórie a menom tvorcu.<br>3. Request GET - Načítanie konkrétneho článku na základe predchádzajúcej odpovede.<br>4. Reply [200] - Vrátenie celého článku vrátane textu a ID obrázkov, ktoré je možné načítať postupným dopytovaním sa na jednotlivé ID. |
| **Výsledok:** PASS ✓ ~~FAIL ✗~~ | |

### 2.2.4. 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.<br>2. Reply [200] - Odpoveď so všetkými otázkami, bez komentárov a znenia samotnej otázky, s hodnotami názov článku, kategórie a menom tvorcu.<br>3. Request GET - Získanie otázky so všetkými komentármi a jej znením.<br>4. Reply [200] - Odpoveď s konkrétnou otázkou, komentármi a jej znením.<br>5. Request DELETE - Požiadavka na vymazanie komentára<br>6. Reply [401] - Dáta zostanú na databáze nezmenené. |
| **Výsledok:** PASS ✓ ~~FAIL ✗~~ | |

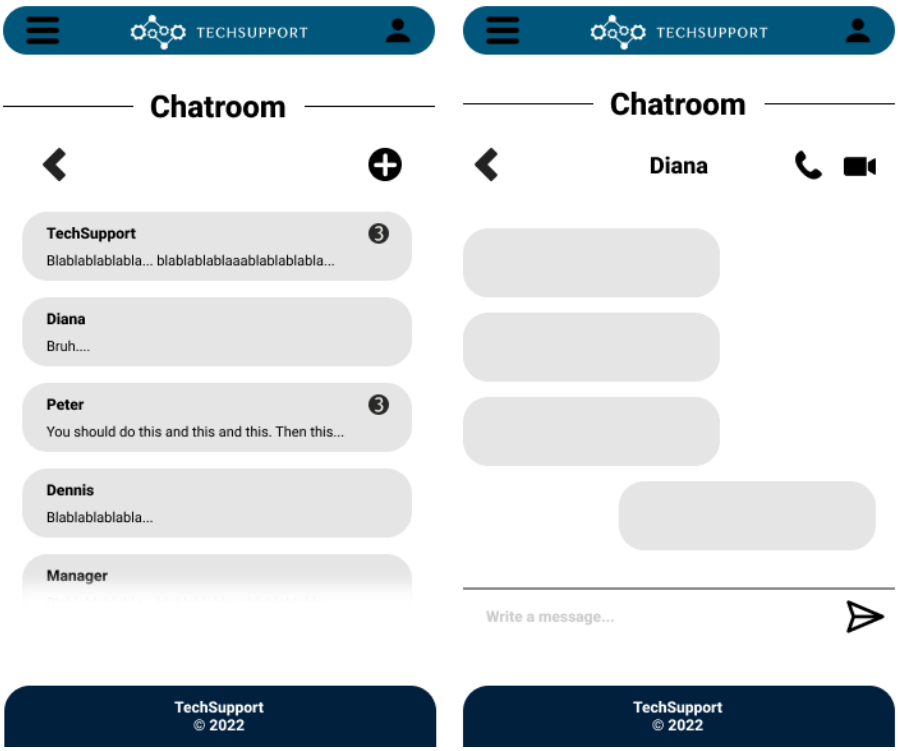
### 2.2.5. Vymazanie obrázku so zadaním nesprávneho ID otázky

| **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.<br>2. Reply [200] - Odpoveď so všetkými otázkami, bez komentárov a znenia samotnej otázky, s hodnotami názov článku, kategórie a menom tvorcu.<br>3. Request GET - Získanie otázky so všetkými komentármi, jej znením a ID príloh.<br>4. Reply [200] - Odpoveď s konkrétnou otázkou, komentármi a jej znením.<br>5. Request DELETE - Odoslanie požiadavky obsahujúcu ID prílohy a nesprávnu hodnotu otázky.<br>6. Reply [404] - Chybová hláška 404. |
| **Výsledok:** PASS ✓ ~~FAIL ✗~~ | |

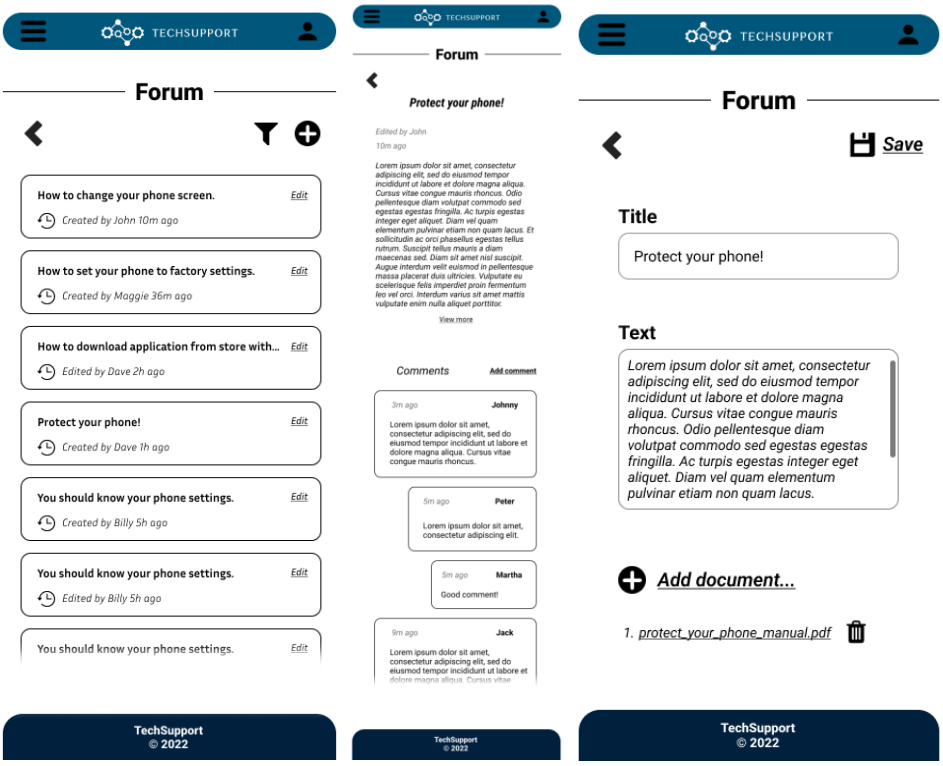
# 3. Obrazovky

Jednotlivé návrhy obrazoviek sú uvedené nižšie podľa kategórii ktorú časť aplikácie zobrazujú. Časť s chatom 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.

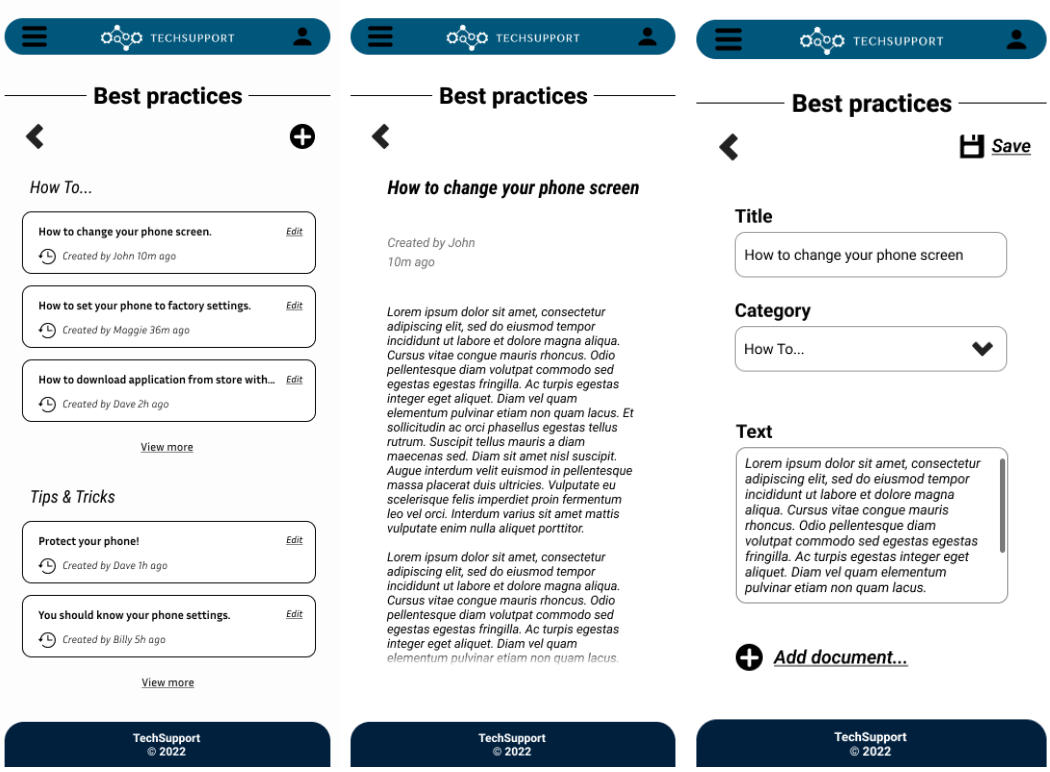
## 3.1. Chat



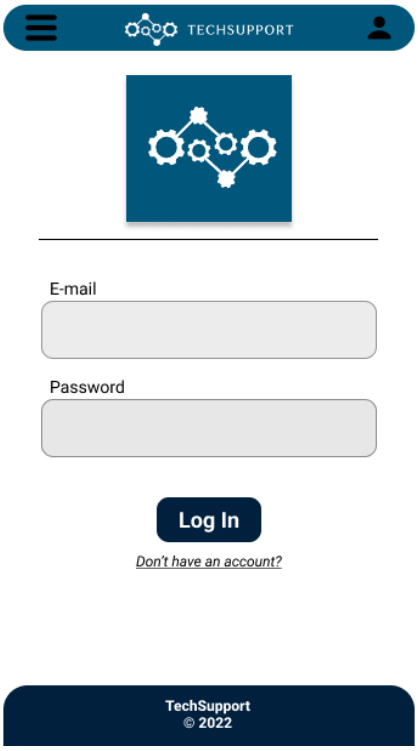
## 3.2. Fórum

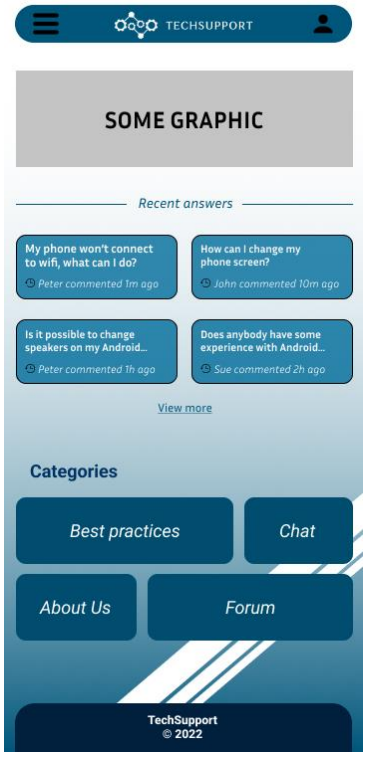


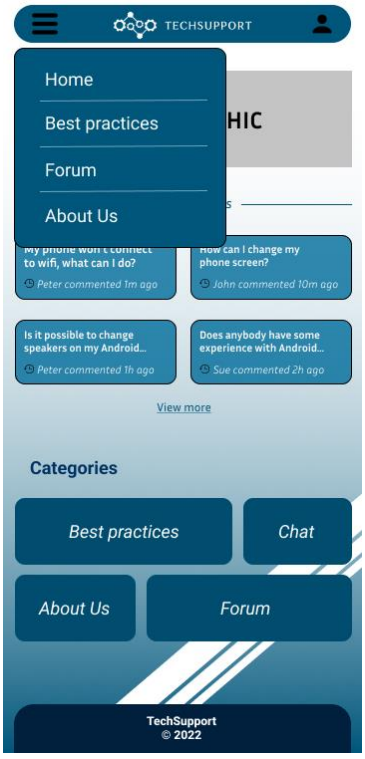

## 3.3. Články

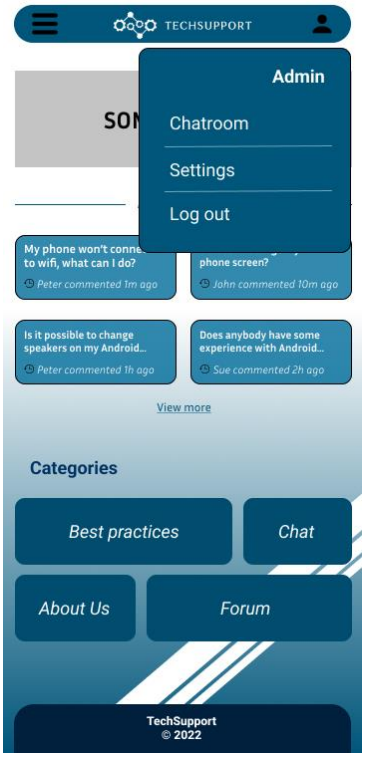


## 3.4. Ostatné


TECHSUPPORT
E-mail
Password
Log In
Don't have an account?
TechSupport
© 2022

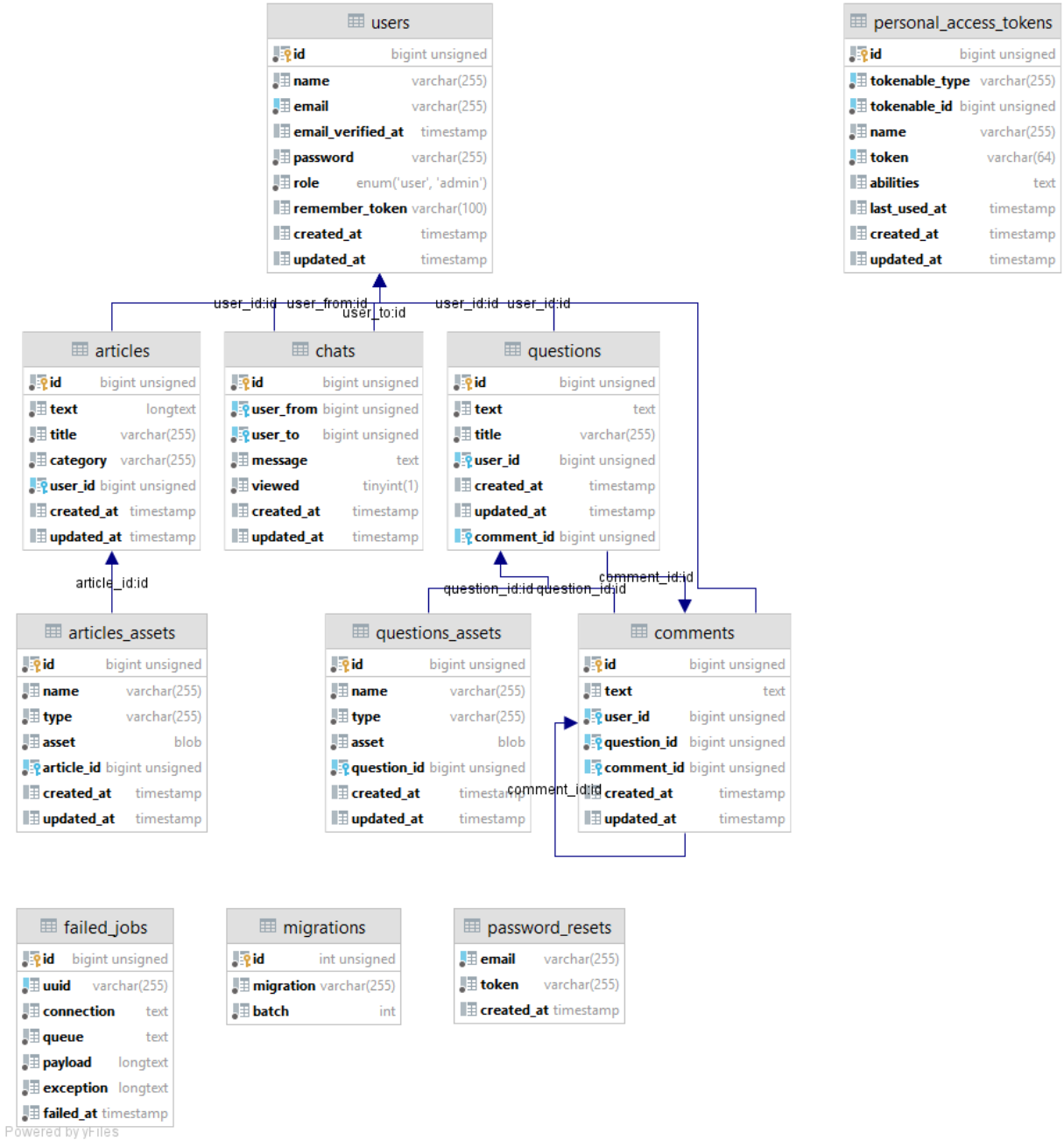

TECHSUPPORT
SOME GRAPHIC
Recent answers
My phone won't connect to wifi, what can I do?
Peter commented 1m ago
How can I change my phone screen?
John commented 10m ago
Is it possible to change speakers on my Android...
Peter commented 1h ago
Does anybody have some experience with Android...
Sue commented 2h ago
View more
Categories
Best practices
Chat
About Us
Forum
TechSupport
© 2022


TECHSUPPORT
Home
Best practices
Forum
About Us
Peter commented 1m ago
John commented 10m ago
Peter commented 1h ago
Sue commented 2h ago
View more
Categories
Best practices
Chat
About Us
Forum
TechSupport
© 2022


TECHSUPPORT
Admin
Chatroom
Settings
Log out
Peter commented 1m ago
John commented 10m ago
Peter commented 1h ago
Sue commented 2h ago
View more
Categories
Best practices
Chat
About Us
Forum
TechSupport
© 2022

# 4. 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. 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.



# 5. 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 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.

## 5.1. 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.

## 5.2. 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:* Pridanie komentára *{comment_id}* pre otázku *{question_id}.*

*/api/questions/{question_id}/comments/{comment_id}*

*PUT:* Úprava komentára *{comment_id}* pre otázku *{question_id}*.

*DELETE:* Vymazanie komentára *{comment_id}* pre otázku *{question_id}.*

## 5.3. 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}*.

## 5.4. 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.

# 6. Front end

Front end aplikácie je implementovaný vo framework-u React Native a primárne určený pre platformu Android. Zdrojový kód je dostupný na odkaze.

*Aktuálne je front end časť v procese implementácie, a preto tu nie sú uvedené dodatočné informácie.*

# 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<br>2. Používateľ stlačí tlačidlo NOVÝ ČLÁNOK<br>3. Zobrazí sa formulár na pridanie článku<br>4. Používateľ vyplní nadpis, text článku, a vyberie obrázok<br>5. Používateľ klikne na tlačidlo ULOŽIŤ<br>6. Zobrazí sa informácia o vytvorení článku<br>7. Po 3 sekundách je používateľ v aplikácií presmerovaný na nový článok |
| **Výsledok:** PASS / FAIL | |

**Ukážka akceptačného testu back-end:**

| **Test 1: Úprava existujúceho článku (negative test)** | |
|---|---|
| **Vstupné podmienky:** | Súčasťou každého volania musí byť autorizačný token |
| **Výstupné podmienky:** | Existujúci článok v databáze sa nezmení |
| **Postup:** | 1. Aplikácia zavolá volanie HTTP GET /getArticleData/[id] pričom poskytne platné id existujúceho článku.<br>2. Volanie vráti kód 200 OK a v HTTP body odpovedi budú prítomné polia „heading“, „body“, „imageUrl“, „author“.<br>3. Aplikácia zavolá volanie HTTP PUT /saveChangedArticle/[id] a v tele požiadavky uvednie zmenené pole „heading“ a neuvedie žiadne pole „body“. Ostatné polia ostanú pôvodné.<br>4. Volanie vráti kód 400 Bad Request a článok v databaze ostane pôvodný, bez zmeny |
| **Výsledok:** PASS / FAIL | |