

Vývoj mobilní aplikace pro stolní hru Dungeons & Dragons

Michal Kotolan

Bakalářská práce
2025



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně

Fakulta aplikované informatiky

Ústav informatiky a umělé inteligence

Akademický rok: 2024/2025

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: Michal Kotolan
Osobní číslo: A21059
Studijní program: B0613A140020 Softwarové inženýrství
Forma studia: Prezenční
Téma práce: Vývoj mobilní aplikace pro stolní fantasy hru
Téma práce anglicky: Development of a mobile application for a fantasy board game

Zásady pro vypracování

- V teoretické části popište klíčové herní prvky hry Dungeons & Dragons potřebné pro návrh a implementaci aplikace, včetně správy postav, herních prvků a soubojových scénářů.
- Popište použité technologie framework Ionic, Angular a open-source API.
- Navrhněte strukturu aplikace a způsob shromažďování herních informací jako vybavení, nepřátel, a dalších herních prvků.
- Vytvořte aplikaci s využitím frameworku Ionic a Angular. Implementujte základní funkce, jako je správa postav, hledání a filtrování herních prvků, a zohledněte intuitivní uživatelské rozhraní.
- Otestujte aplikaci, zohledněte její funkčnost a přínos pro uživatele. Navrhněte možnosti dalšího rozšíření aplikace.

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

1. Dungeons & Dragons: Player's Handbook. Wizards of the Coast, 2014. ISBN 9780786965601.
2. GOLDBERG, Josh. Learning TypeScript: enhance your web development skills using type-safe JavaScript [online]. Beijing: O'Reilly, 2022 [cit. 2024-10-24]. ISBN 978-1-098-11033-8. Dostupné z: <https://learning.oreilly.com/library/view/learning-typescript/9781098110321/>
3. GRIFFITH, Chris. Mobile App Development with Ionic, Revised Edition [online]. O'Reilly Media, 2017 [cit. 2024-10-24]. ISBN 978-1-491-99812-0. Dostupné z: <https://learning.oreilly.com/library/view/mobile-app-development/9781491998113/>
4. SESHADRI, Shyam. Angular: Up and Running [online]. O'Reilly Media, 2018 [cit. 2024-10-24]. ISBN 978-1-491-99983-7. Dostupné z: <https://learning.oreilly.com/library/view/angular-up-and/9781491999820/>
5. FIRTMAN, Maximiliano. High Performance Mobile Web [online]. O'Reilly Media, 2016 [cit. 2024-10-24]. ISBN 9781491913673. Dostupné z: <https://learning.oreilly.com/library/view/high-performance-mobile/9781491912546/>

Vedoucí bakalářské práce: **Ing. Adam Ulrich**
Ústav matematiky

Datum zadání bakalářské práce: **27. října 2024**
Termín odevzdání bakalářské práce: **2. června 2025**



L.S.

doc. Ing. Jiří Vojtěšek, Ph.D. v.r.
děkan

prof. Mgr. Roman Jašek, Ph.D., DBA v.r.
ředitel ústavu

Ve Zlíně dne 17. listopadu 2024

PROHLÁŠENÍ AUTORA BAKALÁŘSKÉ PRÁCE

Beru na vědomí, že

- odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb., v platném znění bez ohledu na výsledek obhajoby;
- bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému a bude dostupná k nahlédnutí;
- jedno vyhotovení bakalářské práce v listinné podobě bude ponecháno Univerzitě Tomáše Bati ve Zlíně k uložení;
- na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- podle § 60 odst. 1 autorského zákona má Univerzita Tomáše Bati ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- podle § 60 odst. 2 a 3 mohu užít své dílo – bakalářskou práci – nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tj. k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá; neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji, že

- jsem na bakalářské práci pracoval(a) samostatně a použitou literaturu jsem řádně citoval(a); v případě publikace výsledků budu uveden(a) jako spoluautor;
- odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou obsahově totožné.

Ve Zlíně, dne

.....
podpis autora

ABSTRAKT

Tato bakalářská práce se zabývá návrhem a vývojem mobilní aplikace určené pro podporu hráčů stolní fantasy hry Dungeons & Dragons. Hlavním cílem bylo vytvořit uživatelsky přívětivou a přehlednou aplikaci, která umožní správu postav, procházení herních prvků a vytváření herních kampaní. Pro vývoj byl zvolen framework Ionic ve spojení s Angular, který umožňuje multiplatformní vývoj. Aplikace využívá otevřené API, z něhož načítá herní data jako jsou vybavení, monstra a magické předměty. Důraz byl kladen na interaktivitu, modularitu a možnost budoucího rozšiřování. Součástí aplikace jsou také doplňkové moduly, například nástroj pro házení kostkou přímo v textu nebo parser tabulek s náhodnými výsledky. Aplikace byla testována v uživatelském prostředí, kde získala pozitivní ohlas a důležitou zpětnou vazbu. Výsledkem je funkční nástroj, který může být v budoucnu dále rozšiřován například o databázové propojení nebo rozšířené možnosti spolupráce mezi hráči.

Klíčová slova: Dungeons & Dragons, Ionic, Angular, mobilní aplikace, stolní hra

ABSTRACT

This bachelor's thesis focuses on the design and development of a mobile application intended to support players of the tabletop fantasy game Dungeons & Dragons. The main goal was to create a user-friendly and well-organized application that enables character management, browsing of game elements, and the creation of game campaigns. The application was developed using the Ionic framework in combination with Angular, allowing for cross-platform deployment. It uses an open-source API to retrieve game data such as equipment, monsters, and magic items. Emphasis was placed on interactivity, modularity, and the possibility of future expansion. The application also includes additional modules, such as a tool for rolling dice directly in the text and a parser for tables with random outcomes. It was tested in a user environment where it received positive feedback and valuable suggestions. The result is a functional tool that can be further extended in the future, for example, with database integration or enhanced features for player collaboration.

Keywords: Dungeons & Dragons, Ionic, Angular, mobile application, tabletop game

Tímto bych rád poděkoval vedoucímu práce za jeho trpělivost, čas a nadšení, které věnoval této práci. Zároveň děkuji za cenné rady, podporu při vývoji a zpětnou vazbu, která významně přispěla k jejímu dokončení.

Také chci poděkoval lidem a přátelům, kteří se podíleli na testování aplikace a projevíli zájem, který doložili zpětnou vazbou.

OBSAH

ÚVOD.....	11
I TEORETICKÁ ČÁST	12
1 MOTIVACE.....	13
1.1 HRANÍ S PŘÁTELI.....	13
1.2 PRŮZKUM TRHU A EXISTUJÍCÍ NÁSTROJE	14
1.2.1 D&D Beyond	14
1.2.2 Roll20	15
1.3 NEDOSTATKY SOUČASNÝCH ŘEŠENÍ.....	17
2 POŽADAVKY NA APLIKACI	19
2.1 FUNKČNÍ POŽADAVKY	19
2.2 NEFUNKČNÍ POŽADAVKY	20
2.3 PŮVOD POŽADAVKŮ	23
3 VÝBĚR TECHNOLOGIÍ.....	24
3.1 IONIC.....	24
3.2 ANGULAR.....	24
3.3 VEŘEJNÁ D&D API	25
3.3.1 Lokální hostování pomocí Dockeru	25
4 SHRUTÍ A PŘECHOD K IMPLEMENTACI	26
II PRAKTICKÁ ČÁST.....	27
5 NÁVRH APLIKACE VE FIGMĚ.....	28
5.1 NÁVRH KLÍČOVÝCH OBRAZOVEK	28
5.1.1 Domovská stránka (HomePage).....	28
5.1.2 Stránka kampaní (CampaignsPage)	29
5.1.3 Stránka postav (CharacterPage)	30
5.1.4 Stránka vyhledávání (SearchPage).....	32
5.2 NÁVRH STRÁNEK PRO SPRÁVU KAMPANÍ.....	34
5.2.1 Stránka pro detail kampaně	34
5.2.2 Stránka pro soubojový scénář	36
5.3 STRÁNKA PRO DETAIL POSTAVY	39
5.4 STRÁNKA PRO DETAIL VYBAVENÍ.....	41
5.5 STRÁNKA PRO DETAIL MONSTER	42
5.6 STRÁNKA PRO VYTVÁŘENÍ POSTAVY	43
6 MAPA APLIKACE	47
7 PROCES VÝVOJE A PRÁCE S DATY Z API.....	48
7.1 PRÁCE S DATY A DATOVÁ VRSTVA	48

7.1.1	Struktura odpovědi z API a návrh kořenových rozhraní	48
7.1.2	Práce s prostředím a konfigurací API.....	50
7.1.3	Načítání a zpracování dat z API	51
7.1.4	Načítání a práce s daty ve službách a komponentách	52
7.1.5	Navigace a načtení detailních dat.....	53
7.1.6	Struktura rozhraní.....	54
8	ARCHITEKTURA APLIKACE A IMPLEMENTACE KOMPONENT	56
8.1	STRUKTURA APLIKACE A HLAVNÍ SEKCE	56
8.1.1	Navigační struktura (tabs)	56
8.1.2	Domovská stránka (Home).....	57
8.1.3	Vyhledávání (Search)	59
8.1.4	Stránka kampaní (Campaigns)	61
8.1.5	Stránka postav (Characters).....	63
8.2	KLÍČOVÉ KOMPONENTY APLIKACE	64
8.2.1	Vytváření postav.....	65
8.2.2	Interaktivnost aplikace	70
9	TESTOVÁNÍ.....	78
9.1	CHYBY	78
ZÁVĚR.....		80
SEZNAM POUŽITÉ LITERATURY		81
SEZNAM OBRÁZKŮ		83
SEZNAM TABULEK.....		85
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK		86
SEZNAM PŘÍLOH.....		CHYBA! ZÁLOŽKA NENÍ DEFINOVÁNA.

ÚVOD

V posledních letech jsem spolu s přáteli, podobně jako mnoho dalších lidí, začal hrát stolní hru Dungeons & Dragons (D&D) – fantasy role-playing hra, která hráčům umožňuje vstoupit do dobrodružného světa plného magie, monster, strategického myšlení a především příběhů. S nástupem digitalizace se způsob hraní výrazně posunul. Problematika spojená se hrou, která se dříve hrála převážně s papírem a tužkou, se začala řešit digitálními nástroji, které dnes pomáhají jak novým, tak zkušenějším hráčům v různých oblastech hraní – od vytváření postav až po tvorbu interaktivních map.

Tato práce se zaměřuje na vývoj mobilní aplikace, která slouží jako pomocník pro tyto hráče. Hlavním cílem bylo vytvořit nástroj pro intuitivní a rychlé vytváření postav a herních kampaní, včetně vyhledávání a interaktivního zobrazování herních dat, s důrazem na rychlost a přehlednost během hraní.

Při vývoji byly využity mimo jiné technologie Ionic a Angular, které umožňují tvorbu multiplatformních aplikací pro web i mobilní zařízení. Práce je zároveň koncipována tak, aby umožňovala další rozšiřování a úpravy, například na základě zpětné vazby z testování.

Práce je rozdělena na teoretickou část, která popisuje problematiku, motivaci ke vzniku aplikace, požadavky na řešení a výběr použitých technologií, a na část praktickou, která se zabývá návrhem aplikace, její implementací, řešením konkrétních problémů při vývoji a testováním funkčnosti.

I TEORETICKÁ ČÁST

1 MOTIVACE

Motivace pro vznik této práce vychází ze snahy zjednodušit a zefektivnit způsob, jakým hráči D&D přistupují k organizaci hry. V posledních letech se hraní výrazně proměnil díky nástupu digitálních nástrojů, které postupně nahrazují tradiční papírové pomůcky. Tyto nástroje navíc umožňují hrát i na dálku, bez nutného fyzického kontaktu, kdy lze veškeré herní interakce provádět bez použití papíru, tužky a fyzických herních kostek – čistě elektronicky.

Ačkoli dnes existuje několik populárních digitálních řešení, jejich použití s sebou často přináší komplikace – složitější uživatelské rozhraní, nedotáhnutá interaktivnost nebo přehlcení obsahem, který není relevantní pro všechny uživatele. Cílem této práce bylo vytvořit alternativní aplikaci, která by se pokusila vyřešit určité nedostatky, které ovlivňují uživatelské pohodlí, zohlední osobní zkušenosti z vlastního hraní a nabídne přehledné a jednoduché rozhraní, zejména pro začínající hráče.

1.1 Hraní s přáteli

Moje první setkání se stolní hrou na hrdiny přišlo již na základní škole, kdy mě spolužáci přivedli do kroužku, kde se hrála česká fantasy hra Dračí doupě, vycházející z amerického systému Dungeons & Dragons [1]. Po základní škole jsem se hraní těchto her na několik let nevěnoval, až v pozdějších vysokoškolských letech přišel kamarád s nápadem, že bychom se mohli pravidelně scházet s dalšími přáteli a začít hrát.

Začínali jsme bez jakýchkoli digitálních nástrojů, pouze s papírem, tužkou a klasickými hracími kostkami. Postupem času jsme však začali narážet na různé organizační problémy – například si někdo zapomněl list s postavou, jednotlivá kola hráčů se protahovala kvůli zdoluhavému hledání informací, nebo příprava soubojového scénáře zabrala příliš mnoho času. To vše mělo negativní dopad na plynulost hry a snižovalo míru ponoření do herního světa.

Tyto problémy nás vedly k hledání nástrojů, které by pomohly organizaci a průběh hry zefektivnit. Hráči, kteří měli tendenci zapomínat svou postavu, začali používat například D&D Beyond, digitální nástroj pro správu postav, který umožňuje i házení kostkami a přehledné sledování statistik [2]. Pro rychlé vyhledávání herních pravidel a informací o monstrech jsme využívali komunitní projekt 5etools [3]. Aplikace Roll20 zase nabízela

komplexní prostředí pro hraní na dálku, včetně tvorby map, správy postav i herních pravidel [4].

Tyto nástroje nám výrazně pomohly. Postupem času však začalo přibývat různých samostatných řešení – od jednoduchých nástrojů pro zobrazování obrázků až po komplexní systémy pro správu herních map. Hraní se tak stávalo roztržitém, přepínání mezi jednotlivými aplikacemi zdržovalo a snižovalo přehlednost. To mě vedlo k myšlence vytvořit vlastní nástroj, který by reflektoval naše potřeby a v rámci jedné aplikace je efektivně propojil.

Tato zkušenost vedla k hlubšímu zkoumání existujících řešení a dostupných nástrojů, které by bylo možné využít nebo se jimi inspirovat při vývoji vlastního řešení.

1.2 Průzkum trhu a existující nástroje

V současnosti existuje řada nástrojů, které hráčům D&D usnadňují hraní hry. V rámci této kapitoly jsou stručně představeny nejčastěji používané platformy, které slouží k tvorbě postav, správě kampaní, online hraní a přístupu k pravidlům. Zvláštní důraz je kladen na dva nejrozšířenější nástroje – D&D Beyond a Roll20.

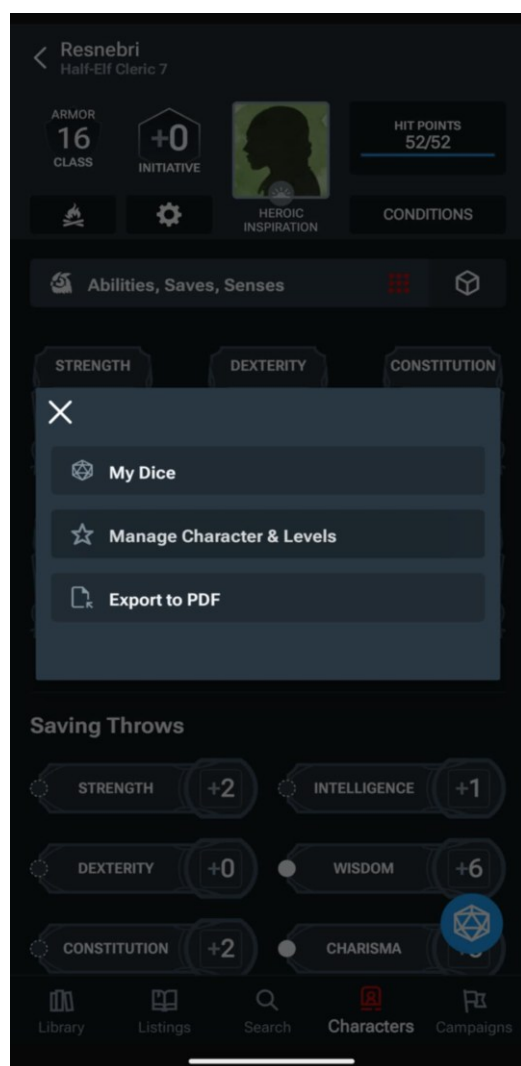
1.2.1 D&D Beyond

D&D Beyond je oficiální digitální platforma společnosti Wizards of the Coast [5], která poskytuje přístup ke kompletním pravidlům, předpřipraveným dobrodružstvím i interaktivnímu nástroji pro tvorbu postav. Uživatelé zde mají možnost [2]:

- Tvořit a spravovat postavy pomocí intuitivního průvodce.
- Ukládat si vybavení, schopnosti, kouzla apod.
- Využívat digitální kostky pro hody během hry.
- Prohlížet pravidla a herní obsah (základní obsah zdarma, dodatečný obsah však placený).
- Možnost vytváření map (momentálně v beta verzi, uzavřené za předplatným) [6].

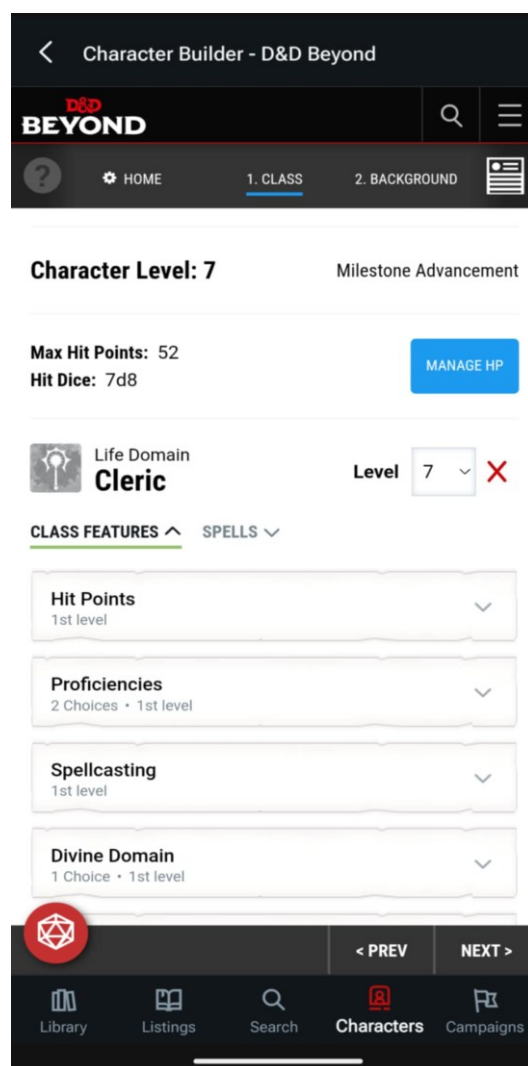
Nástroj disponuje jak webovým rozhraním, tak i mobilní aplikací, která nabízí téměř všechny funkce dostupné ve webové verzi. Některé činnosti však nelze v mobilní aplikaci provádět přímo – uživatel je v těchto případech přesměrován na webové rozhraní. Typickým

příkladem je editace postavy při postupu na vyšší úroveň, kdy je nutné otevřít webovou aplikaci k dokončení změn, rozdíl je vidět níže u Obrázků 1 a 2.



Obrázek 2 Aplikace D&D Beyond

(zdroj vlastní)



Obrázek 1 Přechod na webové rozhraní v aplikaci

(zdroj vlastní)

Mimo běžného nakupování obsahu, má D&D Beyond možnost předplatného, které odemyká možnost vytvářet neomezený počet postav, předběžný přístup k novým knihám a další měsíční výhody. Při předplácení nejvyššího stupně, má uživatel přístup i k tvorbě interaktivních map [7].

1.2.2 Roll20

Roll20 je online platforma určená pro hraní stolních RPG her na dálku. Funguje přímo ve webovém prohlížeči a umožňuje komplexní správu herního prostředí včetně tvorby map a interakce s hráči. Uživatelé mohou pracovat s následujícími funkcemi [8; 9]:

- Virtuální herní plocha – umožňuje nahrávání map, umisťování tokenů, kreslení, nastavování viditelnosti nebo tzv. fog of war.
- Integrovaný chat a hlasová komunikace – hráči mohou komunikovat přímo v rámci prostředí, včetně videohovorů.
- Nástroje pro házení kostkami.
- Správa postav a herních listů – pro mnoho herních systémů jsou k dispozici šablony postav, včetně D&D [10].

Ne všechny herní systémy jsou v plné formě dostupné všem uživatelům. Značná část obsahu je uzamčena nebo omezena modelem předplatného. Níže jsou uvedeny hlavní omezení bezplatné verze [9]:

- Limitovaný prostor pro upload – v základním (free) účtu je omezené množství dat pro nahrávání map, tokenů a dalších médií.
- Chybějící pokročilé funkce – například dynamické osvětlení, line of sight nebo automatické skripty jsou dostupné pouze s předplatným.
- Omezené sdílení a přizpůsobení – sdílení obsahu a pokročilé přizpůsobení hry je často možné až v placené verzi.

Roll20 také nabízí mobilní aplikaci určenou pro správu postav [11; 12]. Podle údajů z Google Play však byla naposledy aktualizována 1. září 2023 a pro zařízení s operačním systémem Android 15 již není její instalace dostupná [11].

Pro přehlednost jsou vybrané vlastnosti nástrojů shrnuty v následující tabulce:

Tabulka 1 Srovnání D&D Beyond a Roll20

Funkce / Vlastnosti	D&D Beyond	Roll20
Oficiální podpora	Ano (Wizards of the Coast)	Ne
Platforma	Web, mobilní aplikace	Webová aplikace
Tvorba postav	Ano	Ano
Správa kampaní	Poznámky	Virtuální mapy, poznámky
Tvorba map	Ano, schovaná za předplatným (Beta verze)	Ano, s mnoha funkcemi

Obsah	Základní zdarma, ostatní obsah placený	Základní systémy dostupné
Multiplayer a komunikace	Ne	Integrovaný chat, video
Model předplatného	Ano (mapy, neomezené vytváření postav)	Ano (více funkcionalit, větší upload, vypnutí reklam)
Přesměrování mezi aplikacemi	Ano (Např. při editaci postavy)	Ne, vše v prohlížeči

(Zdroj vlastní)

1.3 Nedostatky současných řešení

Přestože současné digitální nástroje jako D&D Beyond či Roll20 poskytují širokou škálu funkcí pro hraní D&D, jejich používání může pro některé hráče představovat řadu překážek. Ty často vyplývají z komplexního uživatelského rozhraní, nedostatečné propojenosti funkcí nebo neintuitivního ovládání. Začínající hráče pak může odradit i model předplatného, který omezuje přístup k pokročilejším funkcím.

Jedním z hlavních nedostatků stávajících nástrojů je omezená interaktivita s herními daty. Například při práci s monstrem nebo předmětem, který obsahuje popis útoku či tabulku s náhodnými výsledky, není obvykle možné provést herní hod přímo z textu. Uživatel tak musí hod zadat ručně mimo daný kontext, což zpomaluje hru a snižuje plynulost interakce. Možnost provést hod (například na poškození nebo z tabulky efektů) přímo z rozhraní by výrazně zlepšila uživatelský komfort a zrychlila průběh hry.

Další častou slabinou je nedostatečně propojená navigace napříč funkcemi aplikace. Pokud si hráč vytvoří postavu a přiřadí ji do kampaně, není v některých nástrojích možné pohodlně přecházet mezi jednotlivými částmi – například z probíhajícího souboje přímo na detail postavy nebo monstra a zpět. Chybí plynulost pohybu mezi propojenými entitami. Tyto nedostatky vedou k frustraci zejména při dynamickém hraní a komplikují správu herních prvků.

Přestože současné nástroje nabízejí široké možnosti, často jim chybí dostatečná provázanost funkcí, jednoduchost ovládání nebo uživatelsky přívětivá práce s daty. Právě

proto jsem se rozhodl navrhnout vlastní aplikaci a pokusit se zhotovit řešení na základě požadavků, které reflektují mé zkušenosti se hrou.

2 POŽADAVKY NA APLIKACI

Cílem navrhované aplikace je pokrýt základní oblasti spojené s hraním stolní RPG (Role-playing game) hry D&D. Mezi klíčové funkce patří tvorba a správa postav, organizace kampaní včetně přípravy soubojových scénářů, vyhledávání monster a jejich integrace do hry, a rovněž správa herního vybavení. Aplikace je navržena s důrazem na rychlý a intuitivní přístup k informacím a plynulou navigaci mezi jednotlivými částmi systému.

Součástí řešení je i možnost provádění herních hodů přímo z textu – například tam, kde se v popisu vyskytuje notace 2d6 nebo 1d20+3. Uživatel může takový hod vykonat jediným kliknutím, bez nutnosti přecházet na jiný nástroj. Cílem této funkce není nahrazovat fyzické herní kostky, ale nabídnout pohodlnou alternativu pro specifické situace – např. při testování, nebo při hodech zahrnujících více kostek současně.

2.1 Funkční požadavky

- Postavy
 - Uživatel může vytvářet a ukládat postavy.
 - Uživatel může upravovat vybavení postavy a spravovat její měnu.
 - Postavu lze smazat.
 - Postavu lze přidat do kampaně.
- Kampaně a souboje
 - Uživatel může vytvářet kampaně.
 - Do kampaně lze přidávat a odebírat hráčské postavy.
 - V rámci kampaně lze vytvářet bojové scénáře.
 - Do scénáře lze přidat hráčské postavy a monstra.
 - Aktivní scénář lze kdykoli opustit a později se k němu vrátit, pokud není ukončen.
 - U účastníků scénáře lze upravovat:
 - hodnoty životů,
 - stavy (např. otrávený, paralyzovaný),
 - poznámky.

- Uživatel může z bojového scénáře přejít na detail postavy nebo monstra bez přerušení průběhu scénáře.
- Aplikace uchovává historii ukončených scénářů.
- Kampaň lze smazat.
- Vyhledávání
 - Uživatel může vyhledávat položky v kategoriích:
 - Monstra,
 - Vybavení (Equipment),
 - Magické předměty (Magic Items).
 - Vyhledávání lze provádět podle názvu nebo více filtrů současně.
 - Vyhledané položky lze otevřít v detailním zobrazení.
- Zobrazení dat
 - Data se zobrazují podle kategorií v odpovídajících formátech (např. tabulkově).
 - V textu lze kliknutím spouštět herní hody, pokud je přítomen zápis ve formátu typu 2d6, 1d20+3, apod.
 - V rámci detailu postavy je možné provádět příslušné hody na vlastnosti, zbraně, schopnosti apod.

2.2 Nefunkční požadavky

- Použitelnost
 - Aplikace musí být intuitivní a snadno ovladatelná i pro uživatele bez hlubších technických znalostí.
 - Navigace mezi postavami, kampaněmi a jednotlivými funkcemi musí být rychlá a přehledná.
 - Ovládací prvky musí být konzistentní v celém rozhraní.
- Dostupnost
 - Aplikace bude dostupná jako mobilní aplikace.

- Spolehlivost
 - Aplikace nesmí při běžném používání padat.
- Údržba a rozšiřitelnost
 - Aplikace musí být navržena tak, aby bylo možné do budoucna snadno přidávat nové funkce (např. nové typy vybavení, herní systémy).
- Jazyková podpora
 - Aplikace bude dostupná v anglickém jazyce.

Dále je k dispozici tabulkový souhrn.

Tabulka 2 Funkční požadavky

ID	Typ	Kategorie	Popis
F-01	Funkční	Postavy	Uživatel může vytvářet a ukládat postavy.
F-02	Funkční	Postavy	Uživatel může upravovat vybavení postavy a spravovat měnu.
F-03	Funkční	Postavy	Postavu lze smazat.
F-04	Funkční	Postavy / Kampaně	Postavu lze přidat do kampaně.
F-05	Funkční	Kampaně	Uživatel může vytvářet kampaně.
F-06	Funkční	Kampaně	Do kampaně lze přidávat a odebírat postavy.
F-07	Funkční	Souboje	V kampani lze vytvářet bojové scénáře s hráčskými postavami a monstry.
F-08	Funkční	Souboje	Scénář lze opustit a později obnovit, pokud není ukončen.
F-09	Funkční	Souboje	U účastníků scénáře lze upravovat životy, stavy a poznámky.
F-10	Funkční	Souboje	Kliknutím lze zobrazit detail postavy/monstra bez přerušení scénáře.
F-11	Funkční	Souboje	Ukládání historie ukončených scénářů.

F-12	Funkční	Kampaně	Kampaň lze smazat.
F-13	Funkční	Vyhledávání	Uživatel může vyhledávat podle kategorií a názvu.
F-14	Funkční	Vyhledávání	Vyhledané položky lze otevřít v detailním zobrazení.
F-15	Funkční	Zobrazení dat	Data se zobrazují dle kategorií, např. v tabulkách.
F-16	Funkční	Zobrazení dat	Možnost klikacího házení kostkami přímo z textu.
F-17	Funkční	Zobrazení dat	Hody z detailu postavy na schopnosti, zbraně apod.

(Zdroj vlastní)

Tabulka 3 Nefunkční požadavky

NF-01	Nefunkční	Použitelnost	Aplikace musí být intuitivní i pro netechnické uživatele.
NF-02	Nefunkční	Použitelnost	Navigace musí být rychlá a přehledná.
NF-03	Nefunkční	Použitelnost	Konzistentní ovládací prvky v celém rozhraní.
NF-04	Nefunkční	Dostupnost	Aplikace bude dostupná jako mobilní aplikace.
NF-05	Nefunkční	Spolehlivost	Aplikace nesmí při běžném používání padat.
NF-06	Nefunkční	Rozšiřitelnost	Možnost budoucího rozšiřování o nové funkce, typy obsahu apod.
NF-07	Nefunkční	Jazyková podpora	Aplikace bude dostupná v anglickém jazyce.

(Zdroj vlastní)

2.3 Původ požadavků

Požadavky na aplikaci byly definovány na základě mých osobní zkušeností s vedením a hraním kampaní ve hře D&D v rámci mé herní skupiny přátel. Hlavní inspirací pro návrh byly dostupné digitální nástroje, jako jsou D&D Beyond [2] a Roll20 [4]. Během jejich používání byly identifikovány konkrétní nedostatky:

- Omezené interaktivity dat – například nemožnost přímo házet kostkami z textu popisujícího útok nebo efekt.
- Nedostatečné propojení mezi entitami – například komplikovaný přechod mezi postavou, vybavením, kampaní a scénářem bez ztráty kontextu.
- Uživatelského komfortu – některé nástroje jsou nepřehledné, náročné pro začínající hráče.

Funkční a nefunkční požadavky byly tedy sestaveny s cílem tyto slabiny stávajících řešení reflektovat a zároveň poskytnout základ pro uživatelsky přívětivou aplikaci. Cílem je vytvořit nástroj, který bude podporovat jak jednoduché herní operace (např. hod kostkou), tak i pokročilejší funkce jako správu kampaní a postav, a to v prostředí, které je intuitivní a svižné.

3 VÝBĚR TECHNOLOGIÍ

Volba vhodných nástrojů se odvíjela od mých zkušeností nebo od dostupnosti v případě veřejně dostupné API pro D&D.

3.1 Ionic

Ionic je open-source framework určený pro vývoj hybridních mobilních aplikací, které běží na platformách Android, iOS i web z jednoho kódu. Využívá moderní webové technologie jako HTML, CSS a JavaScript, s integrací pro populární frameworky jako Angular, React a Vue, a díky tomu umožňuje rychlý vývoj multiplatformního rozhraní [13].

Pro vývoj této aplikace jsem zvolil Ionic především na základě mých předchozích zkušeností s tímto nástrojem během studia. Jeho jednoduchá integrace s Angular prostředím, dostupná modernizovaná dokumentace urychlují vývoj. Ionic také poskytuje rozsáhlou knihovnu UI komponent, které umožňují tvorbu přehledného a konzistentního uživatelského rozhraní s minimem úsilí. To mi zároveň umožnilo zaměřit se primárně na funkční stránku aplikace bez nutnosti řešit vizuální identitu v současné verzi aplikace [14].

Alternativními řešeními jsou například React Native nebo Flutter. Vzhledem k absenci předchozích zkušeností s těmito nástroji a časové náročnosti při jejich osvojování jsem však zvolil Ionic.

3.2 Angular

Angular je framework od Googlu určený pro tvorbu webových aplikací. Poskytuje silnou typovou kontrolu díky integraci s TypeScriptem [15], výbornou strukturu pro správu stavu i komponent a rozsáhlý ekosystém [16].

Angular jsem zvolil především díky předchozí praxi a hlubší znalosti prostředí. Výhodou byla také možnost využití novějších prvků frameworku, především:

- Standalone komponenty – umožňují psát komponenty bez nutnosti deklarace v NgModule, což zjednodušuje strukturu projektu a zvyšuje modularitu [17; 18].
- Nový control flow (@if, @for) – nahradil původní *ngIf / *ngFor konstrukce. Tento nový přístup je přehlednější, lépe odpovídá imperativnímu stylu a přináší výrazné zlepšení v čitelnosti a udržitelnosti šablon [19].

Díky těmto změnám bylo možné psát aplikaci jednodušeji a efektivněji. K osvojení Angularu také napomohl moderní interaktivní tutoriál v rámci oficiální dokumentace [15].

3.3 Veřejná D&D API

Pro účely této aplikace jsem využil veřejně dostupné D&D 5e API [20], které poskytuje strukturovaný přístup k herním datům systému Dungeons & Dragons 5. edice prostřednictvím moderního RESTful API [21]. Jedná se o open-source projekt dostupný na GitHubu [22], který nabízí širokou škálu herních entit jako jsou monstra, vybavení, kouzla, schopnosti, povolání a další. API je stabilní [23], aktivně spravované komunitou a jeho obsah se drží oficiálně publikovaných SRD materiálů od Wizards of the Coast.

V době psaní této práce byla k dispozici pouze databáze pro pravidla 5e (2014). Tvůrci API však plánují rozšíření o aktualizovanou verzi 2024 (verze Next) [22], jestliže zůstane struktura databáze stejná, neměl by být problém využívat novou aktualizovanou verzi dat bez náročných zásahů do kódu.

3.3.1 Lokální hostování pomocí Dockeru

Namísto odesílání HTTP požadavků typu GET na vzdálený server API přes internet, jsem se rozhodl databázi hostovat lokálně, a to pomocí Dockeru. Docker je platforma pro kontejnerizaci aplikací, která umožňuje snadné zprovoznění softwarového prostředí v izolovaném kontejneru [24]. Výhodou tohoto přístupu je:

- Nižší latenci a vyšší rychlost odezvy během vývoje a testování.
- Nezávislost na síťovém připojení či dostupnosti vzdáleného serveru.
- Možnost plné kontroly nad běžící databází, včetně její případné úpravy.

Oficiální repozitář API obsahuje předpřipravený Dockerfile, který umožňuje rychlé nasazení databáze lokálně. Tento přístup zefektivnil vývoj a minimalizoval možné komplikace s latencí či výpadky. Repozitář také obsahuje přehledný popis nasazení jak na Windows, tak na Mac [22].

4 SHRNUTÍ A PŘECHOD K IMPLEMENTACI

Teoretická část této práce definovala základní požadavky na aplikaci, zhodnotila možnosti technického řešení a zdůvodnila výběr konkrétních nástrojů a technologií. Na tomto základě lze přistoupit k samotnému návrhu a realizaci aplikace, která bude tyto požadavky naplňovat v praxi.

Následující kapitoly se proto zaměřují na návrh architektury, implementační postupy a konkrétní funkce aplikace, včetně zhodnocení jejich funkčnosti a uživatelské přívětivosti.

II PRAKTICKÁ ČÁST

5 NÁVRH APLIKACE VE FIGMĚ

Prvotní návrh aplikace byl vytvořen ve Figmě, která byla zvolena především díky své dostupnosti, možnosti vytvářet rychlé wireframy a osobním zkušenostem. Návrh sloužil jako výchozí bod pro ověření uživatelského toku, rozložení prvků a celkové funkční logiky aplikace.

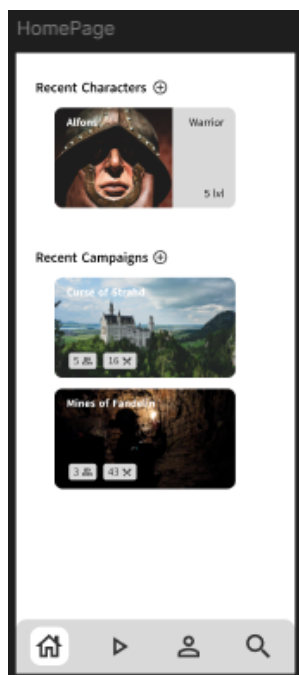
5.1 Návrh klíčových obrazovek

V rámci návrhu uživatelského rozhraní byly ve Figmě vytvořeny čtyři klíčové stránky, které odpovídají hlavním sekcím aplikace. Tyto sekce jsou přístupné prostřednictvím spodní navigační lišty (tabs pro top level navigaci) a tvoří základní strukturu uživatelského rozhraní.

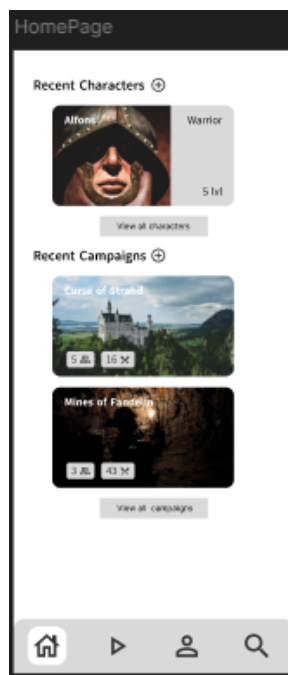
5.1.1 Domovská stránka (HomePage)

Slouží jako výchozí bod aplikace. Uživatel zde vidí přehled dostupných postav a kampaní, včetně možnosti přejít na jejich detailní zobrazení. Kromě toho je zde k dispozici také akce pro vytvoření nové postavy či nové kampaně (Obrázek 3).

Během implementace se ukázalo, že by mohl přístup ke stránkám jednotlivým sekcím být umožněn klikem na tlačítko pod každou sekcí (Obrázek 4).



Obrázek 3 Domovská stránka



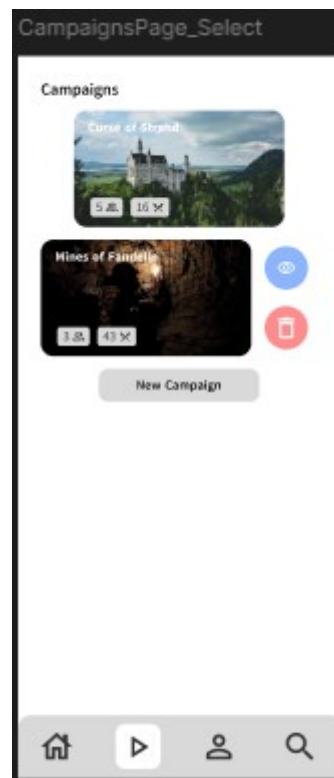
Obrázek 4 Upravená domovská stránka

5.1.2 Stránka kampaní (CampaignsPage)

Zobrazuje seznam všech existujících kampaní včetně údajů o počtu přiřazených postav a existujících scénářů (Obrázek 5). Každá položka v seznamu nabízí interaktivní prvky – tlačítko pro přechod na detail kampaně a tlačítko pro její smazání. Nová kampaň se vytváří pomocí tlačítka umístěného pod seznamem (Obrázek 6).



Obrázek 5 Stránka kampaní

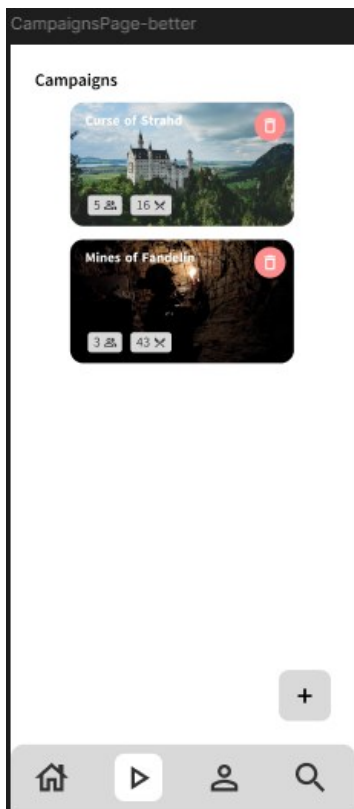


Obrázek 6 Stránka kampaní se
zobrazenými funkcemi

Během revize návrhu se ukázalo, že původní řešení interakce s kartou kampaně nebylo ideální z hlediska uživatelského zážitku. Uživatel musel nejprve kliknout na kartu, aby se mu zobrazila akční tlačítka, a poté zvolit konkrétní akci (například přechod na detail). Tento krok byl navíc zbytečně zdlouhavý a zpomaloval běžné používání aplikace (Obrázek 7).

Proto jsem se rozhodl přístup zjednodušit – kliknutí na celou kartu nyní uživatele okamžitě přesměruje na detail kampaně. Tlačítko pro smazání bylo přesunuto do rohu karty a doplněno o potvrzovací dialog (alert), aby se předešlo nechtěnému odstranění (Obrázek 7).

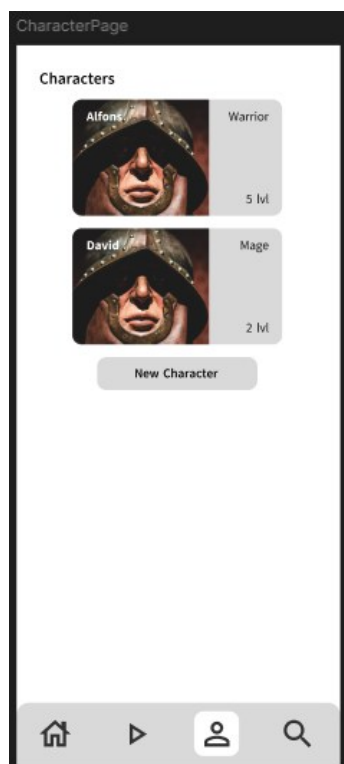
Další úpravou prošlo tlačítko pro vytvoření nové kampaně. Původně bylo umístěno za seznamem existujících kampaní a „cestovalo“ spolu s ním, což při větším počtu záznamů vedlo k horší dostupnosti. V nové verzi je tlačítko zafixované na spodní okraj obrazovky, kde je vždy snadno dosažitelné (Obrázek 7).



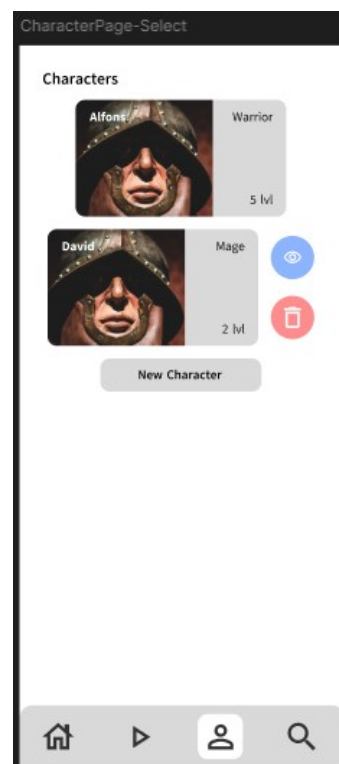
Obrázek 7 Upravená stránka kampaní

5.1.3 Stránka postav (CharacterPage)

Podobně jako u stránky kampaní se i zde zobrazuje seznam všech vytvořených postav, doplněný o klíčové informace, jako jsou jméno, třída a úroveň. Každá karta postavy nabízí možnosti správy – přechod na detail, úpravu nebo odstranění postavy. V dolní části stránky se nachází tlačítko pro přidání nové postavy (Obrázky 8 a 9).



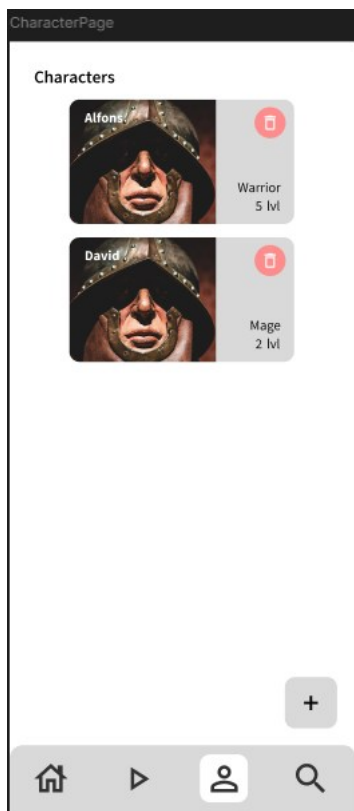
Obrázek 8 Stránka postav



Obrázek 9 Stránka postav se zobrazením funkcí

Při testování návrhu se ukázalo, že uživatelský tok trpí stejnými nedostatky jako v případě kampaní – nutnost dalšího kliknutí pro zobrazení možností a špatná dostupnost akčního tlačítka pro přidání nové postavy.

Řešení proto odpovídá změnám provedeným u kampaní: kliknutí na celou kartu nyní vede přímo na detail postavy a tlačítko pro odstranění je umístěno v rohu karty s potvrzovacím dialogem. Tlačítko pro vytvoření nové postavy je také trvale přichycené ve spodním rohu stránky (Obrázek 10).

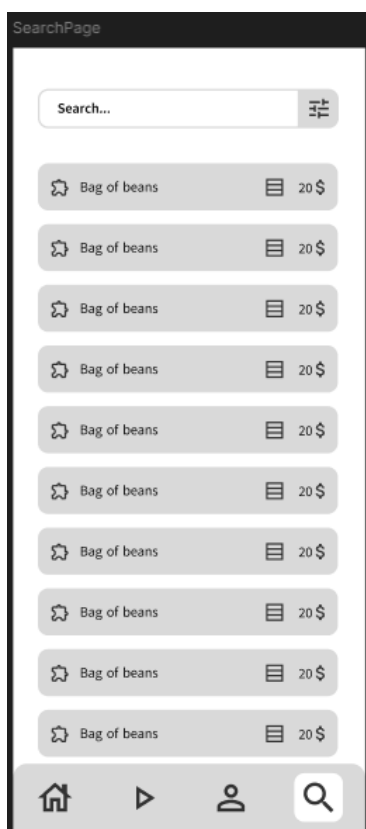


Obrázek 10 Upravená stránka postav

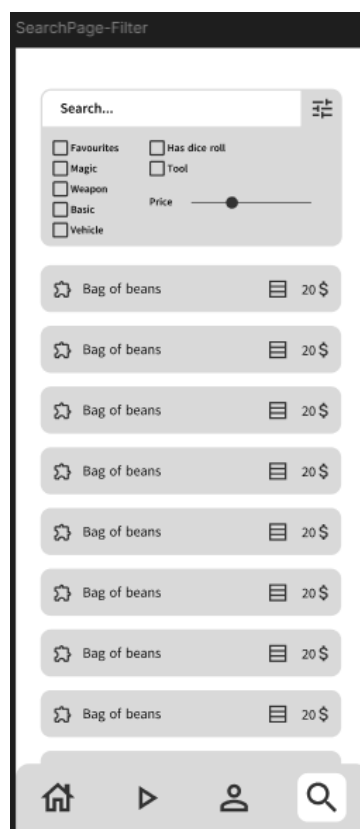
5.1.4 Stránka vyhledávání (SearchPage)

Stránka vyhledávání zobrazuje seznam položek získaných z API – konkrétně monster, vybavení a magických předmětů. Každá karta obsahuje název, případně cenu (pokud je dostupná), a ikonu určující typ položky. Vedle ceny je také tlačítko pro přechod na detail položky (Obrázek 11).

Uživatel má možnost filtrovat data podle názvu, kategorie a minimální ceny (Obrázek 12).



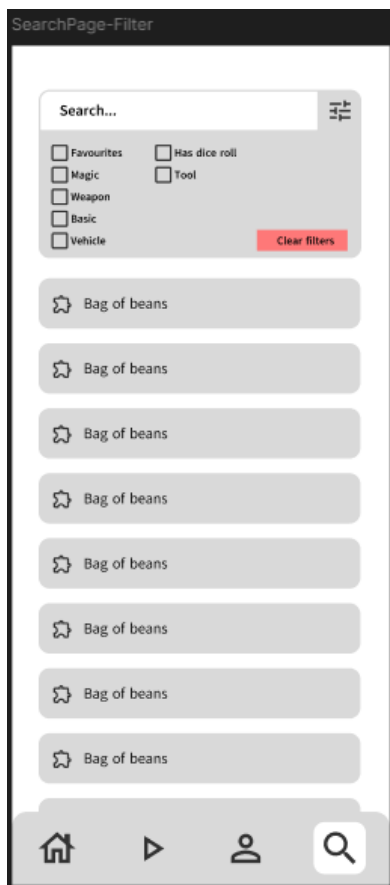
Obrázek 11 Stránka vyhledávání



Obrázek 12 Stránka vyhledáváním s filtry

Při průzkumu API se ukázalo, že cena není univerzální údaj – monstra ji vůbec neobsahují a u některých dalších položek (např. magických předmětů) je její význam zanedbatelný. Proto byla cena z karty odstraněna a s ní i příslušný filtr.

Stejně jako u ostatních stránek bylo zrušeno samostatné tlačítko pro přechod na detail a jeho funkce byla převedena na kliknutí celé karty, což zjednodušuje ovládání. Přibyl také nové tlačítko pro rychlé vymazání všech aktivních filtrů, čímž se zlepšil uživatelský komfort při práci s vyhledáváním (Obrázek 13).



Obrázek 13 Upravená stránka vyhledávání

5.2 Návrh stránek pro správu kampaní

V rámci sekce CampaignsPage bylo kromě hlavního přehledu nutné navrhnout i vnořené stránky pro správu jednotlivých kampaní a soubojových scénářů mezi nimi. Tyto stránky umožňují nejen zobrazit informace o konkrétní kampani, ale také spravovat její obsah – tedy přiřazené postavy a vytvořené soubojové scény. Bylo důležité, aby navigace mezi částmi kampaně byla svižná a popřípadě nenarušovala průběh soubojového scénáře.

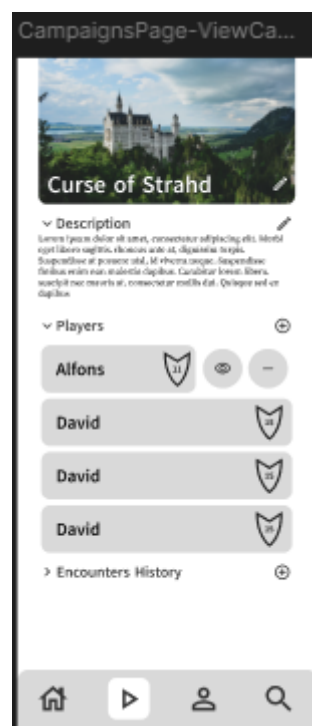
5.2.1 Stránka pro detail kampaně

Stránka zobrazuje detail vytvořené kampaně spolu s informacemi a o ní – jméno, popis, přiřazení hráči a historii soubojových scénářů. Sekce „Players“ a „Encounters History“ mají k dispozici tlačítko pro přidání nové instance, pro „Players“ se jedná o přidání postavy hráče do kampaně u druhé možnosti se jedná o vytvoření soubojového scénáře (Obrázek 14).

V rámci listu hráčů je opět možnost zobrazení možností pro náhled a odstranění hráče z kampaně (Obrázek 15).



Obrázek 14 Detail kampaně

Obrázek 15 Detail kampaně se
zobrazenými možnostmi

Zde se vyskytují obdobné problémy jako u předchozích stránek. Hlavní funkcí téhle stránky je vytváření soubojových scénářů, což by mělo být reflektováno v uživatelském rozhraní. Proto bylo přidáno akční tlačítko do spodního rohu právě pro vytváření scénářů (Obrázek 16).

Dále byly odstraněny možnosti u hráčů. Náhled je teď umožněn kliknutím na kartu hráče a odstranění hráče je umožněno vždy viditelným tlačítkem (Obrázek 16).



Obrázek 16 Opravený detail kampaně

5.2.2 Stránka pro soubojový scénář

Stránka pro soubojový scénář je rozdělena na dvě stránky. První stránka, na kterou uživatel narazí po vytvoření scénáře je stránka pro připravení souboje, kde je možné vložit hráče nebo monstrem, kteří se souboje budou účastnit (Obrázek 18).

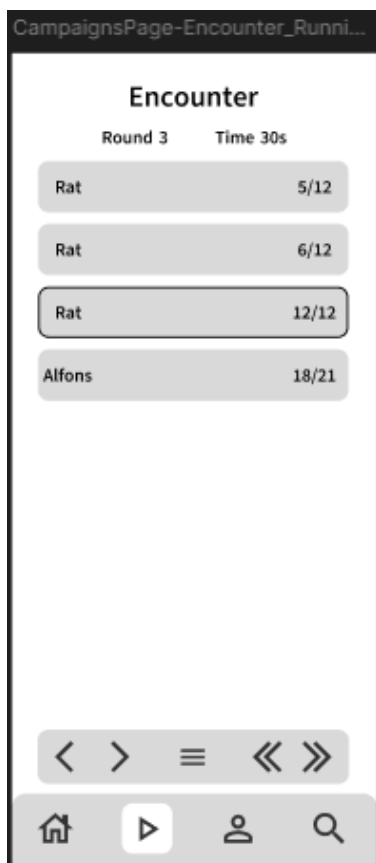
Při kliknutí na červené tlačítko se zobrazí seznam monster z API s vyhledávacím polem. Při kliknutí na tlačítko u jednotlivých monster se konkrétní monstrem přidá do listu účastníků. Obdobně je to u modrého tlačítka, kdy se zobrazí výběr z hráčských postav, kteří jsou v kampani obsažení (Obrázek 19).

Pokud je uživatel spokojený s výběrem účinkujících, může kliknout na zelená tlačítko „Start“, které ho přesune na další stránku, kde už souboj probíhá (Obrázek 18).

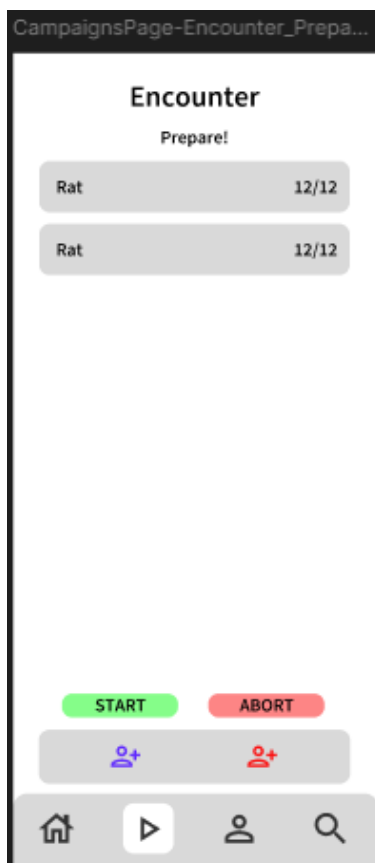
Tahle stránka obsahuje všechny účastníky, kteří jsou zobrazení v pořadí, podle automatizovaného hodu na iniciativu, kdy účinkující s největší iniciativou začínají. Ve spodní části stránky se nachází nástroj pro správu jednotlivých kol a kroků. Uživatel se

v souboji může pohybovat dopředu nebo dozadu, kdykoli je hra posunuta, tak se zvýrazní účastník, který je právě na tahu (Obrázek 17).

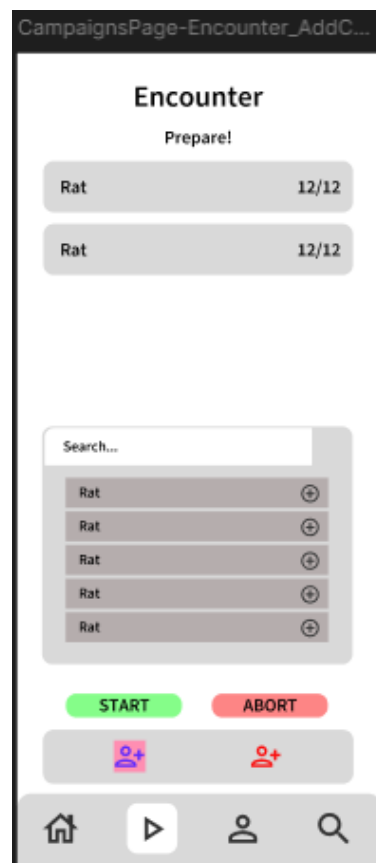
Nad listem účastníku je ukázaný počet kol a čas, který reprezentuje fiktivní herní dobu v rámci herního světa. Podle pravidel se jedno kolo odehraje v rámci šesti sekund v herním světě [25, s. 189], v návrhu je hodnota čistě fiktivní (Obrázek 17).



Obrázek 17 Průběh souboje



Obrázek 18 Příprava souboje



Obrázek 19 Příprava souboje s volbou účinkujících

V tomto přístupu se nachází více problému. Tlačítka, která přidávají účastníky, fungují stejně, jen berou data z jiných zdrojů. Jelikož v kampaních nejsou desítky hráčských postav, je možné volbu přesunout jako list postav pod seznam účastníků, kde stačí kliknout na zvolenou postavu, která se hned přidá do souboje (Obrázek 20).

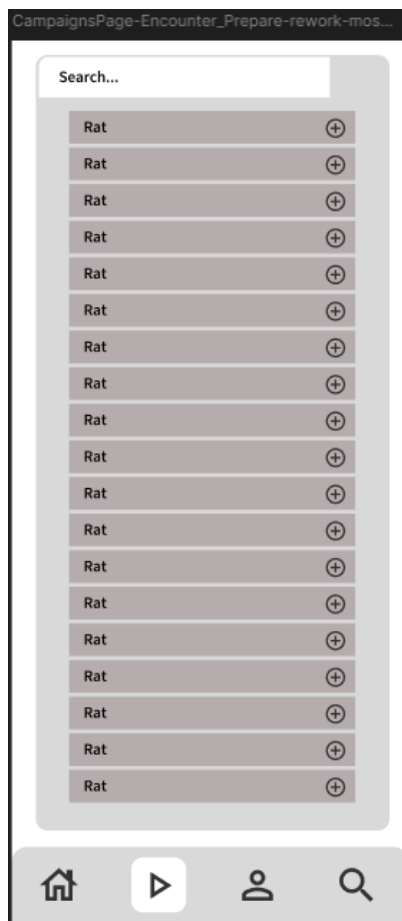
Pro monstra tlačítko zůstalo, ale přesunulo se společně se „Start“ tlačítkem pod list účastníků. Tlačítko „Abort“ není potřeba, jelikož se ze souboje lze dostat skrz spodní

navigaci nebo přidaným tlačítkem „<“ v horním rohu, které přesune uživatele zpět na kampaň (Obrázek 20).

Okno pro výběr monster bylo nutné rozšířit, aby byla práce s velkým množstvím dat příjemnější. Vzhledem k tomu, že API obsahuje stovky záznamů, bylo důležité zajistit více prostoru pro lepší orientaci a efektivnější vyhledávání (Obrázek 21).



Obrázek 20 Opravená příprava souboje

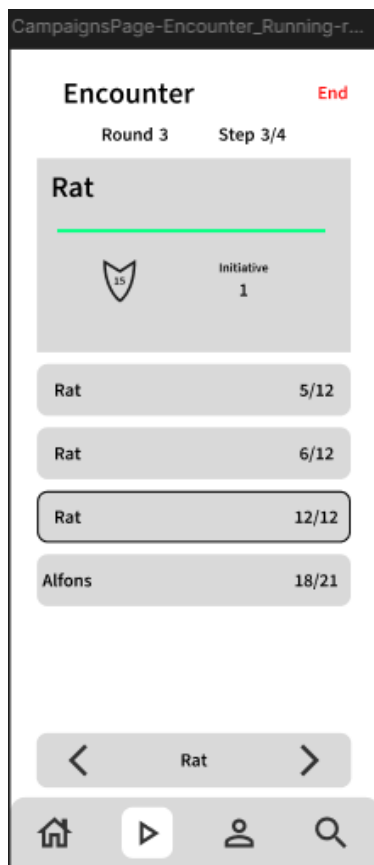


Obrázek 21 Rozšířený list pro volbu monster

Stránku pro průběh souboje bylo potřeba taky upravit. Spodní nástroj pro operování souboje se výrazně zjednodušilo. Menu, kvůli menšímu počtu funkcí, které by obsahovalo, se odstranilo spolu s šípkami, které přeskakovaly celé kolo, zůstali pouze tlačítka pro skok mezi kroky.

Dále přibyl detail pro účastníka, který je právě na řadě, a tlačítko „End“ pro ukončení souboje.

Počítání času se jevilo zbytečné, jelikož by pokaždé inkrementovalo o 10 a z hlediska mých herních zkušeností, není čas tolik zásadní element, který by musel být neustále k dispozici. Proto se nahradil jednoduchým počítáním kroků.



Obrázek 22 Opravená stránka pro průběh souboje

5.3 Stránka pro detail postavy

Stránka se soustředí na rozložení obsahu podle nejvyšší potřeby. V hlavičce jsou základní informace a životy společně s bonusovými životy. Níže potom výpis atributů s modifikátory postavy jako síla, obratnost atd., pod nimi další potřebné informace jako „Armor class“, iniciativa, rychlost a „Proficiency bonus“ (Obrázek 23).

Potom se stránka štěpí do sekcí, které jsou koncipovány jako „dropdown“, každá by řešila např. dovednosti, vybavení, podrobnější popis postavy atd. (Obrázek 23)

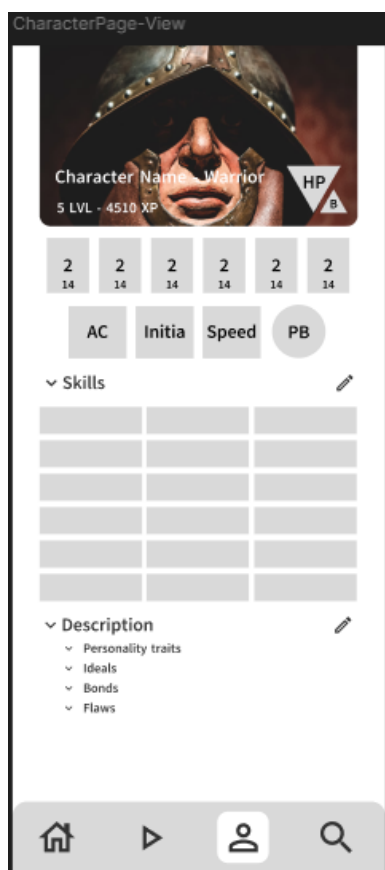
Tento způsob struktury však není dobrý pro přehlednost, jelikož by uživatel musel sjíždět poměrně daleko pro určité informace. Když má uživatel spoustu věcí ve vybavení

může se cestování po stránce zkomplikovat. Proto bylo nutné rozložit sekce do menu, kdy se zobrazí pouze konkrétní část obsahu, který je přehlednější (Obrázek 24).

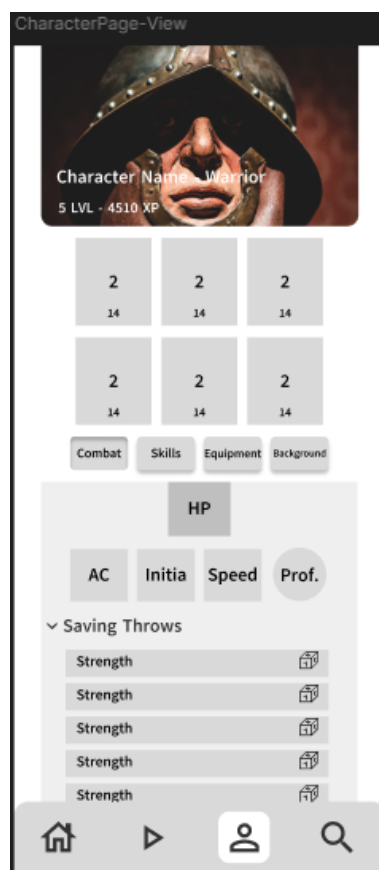
Dále bylo potřeba zpřehlednit seznam atributů, které byly v jednom řádku poměrně malé a nevýrazné, proto se zvětšily a rozložily do dvou řádků (Obrázek 24).

Bonusové životy se zdály být zbytečné, nejedná se o častou herní mechaniku, která by ovlivňovala všechny hráče, proto byl jejich náhled odstraněn a o jejich zobrazení se stará vizuální změna zobrazení klasických životů (například barevně) (Obrázek 24).

Stejný problém jako u atributů nastával i u dalších prvků jako dovednosti, záchranné hody nebo rozložení vybavení v inventáři, kdy bylo rozložení do více sloupců nepřehledné, proto se zobrazení dále koncipovalo do maximálně dvou sloupců v závislosti na obsahu (Obrázek 24).



Obrázek 23 Detail postavy

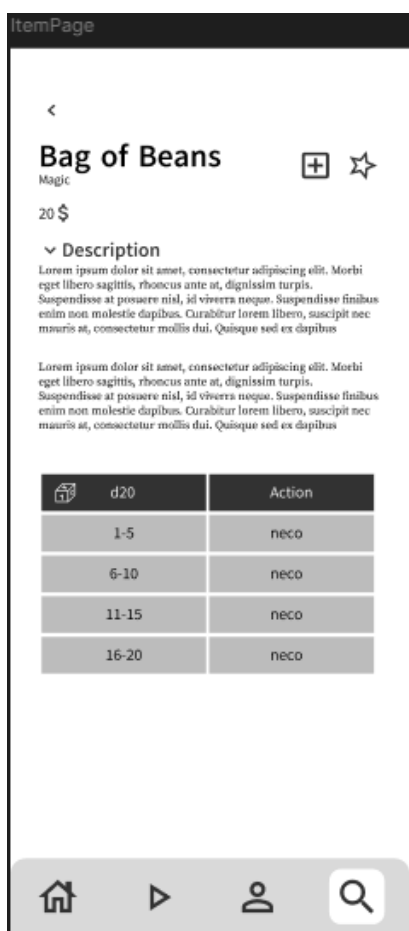


Obrázek 24 Opravený detail postavy

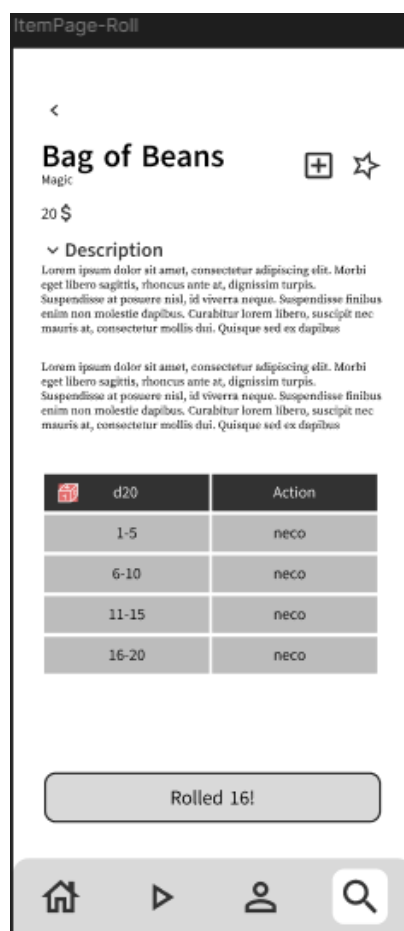
Pro určité elementy, na které se v rámci hry hází kostkou, byly přidány tlačítka s ikonkou kostky, které provedou příslušný hod (Obrázek 24).

5.4 Stránka pro detail vybavení

Důležitou součástí stránky, mimo správného zobrazení informací o vybavení, byla možnost zobrazit interaktivní tabulky pro hod kostkou, která z tabulky možných výsledků vyhodnotí konkrétní akci, která se má provést (Obrázek 25). Výsledek se zobrazí pomocí alertu (Obrázek 26).

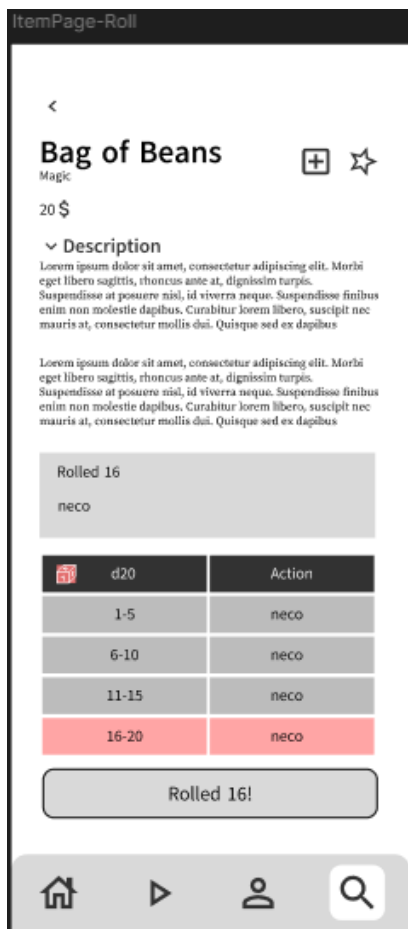


Obrázek 25 Detail vybavení



Obrázek 26 Detail vybavení s hodem na tabulku

Problém mohl nastat, kdyby bylo v textu více tabulek, tak by nebylo přehledné, na jakou tabulku byl hod proveden. Proto bylo nutné přidat označení řádku, na který hozená hodnota spadá a pro lepší čitelnost výsledku byla nad konkrétní tabulku přidána karta s výsledkem hodu a akcí (Obrázek 27).

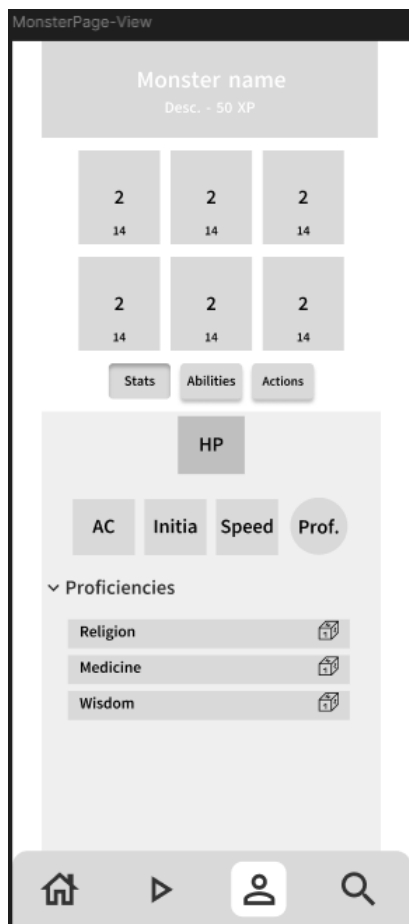


Obrázek 27 Opravená interakce s tabulkou

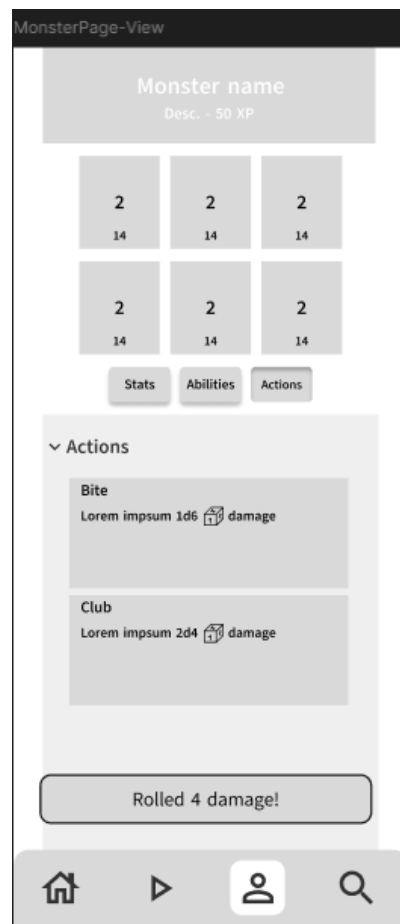
5.5 Stránka pro detail monster

Stránka pro zobrazení monster vychází vizuálně ze stránky pro detail postavy s několika změnami. Menu pro jednotlivé sekce čítá jen tři záložky pro statistiky, schopnosti a možné akce příšery (Obrázek 28).

Důležitou součástí, která se projevuje například i u stránky vybavení, je možnost házet kostkou přímo v rámci popisu akce, schopnosti atd., což vytváří větší interaktivnost s aplikací (Obrázek 29).



Obrázek 28 Detail monstra

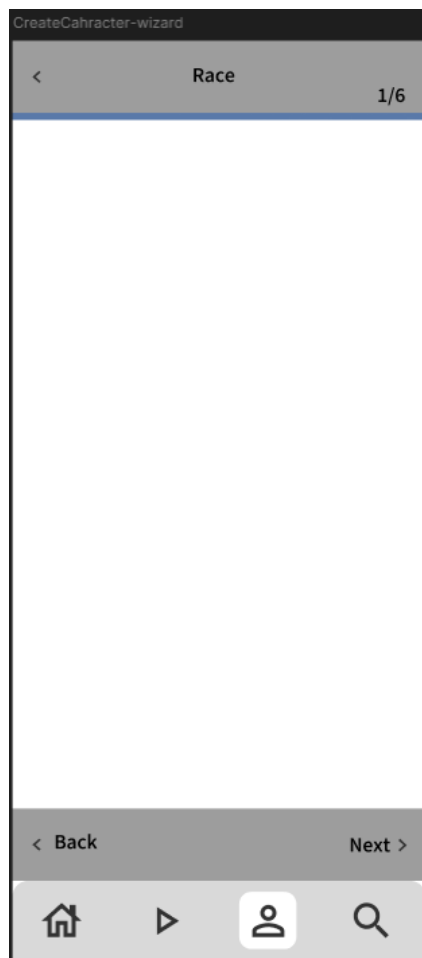


Obrázek 29 Hod kostkou na akci monstra

5.6 Stránka pro vytváření postavy

Tato stránka funguje jako klasický Wizard. Tedy asistent pro vytváření postav, který se dělí na několik kroků. V rámci mé aplikace je tvorba rozdělena do šesti částí – rasa, třída, charakter postavy, detail, atributy a souhrn.

Každá část je vytvořena separátně od rozhraní Wizardu a je jen vložena do rozhraní u příslušného kroku.

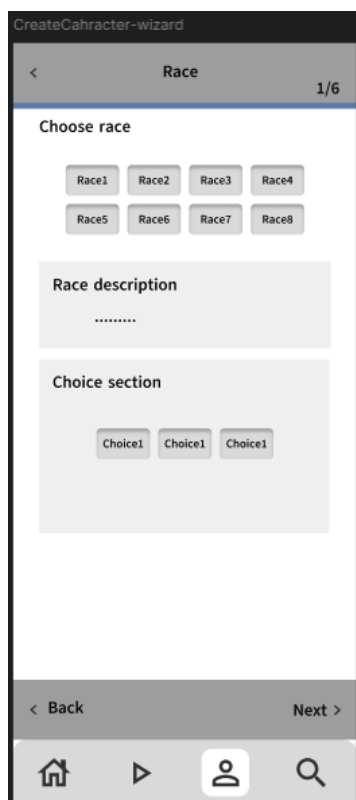


Obrázek 30 Wizard pro vytváření postavy

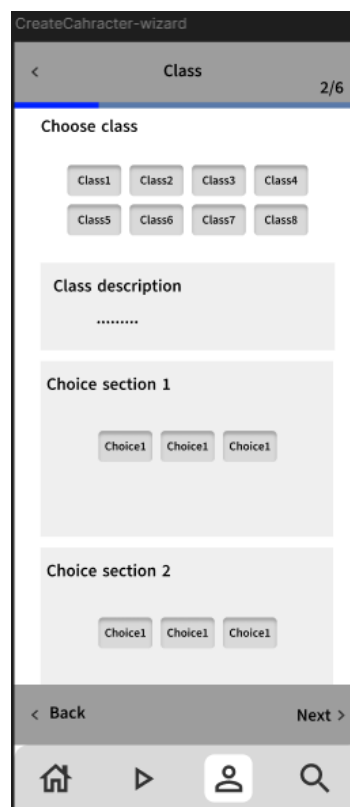
Zde se po jednotlivých krocích zobrazují jednotlivé sekce. Uživatel si v rámci sekcí musí zvolit jednu nebo více možností, aby se mohl posunout do dalšího kroku.

V kroku pro zvolení atributů musí hodit v každé ze šesti karet čtyřmi kostkami (kostka s nejnižší hozenou hodnotou se nezapočítá), aby se vyhodnotila finální hodnota dané karty, každou z hodnot potom přiřadí jednomu z šesti atributů (Obrázek 35).

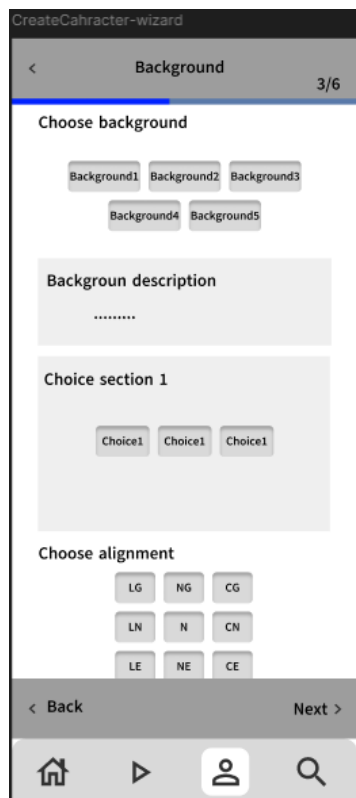
Ve finálním kroku uživatel zkontroluje navolenou postavu a pojmenuje ji (Obrázek 36).



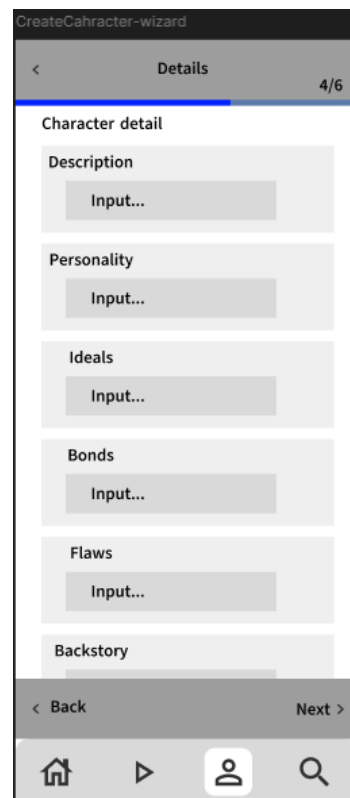
Obrázek 31 Volba rasy



Obrázek 32 Volba třídy



Obrázek 33 Volba charakteru postavy



Obrázek 34 Volba detailů

CreateCahracter-wizard

< Abilities 5/6

Ability scores

 0	 0	 0
 0	 0	
 0	 0	 0
 0	 0	
 0	 0	 0
 0	 0	

...

Assign Scores

STR	select v	INT	select v
DEX	select v	WIS	select v
CON	select v	CHA	select v

< Back Next >

Obrázek 35 Volba atributů

CreateCahracter-wizard

< Finish 6/6

Character summary

2 14	2 14	2 14
2 14	2 14	2 14

Stats

- combat statistics
- > Skills
- > Proficiencies
- > Languages
- > Starting equipment
- > Character details
- > Traits and features

Character name

Name here...

< Back Finish >

Obrázek 36 Rekapitulace a volba jména

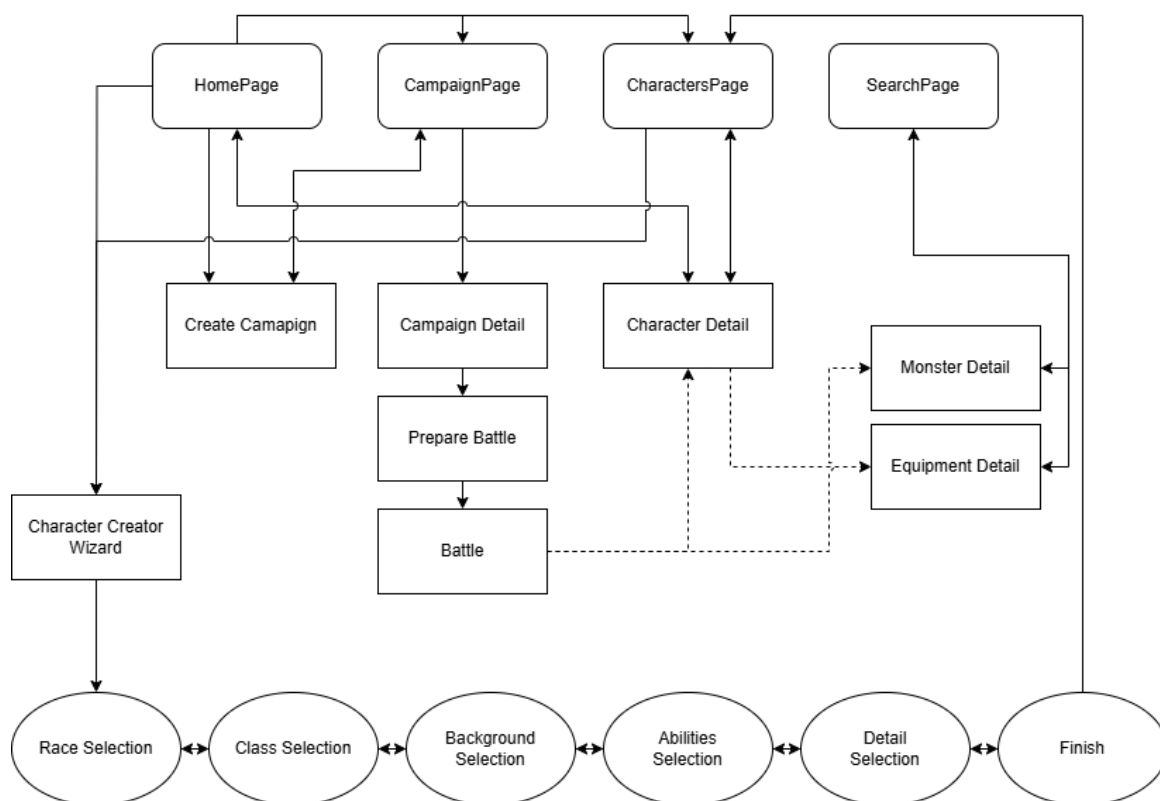
6 MAPA APLIKACE

Následující diagram (Obrázek 37) znázorňuje základní strukturu navigace v aplikaci. Hlavní rozhraní tvoří čtyři klíčové obrazovky dostupné přes spodní navigační lištu: HomePage, CampaignPage, CharactersPage a SearchPage.

Z těchto obrazovek se uživatel může dále dostat na podstránky:

- Detail kampaně (Campaign Detail), příprava bitvy (Prepare Battle) či samotná bitva (Battle).
- Detail postavy (Character Detail).
- Detaily entit z API, jako jsou monstra nebo vybavení (Monster Detail, Equipment Detail).
- Průvodce tvorbou postavy (Character Creator Wizard), který se skládá z jednotlivých kroků.

Prerušované šipky v diagramu označují odkazy vedoucí k náhledovým (např. detaily entit z API), které nepředstavují součást hlavního navigačního toku, ale zobrazují konkrétní informace podle výběru uživatele.



Obrázek 37 Mapa navigace v aplikaci

7 PROCES VÝVOJE A PRÁCE S DATY Z API

Po dokončení návrhu aplikace následovala samotná implementace, jejímž cílem bylo správně uchopit vizuální koncept a vytvořit funkční mobilní aplikaci. Klíčovou součástí bylo napojení na veřejnou D&D API, která poskytuje data ve formátu JSON (JavaScript Object Notation). Tato data bylo potřeba efektivně získat, zpracovat a správně zakomponovat do datové struktury aplikace ve formátu vhodném pro TypeScript.

V této kapitole je popsán způsob komunikace s API, struktura použitých datových typů, a také proces transformace a filtrace dat.

7.1 Práce s daty a datová vrstva

Jádrem aplikace je integrace s veřejně dostupným D&D API [21], které poskytuje rozsáhlé množství datových entit, jako jsou monstra, vybavení, povolání nebo pravidla. API vrací data ve formátu JSON, přičemž každý typ entity má vlastní strukturu. Pro efektivní práci s těmito daty v prostředí Angular a TypeScript bylo nutné provést několik kroků:

- Navrhnout odpovídající interface pro jednotlivé datové typy, které slouží jako statická typová kontrola.
- Vytvořit služby pro stahování dat pomocí HTTP požadavků.
- Zajistit správné zpracování a případnou úpravu struktury JSON při příjmu dat – a to jak kvůli typové přesnosti, tak kvůli optimalizaci pro použití v komponentách aplikace.

V následujících částech je popsán konkrétní způsob, jakým byla data získávána, jak vypadá výstup z API, jak probíhala jejich transformace, a proč bylo potřeba vytvořit vlastní skript pro filtrování a předzpracování těchto dat.

7.1.1 Struktura odpovědi z API a návrh kořenových rozhraní.

API dostupné na adrese <https://www.dnd5eapi.co/api/2014/> poskytuje základní rozdělení do několika entit ve formátu JSON. Tento výstup obsahuje klíče, které reprezentují jednotlivé kategorie dat (např. monsters, equipment, races) a zároveň pro každou kategorii nabízí relativní cestu pro získání konkrétních dat:

```
{
  "monsters": "/api/2014/monsters",
  "equipment": "/api/2014/equipment",
  ...
  "races": "/api/2014/races",
  ...
}
```

Každý tento endpoint po zavolání vrátí seznam položek dané kategorie. Například `/api/2014/equipment` vrátí následující JSON strukturu:

```
{
  "count": 237,
  "results": [
    {
      "index": "abacus",
      "name": "Abacus",
      "url": "/api/2014/equipment/abacus"
    },
    {
      "index": "acid-vial",
      "name": "Acid (vial)",
      "url": "/api/2014/equipment/acid-vial"
    },
    ...
  ]
}
```

Proto bylo nutné navrhnout základní TypeScript interface, který definuje obecnou strukturu odpovědi. Tento interface byl opakovaně používán napříč službami aplikace pro různé entity:

```
// Obecná odpověď pro endpointy
export interface Root {
  count: number
  results: SearchResult[]
}

// Výstup jednoho záznamu
export interface Item {
  index: string
  name: string
  url: string
}
```

Pro flexibilnější práci s daty z hlediska kategorizace a vyhledávání, byl přidán prostředník, kterému byl přidán klíč „source“ pro jednoduché rozčlenění dat.

```
export interface SearchResult extends Item {  
    source: string  
}
```

7.1.2 Práce s prostředím a konfigurací API

Během vývoje aplikace nebylo využíváno veřejné API dostupné online, ale jeho lokální instance běžící na adrese localhost. Pro oddělení prostředí vývoje a produkce byly v rámci frameworku Ionic využity takzvané environmentální konfigurace, které jsou definovány ve dvou samostatných souborech: environment.ts pro vývojové prostředí a environment.prod.ts pro produkční nasazení.

Vzhled těchto konfiguračních souborů je následující:

Vývojové prostředí (environment.ts):

```
export const environment = {  
    production: false,  
    api_root_url: 'http://localhost:3000/api',  
    api_url: 'http://localhost:3000',  
};
```

Produkční prostředí (environment.prod.ts):

```
export const environment = {  
    production: true,  
    api_root_url: 'https://www.dnd5eapi.co/api',  
    api_url: 'https://www.dnd5eapi.co',  
};
```

Při buildu aplikace dochází automaticky k nahrazení vývojového souboru environment.ts za jeho produkční verzi environment.prod.ts, tím se zajistí správné přesměrování požadavků na online dostupnou verzi API.

7.1.3 Načítání a zpracování dat z API

V této části je popsána konkrétní implementace načítání dat z API a jejich zpracování do předem definovaných datových struktur. Jako příklad slouží metoda pro získání všech položek z kategorie equipment, která využívá službu HttpClient a rozhraní Observable z knihovny RxJS.

Metoda getAllEquipment() zavolá příslušný endpoint API a výstup uloží do rozhraní Root. Každému načtenému prvku je následně doplněn atribut source, který určuje původ položky, pro pozdější rozdělení při filtraci:

```
getAllEquipment(): Observable<Root> {
    return this.http.get<Root>(`${API_URL}${BASE_EQUIPMENT_URL}`).pipe(
        map(response => ({
            count: response.count,
            results: response.results.map(item => ({
                ...item,
                source: "equipment"
            }))
        )))
    );
}
```

Podobná implementace je definována i pro další entity jako například monsters nebo magic-items.

Pro získání konkrétních objektu příslušné kategorie jsou potom definovány dvě funkce. Jedna getEquipment(equipment_url: string), která pracuje s adresou objektu a druhá getEquipmentByIndex(index: string), která pracuje pouze s indexem objektu.

```
getEquipment(equipment_url: string = "/api/equipment/abacus") {
    return this.http.get(`${API_URL}${equipment_url}`);
}

getEquipmentByIndex(index: string) {
    return this.http.get(`${API_URL}/api/equipment/${index}`);
}
```

Obdobná implementace je i v dalších službách aplikace. Výjimkou pro EquipmentService je funkce getAll(), která zkombinuje výstup z getAllEquipment() a getMagicItems():

```

getAll(): Observable<Root> {
    return forkJoin({
        equipment: this.getAllEquipment(),
        magicItems: this.getMagicItems()
    }).pipe(
        map(result => {
            // Combine results from both endpoints
            return {
                count: (result.equipment.count || 0) + (result.magicItems.count || 0),
                results: [
                    ...(result.equipment.results || []),
                    ...(result.magicItems.results || [])
                ]
            };
        })
    );
}

```

7.1.4 Načítání a práce s daty ve službách a komponentách

Na základě výše popsané struktury odpovědi z API a navržených rozhraní je možné jednoduše zpracovávat data přímo v komponentách aplikace. Jako příklad konkrétního využití lze uvést komponentu stránky Search, kde dochází k načítání dat z více kategorií prostřednictvím služeb EquipmentService a MonsterService.

Klíčová metoda loadData() zajišťuje načtení dat a jejich následné sloučení do jedné výsledkové struktury allResults, která je dále využita například pro vyhledávání či zobrazení výsledků. Při implementaci je rovněž ošetřena případná chyba během načítání a podporováno tzv. infinite scroll načítání obsahu.

```

loadData(event?: InfiniteScrollCustomEvent) {
    this.error = null;
    this.allResults = [];
    this.searchResults = [];
    this.currentPage = 0;

    if (!event) {
        this.isLoading = true;
    }
}

```

```
forkJoin({
  equipment: this.equipmentService.getAll(),
  monsters: this.monsterService.getMonsters()
}).pipe(
  finalize(() => {
    this.isLoading = false;
    if (event) {
      event.target.complete();
    }
    // Load first chunk of results
    this.loadMoreResults();
  }),
  catchError((err) => {
    console.log(err);
    this.error = err;
    return of({ equipment: { results: [] }, monsters: { results:
[] } });
  })
).subscribe({
  next: (results) => {
    // Store all results
    if (results.equipment && results.equipment.results) {
      this.allResults.push(...results.equipment.results);
    }
    if (results.monsters && results.monsters.results) {
      this.allResults.push(...results.monsters.results);
    }
  }
});
}
```

7.1.5 Navigace a načtení detailních dat

Navigaci na detail konkrétní entity zajišťuje Angular Router. K přechodu na detailovou stránku (např. monster-detail) dochází pomocí metody navigate, která zároveň předává stav (data a typ zdroje):

```
this.router.navigate(['/tabs/search/monster-detail', item.index], {
  state: {
    item: item,
    sourceType: item.source
  }
});
```

Na cílové stránce (např. monster-detail) je poté entita načtena dle parametru index pomocí příslušné služby a uložena do konkrétního rozhraní:

```
loadMonster() {
  if (!this.monsterId) return;

  this.isLoading = true;

  this.monstersService.getMonster(this.monsterId).subscribe({
    next: (monster: Monster) => {
      this.monster = monster;
      this.isLoading = false;
    },
    error: (err) => {
      console.error('Error loading monster:', err);
      this.error = 'Failed to load monster';
      this.isLoading = false;
    }
  });
}
```

7.1.6 Struktura rozhraní

Pro zachycení struktury odpovědí z API byla vytvořena rozhraní, která zajišťují typovou bezpečnost napříč aplikací. Rozhraní jsou rozdělena do souborů podle kategorie používání a exportována pro další použití.

Například základní rozhraní pro vybavení (BaseEquipment) vypadá následovně:

```
export interface BaseEquipment extends SearchResult {
  desc: string[]
  special: string[]
  equipment_category: Item
  gear_category?: Item
  cost: Cost
  weight: number
}
```

```
    contents: Content[]
    properties: Item[]
}
```

Z tohoto rozhraní následně dědí konkrétní typy jako `Armor` nebo `Weapon`, které rozšiřují základní strukturu o specifické vlastnosti:

```
export interface Armor extends BaseEquipment {
    armor_category: string
    armor_class: ArmorClass
    equipment_category: Item
    stealth_disadvantage: boolean
    str_minimum: number
}

export interface Weapon extends BaseEquipment {
    category_range: string
    damage: Damage
    range: Range
    throw_range: ThrowRange
    two_handed_damage: TwoHandedDamage
    weapon_category: string
    weapon_range: string
}
```

Podobná rozhraní byla vytvořena také pro další typy dat – například postavy (`Character`) nebo kampaně (`Campaign`).

8 ARCHITEKTURA APLIKACE A IMPLEMENTACE KOMPONENT

Tato kapitola se věnuje podrobnějšímu popisu struktury aplikace, jejích hlavních částí a klíčových komponent, které tvoří hlavní část funkcionality. Cílem je přiblížit, jak byly dílčí části navrženy a implementovány v rámci hlavních cílů práce.

Dále je uveden přehled hlavních služeb, komponent a nástrojů, které byly v rámci vývoje vytvořeny.

Jednotlivé části aplikace jsou organizovány do adresářů podle typu pod kořenovým adresářem `app/`:

- `pages/` - obsahuje jednotlivé stránky jako např. `search`, `characters`, `home` apod.,
- `services/` - zajišťují komunikaci s API a správu dat (např. `monster.service.ts`),
 - `interfaces` – obsahuje rozhraní pro práci s daty skrz aplikaci,
- `components/` - opakovaně použitelné komponenty UI (např. `dice-chip`, `interactive-table`)
- `pipes/` - vlastní transformace dat (pouze `dice-notation.pipe.ts`),
- `tabs/` - hlavní navigační komponent s přepínáním mezi jednotlivými sekcemi aplikace

8.1 Struktura aplikace a hlavní sekce

Aplikace je rozdělena do několika hlavních sekcí, které jsou přístupné skrz spodní tabovou navigaci (implementována v adresáři `/tabs/`). Každá sekce je implementována jako samostatná stránka s vlastní logikou a komponentami.

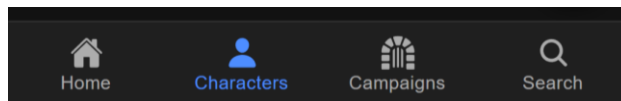
8.1.1 Navigační struktura (tabs)

Navigace mezi sekcemi je řešena pomocí Ionic Tabs s definovaným trasováním v souboru `tabs.routes.ts`, kde jsou specifikovány cesty a načteny komponenty jednotlivých stránek.

Ukázka routovací konfigurace:

```
export const routes: Routes = [
  {
    path: '',
    component: TabsPage,
    children: [
      {
        path: 'home',
        children: [
          {
            path: '',
            loadComponent: () =>
import('../pages/home/home.page').then(m => m.HomePage),
          }
        ]
      },
      ...
    ],
  },
];
```

Zde je vidět výsledek implementace tabů v aplikaci:



Obrázek 38 Hlavní navigace v aplikaci

8.1.2 Domovská stránka (Home)

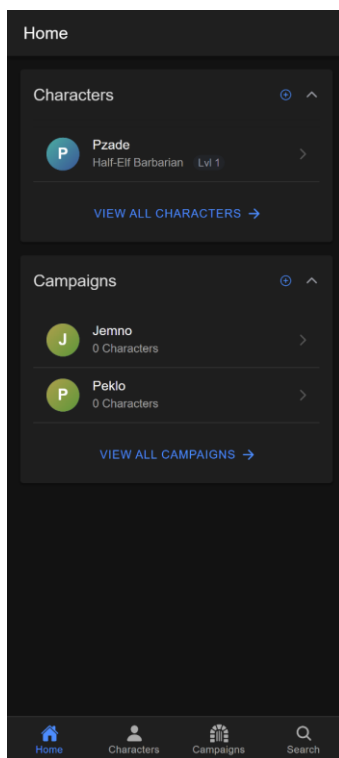
Stránka slouží jako úvodní přehled aplikace a rychlí přístup k nejčastěji používaným entitám v rámci kampaní a postav. Zobrazují se zde dvě samostatné karty pro postavy a kampaně, kdy je přes ně možné přejít do příslušných sekcí, k vytvoření nové kampaně či postavy nebo na detail dané entity. Důležitý je zde přehled a rychlí přístup po spuštění aplikace.

Pro zobrazování nejčastějších položek je implementovaná následující třídící metoda:

```
loadCampaigns() {  
  this.loadingCampaigns = true;  
  
  const sub = this.campaignService.campaigns.subscribe(campaigns =>  
  {  
    this.campaigns = campaigns  
    .sort((a, b) => {  
      const dateA = a.updatedAt || 0;  
      const dateB = b.updatedAt || 0;  
      return dateB - dateA;  
    })  
    .slice(0, 3);  
  
    this.loadingCampaigns = false;  
  });  
  
  this.subscriptions.add(sub);  
}
```

Metoda získá data ze služby CampaignService, rozřídí je podle data aktualizace a vezme z nich první tři záznamy. Obdobně funguje implementace pro postavy.

Zde je potom vidět hlavní stránka v aplikaci:



Obrázek 39 Hlavní stránka

8.1.3 Vyhledávání (Search)

Stránka pro vyhledávání umožňuje uživateli procházet a vyhledávat entity pocházející z několika kategorií dostupných přes API, v aplikaci jsou dostupné tři kategorie – vybavení, monstra a kouzelné předměty.

Hlavní součástí je načítání a zpracování dat, které je implementováno v metodě `loadData()`:

```
loadData(event?: InfiniteScrollCustomEvent) {
    this.error = null;
    this.allResults = [];
    this.searchResults = [];
    this.currentPage = 0;

    if (!event) {
        this.isLoading = true;
    }
    forkJoin({
        equipment: this.equipmentService.getAll(),
        monsters: this.monsterService.getMonsters()
    }).pipe(
        finalize(() => {
            this.isLoading = false;
            if (event) {
                event.target.complete();
            }
            this.loadMoreResults();
        }),
        catchError((err) => {
            console.log(err);
            this.error = err;
            return of({ equipment: { results: [] }, monsters: { results:
[] } });
        })
    ).subscribe({
        next: (results) => {
            // Store all results
            if (results.equipment && results.equipment.results) {
                this.allResults.push(...results.equipment.results);
            }
        }
    });
}
```

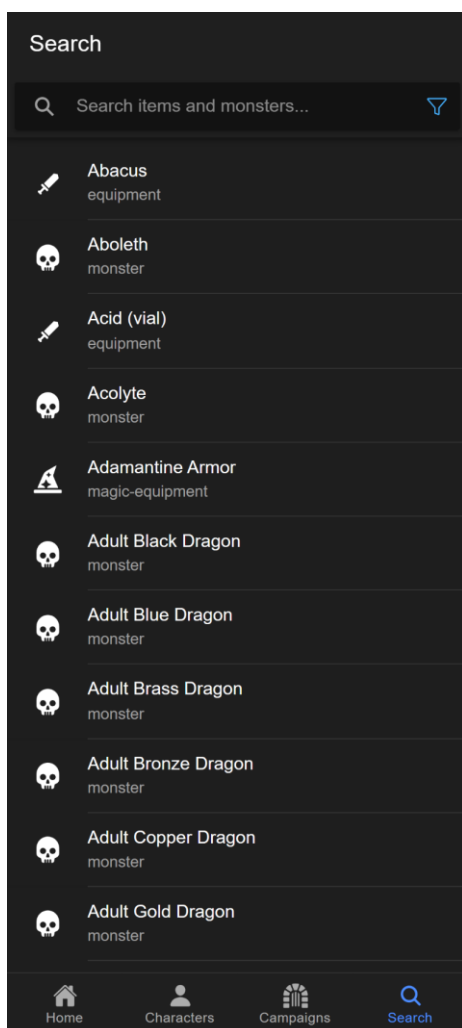
```
    if (results.monsters && results.monsters.results) {  
        this.allResults.push(...results.monsters.results);  
    }  
  
    this.allResults.sort((a, b) => a.name.localeCompare(b.name));  
}  
});  
}
```

Metoda prostřednictvím služeb `EquipmentService` a `MonstersService` načítá data pomocí `forkJoin`, z více různých API endpointů současně. Výsledky jsou poté sloučeny do jedné společné struktury `allResults`, ze které se dále čerpá pro zobrazení.

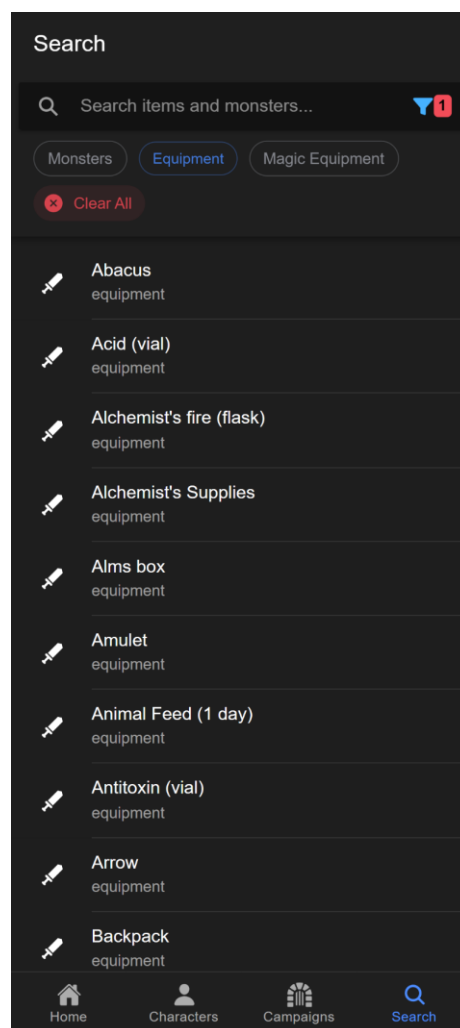
Vyhledávací pole umožňuje filtrování zobrazených výsledků podle názvu entity. Filtrování je realizováno v reálném čase, a tedy s každou změnou vstupu se upravuje zobrazený seznam výsledků. Každý výsledek v seznamu je klikací prvek, který uživatele přesměruje na detailní stránku dané entity pomocí Router služby. Při navigaci je navíc kromě identifikátoru entity předávána i informace o jejím původu (`source`), což umožňuje správné načtení detailu pomocí příslušné služby.

Vyhledávání je navrženo tak, aby bylo snadno rozšiřitelné o další typy entit v budoucnu (stačí přidat nový service a zapojit jej do `forkJoin`).

Dále je vidět náhled stránky bez a s filtrací:



Obrázek 40 Stránka vyhledávání



Obrázek 41 Stránka vyhledávání s filtrací

8.1.4 Stránka kampaní (Campaigns)

Stránka kampaní slouží uživateli k prohlížení, vytváření a správě vlastních herních kampaní. Základní obrazovka zobrazuje seznam všech kampaní uložených v paměti aplikace. Každá kampaň je reprezentována kartou obsahující její název, datum vytvoření, jméno správce kampaně a počet importovaných postav.

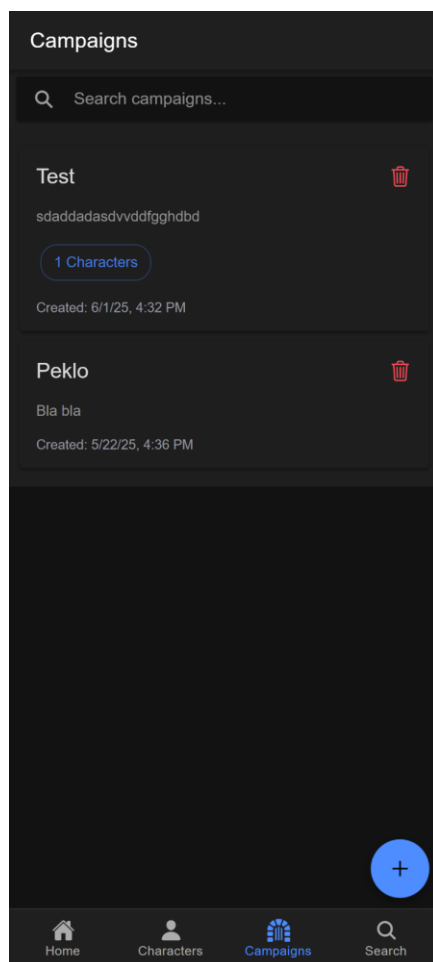
Uživatel může pomocí tlačítka (tzv. floating action button) přejít na stránku pro vytvoření nové kampaně nebo kliknutím na kampaň dojde k navigaci na detail kampaně, kde jsou zobrazeny podrobnosti o ní – například seznam přidružených postav.

Uživatel může vyhledávat kampaně pomocí jejich jména.

Zde je implementace pro načítání kampaní ze služby a navigace na detail kampaně:

```
loadCampaigns() {  
  this.isLoading = true;  
  this.subscription.add(  
    this.campaignService.campaigns.subscribe(campaigns => {  
      this.campaigns = campaigns;  
      this.filteredCampaigns = [...campaigns];  
      this.isLoading = false;  
    })  
  );  
}  
  
viewCampaign(id: string) {  
  this.router.navigate(['/tabs/campaigns', id]);  
}
```

Zde je náhled na stránku v aplikaci:



Obrázek 42 Stránka kampaní

8.1.5 Stránka postav (Characters)

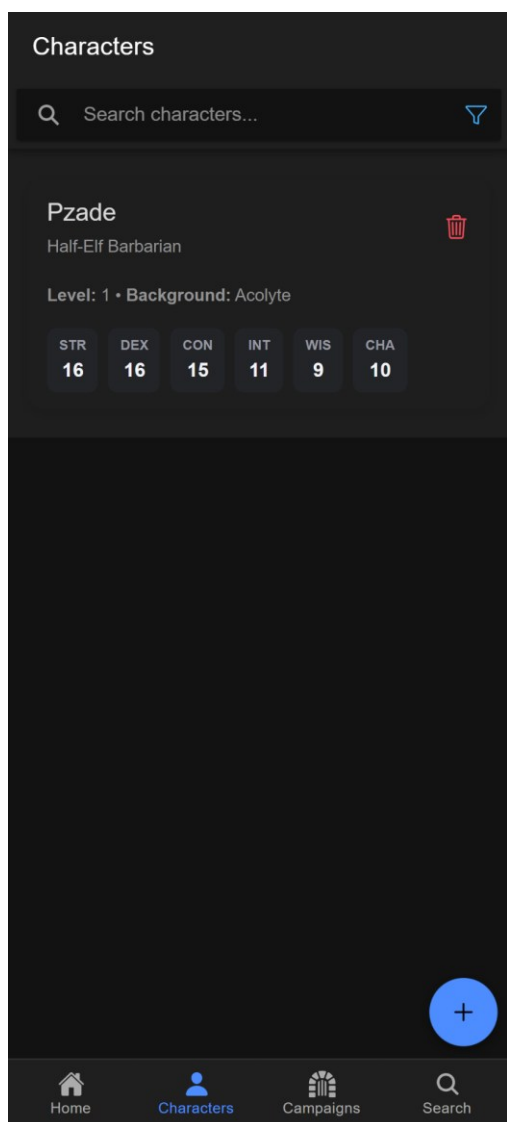
Stránka postav umožňuje uživateli zobrazovat, vytvářet a mazat vlastní herní postavy. Sekce zobrazuje seznam všech uložených postav formou karet s názvem postavy a stručnými informacemi, jako je rasa, třída, úroveň, zázemí a základní atributy postavy.

Stejně jako na stránce kampaní lze i zde vyhledávat postavy pomocí názvu, ale i pomocí filtrů tříd.

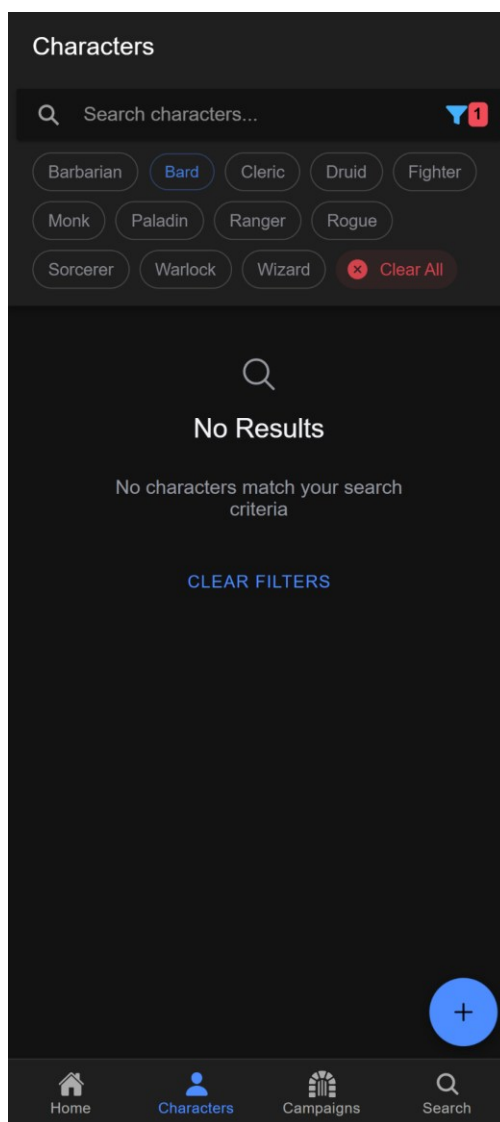
Oproti stránce kampaní se zde načítají kategorie pro filtrování (všechny dostupné třídy postav) z API, což je vyřešené následující metodou.

```
async loadCategories() {  
  try {  
    const classes = await  
    firstValueFrom(this.creatorService.getClasses());  
    console.log('Classes:', classes);  
    this.characterCategories = classes.map((cls: Item) => ({  
      value: cls.index.toLowerCase(),  
      label: cls.name  
    }));  
  } catch (error) {  
    console.error('Error loading categories:', error);  
  }  
}
```

Jelikož metoda používá `await` kvůli čekání na data z API, je nutné, aby byla asynchronní. Třídy se potom uloží do `characterCategories`, které jsou následně použity ve filtrech pro vyhledávání, jak je vidět na následujících obrázcích:



Obrázek 43 Stránka postav



Obrázek 44 Stránka postav s filtrací

8.2 Klíčové komponenty aplikace

Tahle část je zaměřena na popis klíčových funkcionalit aplikace, které přímo odpovídají požadavkům zadání práce a zároveň přináší uživatelsky významné přínosy. Místo popisu každé jednotlivé komponenty je kladen důraz na části týkající se řešením zajímavých problémů, které se při implementaci naskytly, a interaktivních herních prvků, které pozitivně ovlivnily chod aplikace a uživatelský zážitek.

8.2.1 Vytváření postav

Proces vytváření postavy je realizován pomocí služby character-creator, která je zodpovědná za načítání dat z API a zároveň za průběžné sledování a ukládání voleb uživatele během jednotlivých kroků tvorby postavy. Tato služba funguje jako centrální zdroj dat, ze kterého jednotlivé komponenty získávají potřebné informace a kam jsou ukládány výsledné volby.

V rámci aplikace je vytvořena stránka character-wizard, což je průvodce vytvářením postavy rozdělený do šesti kroků:

- Race (rasa)
- Class (povolání)
- Background (životní příběh/zázemí a tzv. „alignment“ – volba z devíti předvoleb, která určuje, jaký má postava charakter)
- Detail (základní údaje – cíl, popis, atd.)
- Abilities (házení kostkami na atributy postavy)
- Finish (shrnutí, pojmenování a uložení postavy)

Každý krok je reprezentován vlastní komponentou, která je dynamicky vkládána do wizardu přes router-outlet.

V průběhu vyplňování wizardu je zajištěno, že uživatel vyplní všechna povinná pole (například výběr rasy nebo povolání, zadání jména), což je validováno přímo v dané komponentě. Pokrok uživatele je vizualizován pomocí progres baru, který ukazuje, kolik kroků je dokončeno.

Po dokončení posledního kroku jsou data uložena podle předem definovaného rozhraní, které zahrnuje všechny atributy postavy jako například jméno, rasa, povolání, vlastnosti, vybavení atd.

Zde je část kódu pro validaci všech povinných voleb, které musí uživatel udělat:

```
private checkRequirements() {  
  // Must have selected a race  
  const raceSelected = !!this.selectedRace;  
  
  // Must have selected required languages if applicable  
  let languagesSelected = true;  
  if (this.selectedRaceDetail?.language_options) {
```

```

        const requiredCount =
        this.selectedRaceDetail.language_options.choose;

        languagesSelected = this.selectedLanguageIndexes.length >=
        requiredCount;
    }

    let bonusSelected = true;
    if (this.selectedRaceDetail?.ability_bonus_options) {
        const requiredCount =
        this.selectedRaceDetail.ability_bonus_options.choose;
        bonusSelected = this.selectedAbilityBonusIndexes.length >=
        requiredCount;
    }

    let proficiencySelected = true;
    if (this.selectedRaceDetail?.starting_proficiency_options) {
        const requiredCount =
        this.selectedRaceDetail.starting_proficiency_options.choose;
        proficiencySelected = this.selectedProficiencyIndexes.length
        >= requiredCount;
    }

    // Combine all requirements
    this.allRequirementsMet = raceSelected && languagesSelected &&
    bonusSelected && proficiencySelected;

    // Update the wizard
    this.updateStepValidity();
}

private updateStepValidity() {
    this.wizard.updateStepValidity('race', this.allRequirementsMet);
}

```

Kdykoli se provede nějaké změna nebo se krok právě načetl, tak se zavolá metoda `checkRequirements`, která zkontroluje všechna povinná pole, aktualizuje proměnnou `allRequirementsMet` a zavolá metodu `updateStepValidity`, které spravuje wizarďa, pokud je vše v pořádku wizarď umožní uživateli přejít na další krok.

Obdobná implementace je napříč komponentami všech kroků.

Důležitou součástí tvorby postavy je výběr mezi několika možnostmi jako výběr dvou dovedností nebo výběr začátečního vybavení. Občas je výběr jednoduchý (např. vyber dvě

možnosti), ale občas je volba vnořená (např. vyber mezi dvěma, ale jedna z nich umožní další rozhodnutí mezi více možnostmi). Proto bylo nutné vytvořit komponent, který by tento problém řešil skrz všechny kroky wizardu.

V rámci komponentu option-selector bylo nutné vytvořit tzv. input dekorátory, které ovlivňují chování komponenty.

```
@Input() title: string = 'Select Options';
@Input() description: string = '';
@Input() required: boolean = true;
@Input() multiple: boolean = false;
@Input() chooseCount: number = 1;
@Input() options: OptionChoice[] = [];
@Input() selectedValues: string[] = [];
@Input() displayMode: 'select' | 'checkboxes' | 'chips' | 'equipment'
  = 'select';
@Input() placeholder: string = 'Select options...';
```

Například chooseCount ovlivňuje, kolik možností se musí zvolit, displayMode ovlivňuje, jak bude volba reprezentována v UI, required jestli je volba povinná a options je pole struktury, která popisuje, o jakou volbu se jedná. Zde je rozhraní pro danou strukturu:

```
export enum EquipmentOptionType {
  Standard = 'standard',
  CountedReference = 'counted_reference',
  Multiple = 'multiple',
  Choice = 'choice'
}

export interface OptionChoice {
  index: string;      // Unique identifier
  name: string;       // Display name
  value: any;         // The actual object
  disabled?: boolean; // Whether this option can be selected
  type?: EquipmentOptionType; // Added for equipment options
  count?: number;     // For CountedReference type
  items?: any[];      // For Multiple type options
  category?: any;     // For Choice type options
  subChoiceSelection?: string; // name of the selected subchoice
  subChoices?: OptionChoice[]; // available subchoices
}
```

Hlavní funkcí tohoto rozhraní je rozlišovat mezi typy voleb. Někdy je možné zvolit vybavení, které je buď předáno v rámci JSON objektu nebo je předána pouze kategorie, ze které lze potom dané možnosti vytáhnout.

- Standard je klasická volba, uživatel ji vybere a její hodnota se zvolí.
- CountedReference je možnost, která předá item (třeba sekeru), ale k ní ještě počet (například 1x sekara).
- Multiple je možnost, kdy z jedné volby dostane uživatel více itemů (například si může zvolit zbraň, a ještě k tomu dostane jeden štít).
- Choice je možnost, která dovolí volbu z kategorie v API např. kategorie „martial-melee-weapons“.

Taková možnost může v JSON formátu vypadat následovně:

```
{
  "option_type": "choice",
  "choice": {
    "desc": "any martial melee weapon",
    "choose": 1,
    "type": "equipment",
    "from": {
      "option_set_type": "equipment_category",
      "equipment_category": {
        "index": "martial-melee-weapons",
        "name": "Martial Melee Weapons",
        "url": "/api/equipment-categories/martial-melee-weapons"
      }
    }
  }
}
```

Nasazení takové komponenty může vypadat v souboru HTML následovně:

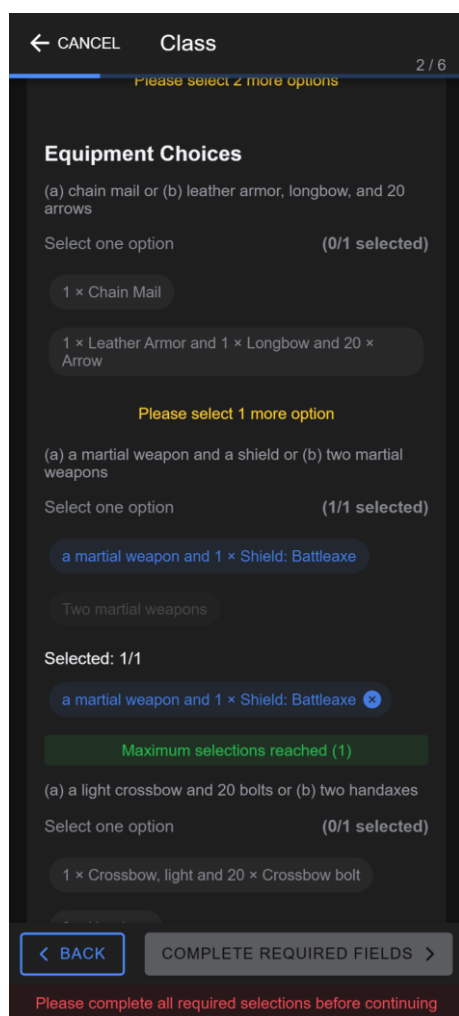
```
<app-option-selector
  title="Choose Proficiencies"
  [description]="getProficiencyChoiceText()"
  [multiple]="
    selectedRaceDetail.starting_proficiency_options.choose > 1"
  [chooseCount]="
    selectedRaceDetail.starting_proficiency_options.choose"
```

```

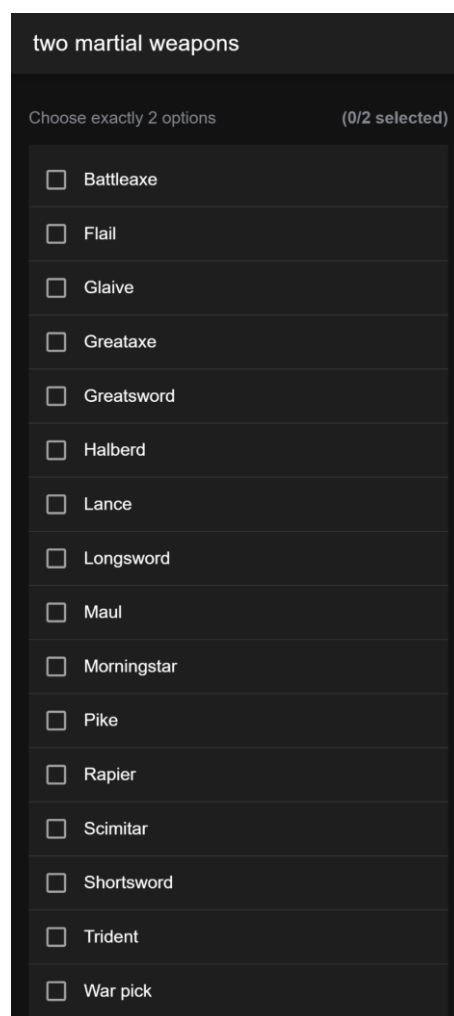
[options]="getProficiencyOptions()"
[selectedValues]="selectedProficiencyIndexes"
(selectionChange)="onProficiencySelectionChange($event)"
displayMode="chips"
>
</app-option-selector>

```

V UI se komponent zobrazuje jako sekce, kde je popis volby, indikátor, kolik voleb lze učinit, a ion-chipy, které reprezentují možnosti (Obrázek 45). U možností jako Choice, se po zvolení otevře modul s dalším výběrem (Obrázek 46).



Obrázek 45 Vizualizace OptionSelector komponentu



Obrázek 46 Možnost Choice v rámci OptionSelector komponentu

8.2.2 Interaktivnost aplikace

Pro uživatelské pohodlí a větší interaktivnost aplikace byl navržen systém pro parsování textu tzv. description-renderer, který prochází textová pole (například popisy u entit jako vybavení nebo magické předměty) a snaží se detekovat, zda obsahují tabulku určenou k náhodnému hodu kostkami.

Tento mechanismus umožňuje uživatelům rychle simulovat výběr z předdefinovaných možností podle hodu (např. 1d100). Příklad výstupu z API v JSON formátu:

```
[
  "Wondrous item, rare",
  "Inside this heavy cloth bag are 3d4...",
  "If you dump the bag's contents...",
  "If you remove a bean from the bag,...",
  "| d100 | Effect |",
  "|---|---|",
  "| 01 | 5d4 toadstools sprout... |",
  "| 02-10 | A geyser erupts...|",
  "| 11-20 | A treant sprouts...|",
  "| 21-30 | An animate, immobile ...|",
  "| 31-40 | A campfire with blue ... |",
  "| 41-50 | 1d6 + 6 shriekers sprout |",
  "| 51-60 | 1d4 + 8 bright pink...|",
  ...
  "| 100 | A giant beanstalk sprouts, ... |"
]
```

Detekce tabulky pracuje na následujícím algoritmu:

```
while (i < this.description.length) {
  const line = this.description[i].trim();

  if (line.startsWith('|') && line.endsWith('|')) {
    // Jedná se o tabulku - najdi konec tabulky a ulož ho jako
    typ: Table
  } else {
    // Standardní text - ulož jako typ text
    this.processedContent.push({ type: 'text', content: line });
    i++;
  }
}
```

Výstup potom vypadá následovně:

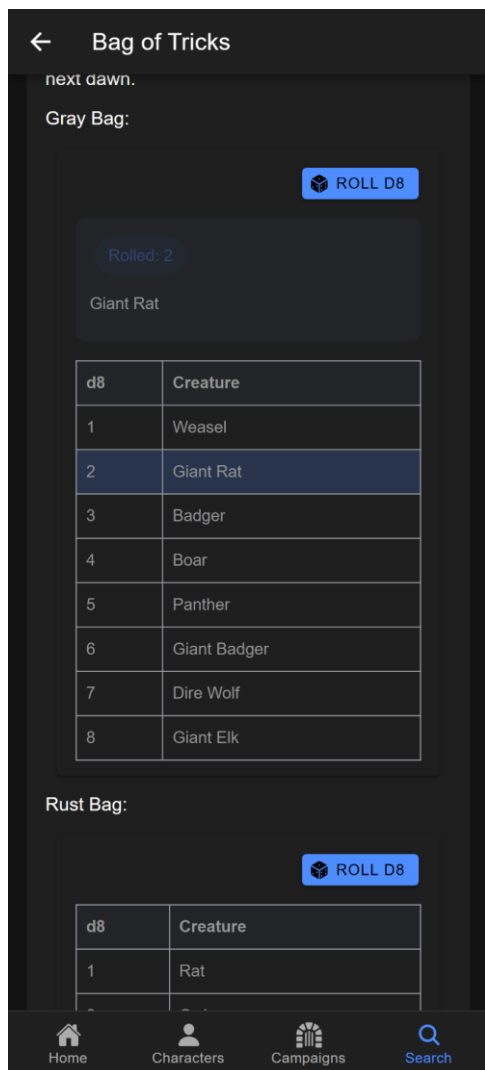
```
[
  {
    "type": "text",
    "content": "Wondrous item, rare"
  },
  {
    "type": "text",
    "content": "Inside this..."
  },
  {
    "type": "text",
    "content": "If you dump the..."
  },
  {
    "type": "text",
    "content": "If you remove..."
  },
  {
    "type": "table",
    "content": [
      "| d100 | Effect |",
      "|---|---|",
      "| 01 | 5d4 toadstools sprout... |",
      "| 02-10 | A geyser erupts...|",
      "| 11-20 | A treant sprouts...|",
      "| 21-30 | An animate, immobile ...|",
      "| 31-40 | A campfire with blue ... |",
      "| 41-50 | 1d6 + 6 shriekers sprout |",
      "| 51-60 | 1d4 + 8 bright pink...|",
      "...",
      "| 100 | A giant beanstalk sprouts, ... |"
    ]
  }
]
```

Tento výstup se následně použije v HTML šabloně komponenty:

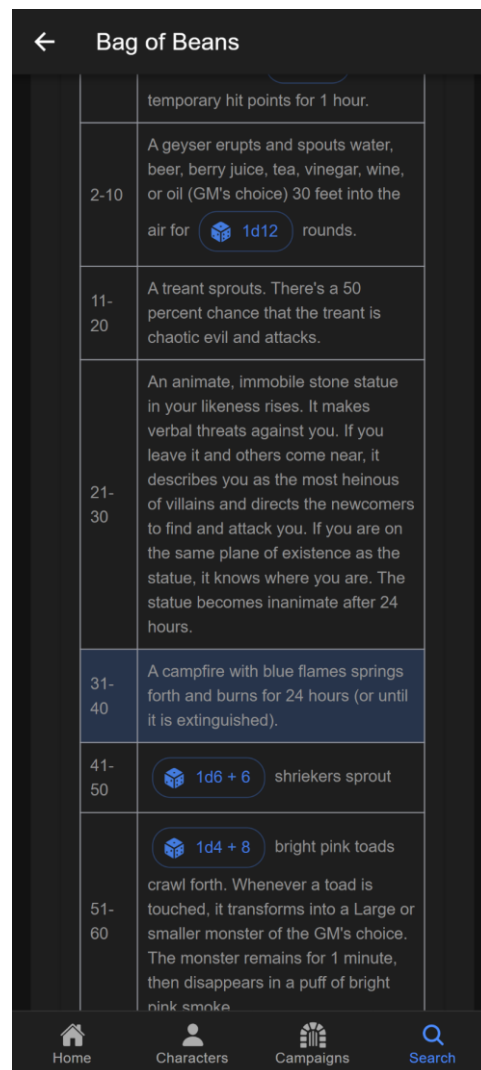
```
<div class="full-description">
  @for (item of processedContent; track $index) {
    @if (item.type === 'text') {
      <p>
        @for (segment of item.content | diceNotation; track $index)
          {
            @if (typeof segment === 'string') {
              {{ segment }}
            } @else {
              <app-dice-chip
                [notation]="segment.notation"></app-dice-chip>
            }
          }
        </p>
      } @else {
        <!-- 'table' -->
        <app-interactive-table
          [tableData]="getTableContent(item)"></app-interactive-table>
      }
    }
  }
</div>
```

S touthle implementací nadále souvisí komponent `interactive-table`, který se nadále stará o správné zobrazení tabulky v HTML a služba `table-parser`, která zajišťuje rozdělení tabulky na řádky a buňky, a také umožňuje házet kostkou na tabulku a vyhodnocovat výsledek hodů, buď podle rozsahu (20-30, 1-5) nebo přímé hodnoty (5, 6, 2).

Tato zhotovená funkční tabulka může v aplikaci vypadat následovně:



Obrázek 47 Tabulka v popisu entity 1



Obrázek 48 Tabulka v popisu entity 2

Součástí této problematiky je i možnost házení kostek v rámci textu, jak je možné vidět na obrázku (Obrázek 48). Tyto hody je možné provést kliknutím na dice-chip komponent, který má ikonku kostky a formát pro hod kostkou (například $1d6 + 6$).

Pro správu této funkce slouží služba `dice.service`, která definuje funkci `findDiceNotations(text: string)`, která prohledá text pomocí regex výrazu a vrátí pole všech notací na hod kostkou.

```
findDiceNotations(text: string): string[] {
  if (!text) return [];
  const diceRegex = /\b(\d*d\d+)(\s*[+-]\s*\d*d\d+)*(\s*[+-]\s*\d+)*\b|[\s*][+-]\d+\b/gi;
  return text.match(diceRegex) || [];
}
```

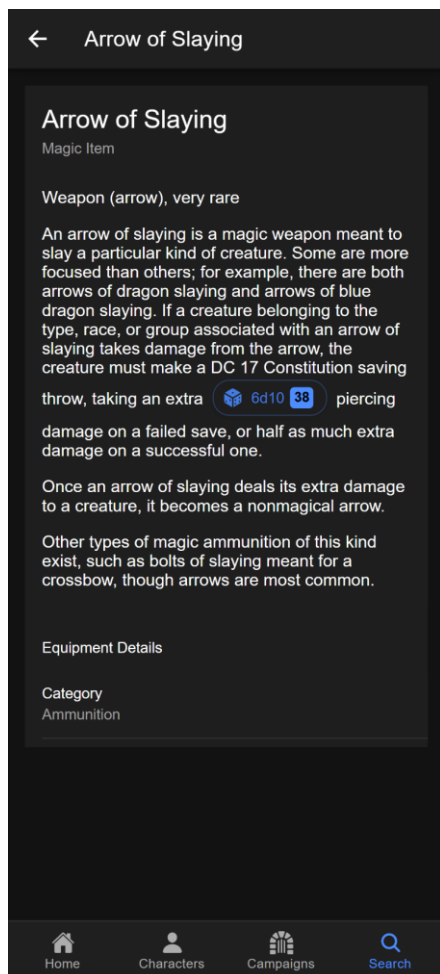
Služba se také stará o samotné hody kostkou za pomoci funkce roll(notation: string), která přijímá notaci na hod kostkou (například 1d6), ta notaci zpracuje a vrátí výsledek jako strukturu rozhraní DiceRoll, která vypadá následovně:

```
export interface DiceRoll {  
  notation: string;  
  rolls: number[];  
  total: number;  
  originalNotation?: string; // To store the original input notation  
}
```

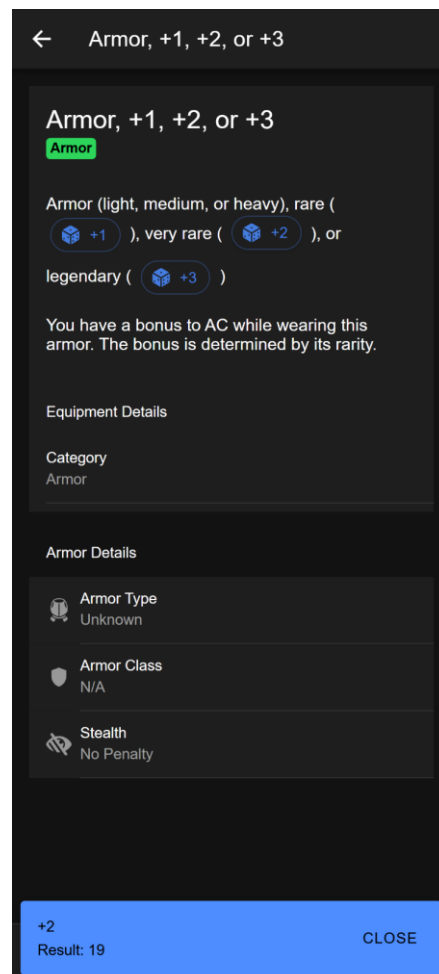
Komponent dice-chip má následující dekorátory pro zvolení fungování:

```
@Input() notation: string = '';  
@Input() disabled: boolean = false;  
@Input() showResult: boolean = true;  
@Input() rollOnce: boolean = false;  
@Input() interface: string = 'dice'; // Default to dice interface  
@Input() showType: boolean = false;  
@Input() type: string = ''; // For damage type  
@Input() showIcon: boolean = true; // Show icon or not  
@Input() color: string = "primary";  
@Input() autoHideDelay: number = 3000; // Time in ms before hiding result
```

Důležitá je notace, která dovoluje házení kostky, dále rollOnce input, který umožňuje hodit kostkou pouze jednou, potom se uzamkne. Možnost je také zvolit interface, který určuje, jak se bude výsledek zobrazovat, buď vedle kostky nebo jako „toast“ modul, jak je vidět na následujících obrázcích:



Obrázek 49 Hod s interface Dice



Obrázek 50 Hod s interface Toast

Tento chip, lze v HTML použít například následovně:

```
<app-dice-chip
  notation="{{ formatNumericModifier(skill.bonus.bonus || 0) }}"
  interface="dice"
  [color]="skill.bonus.proficient ? 'primary' : 'medium'"
  [showIcon]="true"
></app-dice-chip>
```

V některých případech se chip do HTML musí dostat dynamicky. To je právě příklad textu v tabulce na obrázku (Obrázek 48) nebo na obrázku (Obrázek 49 a 50).

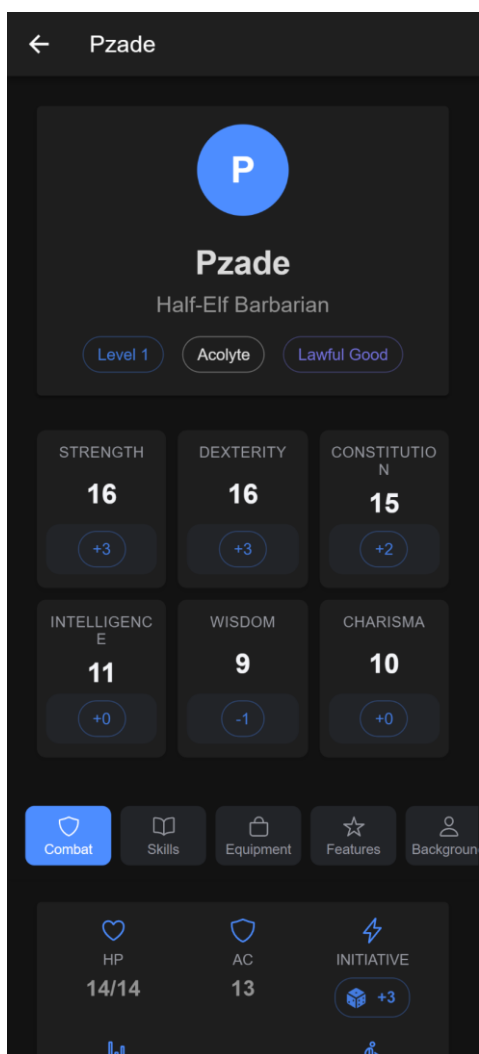
Proto bylo nutné udělat vlastní dice-notation-pipe, která se stará o transformaci textu a vkládá do něj objekt s notací, výstup v JSON formátu vypadá následovně:

```
[
  "Armor (light, medium, or heavy), rare (",
  {
    "type": "dice",
    "notation": "+1"
  },
  ")", very rare (",
  {
    "type": "dice",
    "notation": "+2"
  },
  ")", or legendary (",
  {
    "type": "dice",
    "notation": "+3"
  },
  ")]"
```

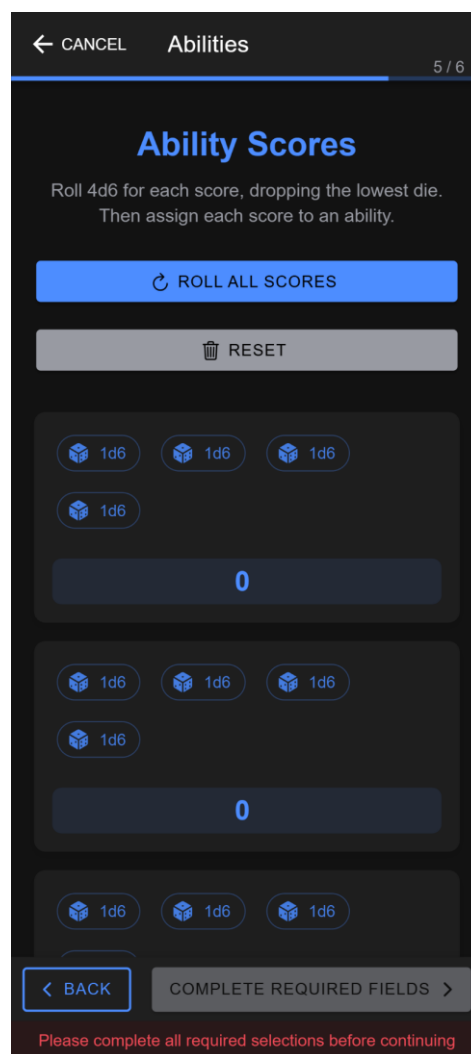
Když se vrátíme ke komponentu `description-renderer`, který definuje následující kód, tak zjistíme že pracuje s `diceNotation` pipe, která vrací objekt segmentů. Dále jen kontroluje každý segment – jestli se jedná o typ „string“ vloží do elementu celý segment, ale pokud se jedná o notaci, tak vloží `dice-chip` komponent s příslušnou notací:

```
@for (segment of item.content | diceNotation; track $index) {
  @if (typeof segment === 'string') {
    {{ segment }}
  } @else {
    <app-dice-chip [notation]="segment.notation"></app-dice-chip>
  }
}
```

Tento komponent se dá využít různými způsoby, například házení na záchranné hody postavy (Obrázek 51) nebo házení na atributy při vytváření postavy (Obrázek 52).



Obrázek 51 Využití komponentu dice-chip v detailu postavy



Obrázek 52 Využití komponentu dice-chip při vytváření postavy

9 TESTOVÁNÍ

Aplikace jsem testoval osobně, tak i s dalšími uživateli z řad přátel a hráčů D&D. Díky jejich zpětné vazbě se podařilo identifikovat nejen drobné chyby, ale také příležitosti k vylepšení uživatelského zážitku.

Funkcionality jako interaktivní házení kostkami a tabulky s možností náhodného výběru hodnot byly velmi pozitivně hodnoceny. Uživatelé ocenili zejména jejich jednoduché použití a přímou návaznost na herní mechaniky.

Navigace v aplikaci byla většinou vnímána jako intuitivní, s výjimkou některých specifických scénářů – například v případě, kdy bylo nutné vybrat povinnou možnost a uživatel ji přehlédl. V takových situacích by aplikace mohla automaticky posunout pohled na daný prvek, čímž by se předešlo špatné orientaci.

Z pohledu designu byla vznesena připomínka k chování spodní navigace (tabs). Ta v některých situacích mizela, což bylo v rozporu s očekáváním uživatelů i zásadami Material designu.

Další návrh padl na ukládání dat, která se, v době psaní práce, ukládají lokálně na zařízení, což může způsobit ztrátu dat. Byl návrh na implementaci databáze i z hlediska lepšího propojení uživatelů mezi sebou, například pro načtení postavy jiného hráče do souboje, místo importování hráče přes kód nebo „opisování“ dat zpět na původní postavu po skončení souboje. Data by se odeslala zpět na postavu po skončení souboje.

9.1 Chyby

Během testování se vyskytlo několik technických problémů:

- Problém s aktualizací seznamu v soubojích – Při přidávání monster se nová monstra nezobrazila ihned v seznamu. Až po přidání dalšího prvku (hráče nebo dalšího monstra) došlo k jeho aktualizaci.
- Stavy soubojů – „Preparing“ - Pokud měl souboj stav „Preparing“, nebylo možné se do něj vrátit. Tento stav uzavíral přístup k rozpracovanému souboji.
- Obnovení souboje – V některých případech se při návratu do souboje stránka nenačetla správně a zůstala prázdná.

- Další problém byl s aktualizací postavy – Když se postavě přidalo vybavení nebo změnil stav peněz, tak se stav neuložil a při dalším navštívení byla postava ve stejném stavu jako předtím (již opraveno).

ZÁVĚR

Cílem této bakalářské práce bylo navrhnout a implementovat mobilní aplikaci pro podporu hraní stolní hry Dungeons & Dragons (5. edice), která by poskytovala uživatelům nástroje pro správu postav, vytváření kampaní a soubojových scénářů. Zároveň umožňovala získávání a filtraci informací o klíčových složkách potřebných ke hraní – vybavení, monstra.

Při vývoji byly použity moderní technologie framework Ionic a Angular, které umožnily vytvořit řešení s důrazem na intuitivní uživatelské rozhraní. Pro získání dat o herních prvcích bylo využito open-source API DnD 5e.

Výrazným přínosem projektu byly komponenty jako dice-chip, pro možnost házet kostkami přímo v textu, table-parser, pro automatické rozpoznávání a interaktivní zpracování tabulek, nebo systém pro tvorbu postavy. Tyto části rozšířily funkcionalitu aplikace, ale přinesly i zajímavou implementační problematiku během vývoje, která přispěla ke zlepšení znalostí v rámci implementaci aplikací.

Přestože aplikace plní požadavky, v rámci návrhu by bylo možné zvolit konzistentnější přístup například dodržováním zásad Material Designu, což by pozitivně ovlivnilo uživatelský zážitek. Zároveň by bylo vhodné architekturu aplikace rozdělit do více menších komponent, což by přineslo větší modularitu, lepší přehlednost a snazší údržbu v budoucnu.

Aplikace jako celek funguje dobře jako digitální společník při hraní D&D, a její potenciál lze dále rozvíjet. Do budoucna by bylo vhodné rozšířit ji o více dat, zvýšit interaktivitu například hlubší integrací s postavou, nebo zavést možnost proklikávat se k definicím herních prvků přímo z aplikace bez nutnosti dohledávání z externích zdrojů. Logickým krokem by bylo také nasazení databáze, která by umožnila přetrvávající ukládání herního stavu, sdílení mezi hráči a celkově vyšší komfort při používání.

Vývoj aplikace byl skvělou zkušeností nejen z technického hlediska, ale také z pohledu návrhu uživatelsky přívětivého prostředí a praktického využití otevřených dat. I když je aplikace stále otevřená dalšímu rozšiřování, již nyní představuje funkční základ pro efektivní hraní Dungeons & Dragons.

SEZNAM POUŽITÉ LITERATURY

- [1] *Dračí doupeř*. Online. In: Wikipedia: the free encyclopedia. San Francisco (CA): Wikimedia Foundation, 2001-. Dostupné z: https://cs.wikipedia.org/wiki/Dra%C4%8D%C3%AD_doup%C4%9B. [cit. 2025-05-12].
- [2] *D&D Beyond*. Online. Dndbeyond. 2017-2025. Dostupné z: <https://www.dndbeyond.com/>. [cit. 2025-05-12].
- [3] *5etools*. Online. 5etools. C2025. Dostupné z: <https://5e.tools/>. [cit. 2025-05-12].
- [4] *Roll20*. Online. Roll20. Dostupné z: <https://app.roll20.net/>. [cit. 2025-05-12].
- [5] *Wizards of the Coast*. Online. Wizards of the Coast. C1993-2025. Dostupné z: <https://company.wizards.com/en>. [cit. 2025-05-14].
- [6] *D&D Beyond Maps: Beta*. Online. Dndbeyond. 2017-2025. Dostupné z: <https://www.dndbeyond.com/games>. [cit. 2025-05-12].
- [7] *Choose your plan*. Online. Dndbeyond. 2017-2025. Dostupné z: <https://marketplace.dndbeyond.com/subscribe#tiers-comparison-table>. [cit. 2025-05-12].
- [8] *Roll20 Crash Course*. Online. Help Roll20. [2023]. Dostupné z: <https://help.roll20.net/hc/en-us/articles/360039223834-Roll20-Crash-Course>. [cit. 2025-05-15].
- [9] *Getting Started for Players*. Online. Help Roll20. 1 September 2023. Dostupné z: <https://help.roll20.net/hc/en-us/articles/360046621893-Getting-Started-for-Players#01H1RJCV672F0YPANWW7JDG6R9>. [cit. 2025-05-15].
- [10] *Roll20Characters*. Online. Roll20. 2012. Dostupné z: <https://app.roll20.net/characters>. [cit. 2025-05-15].
- [11] *Roll20 - Character Sheets*. Online. Google Play. C2025. Dostupné z: https://play.google.com/store/apps/details?id=net.roll20.player&__hstc&__hssc&__li=1. [cit. 2025-05-15].
- [12] *Your Games, Anywhere*. Online. Roll20. 2021. Dostupné z: <https://pages.roll20.net/companionapp>. [cit. 2025-05-15].
- [13] *Ionic framework*. Online. Ionic framework. C2025. Dostupné z: <https://ionicframework.com/>. [cit. 2025-05-18].
- [14] *UI Components*. Online. Ionic framework. C2025. Dostupné z: <https://ionicframework.com/docs/components>. [cit. 2025-05-18].
- [15] *Components in Angular*. Online. Angular. C2010-2025. Dostupné z: <https://angular.dev/tutorials/learn-angular/1-components-in-angular>. [cit. 2025-05-18].
- [16] *What is Angular?* Online. Angular. C2010-2025. Dostupné z: <https://angular.dev/overview>. [cit. 2025-05-18].
- [17] *Getting started with standalone components*. Online. Angular. C2010-2025. Dostupné z: <https://v17.angular.io/guide/standalone-components>. [cit. 2025-05-18].
- [18] *Anatomy of a component*. Online. Angular. C2010-2025. Dostupné z: <https://angular.dev/guide/components>. [cit. 2025-05-18].

- [19] *Control flow*. Online. Angular. C2010-2025. Dostupné z: <https://angular.dev/guide/templates/control-flow>. [cit. 2025-05-18].
- [20] *Introduction*. Online. D&D 5e SRD API. C2025. Dostupné z: <https://5e-bits.github.io/docs/introduction>. [cit. 2025-05-19].
- [21] *D&D 5e API*. Online. C2025. Dostupné z: <https://www.dnd5eapi.co/>. [cit. 2025-05-19].
- [22] *5e-srd-api*. Online. Github. C2025. Dostupné z: <https://github.com/5e-bits/5e-srd-api?tab=readme-ov-file>. [cit. 2025-05-19].
- [23] *Status tracks the uptime of the D&D REST API*. Online. D&D API Status. C2025. Dostupné z: <https://5e-bits.github.io/dnd-uptime/>. [cit. 2025-05-19].
- [24] *Docker Desktop*. Online. C2025. Dostupné z: <https://www.docker.com/products/docker-desktop/>. [cit. 2025-05-19].
- [25] *Dungeons & Dragons: Player's Handbook*. Wizards of the Coast, 2014. ISBN 9780786965601.

SEZNAM OBRÁZKŮ

Obrázek 1 Přejít na webové rozhraní v aplikaci.....	15
Obrázek 2 Aplikace D&D Beyond	15
Obrázek 3 Domovská stránka	28
Obrázek 4 Upravená domovská stránka	28
Obrázek 5 Stránka kampaní	29
Obrázek 6 Stránka kampaní se zobrazenými funkcemi	29
Obrázek 7 Upravená stránka kampaní	30
Obrázek 8 Stránka postav	31
Obrázek 9 Stránka postav se zobrazením funkcí	31
Obrázek 10 Upravená stránka postav	32
Obrázek 11 Stránka vyhledávání	33
Obrázek 12 Stránka vyhledávání s filtry	33
Obrázek 13 Upravená stránka vyhledávání	34
Obrázek 14 Detail kampaně.....	35
Obrázek 15 Detail kampaně se zobrazenými možnostmi	35
Obrázek 16 Opravený detail kampaně.....	36
Obrázek 17 Průběh souboje	37
Obrázek 18 Příprava souboje.....	37
Obrázek 19 Příprava souboje s volbou účinkujících	37
Obrázek 20 Opravená příprava souboje.....	38
Obrázek 21 Rozšířený list pro volbu monster	38
Obrázek 22 Opravená stránka pro průběh souboje	39
Obrázek 23 Detail postavy.....	40
Obrázek 24 Opravený detail postavy.....	40
Obrázek 25 Detail vybavení	41
Obrázek 26 Detail vybavení s hodem na tabulku	41
Obrázek 27 Opravená interakce s tabulkou	42
Obrázek 28 Detail monstra	43
Obrázek 29 Hod kostkou na akci monstra	43
Obrázek 30 Wizard pro vytváření postavy	44
Obrázek 31 Volba rasy	45
Obrázek 32 Volba třídy.....	45
Obrázek 33 Volba charakteru postavy	45
Obrázek 34 Volba detailů	45

Obrázek 35 Volba atributů.....	46
Obrázek 36 Rekapitulace a volba jména.....	46
Obrázek 37 Mapa navigace v aplikaci	47
Obrázek 38 Hlavní navigace v aplikaci	57
Obrázek 39 Hlavní stránka.....	58
Obrázek 40 Stránka vyhledávání	61
Obrázek 41 Stránka vyhledávání s filtrací	61
Obrázek 42 Stránka kampaní	62
Obrázek 43 Stránka postav	64
Obrázek 44 Stránka postav s filtrací	64
Obrázek 45 Vizualizace OptionSelector komponentu	69
Obrázek 46 Možnost Choice v rámci OptionSelector komponentu	69
Obrázek 47 Tabulka v popisu entity 1	73
Obrázek 48 Tabulka v popisu entity 2	73
Obrázek 49 Hod s interface Dice	75
Obrázek 50 Hod s interface Toast.....	75
Obrázek 51 Využití komponentu dice-chip v detailu postavy	77
Obrázek 52 Využití komponentu dice-chip při vytváření postavy	77

SEZNAM TABULEK

Tabulka 1 Srovnání D&D Beyond a Roll20	16
Tabulka 2 Funkční požadavky	21
Tabulka 3 Nefunkční požadavky	22

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

D&D Dungeons & Dragons

VTT Virtual Tabletop

RPG Role-playing game

API Application Programming Interface

HTTP Hypertext Transfer Protocol

JSON JavaScript Object Notation