

MILESTONE 2

Backend

Predmet: Mobilné technológie a aplikácie

Kód predmetu: MTAA_B

Cvičiaci: doc. Ing. Marek Galinski, PhD.

Vypracovali: Kanoš Zdenko, Majerčík Tomáš

Študijný odbor: informatika

Študijný program: informatika

Ročník štúdia: druhý

Forma štúdia: denná

Semester: letný

Akademický rok: 2024/25

Wireframes + Prototype:

<https://www.figma.com/design/oHdGSESOHcrSatMFrXhwB2/WIREFRAMES?node-id=0-1&t=qPCZvWVgX1DzYbnD-1>

Popis aplikácie:

Naša aplikácia uľahčuje organizovanie turnajov a ponúka nástroje na ich vyhľadávanie, registráciu a správu. Používateľ môže vytvárať, upravovať a spravovať vlastné turnaje, ako aj zaznamenávať ich výsledky. Zároveň sa môže prihlásiť na existujúce turnaje vytvorené inými používateľmi. Turnaje je možné vyhľadávať podľa kategórií, ako sú napr. tenis, futbal a ďalšie. Po výbere kategórie sa používateľovi zobrazí zoznam dostupných turnajov, na ktoré sa môže používateľ jednoducho prihlásiť.

Use Cases:

Vytvorenie turnaja – aplikácia umožňuje používateľovi vytvoriť turnaj pomocou tlačidla „Create“ v spodnej lište. Po vyplnení požadovaných informácií sa turnaj vytvorí a zobrazí používateľovi v sekcii „Events“ odkiaľ môže turnaj spustiť a upravovať.

Prihlásenie na turnaj – používateľ sa môže prihlásiť na už existujúci turnaj, ktorý bol vytvorený iným používateľom, ~~aplikácia umožní používateľovi pridať aj ďalších účastníkov do tímu.~~ Ostatní používatelia sa budú môcť do tímu pripojiť pomocou unikátneho kódu vygenerovaného pre tím do, ktorého sa môžu pripojiť. Tento kód môžu získať od členov tímu.

Upravenie umiestnení tímov – Organizátor turnaja (ten čo turnaj vytvoril), môže turnaj spustiť, po spustení sa tomuto používateľovi umožní meniť umiestnenia účastníkov a status turnaja (Ongoing, Closed, ~~Ended~~, ~~Suspended~~).

Prehliadanie histórie turnajov – Na hlavnej obrazovke používateľ vidí sekciu história, v tejto sekcii sa zobrazia všetky turnaje, ktorých sa používateľ zúčastnil. Po kliknutí na ikonu „informácie“ vidí používateľ jeho umiestnenie a tabuľku s umiestneniami ostatných tímov.

Lístky pre účastníkov turnaja – Aplikácia generuje vstupenku na turnaj pri každej registrácii v podobe QR kódu. Tento QR kód slúži na overenie registrácie organizátorom turnaja. Lístky sú používateľovi dostupné aj bez internetového pripojenia.

Kontrolovanie lístkov organizátorom turnaja – Organizátor má pri každom turnaji, ktorý vytvoril možnosť kontrolovať vstupenky pomocou čítačky QR kódov, ktorý porovná unikátny string znakov a vyhodnotí či je vstupenka platná.

Vyhľadávanie turnajov podľa názvu – Vyhľadávanie dostupných turnajov prebieha cez kartu explore v aplikácii. Tá zobrazuje najbežnejšie kategórie športov. Daná kategória zobrazuje už

konkrétne športy na ktoré sa dá prihlásiť. Pri každom turnaji je zobrazené za koľko dní začína a ako ďaleko sa turnaj koná od používateľovej polohy.

Odporúčanie turnajov na základe preferencie a polohy používateľa – Po registrácii do aplikácie je používateľ vyzvaný na vyplnenie preferovaného športu a mesta, na základe, ktorého sa mu na hlavnej obrazovke budú odporúčať turnaje, v týchto športoch, ktoré sú k používateľovi najbližšie. Ak používateľ odmietne zdieľať polohu zobrazia sa mu turnaje v blízkosti ním zvoleného mesta.

Upozornenie používateľa pomocou notifikácie – aplikácia upozorní používateľa o nadchádzajúcom turnaji 24 hodín, 1 hodinu a 15 minút pred začatím podujatia pomocou push-notifikácie.

Používateľské role:

V našej aplikácii môže mať používateľ dve rôzne role: organizátor turnaja alebo jeho účastník. Súčasne môže zastávať obe role, avšak vždy v rámci odlišných turnajov.

To znamená, že používateľ:

- Môže organizovať turnaj A a zároveň sa zúčastniť turnaja B
- Nemôže byť organizátor aj účastník toho istého turnaja

Používateľ ako organizátor turnaja:

Po vytvorení turnaja sa používateľovi zobrazí turnaj v sekcii „Events“ pod kategóriou „Hosted Tournaments“, odtiaľto môže používateľ upravovať, spustiť a odstrániť turnaj. Zároveň po spustení turnaja má možnosť nastaviť umiestenia tímov a status turnaja. Tieto operácie vie vykonať len používateľ, ktorý turnaj vytvoril.

Používateľ ako prihlásený účastník turnaja:

Používateľ si môže vyhľadať turnaj v sekcii „Explore“ a prihlásiť sa naň. Po prihlásení na takýto turnaj bude používateľ disponovať vstupenkou vo forme QR kódu, ktorá môže byť skontrolovaná používateľom v roli *organizátora turnaja*.

Platforma a technológie:

Platforma – *Cross (iOS/Android)*

Frontend – *ReactNative s využitím expo*

Backend – *Node.js (Express.js)*

Databáza – *PostgreSQL*

API:

- Linky na API sa zobrazia po kliknutí na názov

Systémové:

- [Geolokácia](#) – poloha zariadenia
- [Akcelometer](#) – zatrasenie pre obnovenie vyhľadávania
- [Kamera](#) – skenovanie QR kódov
- [Úložisko](#) – lokálne uloženie informácií o prihlásených turnajoch a vstupenke
- [Systémové súbory](#) – nahranie profilovej fotografie
- [Notifikácie](#) – notifikácia používateľa o nastávajúcim turnaji
- [Internetové pripojenie](#) – zisťovanie či je zariadenie pripojené k internetu
- [Povolenia aplikácie](#) – správa oprávnení pre kameru a polohu

Externé:

- [Mapy a poloha](#) – Google Maps API – na zistenie longitude a latitude, zobrazenie ukážky mapy pri turnaji
- [Generovanie QR kódu](#) – pomocou jednoduchého volania vygeneruje QR kód

Technologický stack:

React Native + Expo / Node.js ([Express.js](#)) / PostgreSQL stack

Povinne-voliteľné súčasti:

- Podpora accessibility – umožnenie používateľovi nastaviť aplikáciu do čierneho-bieleho režimu
- GPS – využitie GPS pre určenie vzdialenosti používateľa od eventu
- Integrácia akcelometra – používateľ bude môcť obnoviť stránku vyhľadávaných turnajov pomocou zatrasenia zariadenia

UATy:

Pozitívne:

Test 1: Registrácia nového používateľa a nastavenie preferencií	
Vstupné podmienky:	Používateľ nie je prihlásený a nie je registrovaný email s jeho adresou
Výstupné podmienky:	Nový účet je úspešne vytvorený a používateľ je automaticky prihlásený
Postup:	<ol style="list-style-type: none">1. Otvorí sa aplikácia2. Používateľ klikne na tlačidlo "SIGN UP"3. Vyplní email, heslo a potvrdí heslo4. Klikne na tlačidlo "REGISTER"5. Vyplní preferované športy a mesto6. Zobrazí sa hlásenie o úspešnej registrácii7. Používateľ je automaticky prihlásený a presmerovaný na hlavnú stránku
Výsledok: PASS / FAIL	

Test 2: Pripojenie sa k turnaju	
Vstupné podmienky:	Používateľ je prihlásený do aplikácie
Výstupné podmienky:	Používateľ je prihlásený na turnaj
Postup:	<ol style="list-style-type: none">1. Používateľ klikne na tlačidlo "explore" v spodnej lište2. Zvolí kategóriu športu na ktorý sa chce prihlásiť3. Vyhľadá konkrétny turnaj pomocou vyhľadávania podľa názvu4. Stlačí tlačidlo na zobrazenie informácii o turnaji5. Pridá seba a ďalšieho člena tímu a potvrdí prihlásenie6. V karte "events" vidí turnaj na ktorý sa práve registroval
Výsledok: PASS / FAIL	

Test 3: Zobrazenie histórie turnaja	
Vstupné podmienky:	Používateľ je prihlásený a zúčastnil sa turnaja, ktorý je už ukončený a má pridané umiestnenia
Výstupné podmienky:	Používateľ si vie pozrieť umiestnenie a informácie o turnaji, ktorého sa zúčastnil
Postup:	<ol style="list-style-type: none">1. Používateľ na hlavnej obrazovke klikne v sekcii histórie turnajov na tlačidlo "VIEW ALL"2. Zobrazia sa všetky turnaje, ktorých sa používateľ zúčastnil a boli ukončené3. Používateľ vidí názov turnajov aj s pozíciou na ktorej sa umiestnil4. Klikne na ikonu so symbolom "i"5. Zobrazia sa podrobnosti o turnaji
Výsledok: PASS / FAIL	

Test 4: Vytvorenie turnaja	
Vstupné podmienky:	Používateľ je prihlásený
Výstupné podmienky:	Turnaj je úspešne vytvorený a zobrazí sa v sekcii events
Postup:	<ol style="list-style-type: none"> 1. Používateľ klikne na tlačidlo “Create“ na spodnej lište 2. Vyplní názov turnaja, dátum, typ hry, lokalitu a ďalšie potrebné informácie 3. Klikne na tlačidlo “SUBMIT” 4. Zobrazí sa potvrdenie o úspešnom vytvorení turnaja 5. Používateľ je presmerovaný na obrazovku turnaja 6. Vidí turnaj v zozname vytvorených turnajov
Výsledok: PASS / FAIL	

Test 5: Offline režim	
Vstupné podmienky:	Používateľ je prihlásený a zaregistrovaný na turnaj, ale nemá prístup k internetu
Výstupné podmienky:	Používateľ vidí svoju vstupenku aj bez internetu
Postup:	<ol style="list-style-type: none"> 1. Používateľ otvorí aplikáciu bez internetu 2. Vidí v hlavnej obrazovke sekciu s lístkami 3. Kliknutím vyberie konkrétny lístok 4. Zobrazí sa jeho vstupenka so základnými informáciami a QR kódom 5. QR kód je možné zväčšiť kliknutím naň 6. Po opätovnom pripojení má prístup ku všetkým funkciám bez zatvorenia aplikácie
Výsledok: PASS / FAIL	

Test 6: Zaznamenanie umiestnenia	
Vstupné podmienky:	Turnaj je vytvorený
Výstupné podmienky:	V turnaji budú upravené umiestnenia
Postup:	<ol style="list-style-type: none"> 1. Používateľ klikne tlačidlo “START” a spustí turnaj ak už nie je spustený 2. Používateľ klikne na “EDIT STATISTICS” a otvorí úpravu umiestnení 3. Klikne na tlačidlo pridať pri umiestnení 4. Vyhľadá a vyberie tím podľa názvu 5. Tím sa pridá do daného umiestnenia 6. Po stlačení tlačidla “SAVE“ sa umiestnenia uložia a zobrazia aj ostatným používateľom v histórii po ukončení turnaja
Výsledok: PASS / FAIL	

Test 7: Notifikácia o nastávajúcom turnaji	
Vstupné podmienky:	Používateľ je prihlásený do aplikácie
Výstupné podmienky:	Používateľovi sa zobrazí upozornenie v podobe notifikácie
Postup:	<ol style="list-style-type: none"> 1. Používateľ sa registruje na turnaj 2. Systém to zaznamená a pridá používateľa do zoznamu upozornení 3. 1 hodinu pred začiatkom turnaja, backend generuje notifikáciu na nasledujúcu udalosť 4. Používateľ dostane push notifikáciu na svojom zariadení 5. Po kliknutí na notifikáciu sa otvorí aplikácia
Výsledok: PASS / FAIL	

Negatívne:

Test 8: Pripojenie sa druhý krát na ten istý turnaj	
Vstupné podmienky:	Používateľ je prihlásený na daný turnaj
Výstupné podmienky:	Používateľ dostane chybové hlásenie.
Postup:	<ol style="list-style-type: none"> 1. Používateľ vyhľadá turnaj 2. Iný používateľ sa ho snaží prihlásiť do svojho tímu 3. Klikne registrovať 4. Zobrazí sa chybové hlásenie, že daný používateľ sa už na tento turnaj registroval 5. Systém nedovolí aby bola osoba prihlásená 2 krát
Výsledok: PASS / FAIL	

Test 9: Pokus o pripojenie sa do plného turnaja	
Vstupné podmienky:	Turnaj má nastavenú kapacitu, kapacita je plná
Výstupné podmienky:	Systém nedovolí používateľovi sa prihlásiť
Postup:	<ol style="list-style-type: none"> 1. Používateľ vyhľadá turnaj 2. Stlačí tlačidlo na zobrazenie informácií o turnaji 3. Vyplní údaje a potvrdí prihlásenie 4. Systém zistí, že turnaj je plný a nedovolí používateľovi sa pridať 5. Vypíše chybovú hlášku, že turnaj je plný
Výsledok: PASS / FAIL	

Test 10: Pridanie turnaja s názvom, ktorý už existuje	
Vstupné podmienky:	Používateľ má už aspoň jeden vytvorený turnaj a vytvára ďalší
Výstupné podmienky:	Používateľ dostane chybové hlásenie
Postup:	<ol style="list-style-type: none"> 1. Používateľ začne vytvárať nový turnaj 2. Zadá rovnaký názov turnaja, taký aký už má iný turnaj 3. Vyskočí chybová hláška 4. Používateľ má možnosť zmeniť meno 5. Po zmene názvu, môže byť turnaj pridaný.
Výsledok: PASS / FAIL	

Backend – Milestone 2:

GitHub: <https://github.com/zdenkokanos/MTAA-backend>

Swagger, priložený v ZIP súbore ako HTML.

Použité technológie:

Programovací jazyk a framework

Backend aplikácie je implementovaný v JavaScripte, s využitím frameworku Node.js a knižnice Express.js pre tvorbu REST API.

Databáza

Na perzistenciu dát je použitá relačná databáza PostgreSQL, ktorá beží lokálne na laptope. Aplikácia komunikuje s databázou priamo prostredníctvom knižnice pg bez použitia ORM.

Bezpečnosť

Autentifikácia a autorizácia sú riešené pomocou JWT (JSON Web Token). Prihlasovacie údaje sú overené a následne je používateľovi vygenerovaný JWT token, ktorý je potrebný pre prístup k chráneným endpointom.

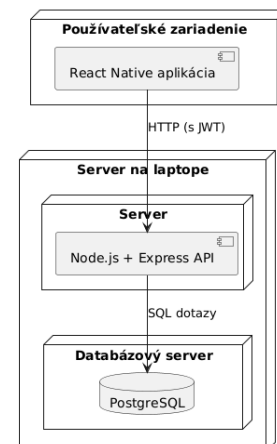
Deployment a architektúra

V aktuálnej fáze vývoja beží celá aplikácia lokálne:

- Backend (Node.js + Express) beží na `http://localhost:3000`
- Databáza (PostgreSQL) beží na `localhost:5432`
- Frontend je vo vývoji v React Native, ktorý bude komunikovať priamo s backendom pomocou protokolu HTTP.

Komunikácia medzi komponentmi prebieha nasledovne:

- Frontend posíla požiadavky na REST API.
- Backend spracuje požiadavky, prípadne pristupuje k databáze a vracia odpoveď.
- Pre chránené operácie je vyžadovaný JWT token.



Databázový návrh – diagram

