STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

ALBI hry

Daniel Kopecký

Pardubický kraj

Pardubice, 2023

STŘEDOŠKOLSKÁ ODBORNÁ ČINNOST

Obor č. 18: Informatika

ALBI hry

ALBI games

Autor: Daniel Kopecký
Škola: DELTA - Střední škola informatiky a ekonomie, s.r.o., Ke Kamenci 151, 530 03 Pardubice
Kraj: Pardubický kraj
Konzultant: RNDr. Jan Koupil, Ph.D.

Pardubice, 2023

Prohlášení

Prohlašuji, že jsem svou práci SOČ vypracoval samostatně a použil jsem pouze prameny a literaturu uvedené v seznamu bibliografických záznamů.

Nemám závažný důvod proti zpřístupňování této práce v souladu se zákonem č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších předpisů.

V Pardubicích dne 30. 3. 2023 .....

Daniel Kopecký

Poděkování

Chtěl bych poděkovat mému učiteli RNDr. Janu Koupilovi, Ph.D. za vedení projektu a společnosti ALBI Česká republika a.s. za možnost spolupráce na tomto projektu.

Abstrakt

Práce SOČ dokumentuje návrh a vývoj mobilní aplikace zaměřené na reklamní marketing společnosti ALBI Česká republika a.s. a propojení fanoušků deskových her. Realizace projektu zahrnuje backendovou část a mobilní aplikaci.

Klíčová slova

Programování; Flutter; Laravel; mobilní aplikace; deskové hry

Abstract

The thesis documents the design and development of a mobile application focused on the promotional marketing of ALBI Czech Republic a.s. and connecting board game fans. The implementation of the project includes a backend part and a mobile app.

Keywords

Programming; Flutter; Laravel; mobile app; board games

Zadání maturitního projektu z informatických předmětů

Jméno a příjmení:	*Daniel Jaroslav Kopecký*
Pro školní rok:	*2022/2023*
Třída:	*4.*
Obor:	*Informační technologie 18-20-M/01*
Téma práce:	*Online systém achievementů pro firmu Albi*
Vedoucí práce:	*RNDr. Jan Koupil, Ph.D.*

Způsob zpracování, cíle práce, pokyny k obsahu a rozsahu práce:

Cílem práce je navrhnout a implementovat online prostředí pro správu vlastněných deskových her, dosažených achievementů v nich a na nich založený promo-marketing firmy Albi. Na přesné specifikaci a funkcionalitě bude student průběžně spolupracovat se zadavatelem.

Stručný časový harmonogram (s daty a konkretizovanými úkoly):

- *Září*: rešerše – Analýza potřeb zadavatele, průzkum existujících řešení, volba technologií
- *Říjen*: Návrh datového modelu, jeho fyzická realizace
- *Listopad*: Návrh uživatelského rozhraní
- *Prosinec-Leden*: Implementace funkcionality
- *Únor*: Testování a ladění
- *Březen*: Dokumentace maturitního projektu

Obsah

1 Úvod

Nápad na tento projekt vznikl ve spolupráci se společností ALBI Česká republika a.s. Cílem projektu je vytvořit mobilní aplikaci, která má za úkol propojit komunitu fanoušků a hráčů deskových her od Albi.

Aby aplikace získala nové uživatele, je třeba navrhnout systém úspěchů a odměn k deskovým hrám. Vlastněné hry si uživatelé budou zakládat do knihovny s použitím čárového kódu a k nim přiřazovat dosažené úspěchy. Za ně pak může ALBI odměnit hráče i slevou na nákup dalších her.

1.1 Existující řešení

V tuto chvíli, v České republice, žádná podobná aplikace neexistuje. V zahraničí je známá podobná aplikace, která odměňuje uživatele za vykonanou činnost, ale v ní se jedná o sportovní aktivity. Pro komunitu hráčů deskových her tedy takový aplikace není. Inspiroval jsem se pouze platformou Steam [1], která obsahuje úspěchy, nikoliv deskových, ale digitálních her.

2 Teoretická část

2.1 Pojmy

2.1.1 Kontejner

Kontejner je standardizovaná instalace celého operačního systému, ve které je zabalený kód a všechny jeho potřebné závislosti, aby bylo možné aplikaci rychle spouštět a aby byla aplikace konzistentní mezi různými prostředími. [2]

2.1.2 ORM

ORM neboli Object-Relational Mapping (Mapování objektů na relační databáze) je to technologie pro usnadnění práce s relačními databázemi v objektově orientovaných programovacích jazycích. ORM umožňuje převést data v databázi do objektů v programu a naopak, což umožňuje snadnější manipulaci s daty a snižuje potřebu psát složité dotazy na databázi. [3]

2.1.3 Hash

V oblasti počítačového zpracování a kryptografie se jako hash označuje matematická funkce, která bere vstupní data libovolné délky (např. heslo) a produkuje z nich pevnější výstup, který má obvykle pevnou délku. Výstupem hashovací funkce je tzv. hash, který může být použit jako krátká reprezentace původních dat, ale už se nikdo nemůže podívat na původní hodnotu dat. [4]

2.2 Použité technologie

2.2.1 Pro backend

2.2.1.1 Laravel

Laravel je open source framework pro vývoj webových i serverových aplikací za použití PHP. Je velmi populární díky své jednoduchosti, flexibilitě a širokému spektru nástrojů pro vývojáře. Poskytuje mnoho užitečných funkcí pro práci s databázemi, správu uživatelů a autorizaci, integraci s různými službami, práci s emaily a mnoho dalšího. [5]

2.2.1.2 MySQL

MySQL je relační databázový systém, který umožňuje ukládání, správu a manipulaci s daty. Je to open source software, který je široce používán v různých aplikacích a webových stránkách, pro ukládání a získávání dat. [6]

2.2.1.3 Docker

Docker je platforma pro vytváření, správu a běh kontejnerů. Pomocí takzvaných „Dockerfile“ je možné definovat, jak bude výsledný kontejner sestaven a jak bude fungovat. Lze pomocí něj také spouštět samotné kontejnery, sestavené ať už námi, nebo komunitou. Díky němu lze spustit v podstatě jakoukoliv aplikaci pomocí jednoho příkazu. [7]

2.2.2 Pro frontend

2.2.2.1 Flutter

Flutter je open source framework pro vývoj aplikací pro mobilní zařízení s operačním systémem Android i iOS. Díky svému jednoduchému a intuitivnímu designu umožňuje rychle vyvíjet aplikace s atraktivním vzhledem a příjemným uživatelským rozhraním. Navíc Flutter umožňuje vyvíjet aplikace pro obě platformy najednou (i pro web), což ušetří čas i úsilí. [8]

2.2.3 Principy návrhů mobilních aplikací

Model-View-Controller (MVC) je architektonický vzor, který se používá při vývoji softwaru, včetně mobilních aplikací. MVC odděluje aplikaci na tři základní části: model, pohled a kontrolér.

Model představuje datovou část aplikace, která uchovává a spravuje informace.

Pohled je to, co uživatel vidí na obrazovce, jako jsou formuláře, texty a grafika.

Kontrolér představuje logiku aplikace, která řídí interakci mezi modelem a pohledem. Kontrolér zodpovídá za zpracování uživatelských akcí, jako je kliknutí na tlačítko a volání příslušných funkcí modelu.

Výhodou MVC je, že oddělení těchto tří částí umožňuje vývojářům lépe organizovat a upravovat kód, a také usnadňuje opravy či úpravy bez nutnosti zásahu do celého kódu. To v konečném důsledku přispívá ke snadnějšímu a rychlejšímu vývoji kvalitních aplikací. [9]

3 Praktická část

Aplikace má za cíl propojit komunitu hráčů deskových her. Mimo toho si uživatelé mohou přidávat deskové hry do své knihovny her naskenováním čárového kódu na zadní straně krabice hry. Aplikace obsahuje gamifikační prvky: uživatelé plní úspěchy v jednotlivých hrách a za průběžné plnění úspěchu mohou získat slevové kupóny na nákup dalších her od ALBI.

Komunitní prvek aplikace je vytvořena menší sociální síť, kde si uživatel může přidat přátelé pomocí přezdívek a soutěžit o to, kdo má více úspěchů v deskových hrách. Dále si uživatel může založit herní klub, nebo se do nějakého připojit a sledovat nastávající herní akce klubu, a domlouvat se na herních schůzkách pomocí klubového chatu.

3.1 Realizace

3.1.1 Architektura

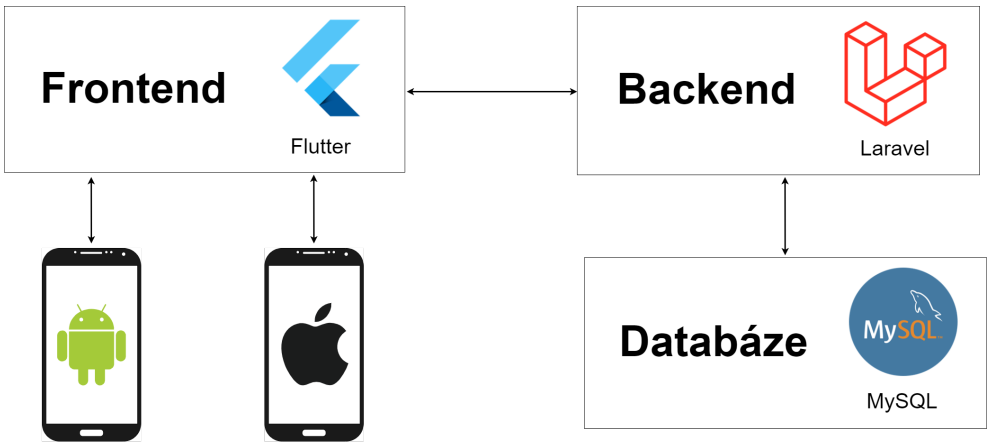
Cílem této práce je vytvořit komplexní aplikaci, která využívá mobilních zařízení s operačním systémem Android a iOS. Aplikace na mobilních zařízeních slouží pro interpretaci a přijímání vstupních dat uživatele. Na serverové straně se bude s těmito daty dále pracovat a ukládat do databáze.

Z tohoto rozdělení úkolů vychází základní nároky na architekturu systému skládající se ze dvou částí. Frontend jako mobilní aplikace sloužící pro uživatelské rozhraní, která úzce spolupracuje se serverovým backendem. Ten bude zastávat většinu funkcionality, jako práci se vstupními daty a jejich uložení do databáze, autorizaci a posílání dat zpět na frontend.

Pro frontend jsem vybral framework Flutter, který umožňuje vyvíjet aplikace pro různé platformy s jedním kódem. Pro backend je použitý framework Laravel, který pracuje s MySQL databází. Laravel je na serverové straně a zajišťuje veškerou logiku aplikace, jako je správa dat, autorizace a ukládání dat do databáze. Komunikace mezi frontendem a backendem probíhá pomocí API endpointů, které jsou vytvořeny na straně backendu. Architekturu názorně shr-

nuje schéma na obr. 3.1.

Obrázek 3.1: Architektura aplikace.



Tato architektura typu klient-server umožňuje oddělení logiky aplikace od uživatelského rozhraní, což usnadňuje údržbu a vývoj aplikace. Backend může být aktualizován a měnit svou funkcionalitu, aniž by se to negativně odrazilo na uživatelském rozhraní. V případě, že bude potřeba rozšířit aplikaci o další funkcionality, může tak být snadno provedeno přidáním nových funkcí do backendu a poskytnutím odpovídajícího API, které bude poskytovat data a informace novým funkcím.

Kromě toho, tato architektura umožňuje snadné škálování aplikace. Backend a databáze mohou být umístěny na různých serverech nebo dokonce v cloudových prostředích, což umožňuje rychlé a efektivní škálování aplikace v závislosti na požadavcích uživatelů.

3.1.1.1 Zabezpečení

Důležitou součástí této architektury je také bezpečnost. Laravel poskytuje výkonné nástroje pro autentizaci a autorizaci uživatelů, což zajišťuje, že pouze oprávnění uživatelé mají přístup k chráněným informacím. Backend obsahuje další bezpečnostní prvky, například ochranu proti útokům typu SQL injection.

Aplikace dbá na bezpečnost a soukromí uživatelů. Používá token pro zabezpečení komunikace mezi uživatelem a backendem. Token se přiřadí uživateli po přihlášení nebo registraci a slouží jako unikátní identifikátor, který umožňuje rozpoznat uživatele a udržet jeho data v bezpečí.

Kromě toho aplikace používá funkci hash pro ochranu citlivých informací uživatelů, jako jsou například přihlašovací údaje. Díky těmto opatřením jsou uživatelská data v aplikaci v bezpečí.

3.1.2 Databáze

Základní kámen aplikace je její databáze, zde se ukládají kolekce dat do tabulek, se kterými následně pracuje backend. Schéma databáze tohoto projektu je komplexnější viz obr. 3.3.

Pro tento projekt jsem vybral databázi MySQL, která je jednoduchá, rychlá a vhodná pro malé či velké systémy. Laravel s touto databází dokáže snadno pracovat a má připravené funkce pro snadnou správu samotné databáze.

Každá tabulka má vlastní strukturu, kde se určuje, co bude jednotlivý záznam v tabulce obsahovat. Příklad základní struktury můžeme vidět na první nejdůležitější tabulce users na obr. 3.2.

Obrázek 3.2: tabulka users.

Sloupec	Typ
id	bigint(20) unsigned *Auto Increment*
first_name	varchar(220)
last_name	varchar(220)
nickname	varchar(220)
email	varchar(100)
email_verified_at	timestamp *NULL*
password	varchar(220)
created_at	timestamp *NULL*
updated_at	timestamp *NULL*

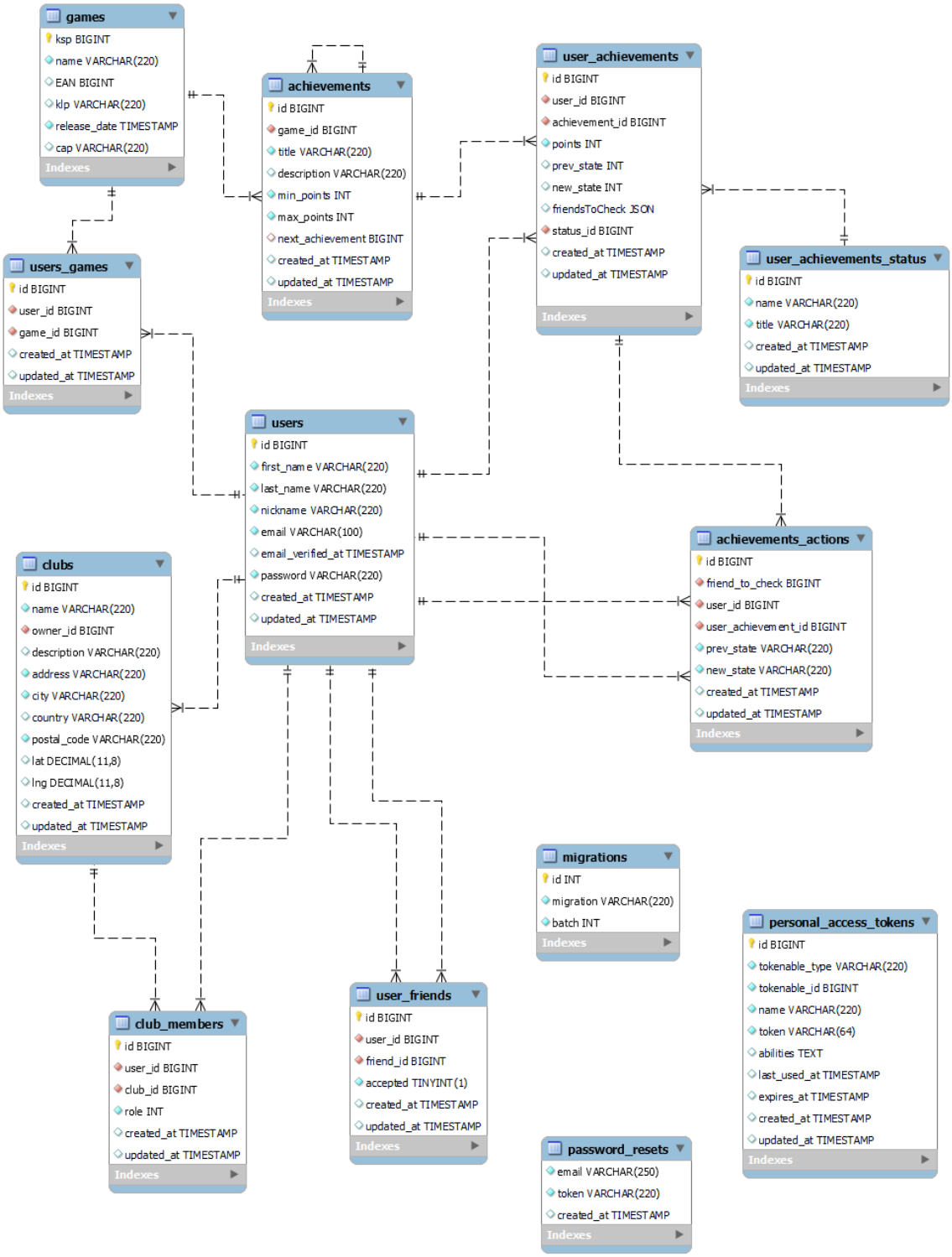
Indexy

PRIMARY	*id*
UNIQUE	*email*

Každá tabulka obsahuje sloupec id, `created_at` a `updated_at`. Sloupec id je primární klíč záznamu, je tvořen automaticky databází a vždy je to unikátní číslo. Sloupce `created_at` a `updated_at` jsou takzvané timestampy, neboli časová razítka, kde `created_at` obsahuje datum vytvoření záznamu a `updated_at` obsahuje datum poslední změny záznamu.

Dále tabulka „\users obsahuje základní informace o uživateli, jako je jméno, příjmení, přezdívka a email. Email je v této tabulce unikátní, z toho vyplývá, že dva uživatelé nemohou mít stejný

Obrázek 3.3: Schéma databáze.



email. Díky tomu v tuto chvíli email, kromě sloupce id, slouží jako identifikátor uživatele. Heslo uživatele se ukládá do tabulky také, ale je zahashované, takže nikdo ani administrátor databáze nemůže heslo uživatele přečíst.

3.1.2.1 Hlavní tabulky

Další hlavní tabulky, které udržují nejdůležitější data aplikace, jsou tabulky klubů, deskových her a úspěchů. Tabulka klubů obsahuje data o jednotlivých deskových hrách, které projektu poskytla společnost Albi. Proto tato tabulka jako jediná neobsahuje základní sloupce id a časová razítka. Strukturu této tabulky zobrazuje obr. 3.4. Zde je primární klíč nahrazen sloupcem ksp. Dále tabulka obsahuje důležitý sloupec EAN, který obsahuje čárový kód deskové hry a samozřejmě sloupec name s názvem deskové hry.

Obrázek 3.4: Tabulka deskových her.

Column	Type
ksp	int
name	varchar(220)
EAN	bigint *NULL*
klp	varchar(220) *NULL*
release_date	timestamp
cap	varchar(220) *NULL*

Indexes

PRIMARY	*ksp*

Tabulka klubů se strukturou znázorněnou na obr. 3.5 obsahuje informace o vytvořeném klubu. V této tabulce jsou nejdůležitější data o názvu, adrese a přesných souřadnicích klubu, kde sloupec `lat` obsahuje zeměpisnou šířku a sloupec `lng` zeměpisnou délku. V této tabulce máme první referenci na jinou tabulku, a to konkrétně sloupec `owner_id`. Ten drží cizí klíč jiného záznamu z tabulky „\users.

Tabulka úspěchů obsahuje informace o úspěších v daných deskových hrách. Zde jsou sloupce o názvu, popisu, maximálních a minimálních možných bodů úspěchu. Dále tabulka obsahuje dvě reference na jiné záznamy. Sloupec `game_id` drží cizí klíč záznamu z tabulky klubů a sloupec `next_achievement`, který drží cizí klíč jiného záznamu z tabulky úspěchů. Strukturu této

Obrázek 3.5: Tabulka klubů.

Column	Type
id	bigint unsigned *Auto Increment*
name	varchar(220)
owner_id	bigint unsigned
description	varchar(220) *NULL*
address	varchar(220)
city	varchar(220)
country	varchar(220) *NULL*
postal_code	varchar(220)
lat	decimal(11,8) *NULL*
lng	decimal(11,8) *NULL*
created_at	timestamp *NULL*
updated_at	timestamp *NULL*

Indexes

PRIMARY	*id*
INDEX	*owner_id*

tabulky můžeme vidět na obr. 3.6.

Obrázek 3.6: Tabulka úspěchů.

Column	Type
id	bigint unsigned *Auto Increment*
game_id	bigint unsigned
title	varchar(220)
description	varchar(220) *NULL*
min_points	int [**0**]
max_points	int
next_achievement	bigint unsigned *NULL*
created_at	timestamp *NULL*
updated_at	timestamp *NULL*

Indexes

PRIMARY	*id*
INDEX	*game_id*
INDEX	*next_achievement*

3.1.3 Backend

Srdce aplikace je její backend vytvořený ve frameworku Laravel. Ten se stará o veškerou logiku aplikace, správu dat a jejich ukládání do databáze a obsluhuje dotazy z frontendu pomocí vytvořených API endpointů. Navíc Laravel používá technologii ORM, což usnadňuje práci s databází. Po dotazu na určitý API endpoint se požadavek předá kontroléru, kde se nejprve zvalidují data, která jsou potřeba k provedení požadavku. Pokud jsou data v pořádku, přesune se požadavek z kontroléru do příslušné servisy. Zde se probíhá logika aplikace a vyhledávají se výsledná požadovaná data z databáze. V případě, že v datech dochází k chybě, pošle se chybová zpráva zpět na frontend, aby uživatel mohl opravit své zadání. Pokud jsou data v pořádku, backend vytvoří odpověď v požadovaném formátu JSON a odešle ji zpět na frontend.

3.1.3.1 Autorizace

Pro přistup k většině dotazů je potřeba být přihlášen. Uživatelé se proto musí zaregistrovat, nebo přihlásit do aplikace, pokud již mají zaregistrovaný účet. Na backendu toto řeší Laravel. Má připravené funkce jak pro pro autentizaci tak i pro autorizaci. Při registraci jsou uživateli zkontrolována vstupní data, je vytvořen záznam v tabulce „`users`“ s hodnotami: „email“, „first_name“, „last_name“ a „password“ (heslo je zaheshováno pomocí bcrypt[10]).

Po vytvoření záznamu je vygenerován token který se následně vrátí uživateli a tím se dále tento uživatel autentizuje pro další dotazy. K přístupu dotazům na backend ke kterým je potřeba být přihlášen se už tedy neposílají přihlašovací údaje, ale pouze token, diký kterému backend pozná o kterého uživatele se jedná. Token je potřeba přidat do vlastností hlavičky dotazu s názvem „authorization“ a hodnotou „Bearer TOKEN“.

Pokud se chce uživatel odhlásit, backend smaže aktuální token uživatele aby se s ním dále nešlo autentizovat.

3.1.3.2 Správa přátel

Přidání přítele probíhá pomocí přezdívky, tu obdrží backend v dotaze a vyhledává uživatele s danou přezdívkou v tabulce „`users`“. Po vyhledání přítele se vytvoří záznam v tabulce „`user_friends`“. Nový záznam obsahuje sloupec `user_id` s referencí na uživatele který poslal dotaz, sloupec `friend_id` s referencí na přítele. Sloupec `accepted` indikuje stav přijatého přátelství a při vytvoření záznamu je tato hodnota nastavena na nulu. Poté backend na frontend pošle zprávu o úspěšné vytvoření žádosti o přátelství.

Potvrzení žádosti o přátelství proběhne přes příslušný dotaz na backend a pouze se nastaví hodnota ve sloupci `accepted` ve vytvořeném záznamu výše na číslo jedna tzn. že žádost je přijata a tito dva uživatelé jsou přátelé. Se zamítnutím žádosti se pouze tento záznam smaže.

3.1.3.3 Správa her a úspěchů

Pro přidání hry do knihovny uživatel pošle na backend dotaz společně s hodnotou skenu čárového kódu. Backend prohledá v databázi sloupec „`EAN`“ v tabulce her, pokud hru pomocí naskenovaného kódu najde, vytvoří záznam v tabulce „`user_games`“, kde spojí hru s uživatelem. Poté prochází úspěchy hry a vytváří hráčské úspěchy v tabulce „`user_achievements`“ s výchozí hodnotou bodů plněného úspěchu nula. Po dokončení tohoto procesu backend vrátí zpět na frontend naskenovanou hru společně s hráčskými úspěchy dané hry. Pokud hru podle naskenovaného kódu nenajde, vrátí frontendu zprávu o chybě, že hra nebyla nalezena.

Plnění úspěchu probíhá posláním dotazu s novým počtem bodů plněného úspěchu a polem s id přátelů po kterých uživatel chce potvrzení změny jeho úspěchu. Backend nastaví novou hodnotu bodů hráčského úspěchu. S každým id přítele z obdrženého pole se vytvoří záznam o akci úspěchu uživatele v tabulce „`achievements_actions`“. Zde se přiřadí hodnota předchozího stavu bodů úspěchu do sloupce `prev_state` a nového stavu bodů do sloupce `new_state` a do sloupce `frient_to_check` se přiřadí reference na id přítele. Na frontend se vrátí odpověď o úspěšné změně úspěchu.

3.1.3.4 Správa herních klubů

3.1.3.4.1 Vytvoření klubu

Uživatel pošle dotaz na backend s informacemi o klubu. Tyto informace se nejprve zvalidují a poté se vytvoří záznam v tabulce „`clubs`“. Zde se doplní data do sloupců `name` - název klubu, `description` - popis klubu, `lat` a `lng` - zeměpisná šířka a délka, adresa klubu a do sloupce `owner_id` se doplní reference na uživatele který klub vytvořil. Po dokončení se zpět pošle zpráva o úspěšném vytvoření a data klubu.

3.1.3.4.2 Vyhledávání nejbližších klubů

Vyhledávání nejbližších klubů podle GPS souřadnic uživatele backend obdrží v dotaze společně s poloměrem rozsahu vyhledávání a limitem počtu klubů, podle limitu se určuje rozmezí výsledného počtu vyhledávaných klubu. Tyto data se zvalidují a poté se pro vyhledání nejbližších klubů z tabulky „`clubs`“ v okolí polohy hráče použije matematická rovnice Haversinova formule. Ta slouží k výpočtu vzdálenosti mezi dvěma body na geometrickém útvaru kouli, například na Zemi [11]. Pokud tedy při procházení klubů v tabulce je vzdálenost klubu od polohy hráče menší než požadovaný rozsah, tento klub se přidá do výsledku vyhledávání. Po dokončeni procesu se výsledné pole nejbližších klubů pošle zpět na frontend.

3.1.3.4.3 Správa členů klubu

V této sekci se řeší přidávání, přijímání, odmítnutí, připojování a povyšování hráčů do konkrétním klubu. Uživatel, pokud se jedná o zakladatele klubu, může přidávat jiné hráče do svého klubu. Na backend pošle dotaz s polem id přátel které chce přidat do svého klubu. S každým id se vytvoří záznam v tabulce „`club_members`“, kde se doplní reference na přidaného uživatele do sloupce `user_id` a do sloupce `club_id` reference na klub do kterého je uživatel přidán. A poslední údaj v záznamu je sloupec `role` který obsahuje číslo které indikuje roli uživatele v daném klubu. S vytvořením tohoto záznamu zakladatelem klubu je role přidaného uživatele nastavena na číslo jedna, tedy se jedná již o přijatého člena klubu. Dále zakladatel klubu může odebírat hráče z klubu, při této akci se pouze smaže vytvořený záznam. Také může člena klubu povýšit na tzv. admina klubu. V této funkci se pouze změní v záznamu hodnota sloupce `role` na číslo dva.

Uživatel který není zakladatel klubu se může do klubu připojit. Na backend pošle dotaz s informací o který klub se jedná a zde se vytvoří stejný záznam v tabulce „`club_members`“ s jediným rozdílem, a to ve sloupci `role`, kde se nastaví hodnota na číslo nula, což je indikátor nepřijatého člena klubu. Toho může přijmout zakladatel nebo admin klubu se změnou hodnoty sloupce `role` na číslo jedna.

3.1.3.5 Nasazení na server

Nasazení na server je klíčový krok, který je potřeba pro zpřístupnění a běhu backendu. Pro vytvoření vždy identického prostředí je potřeba vytvořit kontejner, který obsahuje pouze data potřebná k běhu služby

Pro vytvoření kontejnerů je použit Docker. Pomocí takzvaného „Dockerfile“ je zadefinován postup jak backend sestavit. Po následném sestavení, dostaneme balík, který je možné kdekoliv spustit

K následnému nasazení na server je použita platform Kubernetes, konkrétně K3s, která se stará o kontejnery, zajišťuje síťovou komunikaci a rozprostívá zátěž mezi servery.

3.1.4 Frontend

Frontend, vytvořený ve frameworku Flutter, zajišťuje běh aplikace na mobilních zařízení s operačním systémem Android a iOS. Tato mobilní aplikace zastává funkci interpretaci dat a zpracovávání vstupních dat uživatele.

Mobilní aplikace je rozdělena do čtyř hlavních sekcí viz spodní navigační panel na obr. 3.7, každá sekce zastává určitou funkcionalitu a má svůj tzv. „action button“ tlačítko, které slouží pro hlavní akci určité sekce.

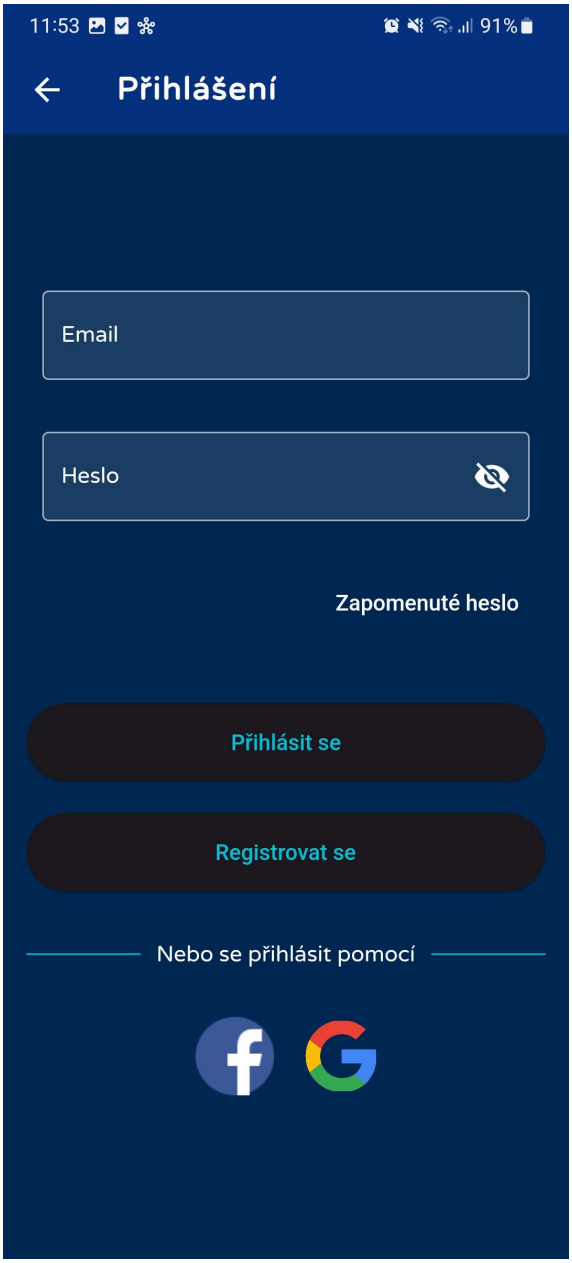
3.1.4.1 Přihlášení a registrace

Pro používání všech sekcí aplikace musí být uživatel přihlášen. Na přihlašovací obrazovku se uživatel dostane přes ikonku přihlášení v horní části aplikace viz obr. 3.7. Zde se uživatel může přihlásit do aplikace pomocí přihlašovacích údajů viz obr. 3.8, nebo pokud není v aplikaci registrovaný, může provést registraci přes tlačítko „Registrovat se“. Tím se přesune na registrační formulář viz obr. 3.9 a po vyplnění osobních údajů se data v aplikaci zkontrolují, např. jestli je email vyplněný ve správném formátu. Poté se tyto data pošlou ke zpracování na backend a při úspěšné registraci aplikace automaticky uživatele přihlásí.

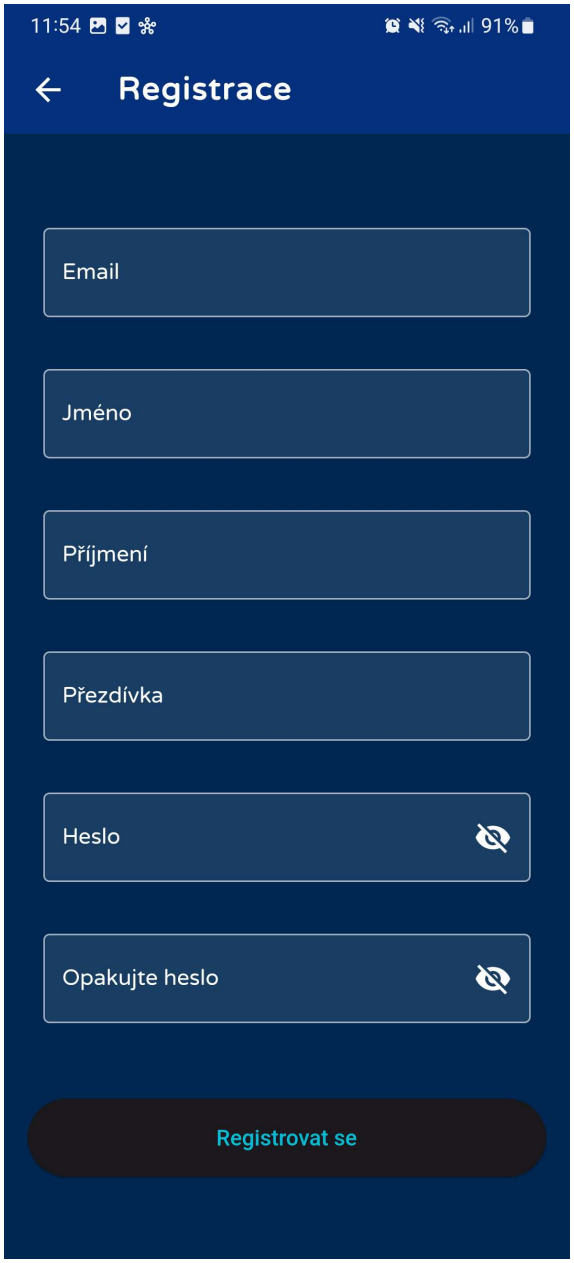
Při úspěšném přihlášení nebo registraci, aplikace obdrží z backendu autentizační token uživatele, který si uloží do místního úložiště mobilního telefonu zvaný „SharedPreferences“ [12]. Následné se tento token, vždy při posílání dotazu na backend na který je potřeba být přihlášen, přidá do hlavičky dotazu.

Obrázek 3.7: Hlavní obrazovka.

Obrázek 3.8: Přihlašovací obrazovka.



Obrázek 3.9: Registrační obrazovka.



3.1.4.2 Knihovna her

První sekce aplikace je knihovna deskových her viz obr. 3.11. V této sekci má uživatel všechny své přidané deskové hry. Každá karta hry obsahuje obrázek (v tuto chvíli je obrázek všude stejný a slouží jen pro představu), název hry a procentuální plnění daných úspěchu ve hře.

V této sekci slouží „action button“ pro naskenování nové hry. Pro skenování slouží knihovna „FlutterBarcodeScanner“ [13]. Po kliknutí na tlačítko se spustí fotoaparát mobilního telefonu a aplikace skenuje obraz pro čárový kód. Po zachycení čárového kódu se výsledek skenu pošle na backend a po úspěšném skenu, tedy vrácení správné naskenované hry z backendu si ji uživatel může přidat do knihovny.

3.1.4.2.1 Plnění úspěchů

Pro plnění úspěchu uživatel klikne na kartu deskové hry v sekci knihovna a tím otevře detail hry viz obr. 3.12. V detailu hry můžeme vidět všechny úspěchy hry a body uživatele. Pro přidání bodů k úspěchu uživatel klikne na daný úspěch který chce upravit a tím vyjede spodní panel s možností přidání nebo odebrání bodu viz obr. 3.13. Po upravení počtu bodů se zobrazí seznam s přáteli a uživatel si vybere po kterém příteli chce daný posun v úspěchu potvrdit viz obr. 3.14. Id vybraných přátel se poté pošle v dotaze na backend pro zpracování.

Po dokončení tohoto procesu se upravený úspěch označí jako nepotvrzený a to červenou tečkou vedle názvu úspěchu viz první úspěch na obr. 3.12. Po potvrzení úspěchu vybraným uživatelem se tato červená tečka vymaže.

3.1.4.3 Notifikace

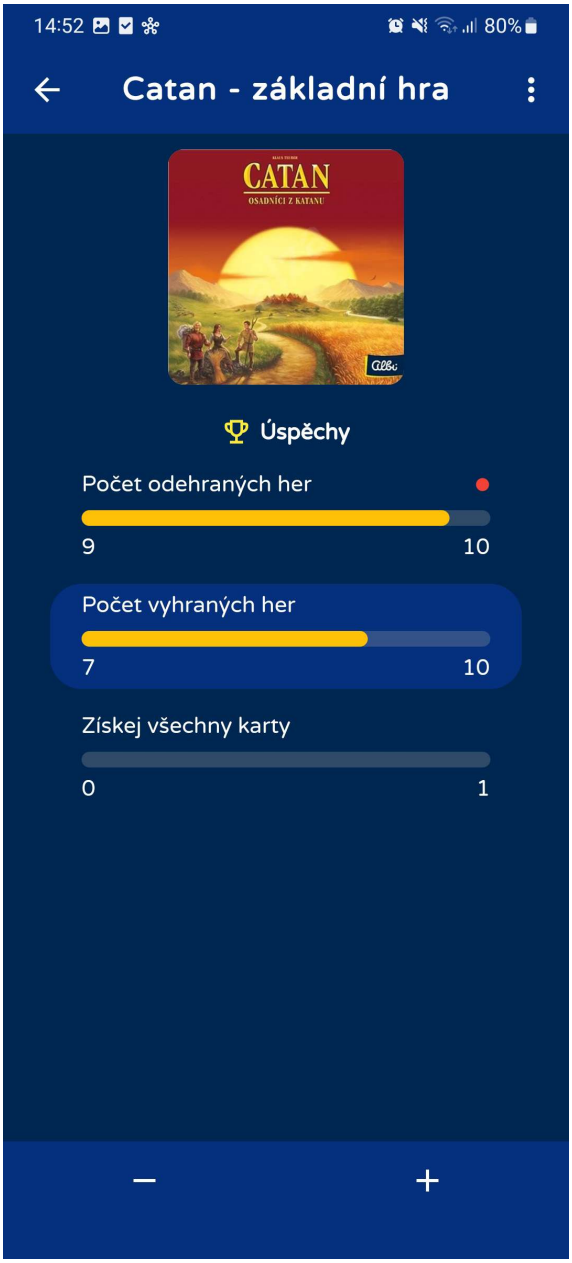
Důležitou součástí nejen plnění úspěchů v aplikaci, ale i předávání informací uživateli, jsou notifikace. Aplikace stahuje tyto data z backendu každých třicet vteřin. Pokud jsou přijaté nové data, aplikace hned upozorní uživatele na hlavní obrazovce přidáním počtu notifikací k ikonce zvonečku v záhlaví. Po kliknutí na tuto ikonku můžeme vidět seznam přijatých notifikací viz obr. 3.10. Žádost o potvrzení posunu v úspěchu můžeme potvrdit přejetím doleva a zamítnout přejetím doprava. Po provedení akce tato notifikace zmizí.

Obrázek 3.10: Notifikace.

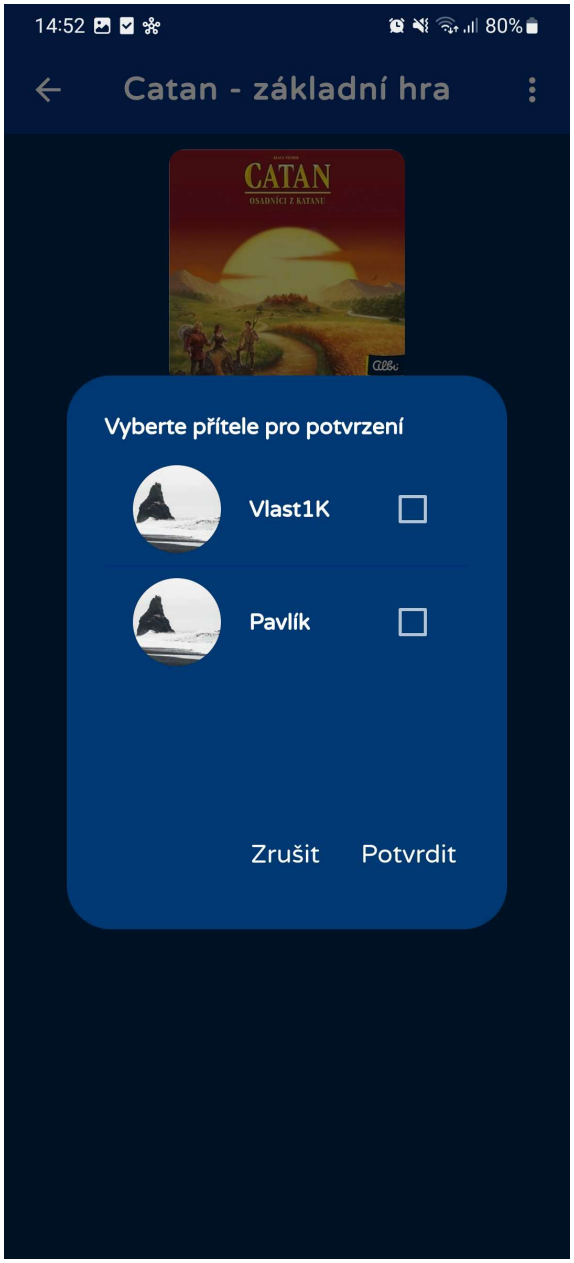
Obrázek 3.11: Knihovna her.

Obrázek 3.12: Detail hry.

Obrázek 3.13: Přidávání bodů.



Obrázek 3.14: Seznam přátel pro výběr.



3.1.4.4 Herní kluby a akce

Další sekce v pořadí je menší sociální síť viz obr. 3.15. Zde můžeme vidět nejbližší skupiny v okolí podle GPS souřadnic uživatele a nastávající akce klubů. Pro zobrazení všech skupin se otevře mapa se všemi kluby viz obr. 3.16. Pro práci s mapou slouží knihovna „FlutterMap“ [14]. Po kliknutí na bod klubu na mapě se zobrazí základní informace s možností přechodu na detail klubu, ten si můžeme otevřít také přes kartu klubu na hlavní stránce komunitní sekce.

V detailu klubu klubu můžeme vidět popis, adresu, akce, členy a chat klubu viz obr. 3.17. Na kartě členů může majitel a admin klubu spravovat členy. Na kartě akcí jsou nastávající akce klubu a na poslední kartě chatu je klubový chat. Uživatel se může připojit do klubu pokud v něm již není a také klub opustit pomocí tlačítek pod adresou. Chat a akce jsou stále ve fázi vývoje.

Dále v této sekci si uživatel může založit vlastní skupinu vyplněním formuláře viz obr. 3.18. Po vyplnění údajů uživatel vybere přesnou lokaci klubu na mapě. Tyto data se pošlou na backend pro další zpracování.

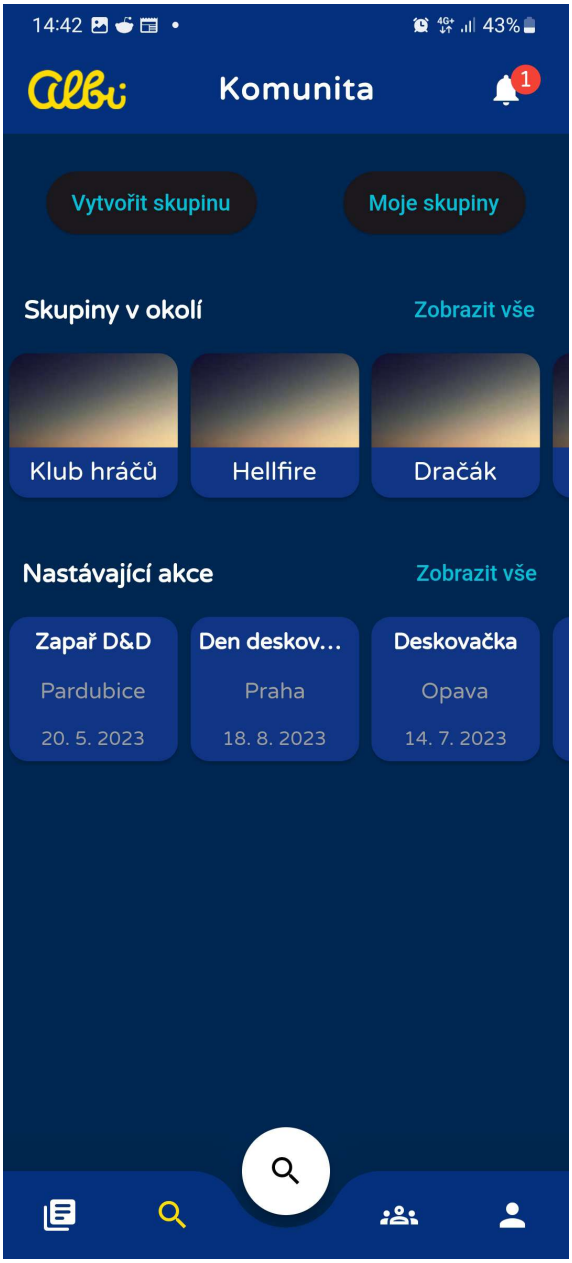
3.1.4.5 Přidávání přátel

Pro přidávání přátel slouží třetí sekce „Přátele“ viz obr. 3.19. Zde jsou tři karty zobrazení přátel. První karta obsahuje seznam přátel kteří potvrdili žádost o přátelství. Na druhé kartě jsou žádosti o přátelství jiných uživatelů. A poslední karta je seznam poslaných, ale ještě nepřijatých žádostí o přátelství. Na každé kartě můžeme provést určitou akci s přítelem. Přejetí doleva na kartě „Přátelé“ můžeme přítele odebrat, v žádostech tím přijmeme konkrétní žádost a v nevyřízených žádostech o přátelství tím poslanou žádost zrušíme. Přejetím doprava na kartě „Žádosti“ žádost o přátelství odmítneme.

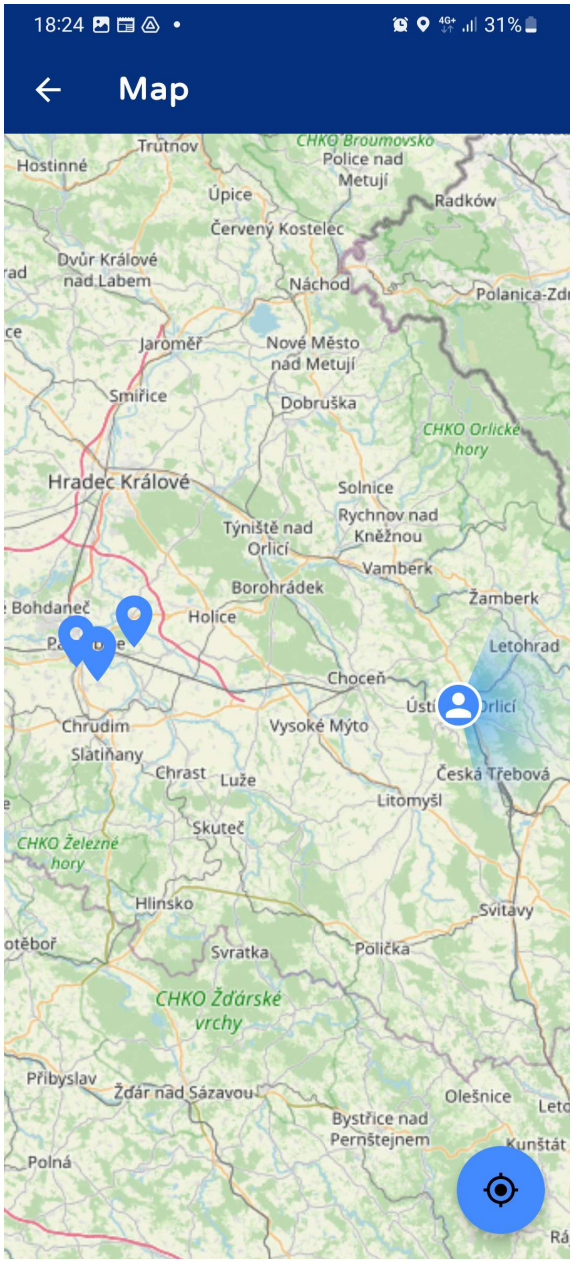
3.1.4.6 Profil

Poslední sekce je uživatelský profil viz obr. 3.20. Zde jsou o uživatelovi, jako celkový počet vlastněných her a počet splněných úspěchu ve všech vlastněných hrách. Po posunutí dolů se uživatel může odhlásit tlačítkem „Odhlásit se“, tím se smaže uložený token v mobilním telefonu a aplikace uživatele odhlásí.

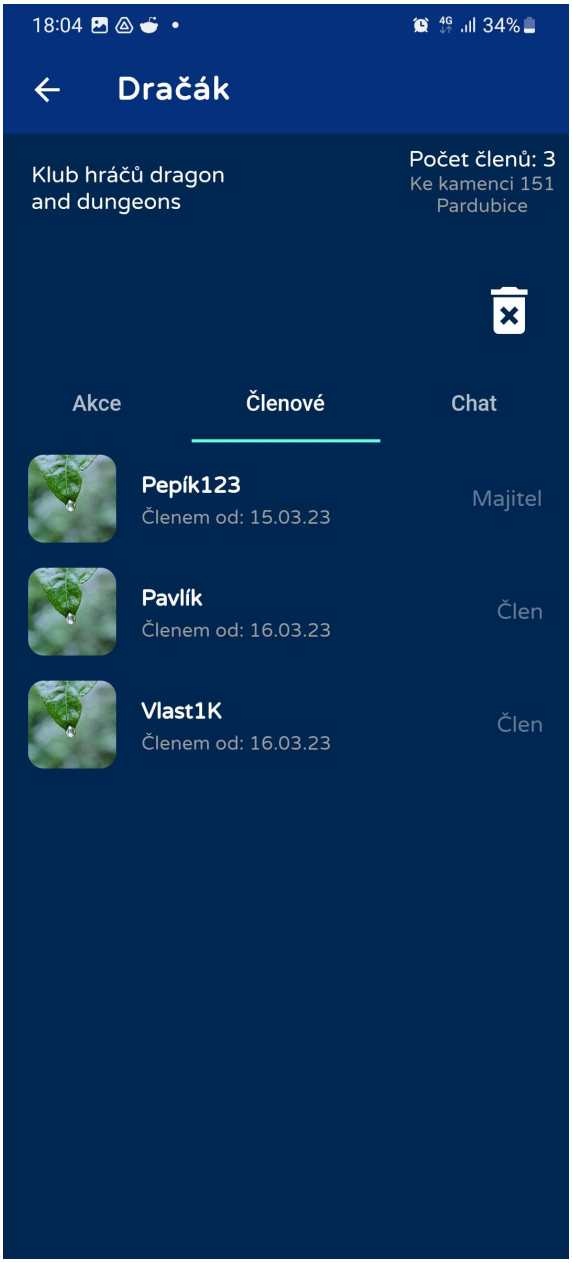
Obrázek 3.15: Herní kluby a akce.



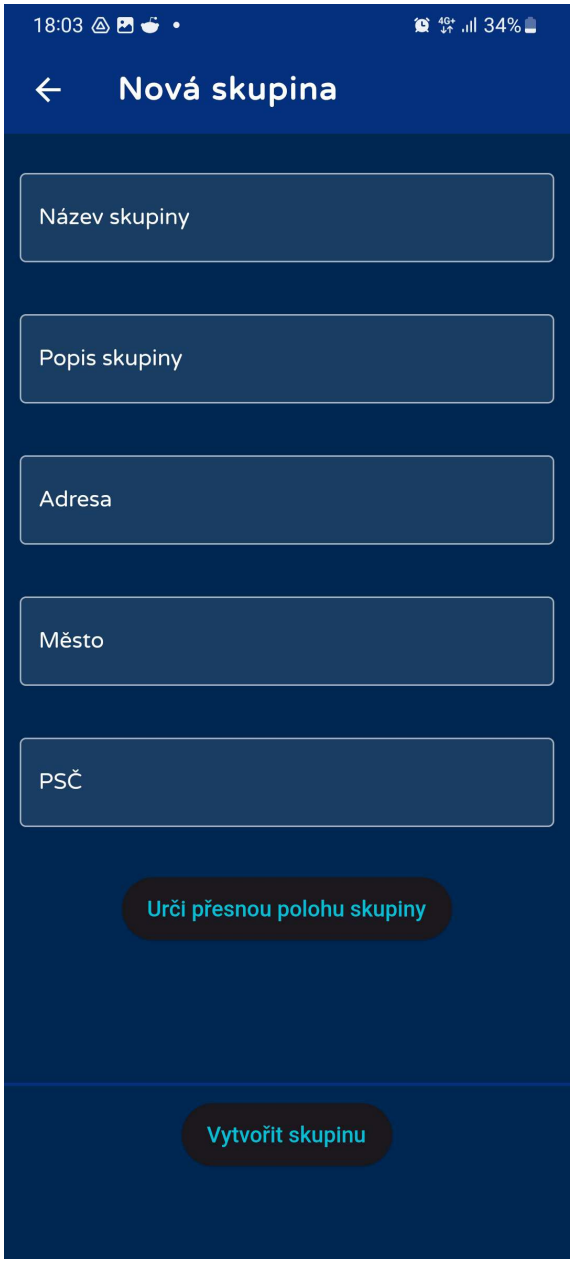
Obrázek 3.16: Mapa klubů.



Obrázek 3.17: Detail klubu.

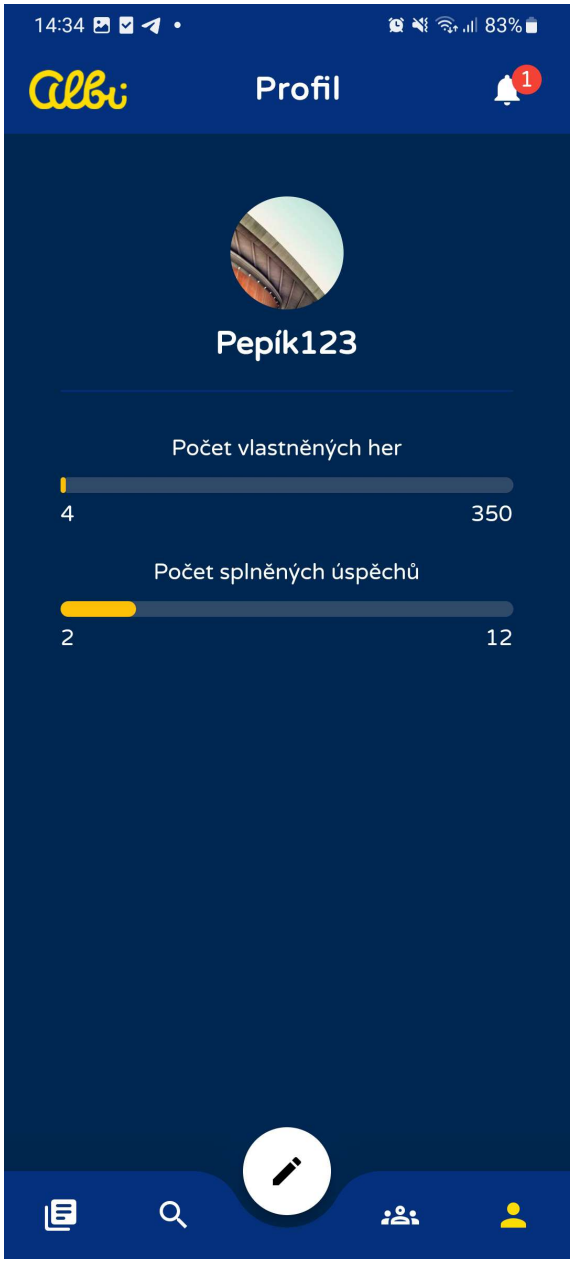


Obrázek 3.18: Formulář pro vytvoření klubu.



Obrázek 3.19: Sekce přátel.

Obrázek 3.20: Profil uživatele.



4 Závěr

V rámci projektu byla navržena a implementována mobilní aplikace a její backend. Aplikace je zaměřená na reklamní marketing společnosti ALBI Česká republika a.s. s prvky menší sociální sítě. Backend zastává veškerou logiku aplikace. Projekt je stále ve fázi vývoje a na dalším postupu rozhodne společnost ALBI.

Literatura

[1] *Steam* [online]. [cit. 2023-03-22]. Dostupné z `https://store.steampowered.com/`

[2] *Kontejner* [online]. [cit. 2023-03-22]. Dostupné z `https://www.docker.com/resources/what-container/`

[3] *What are ORMs ?* [online]. [cit. 2023-03-22]. Dostupné z `https://www.prisma.io/docs/concepts/overview/prisma-in-your-stack/is-prisma-an-orm#what-are-orms`

[4] *What is hashing ?* [online]. [cit. 2023-03-22]. Dostupné z `https://www.educative.io/answers/what-is-hashing`

[5] *Laravel* [online]. [cit. 2023-03-22]. Dostupné z `https://laravel.com/`

[6] *MySQL* [online]. [cit. 2023-03-22]. Dostupné z `https://www.mysql.com/`

[7] *Docker* [online]. [cit. 2023-03-22]. Dostupné z

[8] *Flutter* [online]. [cit. 2023-03-22]. Dostupné z `https://flutter.dev/`

[9] *What is MVC ?* [online]. [cit. 2023-03-22]. Dostupné z `https://developer.mozilla.org/en-US/docs/Glossary/MVC`

[10] *bcrypt* [online]. [cit. 2023-03-22]. Dostupné z `https://auth0.com/blog/hashing-in-action-understanding-bcrypt/`

[11] *Harvensine formula* [online]. [cit. 2023-03-22]. Dostupné z `https://en.wikipedia.org/wiki/Haversine_formula`

[12] *SharedPreferences* [online]. [cit. 2023-03-22]. Dostupné z `https://pub.dev/packages/shared_preferences`

[13] *FlutterBarcodeScanner* [online]. [cit. 2023-03-22]. Dostupné z `https://pub.dev/packages/flutter_barcode_scanner`

[14] *FlutterMap* [online]. [cit. 2023-03-22]. Dostupné z `https://docs.fleaflet.dev/`

Seznam obrázků