

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství



Bakalářská práce

Vývoj serveru webové verze deskové hry v ASP.NET

Václav Steinhauser

© 2024 ČZU v Praze

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Václav Steinhauser

Informatika

Název práce

Vývoj serveru webové verze deskové hry v ASP.NET

Název anglicky

Development of the server of the web version of the board game in ASP.NET

Cíle práce

Cílem práce je návrh a implementace serveru webové aplikace pro hraní obdoby deskové hry. Součástí práce je i návrh databáze pro uložení stavu rozehraných her.

Metodika

Práce sestává ze dvou částí, teoretické a praktické. V rámci teoretické části bude provedeno studium relevantních odborných informačních zdrojů. Zjištěné poznatky pak budou využity při zpracování praktické práce.

V praktické části bude s využitím zjištěných poznatků proveden návrh a implementace aplikace. Bude vytvořeno schéma databáze a jádro aplikace obsahující pravidla. Posléze bude vytvořeno prostředí v ASP.NET sloužící ke komunikaci s již vytvořenými částmi. Bude provedeno nasazení a testování vytvořené aplikace. Budou zváženy možnosti optimalizace kritických částí aplikace a dalšího rozšiřování o možnosti nastavitelnosti parametrů her.

Při tvorbě aplikace bude využit programovací jazyk C# a ASP.NET. Jako databáze bude použita MariaDB.

Doporučený rozsah práce

35-40 stran

Klíčová slova

C#, ASP.NET, .NET framework, databáze, desková hra

Doporučené zdroje informací

GILFILLAN, Ian. *Myslíme v MySQL 4*. Praha: Grada, 2003. ISBN 80-247-0661-.

MASLAKOWSKI, Mark. *Naučte se MySQL za 21 dní*. Praha: Computer Press, 2001. ISBN 80-7226-448-6.

MUELLER, John Paul. *Microsoft ADO.NET Entity Framework Step by Step*. Redmond: Microsoft Press, 2013. ISBN 978-0-735-66416-6

PÍSEK, Slavoj. *ASP.NET : začínáme programovat : podrobný průvodce začínajícího uživatele*. Praha: Grada, 2003. ISBN 80-247-0526-5.

SHARP, John. *Microsoft Visual C# 2012: step by step*. Sebastopol, Calif.: O'Reilly Media, c2012, xxxii, 806 p. Step by step (Redmond, Wash.). ISBN 978-0-735-668010.

Předběžný termín obhajoby

2023/24 LS – PEF

Vedoucí práce

Ing. Jiří Brožek, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 28. 11. 2023

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 9. 2. 2024

doc. Ing. Tomáš Šubrt, Ph.D.

Děkan

V Praze dne 15. 03. 2024

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Vývoj serveru webové verze deskové hry v ASP.NET" jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 15.3.2024

Poděkování

Rád(a) bych touto cestou poděkoval panu Ing. Jiřímu Brožkovi Ph.D. za připomínky a odborné rady při konzultacích a vedení mé bakalářské práce.

Vývoj serveru webové verze deskové hry v ASP.NET

Abstrakt

Tato práce se zaměřuje na návrh a implementaci serverové webové aplikace pro adaptaci deskové hry inspirované hrou Osadníci z Katanu. Hlavním cílem je vytvoření prostředí umožňujícího uživatelům hrát tuto deskovou hru online a ukládat stav rozehraných her. Práce je rozdělena do dvou hlavních částí: teoretické a praktické. V teoretické části je provedeno studium existujících informačních zdrojů v oblasti deskových her a webových technologií. Získané poznatky jsou následně aplikovány při návrhu a implementaci praktické části, která zahrnuje vytvoření schématu databáze pro ukládání stavu her a jádra aplikace obsahujícího pravidla hry. Dále je vytvořeno prostředí v ASP.NET, které umožňuje uživatelům připojení a komunikaci s aplikací. V rámci implementace jsou využity technologie Entity Framework pro efektivní práci s databází a SignalR pro real-time komunikaci mezi hráči a serverem. Po nasazení a testování aplikace jsou zvažovány možnosti optimalizace kritických částí a rozšíření o další funkce, včetně možnosti nastavitelnosti parametrů her. Celkově tato práce představuje komplexní přístup k vytvoření online platformy pro hraní deskových her s důrazem na uživatelskou přívětivost, efektivitu a rozšiřitelnost.

Klíčová slova: C#, ASP.NET, .NET Framework, databáze, desková hra

Development of the server of the web version of the board game in ASP.NET

Abstract

This bachelor's thesis focuses on the design and implementation of a server-side web application for adapting a board game inspired by the game Settlers of Catan. The main objective is to create an environment enabling users to play this board game online and save the state of ongoing games. The thesis is divided into two main parts: theoretical and practical. The theoretical part involves a study of existing information sources in the field of board games and web technologies. The acquired knowledge is subsequently applied in the design and implementation of the practical part, which includes creating a database schema for storing game states and an application core containing the game rules. Furthermore, an environment in ASP.NET is created, allowing users to connect and communicate with the application. Technologies such as Entity Framework for efficient database work and SignalR for real-time communication between players and the server are utilized during implementation. After deploying and testing the application, possibilities for optimizing critical parts and expanding with additional features are considered, including the ability to customize game parameters. Overall, this bachelor's thesis presents a comprehensive approach to creating an online platform for playing board games with a focus on user-friendliness, efficiency, and extensibility.

Keywords: C#, ASP.NET, .NET Framework, database, board game

Obsah

1 Úvod	11
2 Cíl práce a metodika	12
2.1 Cíl práce	12
2.2 Metodika	12
3 Teoretická východiska	13
3.1 Deskové hry a jejich adaptace.....	13
3.1.1 Osadníci z Katanu	14
3.1.2 Adaptace deskových her	14
3.2 Programovací jazyk C# a jeho frameworky.....	15
3.2.1 ASP.NET Core.....	15
3.2.2 Entity Framework Core	17
3.3 Ostatní využití technologie.....	17
3.3.1 JavaScript.....	18
3.3.2 SignalR.....	18
3.3.3 MariaDB	18
3.3.4 Docker.....	19
4 Vlastní práce	20
4.1 Návrh aplikace	20
4.1.1 Analýza požadavků.....	20
4.1.2 Návrh architektury aplikace.....	21
4.1.3 Návrh schématu databáze pro uložení stavu rozehraných her	22
4.1.4 Návrh uživatelského rozhraní uživatelského toku	24
4.2 Implementace aplikace.....	25
4.2.1 Implementace jádra aplikace	25
4.2.2 Implementace komponent pro komunikaci s databází.....	27
4.2.3 Implementace uživatelského rozhraní a interakce s uživatelem.....	27
4.3 Nasazení testování.....	28
4.3.1 Nasazení aplikace na server.....	28
4.3.2 Testování aplikace s reálnými uživateli.....	29
4.3.3 Řešení chyb a oprava nedostatků.....	29
4.4 Optimalizace a rozšíření.....	29
5 Výsledky a diskuse	31
5.1.1 Výsledná grafická stránka aplikace	31
5.1.2 Uživatelská příručka	34
5.1.3 Porovnání s podobnými aplikacemi.....	37

6 Závěr.....	38
7 Seznam použitých zdrojů.....	39
8 Seznam obrázků	41
Přílohy	42

1 Úvod

V dnešní době digitálního věku se stále více oblíbených deskových her přenáší do prostředí webových aplikací, což umožňuje hráčům hrát své oblíbené hry kdykoli a kdekoli. Tato práce se zabývá návrhem a implementací webové aplikace pro hraní období deskové hry, která zahrnuje i návrh databáze pro ukládání stavu rozehraných her. Práce se skládá ze dvou hlavních částí, a to teoretické a praktické. Teoretická část zahrnuje studium relevantních odborných informačních zdrojů, které poskytnou potřebné znalosti a podklady pro návrh a implementaci aplikace. Praktická část se zaměřuje na konkrétní realizaci aplikace s využitím získaných poznatků a metodiky vývoje. V rámci praktické části budou navrženy a implementovány klíčové komponenty aplikace, včetně databáze pro ukládání her, jádra aplikace obsahujícího pravidla hry a prostředí v ASP.NET sloužícího ke komunikaci s ostatními částmi. Dále bude provedeno nasazení a testování aplikace a zvažovány možnosti optimalizace a rozšiřování. Pro implementaci aplikace budou využity programovací jazyk C# a framework ASP.NET. Jako databázový systém bude použita MariaDB. Tímto úvodem byly nastíněné cíle a metodika této práce, která nám poslouží jako orientační rámec pro následující kapitoly.

2 Cíl práce a metodika

2.1 Cíl práce

Cílem práce je návrh a implementace serveru webové aplikace pro hraní obdoby deskové hry. Součástí práce je i návrh databáze pro uložení stavu rozehraných her.

2.2 Metodika

Práce sestává ze dvou částí, teoretické a praktické. V rámci teoretické části bude provedeno studium relevantních odborných informačních zdrojů. Zjištěné poznatky pak budou využity při zpracování praktické práce.

V praktické části bude s využitím zjištěných poznatků proveden návrh a implementace aplikace. Bude vytvořeno schéma databáze a jádro aplikace obsahující pravidla. Posléze bude vytvořeno prostředí v ASP.NET sloužící ke komunikaci s již vytvořenými částmi. Bude provedeno nasazení a testování vytvořené aplikace. Budou zváženy možnosti optimalizace kritických částí aplikace a dalšího rozšiřování o možnosti nastavitelnosti parametrů her.

Při tvorbě aplikace bude využit programovací jazyk C# a ASP.NET. Jako databáze bude použita MariaDB.

3 Teoretická východiska

Teoretická část práce "Vývoj serveru webové verze deskové hry v ASP.NET" se zaměřuje na získání a pochopení potřebných teoretických základů pro návrh a implementaci webové aplikace. Tato část poskytuje užitečné informace o deskových hrách a jejich adaptaci do digitální podoby, jakož i o programovacím jazyce C# a jeho klíčových frameworků pro vývoj webových aplikací.

Deskové hry jsou populární formou zábavy již po staletí a s rozvojem technologie se začaly přenášet do digitálního prostředí. Práce se zabývá studiem adaptace deskových her, zejména Osadníků z Katanu, do webového formátu, což otevírá nové možnosti pro hráče po celém světě. Důležitou součástí této adaptace je vhodný výběr technologií, které umožní efektivní implementaci a fungování webové aplikace.

Programovací jazyk C# a jeho frameworky, jako je ASP.NET Core a Entity Framework Core, hrají klíčovou roli ve vývoji moderních webových aplikací. Tyto technologie poskytují robustní infrastrukturu a nástroje pro vytváření bezpečných, výkonných a uživatelsky přívětivých aplikací. Důkladné pochopení těchto technologií je nezbytné pro úspěšnou realizaci projektu.

Dalšími využitými technologiemi jsou JavaScript a SignalR, které přispívají k interaktivitě a real-time komunikaci ve webových aplikacích. Tyto technologie rozšiřují možnosti aplikace a zlepšují uživatelský zážitek.

Celkově je teoretická část práce klíčová pro pochopení kontextu a získání znalostí nezbytných pro úspěšný vývoj serveru webové verze deskové hry v ASP.NET.

3.1 Deskové hry a jejich adaptace

Podle Zapletala (Zapletal, 1991) je desková hra dobrovolná aktivita více hráčů, která spočívá v posouvání nebo pokládání nových herních kamenů nebo jiných prvků na předem určené herní desce. Herní deska nemusí mít pevně daná, ale celá stejně jako pohyb kamenů se řídí pravidly dané hry.

Dále autor uvádí (Zapletal, 1991), že deskové hry jsou s námi deskové hry už velmi dlouho. Už ve starém Římě nebo i starších civilizacích se našly figurky, které k tomuto účelu zřejmě sloužily. Jednou z nejstarších a zároveň dodnes nejpopulárnějších her, která se dochovaly dodnes jsou Šachy.

3.1.1 Osadníci z Katanu

Osadníci z Katanu neboli Catan, jakožto desková hra vytvořená Klausem Teuberem v roce 1995, okamžitě získala celosvětovou popularitu a stala se jedním z ikonických zážitků v herním světě (Catan, 2024). Pravidla této hry (Catan, 2020) zavádějí hráče do dynamického světa obchodu, stavění a strategie, kde se o vítězství bojuje prostřednictvím budování měst, vesnic a cest na herní ploše skládající se z rozmanitých surovin.

Podle pravidel je princip hry následující: hráči sbírají suroviny z polí, která ovládají jejich stavby, a používají tyto suroviny k budování a rozšiřování svých vesnic a měst. Každá stavba jim přináší výherní body a zároveň strategické výhody, které mohou posílit jejich pozici v dalších fázích hry.

Nicméně, klíčovou součástí strategie jsou i výměnné obchody s ostatními hráči. Tato interakce přináší dynamiku a taktické rozhodování do každé partie. Hráči musí efektivně vyjednávat, aby získali potřebné suroviny pro své plány, zatímco zároveň chrání své zájmy a brání se případným útokům protivníků.

Jedním z dalších zajímavých prvků hry jsou akční karty, které mohou změnit průběh hry v okamžiku. Tyto karty mohou oslabit protivníky, poskytnout hráči výhody nebo přinést neočekávané zvraty do hry. Strategické použití těchto karet může znamenat rozdíl mezi vítězstvím a porážkou.

Podle webu Hrajeme.cz (Hrajeme.cz, 2024) byla základní hra Osadníci z Katanu pouze začátkem, protože po jejím úspěchu vzniklo mnoho rozšíření a odvozených her. Tato rozšíření nejenom upravují herní desku a přidávají nové herní kameny, ale také mění herní mechanismy, či nabízejí možnosti pro hru s různým počtem hráčů. To dává fanouškům možnost objevovat stále nové aspekty této bohaté a dynamické herní zkušenosti.

3.1.2 Adaptace deskových her

Deskové hry mají dlouhou historii a jsou oblíbeným způsobem trávení volného času pro mnoho lidí. S rozvojem počítačových a internetových technologií však začaly vznikat digitální verze těchto her, které nabízejí nové možnosti a výzvy. Adaptace deskových her do digitálního prostředí může přinést mnoho výhod, jako je snadnější přístupnost, možnost hry s hráči po celém světě a větší flexibilita v designu a pravidlech hry.

Například v článku o umělé inteligenci v šachách (Houser, 2017) je vidět, že tento trend digitální adaptace není nový. Již v minulosti byly některé deskové hry, jako například šachy, adaptovány do digitální podoby. Šachy dokonce inspirovaly vývoj specializovaných

programů a celých počítačů, které se zaměřovaly na hru a analýzu tahů. Tato historie ukazuje na dlouhodobý zájem o přenos deskových her do digitálního světa a na jejich potenciál v oblasti technologických inovací.

3.2 Programovací jazyk C# a jeho frameworky

Programovací jazyk C# (C sharp) podle knihy (Microsoft, 2022) vznikl v roce 2000 a je moderním, objektově orientovaným jazykem vyvinutým společností Microsoft. Jeho syntaxe je inspirována jazyky jako C++ a C, což umožňuje rychlou adaptaci pro vývojáře s jinými zkušenostmi. C# nabízí širokou škálu funkcí, které usnadňují vývoj sofistikovaných aplikací a zajišťují bezpečnost a spolehlivost kódu.

V knihách (Microsoft, 2022; Solis, Schrottenboer, 2018) se uvádí, že s rozvojem technologií se C# stal klíčovým jazykem pro vývoj aplikací na platformě .NET. .NET je softwarová platforma vyvinutá společností Microsoft, která umožňuje tvorbu a běh aplikací na různých operačních systémech. Jedná se o framework, který poskytuje širokou škálu knihoven a nástrojů pro vývojáře. Mezi nejznámější součásti platformy .NET patří .NET Framework a .NET Core.

Od autorů knih (Microsoft, 2023; Solis, Schrottenboer, 2018) víme, že .NET Framework byl původní verzí platformy .NET, která byla vydána v roce 2002. Byla určena především pro vývoj aplikací pro operační systémy Windows. .NET Framework obsahuje velké množství knihoven a nástrojů pro vývoj desktopových, webových a serverových aplikací v jazyce C#. Tento framework se stal populárním mezi vývojáři po celém světě.

Dále autoři (Microsoft, 2024; Solis, Schrottenboer, 2018) píšou, že .NET Core je novější verzí platformy .NET, která byla představena v roce 2016. Jedná se o open-source a multiplatformní framework, který podporuje vývoj aplikací pro různé operační systémy, včetně Windows, Linuxu a macOS. .NET Core je modulární a efektivní platformou, která nabízí vysoký výkon a široké možnosti pro vývoj moderních aplikací. Od roku 2020 se .NET Core sloučil s .NET Frameworkem do nového frameworku s názvem .NET 5 a později .NET 6.

3.2.1 ASP.NET Core

ASP.NET Core je webový framework, který jako je psáno v dokumentaci (Microsoft, 2023) poskytuje infrastrukturu a nástroje pro vývoj moderních webových aplikací a služeb.

Jednou z klíčových vlastností ASP.NET Core je použití architektury Model-View-Controller (MVC).

Dle dokumentace Model-View-Controller (MVC) je architektonický vzor, který odděluje různé části aplikace a umožňuje efektivní správu kódu a jeho testovatelnost. Tento vzor se skládá ze tří hlavních komponent.

První komponentou je Model, který představuje datový model aplikace. Jedná se o třídy nebo objekty, které obsahují data a logiku aplikace. Model reprezentuje podnikovou logiku a manipulaci s daty, například přístup k databázi, validaci vstupních dat nebo definici business rules. Modely jsou obvykle nezávislé na uživatelském rozhraní a obsahují pouze logiku aplikace.

Druhou komponentou je View, který představuje uživatelské rozhraní aplikace. Jedná se o stránky HTML, které zobrazují data z modelu uživateli. View je pasivní komponenta, která pouze zobrazuje informace uživateli a neprovádí žádnou logiku aplikace.

Třetí komponentou je Controller, který obsahuje logiku aplikace a zpracovává uživatelské požadavky. Jedná se o prostředníka mezi modelem a viewem. Controller zpracovává uživatelské vstupy (HTTP požadavky), vyvolává odpovídající akce v modelu pro získání potřebných dat a následně předává tyto data viewu pro zobrazení. Controller také zajišťuje správu stavu aplikace a řízení toku dat mezi modelem a viewem.

Díky oddělení modelu, viewu a controlleru je možné jednotlivé části aplikace vyvíjet, testovat a udržovat nezávisle na sobě, což zlepšuje modularitu a rozšiřitelnost aplikace.

Předchozí verzi popsané v knize (Písek, 2003) ASP.NETu byl tzv. "klasický" ASP.NET, který byl založen na architektuře Web Forms. Tento model měl odlišnou architekturu a přístup k vývoji aplikací oproti ASP.NET Core. S příchodem ASP.NET Core byly přinášeny nové možnosti a vylepšení, což umožnilo modernější a efektivnější vývoj webových aplikací.

Podle Freedmana (Freeman, 2021) je ASP.NET Core Identity další klíčovou součástí ASP.NET Core, která poskytuje funkce pro autentizaci a autorizaci uživatelů v aplikacích. Umožňuje správu uživatelských účtů, role a oprávnění, a integruje se s ostatními částmi frameworku pro jednoduchou implementaci bezpečnosti v aplikaci.

3.2.2 Entity Framework Core

Informace v této podkapitole pocházejí z knihy (Vogel, 2021) E. Vogela. Entity Framework Core (EF Core) je ORM (Object-Relational Mapping) framework, který umožňuje vývojářům pracovat s databází pomocí objektově orientovaných konceptů. Jednou z klíčových funkcí EF Core je možnost mapovat objektový model aplikace na strukturu relační databáze a naopak. To znamená, že vývojáři mohou pracovat s databází pomocí objektů a tříd namísto ručního psaní SQL dotazů.

Jedním z hlavních konceptů, který EF Core podporuje, je CRUD (Create, Read, Update, Delete). To znamená, že pomocí EF Core můžete vytvářet nové záznamy v databázi, číst existující záznamy, aktualizovat stávající záznamy a odstraňovat záznamy z databáze. Tyto operace lze provádět pomocí objektů a metod poskytovaných EF Core.

Další důležitou funkcí EF Core je lenivé načítání (lazy loading) a sledování změn (change tracking). Lenivé načítání umožňuje načítat data z databáze až ve chvíli, kdy jsou skutečně potřeba, což může zlepšit výkon aplikace tím, že snižuje množství dat, která jsou načtena najednou. Sledování změn zase umožňuje EF Core sledovat změny provedené na objektech v paměti a automaticky je propagovat do databáze, což usnadňuje práci s daty a udržuje konzistenci mezi objekty a databází.

Tyto funkce EF Core jsou velmi užitečné pro vývoj webových aplikací v ASP.NET Core, protože umožňují jednodušší manipulaci s daty a snižují potřebu psaní ručního SQL kódu. Díky nim mohou vývojáři rychleji vytvářet a udržovat aplikace a zároveň optimalizovat výkon a konzistenci dat.

3.3 Ostatní využití technologie

ASP.NET aplikace a jejich vývoj přinášejí mnoho výhod a možností pro tvorbu robustních a efektivních webových aplikací. Nicméně, aby bylo dosaženo maximálního potenciálu a efektivity, je často potřeba doplnit je o další technologie. Tyto technologie mohou zahrnovat různé programovací jazyky, frameworky, databázové systémy, nástroje pro správu kódu a mnoho dalšího. Kombinace těchto technologií může výrazně ovlivnit vývoj a výslednou kvalitu aplikace. Proto je důležité pečlivě vybírat ty správné technologie a integrovat je do procesu vývoje ASP.NET aplikací tak, aby bylo dosaženo optimálního výsledku.

3.3.1 JavaScript

JavaScript je programovací jazyk, který se často používá pro vývoj interaktivního uživatelského rozhraní webových aplikací. Je to skriptovací jazyk běžící ve webovém prohlížeči klienta, umožňující dynamicky měnit obsah stránky, reagovat na události a komunikovat s uživatelem. JavaScript je nezbytnou součástí vývoje moderních webových aplikací a často se kombinuje s HTML a CSS pro vytvoření kompletního uživatelského rozhraní (MDN, 2024).

3.3.2 SignalR

SignalR je knihovna pro vytváření webových aplikací v reálném čase pomocí technologie WebSockets. Autoři (White, 2014; Vemula, 2017) popisují, jak umožňuje serveru posílat informace klientům až ve chvíli, kdy jsou dostupné, což umožňuje okamžitou aktualizaci obsahu stránky bez nutnosti obnovovat celou stránku. SignalR podporuje různé platformy, včetně ASP.NET Core, a je vhodný pro vývoj aplikací, které vyžadují živou interaktivitu.

SignalR funguje na principu komunikace mezi klientem a serverem, kde server bývá implementován v platformě .NET a klient v JavaScriptu.

3.3.3 MariaDB

V dokumentaci (MariaDB, 2024) se uvádí, že MariaDB je open-source relační databázový systém, který se často používá pro ukládání dat ve webových aplikacích. Jedná se o jednu z nejpopulárnějších relačních databází na světě a nabízí širokou škálu funkcí, včetně podpory SQL dotazů, transakcí, indexování a zabezpečení dat. MariaDB je vhodná pro různé typy aplikací, od malých webových stránek až po velké podnikové systémy.

MariaDB je software, který dle dokumentace poskytuje alternativní řešení k jiným relačním databázím, včetně MySQL a MSSQL. Jedná se o open-source relační databázový systém, který je založen na zdrojovém kódu MySQL a poskytuje kompatibilitu s MySQL, spolu s některými vylepšeními a novými funkcemi. MariaDB je často používána jako náhrada za MySQL nebo MSSQL v různých projektech a je podporována širokou komunitou vývojářů.

3.3.4 Docker

Docker je platforma pro vývoj, nasazení a běh aplikací ve standardních kontejnerech, která je blíže specifikovaná v knize (Docker Inc., 2024). Tento nástroj umožňuje vývojářům balit své aplikace a všechny potřebné závislosti do jednotlivých kontejnerů, což zajistí konzistentní prostředí pro spuštění aplikace na libovolném systému. Docker je často používán pro izolaci aplikací a usnadnění práce s prostředími s různými konfiguracemi.

Jednou z klíčových funkcí Dockeru je možnost snadného nasazení a správy kontejnerů. Díky Dockeru mohou vývojáři snadno spouštět své aplikace na libovolném počítači bez nutnosti instalace a konfigurace všech potřebných závislostí. To znamená, že vývojáři mohou rychle vytvářet a testovat své aplikace v izolovaném prostředí, což zvyšuje efektivitu vývoje a minimalizuje možnost chyb spojených s rozdílnými konfiguracemi.

Další důležitou funkcí Dockeru je možnost provozovat databáze v kontejnerech pro vývoj webových aplikací. Díky tomu mohou vývojáři snadno spouštět a spravovat databáze jako součást svého vývojového prostředí bez nutnosti složité instalace a konfigurace. To zjednodušuje vývoj webových aplikací, protože vývojáři mohou rychle testovat a ladit své aplikace s reálnými daty.

4 Vlastní práce

Tato kapitola je zaměřena na představení návrhu, implementace, nasazení, sběru dat a analýzu výsledků. Každý z těchto kroků hraje klíčovou roli v celkovém procesu a přispívá k porozumění a hodnocení zkoumané problematiky. Nejprve bude provedeno plánování a návrh, který poskytuje strukturu a směřování pro celou práci. Poté bude provedena implementace, což zahrnuje konkrétní realizaci navržených řešení a technických aspektů projektu. Nasazení je dalším důležitým krokem, který se zabývá praktickým uplatněním vytvořených řešení ve skutečném prostředí. Sběr dat je klíčovým prvkem pro získání relevantních informací, které následně budou analyzovány a interpretovány. Nakonec bude provedena analýza výsledků, skládající se ze sběru dat a odvozování závěrů a doporučení. Pomocí této struktury je umožněn systematický a komplexní pohled na celý proces práce a jeho výsledky.

4.1 Návrh aplikace

V této kapitole budou ukázány požadavky na aplikaci podle nich bude zvolena architektura aplikace a použití technologií popsanych v teoretické části. Dále bude navrhnutý způsob ukládání stavu rozehraných her do databáze. Jako poslední bude navrhnuté uživatelské rozhraní a předpokládaný uživatelský tok. Jako poslední bude navrhnutý.

4.1.1 Analýza požadavků

Webová aplikace s deskovou hrou vyžaduje pečlivou analýzu požadavků, aby mohla efektivně splnit očekávání uživatelů. Níže jsou uvedeny klíčové požadavky, které budou aplikovány.

- Přihlašování uživatelů:
 - Uživatelé budou moci přihlásit pomocí svého jména / e-mailu a hesla.
 - Zabezpečení bude základní, bez nadměrných požadavků na komplexní ověření.
- Identita hráče:
 - Každý hráč bude identifikován svým uživatelským jménem, které bude používat po celé aplikaci.
- Zobrazení hry:

- Zobrazení hry poskytne hráčům stejné množství informací a možností jako při fyzickém hraní deskové hry.
- Dostupnost aplikace:
 - Aplikace bude přístupná nejen na počítači, kde je nainstalována, ale také prostřednictvím internetového prohlížeče na síti, kterou si zvolí provozovatel.
- Přístup k databázi:
 - Uživatelé budou mít přístup k datům v databázi pouze prostřednictvím aplikace, což zaručuje bezpečnost a integritu dat.
- Více her současně:
 - Hráči budou moci hrát současně více her a k nim budou mít přístup.
- Obsahová adaptace deskové hry:
 - Jelikož se jedná o adaptaci deskové hry, požadavky se soustředí zejména na obsah a nekladou důraz na složitou grafiku.
- Řízení hry událostmi:
 - Hra bude řízena událostmi a měnit svůj stav jen na základě akcí uživatelů.
- Dynamická aktualizace stránky:
 - Stránka hry se nebude obnovovat po každém kliknutí na herní prvek, což zajišťuje plynulý herní zážitek bez ztráty dat a času.
 - Překreslí se vždy jen část, která se změnila, ale pokud budou velké změny tak dojde k překreslení celé stránky.

Analýza těchto požadavků je klíčovým krokem při návrhu a vývoji aplikace, aby bylo dosaženo očekávaného výsledku a uživatelských potřeb.

4.1.2 Návrh architektury aplikace

Pro efektivní implementaci deskové hry ve webové aplikaci bude využita architektura MVC (Model-View-Controller), která poskytne robustní základ pro oddělení datové logiky, prezentace a řízení toku aplikace. Dále budou využity výchozí funkcionality ASP.NET Core Identity pro řízení uživatelských účtů a autorizaci. Pro dynamickou aktualizaci stavu hry bez nutnosti obnovování stránky bude implementován SignalR Hub. V následujících odstavcích budou popsány jednotlivé komponenty.

Modely v aplikaci budou reprezentovat herní prvky a jejich stav v rozehrané hře. Tyto modely budou vzájemně provázány, což umožní efektivní manipulaci s herními daty.

Každá třída modelu bude obsahovat atributy popisující stav herního prvku a metody pro jeho modifikaci.

Controller bude zodpovědný za příjem požadavků od uživatelů a řízení toku aplikace. Bude obsluhovat akce spojené s přesunem mezi hrami nebo jinými stránkami.

Pomocí Controlleru budou data získávána, upravována a předávána mezi Modelem a View.

View bude zodpovědný za prezentaci dat uživateli. Mimo jiné vykreslí herní desku a všechny související informace. Díky využití SignalR bude možné dynamicky aktualizovat View bez nutnosti obnovování stránky, což zajistí plynulý herní zážitek.

ASP.NET Core Identity poskytne základní funkcionality pro správu uživatelských účtů a autorizaci. Tím bude zajištěno bezpečné přihlašování uživatelů a správa jejich rolí.

SignalR Hub bude integrován do aplikace pro real-time komunikaci mezi serverem a klienty. To umožní okamžitou aktualizaci stavu hry a interakci hráčů bez nutnosti manuálního obnovování stránky. SignalR bude zodpovědný za zpracování akcí spojených s herními událostmi, jako je tah hráče nebo aktualizace stavu hry. Díky této funkcionalitě bude zajištěno okamžité sdílení změn stavu hry mezi všemi hráči a serverem bez nutnosti obnovování herní stránky.

Tímto návrhem architektury aplikace bude dosaženo efektivního oddělení logiky aplikace a prezentace, což umožní snadnou rozšiřitelnost a údržbu aplikace v budoucnosti. Díky využití moderních technologií jako je SignalR bude zajištěno plynulé a interaktivní herní prostředí pro uživatele.

4.1.3 Návrh schématu databáze pro uložení stavu rozehraných her

K dosažení co možná nejpřesnější reprezentace s největší modularitou a rozšiřitelností bylo zvoleno následující schéma tříd.

- Hráč: Reprezentuje hráče v aplikaci.
- Má vlastnosti potřebné k ověření identity při přihlašování.
- Surovina: Reprezentuje surovinu v aplikaci.
 - Obsahuje vlastnosti Název k identifikaci.
 - Dále má Image URL s barvou pozadí sloužící k vykreslení.
- Mapka: Reprezentuje herní mapu.
 - Obsahuje kolekce různých prvků na desce, jako jsou pole, cesty a rozcestí.
- Pole: Představuje pole na herní mapě.
 - Obsahuje vlastnosti Číslo a Blokováno a Surovinu k herní logice.

- Má kolekci rozcestí, která s ním sousedí a referenci na mapku na které se nachází.
- Dále má vlastnosti Pozice X, Pozice Y sloužící k vykreslení na správném místě na desce.
- Rozcestí: Reprezentuje rozcestí na herní mapě.
 - Má referenci na mapku, na které se nachází.
 - Pokud ho obsadí hráč tak obsahuje referenci na něj.
 - Pokud hráč postaví Stavbu tak bude obsahovat referenci i na ni.
 - Obsahuje vlastnosti Pozice X a Pozice Y sloužící k vykreslení na správném místě na desce.
- Cesta: Představuje cestu v herní mapě.
 - Má kolekci koncových rozcestí a referenci na mapku na které se nachází.
 - Pokud ji hráč obsadí tak obsahuje referenci na něj.
 - Obsahuje vlastnosti Pozice X, Pozice Y a natočení sloužící k vykreslení na správném místě na desce.
- Stavba: Reprezentuje stavbu v aplikaci.
 - Obsahuje vlastnosti Název, Zisk a Body k herní logice.
- Aktivita: Reprezentuje aktivitu v aplikaci.
 - Má vlastnosti Akce, a číslo Aktivity k herní logice.
 - Musí mít referenci na hráče.
- Akční karta: Reprezentuje akční kartu v aplikaci.
 - Obsahuje vlastnosti Název, a Počet k herní logice
- Surovinová karta: Reprezentuje kartu se surovinou.
 - Má vlastnost Počet a referenci na surovinu.
- Bodova karta: Reprezentuje karty s vítěznými body
 - Má vlastnosti Název a Body.
- Směna: Představuje nabídnutou směnu v aplikaci.
 - Obsahuje kolekce surovinových karet s nabídkou a poptávkou.
 - Musí mít referenci na hráče, který nabídku vytvořil.
- Stav Hráče: Reprezentuje stav hráče v rámci hry.
 - Obsahuje vlastnosti barva, pořadí, nejdelší cesta a největší vojsko.
 - Má kolekce všech druhů karet.
 - Musí mít referenci na právě jednu hru a jednoho hráče.

- Hra: Představuje herní session.
 - Obsahuje vlastnosti index hráče na tahu, stav Hry (nezačala, probíhá, skončila), a další vlastnosti spojené s průběhem hry.
 - Obsahuje také kolekce různých prvků jako jsou stavy, aktivity, směny a nerozdané karty.

Díky použití Entity Frameworku bude toto navržené schéma převedeno do tabulek v databázi, což umožní efektivní ukládání dat a správu herního stavu. V rámci tohoto schématu jsou definovány vztahy mezi entitami, kde reference představují vztah k jedné entitě jiného druhu a kolekce umožňují volitelný vztah k mnoha entitám jiného druhu. Kardinality z druhé strany vztahů jsou pečlivě zvoleny tak, aby se třídy, které obsahují nějakou měnitelnou vlastnost, vytvářely nové, zatímco neměnné třídy, jako je například třída Surovina a Stavba, zůstávaly stejné a pouze se k nim přidávaly reference.

Pro zajištění jedinečnosti a integrity dat v databázi bude v každé tabulce implementován primární klíč. Tento klíč bude přidán v abstraktní třídě HerniEntita, která bude předchůdcem všech tříd určených k ukládání do databáze. Jako typ primárního klíče bude použita instance třídy Guid, která je dostatečně velká a minimalizuje pravděpodobnost kolizí v rámci tabulky. V případě kolize je řešení velmi jednoduché a efektivní, neboť stačí načíst stránku znovu, čímž se konflikt vyřeší. Tento přístup je ideální pro zajištění bezproblémového ukládání a manipulace s daty v rámci aplikace.

4.1.4 Návrh uživatelského rozhraní uživatelského toku

Uživatelské rozhraní bylo zvoleno následovně. Pokud hráč není přihlášený nemá přístup nikam kromě přihlášení a registrace. Přihlášenému hráči se hned na úvodní stránce zobrazí jeho rozehrané hry, hry, které ještě nezačaly a může se připojit a třetí sekce budou ukončené hry, kterých byl součástí. Samozřejmě bude mít i možnost vytvořit novou hru.

Po kliknutí na připojení k nové hře nebo vytvoření úplně nové hry přejde hráč na stránku, kde se mu zobrazí herní plocha, připojení hráči a jejich zvolené barvy. Poté si hráč musí zvolit barvu, kterou v dané hře chce mít a po potvrzení přejde na stránku průběhu hry. Stránka průběhu hry je hlavní částí grafického rozhraní aplikace. Stránka se bude skládat ze tří panelů.

- Akční panel obsahuje:
 - políčko, kam se hráči vypisují instrukce o aktivitě, kterou musí v danou chvíli provést, nebo pobídky k možným akcím;

- tlačítka nákupu, která se odemknou v případě dostatku surovin po kliknutí na nákup hra hráči odebere cenu stavby ze surovin a pokud se jedná o nákup stavby vygeneruje aktivity související s umístěním nové stavby;
 - tlačítka akčních karet, které může hráč zahrát, když je na tahu a nemá vynucenou žádnou aktivitu;
 - tlačítko pro nabídku směny, po které se otevře okno s nabídkou směny
 - tlačítko ukončení tahu.
- Panel s deskou obsahuje vykreslené pole, cesty, rozcestí a stavby.
 - Stavový panel obsahuje:
 - výpis hráčů a jejich vítězné body, kde aktuální hráč je zvýrazněný;
 - vykreslení bodových karet s hodnotou, kolik bodů, která přináší;
 - vykreslení karet se surovinami;
 - seznam nabídek směn hráčů za poslední kolo.

Všechny elementy reprezentující vykreslené herní prvky by měly být schopné reagovat na kliknutí a aplikace by je kliknutí podle jedinečného Id a typu dohledala a vyhodnotila reakci.

4.2 Implementace aplikace

Samotná implementace aplikace obsahuje kromě již zmíněných tříd ještě několik dalších, a hlavně tyto třídy už budou implementovat metody ovládající logiku hry.

4.2.1 Implementace jádra aplikace

Třídy Cesta, Pole a Rozcestí obsahují jen vlastnosti popsané v datovém modelu, jediná metoda, kterou mají je k vykreslení na správném místě pomocí HTML elementu s absolutním pozicováním a nastavenými parametry a nastavenou reakcí na kliknutí. Tyto třídy jsou vytvořené buď pomocí generování herní desky v třídě Mapka, kde jsou nastavené i jejich vztahy.

Třída Surovina a Stavba reprezentují neměnné instance, jejich vlastnosti jsou popsané v datovém modelu. Budou vytvořeny se pouze jednou, pokud v databázi instance se stejným názvem ještě neexistuje. Samotné tyto třídy se nevykreslují, ale poskytují data potřebné k vykreslení jiných herních prvků, které na ně mají referenci.

Třída Karta je abstraktní. Obsahuje jen vlastnost Počet. Díky tomu lze množinu karet stejného druhu reprezentovat pomocí jedné instance

Třídy `BodovaKarta`, `SurovinaKarta` jsou jen pro ukládání herních prvků a jejich vztahů. Jsou generované při vytvoření `StavHrace`, ke kterému dochází při připojení hráče do hry a zůstávají stejné po celou dobu, jen se podle stavu hry mění parametr `Pocet`. Dále se instance třídy `SurovinaKarta` vytváří při nabídce směny, nebo potřebě reprezentace nějakého počtu karet suroviny. Nemají žádné metody měnící jejich stav, jen metodu k jejich vykreslení jako HTML element s nastavenou reakcí na kliknutí.

Třída `AkcniKarta` má vlastnost `nazev`. Dále má metodu sloužící k zahrání karty, které v případě počtu většího než 0 podle názvu karty odpovídajícím způsobem změní stav hry a zmenší počet o 1. Má také metodu k vykreslení jako HTML element s nastavenou reakcí na kliknutí.

Třída `Smena` opět neobsahuje žádné metody, slouží k reprezentaci směny jedné množiny surovinových karet za druhou, kterou hráč nabízí ostatním.

Třída `Aktivita` reprezentuje aktivitu, kterou je potřeba ve hře vykonat před tím než lze ukončit tah, vykonat nákup, nebo zahrát akční kartu. Instance této třídy jsou vytvářeny například při nákupu stavby nebo zahrání akční karty. Následně je přidána na konec kolekce ve třídě `Hra`. K zachování pořadí i při restartování serveru slouží vlastnost `CisloAktivity`. Pokud existuje nějaká nevykonaná aktivita tak nemůže žádný hráč vytvořit novou aktivitu a jen hráč, kterému aktuální aktivita patří může kliknutím na herní prvek podle vypsané instrukce tuto aktivitu splnit a odstranit z kolekce. Tato třída slouží je ukládána do kolekce kvůli tomu, že pokud je ve hře nutně vyžadováno provedení více kroků bezprostředně za sebou jako například úvodní rozestavení kamenů stačí přidat aktivity do kolekce a hráči toto pořadí nebudou moct narušit.

Třída `StavHrace` obsahuje všechny informace o hráči v dané hře, hodnotu má uloženou u vlastností `barva`, `poradi` a několik vlastností, které určují, jestli je v nějakém hodnoceném aspektu hry nejlepší. Dále obsahuje kolekci pro každý ze tří druhů herních karet a vlastnosti s vypočítávanou hodnotou, kde pomocí reference na hru získá kolekci všech hráčových staveb, rozcestí a cest. Tyto vlastnosti jsou využívány při výpočtu aktuálního počtu výherních bodů. Dále tato třída obsahuje metodu pro výpočet nejdelší posloupnosti navazujících neopakujících se cest.

Třída `Mapka` obsahuje kolekci pro všechny prvky na herní desce ale i metody a výchozí vlastnosti potřebné k vygenerování nové desky.

Třída `Hra` obsahuje právě jednu instanci třídy `mapka`. Vlastnost `StavHrace` určuje, jestli hra ještě nezačala, kdy se hráči mohou připojovat, když hra probíhá tak jsou aktivní

všechny ovládací prvky a v případě kdy hra už skončila jsou ovládací prvky deaktivovány a hráči si mohou jen zobrazit stav desky a karet. Vlastnost `HracNaTahu` určuje pořadí hráče, který je právě na tahu, pokud byly splněny všechny čekající aktivity. Tato třída obsahuje kolekce pro nerozdané karty pro akční a bodové karty, kde je omezený počet od každého druhu. Pak obsahuje kolekci stavů jednotlivých hráčů. Obsahuje jak metody pro získání stavu rozehrané hry, tak metody pro vykreslení větších prvků herní stránky ale i metody měnící stav hry, jako přidání hráče, začátek hry, nebo přechod na dalšího hráče.

4.2.2 Implementace komponent pro komunikaci s databází

Pro komunikaci s databází bude využita třída odvozená od `DbContext` z `Entity Framework`. Tato třída obsahuje kolekce tříd popsaných v datovém modelu. Pro jednotlivé třídy pak obsahuje výpis ukládaných vlastností a kardinalitu všech vztahů. `Entity Framework` vytvoří pro každou kolekci tříd tabulku v databázi a v případě kardinalit `Many-to-Many` vytvoří i propojovací tabulky.

4.2.3 Implementace uživatelského rozhraní a interakce s uživatelem

O implementaci uživatelského rozhraní se starají `Controllery`, `View` a `SignalR Hub`. V následující kapitole budou představeny komponenty, které se starají o doručení vykreslených stránek uživateli.

`HomeController` se stará o výchozí stránky. Obsahuje jedinou akci `Index` s jedním `View`, která je výchozí bod celé aplikace a zároveň i místo, kam je uživatel přesměrován, pokud se pokusí o přístup do části aplikace, ke které nemá oprávnění. Pokud hráč není přihlášený tak dostane jen odkazy na přihlášení/registraci zprostředkovanou pomocí `ASP.NET Core Identity`. `Index` pro přihlášeného uživatele obsahuje odkaz na vytvoření nové hry a následující 3 sekce:

- Moje rozehrané hry
 - Obsahuje odkazy na hry, které už probíhají a hráč je k nim už připojený
 - Obsahuje i odkazy na hry, které ještě neprobíhají, ale hráč už se připojil a čeká na prvního hráče až začne hru
- Hry kam se můžu připojit
 - Obsahuje odkazy na hry, které ještě neprobíhají a hráč se k nim může připojit
- Moje ukončené hry
 - Obsahuje odkazy na hry, ke kterým se hráč připojil a už skončily

Odkaz na hru vykreslí ilustrační obrázek a seznam hráčů, na kterém hráč na tahu je zvýrazněný tučně.

HraController obsluhuje akce přímo související se stavem her. Tyto akce vykreslují stránky pomocí 3 View:

- Mapka.cshtml je partial View, který vykreslí herní desku do dvou ostatních View;
- Pripojit.cshtml je View, ve kterém se zobrazí herní deska a hráč dostane na výběr s kterou barvou se chce připojit;
- Prubeh.cshtml je View, který připojenému hráči zobrazí herní desku stav karet a ostatní ovládací prvky.

Dále obsahuje 4 akce:

- Nova vytvoří novou instanci hry a přesměruje hráče na Pripojit;
- Pripojit zobrazí hráči View Pripojit a po odeslání odkazuje na AddStavHrace;
- AddStavHrace zavolá na instanci třídy hra metodu k přidání hráče a přesměruje na Prubeh;
- Prubeh zobrazí View Prubeh.

ClickHub je SignalR Hub, ke kterému se připojí klienti pomocí Javascriptu odeslaného ve View. po kliknutí na herní prvky bude zavolána správná JavaScript metoda, která hubu odešle informaci o aktuální hře hráči a kliknutém elementu. Hub tato volání vyhodnotí, podle Id najde správnou hru, vyhodnotí, jestli daný hráč může dané kliknutí provést a pokud ano tak správným způsobem změní stav hry.

4.3 Nasazení testování

Po implementaci proběhlo nasazení aplikace do provozu, testování na uživatelích a sběr zpětných vazeb následované opravou chyby a nedostatků. Tento postup se několikrát zopakoval, až se implementace dostala do stavu, ve kterém již uživatelé neměli výtky.

4.3.1 Nasazení aplikace na server

Aplikace byla testována na osobním počítači s lokálním databázovým serverem MariaDB v Docker kontejneru. Aplikace byla přístupná z libovolného počítače na virtuální lokální síti. Součástí příloženého kódu aplikace je i Docker compose soubor, který byl využitý k vytvoření druhého Docker kontejneru phpMyAdmin sloužícímu k přístupu do databáze a kontrole správnosti a kompletnosti ukládaných dat mimo aplikaci.

4.3.2 Testování aplikace s reálnými uživateli

Několik uživatelů, patřících do cílové skupiny mělo přístup, registrovalo se a v průběhu několika dní sehrálo několik her. Během toho byl server i databáze monitorovaný jak z hlediska výkonu, spotřeby paměti ale i korektnosti chování.

4.3.3 Řešení chyb a oprava nedostatků

Během testování bylo nalezeno několik nedostatků. Zejména se týkají výkonu aplikace a jediná, která se týkala uživatelského rozhraní:

Není vidět, jaká čísla padla – vyřešeno přidáním panelu s posledním hodem vedle panelu s body hráčů.

Aplikace se po kliknutí dlouho načítá a nedočkavý hráč klikáním aplikaci ještě víc zahlcuje – Příčina byla příliš časté načítání složité struktury závislostí z databáze. Vylepšeno pomocí zobrazení hlášky o načítání uživateli, aby věděl, že je vše v pořádku a nejedná se o chybu. Rychlost byla zvýšena pomocí využití lazy loading přístupu při načítání z databáze. A ukládání her načtených z databáze do paměti, která se uchovává po celou dobu běhu aplikace. Pro zlepšení využití paměti jsou hry, u kterých nedošlo k žádné interakci během posledních 6 hodin jsou z paměti uvolněny a příště budou opět načítány z databáze. Tyto problémy a řešení byly zjištěny a opraveny během několika iterací testování a oprav.

Dále se někdy stalo že při delším nepoužívání počítače klienta a například jeho uspání se stane že se prohlížeč odpojí od SignalR hubu a aplikace tak přestane dostávat vstupy a odesílat výstupy. Aplikace je ale navržena tak, aby obnovení stránky v libovolné chvíli neztratilo uživateli jakoukoliv část rozehraného postupu. Jediná úprava byla že v JavaScriptovém skriptu odeslaném klientovi s View je přidána metoda, která po detekování odpojení od serveru klientovi vypíše instrukci o nutnosti obnovit stránku

4.4 Optimalizace a rozšíření

Během implementace, ale i testování bylo zjištěno že kritickou částí aplikace je jednoznačně načítání dat z databáze. Problematické je, že optimalizace a rozšíření jsou dva protichůdné směry, protože optimalizace by znamenala zjednodušování načítané struktury, zatímco rozšiřitelnost by znamenala větší dělení méně parametrů nastavených staticky v aplikaci, a naopak větší množství načítaných dat a rozhodování na straně entit načtených z těchto dat.

Kritická část aplikace je jednoznačně náročnost načítání dat, které by určitě šlo optimalizovat načtením pouze vybraných vlastností a vazeb potřebných právě v danou chvíli. Tento problém částečně řeší lazy loading přístup a částečně je vyřešený ukládáním načtených her do paměti aplikace. Stále zůstává problém, že když se hráč chce podívat na hru, ke které dlouho nikdo nijak nepřistupoval, tak potřebuje stále načíst většinu informací v jednu chvíli, protože ke zobrazení herní stránky jsou potřeba načtené doopravdy všechny vykreslované entity i jejich vlastnosti.

Rozšiřitelnost je druhý přístup, který by naopak vytvářel ještě složitější počty a vazby entit potřebné k načítání. Směrem k rozšiřitelnosti míří definice surovin a staveb jako entit, které se dynamicky načítají z databáze a program je bez editace kódu přijme a zobrazí. Další rozšíření připadající v úvahu jsou:

- dynamicky generované tvary mapy, nebo tvary mapy načítající se z databáze;
- přidání více hráčů, a větších herních desek.

Návrhy na nové funkcionality jsou:

- přidání nových druhů políček, jako například výběr ze dvou nebo více surovin;
- kontrola podmínek pro stavbu staveb zakomponovaná přímo v entitě stavba a jejích vlastnostech ukládaných do databáze;
- přidání dalších entit ovlivňujících chování políček nebo staveb;
- přidání nových akčních karet, nebo bodovaných aspektů pro bodové hry.

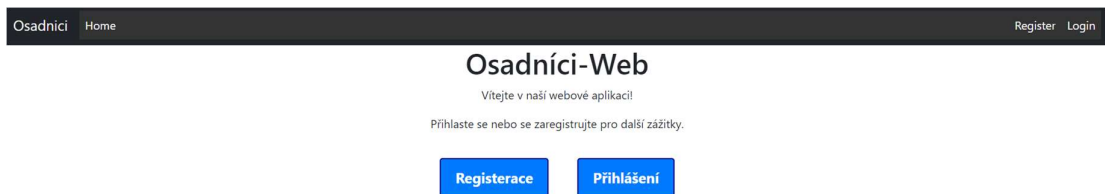
5 Výsledky a diskuse

V této kapitole bude předvedena výsledná aplikace, předvedena uživatelská příručka a aplikace bude porovnána s nalezenými srovnatelnými aplikacemi a z toho budou odvozené závěry.

5.1.1 Výsledná grafická stránka aplikace

Při první návštěvě stránky se uživateli zobrazí první úvodní obrazovka informující o nutnosti mít založený účet.

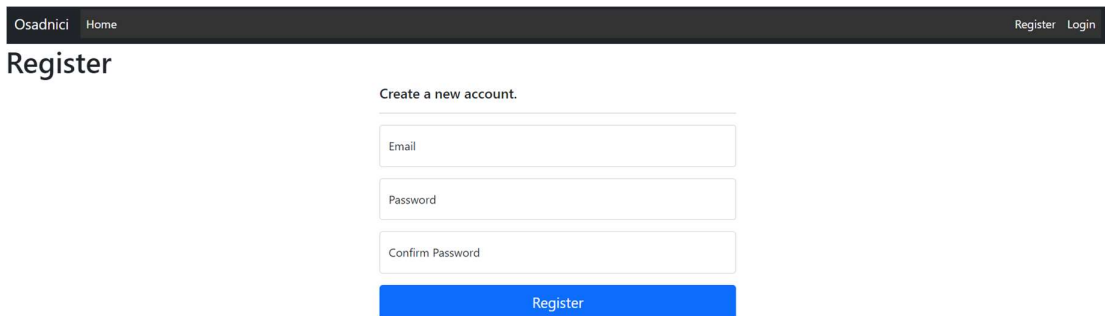
Obrázek 1 Úvodní stránka nepřihlášeného uživatele



Bakalářská práce Václav Steinhauser

Při registraci si uživatel zvolí uživatelské jméno ve formě e-mailu a heslo.

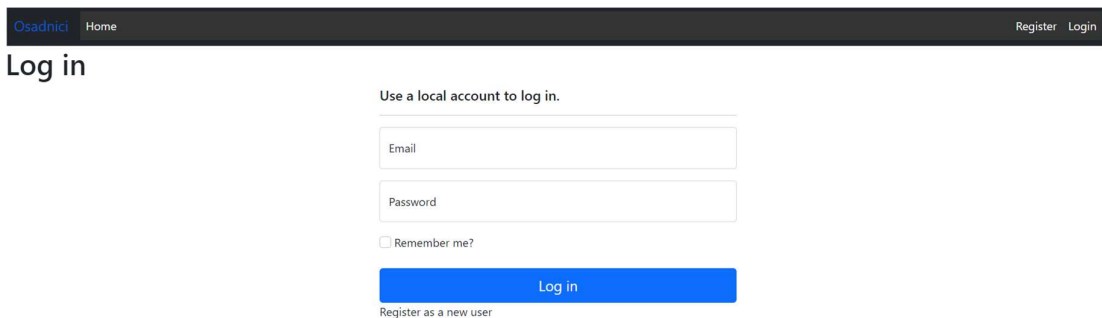
Obrázek 2 Stránka s registrací



Bakalářská práce Václav Steinhauser

Pokud uživatel už založený účet má může se přihlásit na stránce přihlášení. Volitelnou možností je uložit přihlášení, která způsobí, že si prohlížeč uloží přihlašovací token formou cookies a při příští návštěvě už uživatel bude přihlášený.

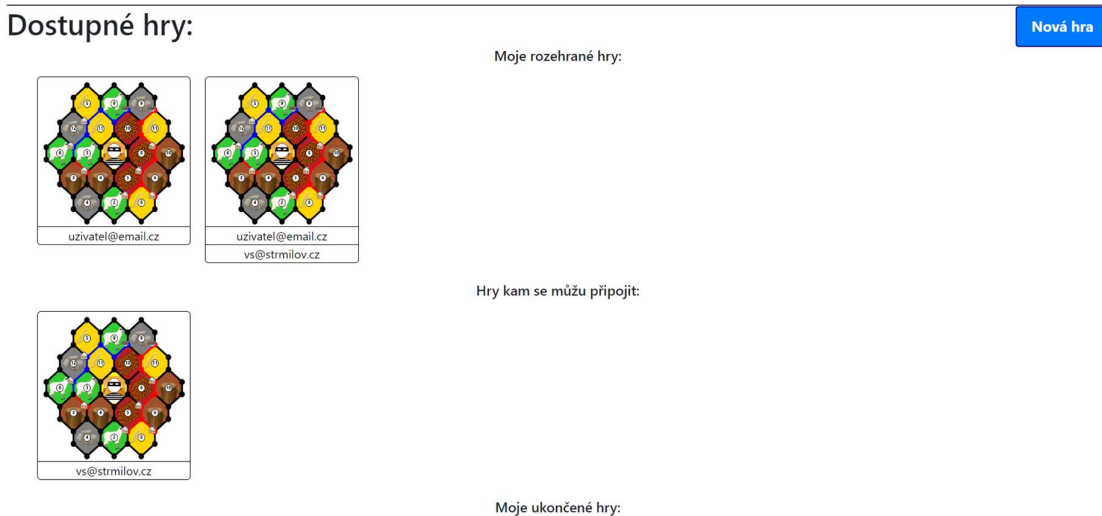
Obrázek 3 Přihlašovací stránka



Bakalářská práce Václav Steinhauser

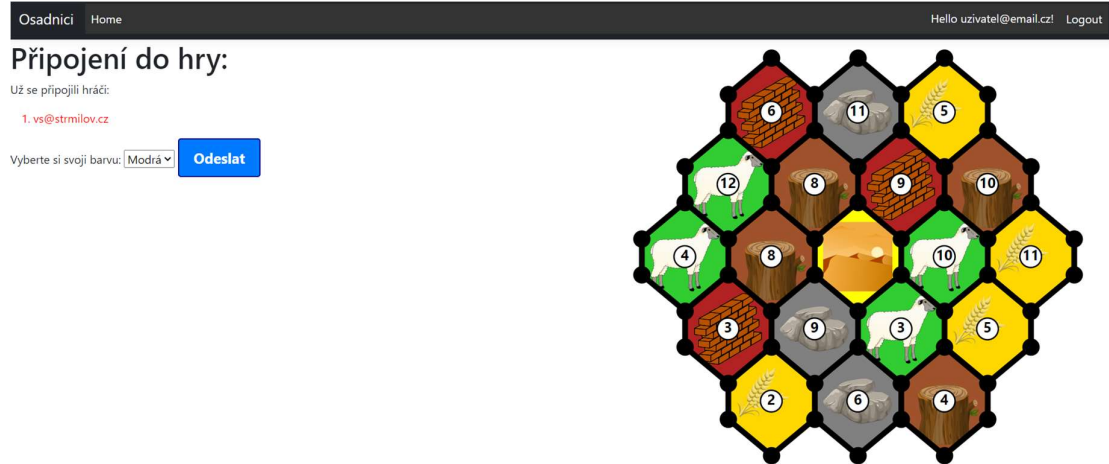
Po přihlášení se uživateli zobrazí úvodní stránka se seznamem her, ke kterým je připojený, nebo se může připojit.

Obrázek 4 Úvodní obrazovka přihlášeného uživatele



Pokud uživatel zvolí možnost připojení k nové hře zobrazí se mu seznam hráčů, kteří se už připojili s jejich barvami a na hráči je volba jeho barvy.

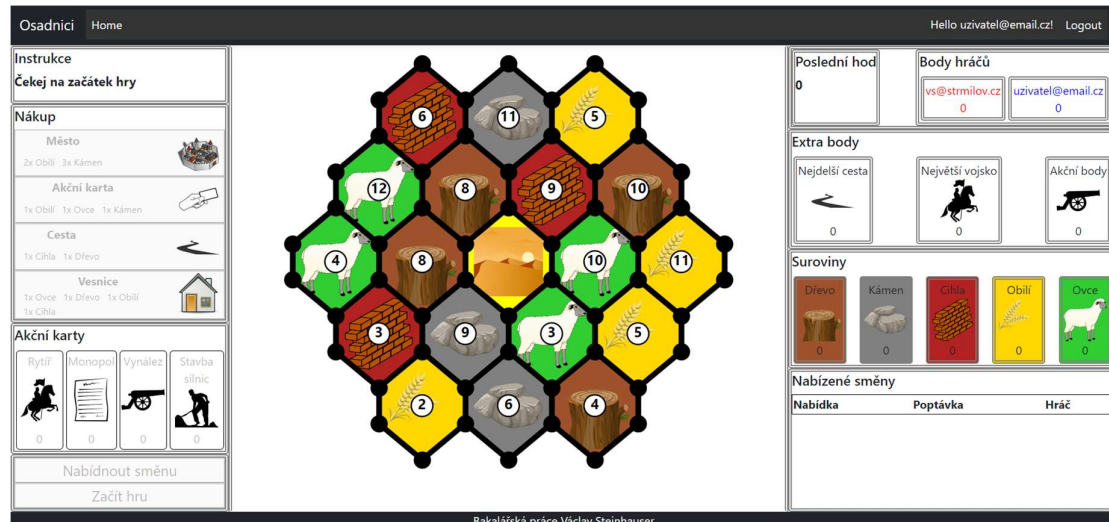
Obrázek 5 Připojení se ke hře



Bakalářská práce Václav Steinhauser

Po připojení (nebo přímém přechodu na připojenou hru z úvodní stránky) se hráči zobrazí stránka průběhu hry. Pak následuje průběh hry jako je popsáný v následující kapitole.

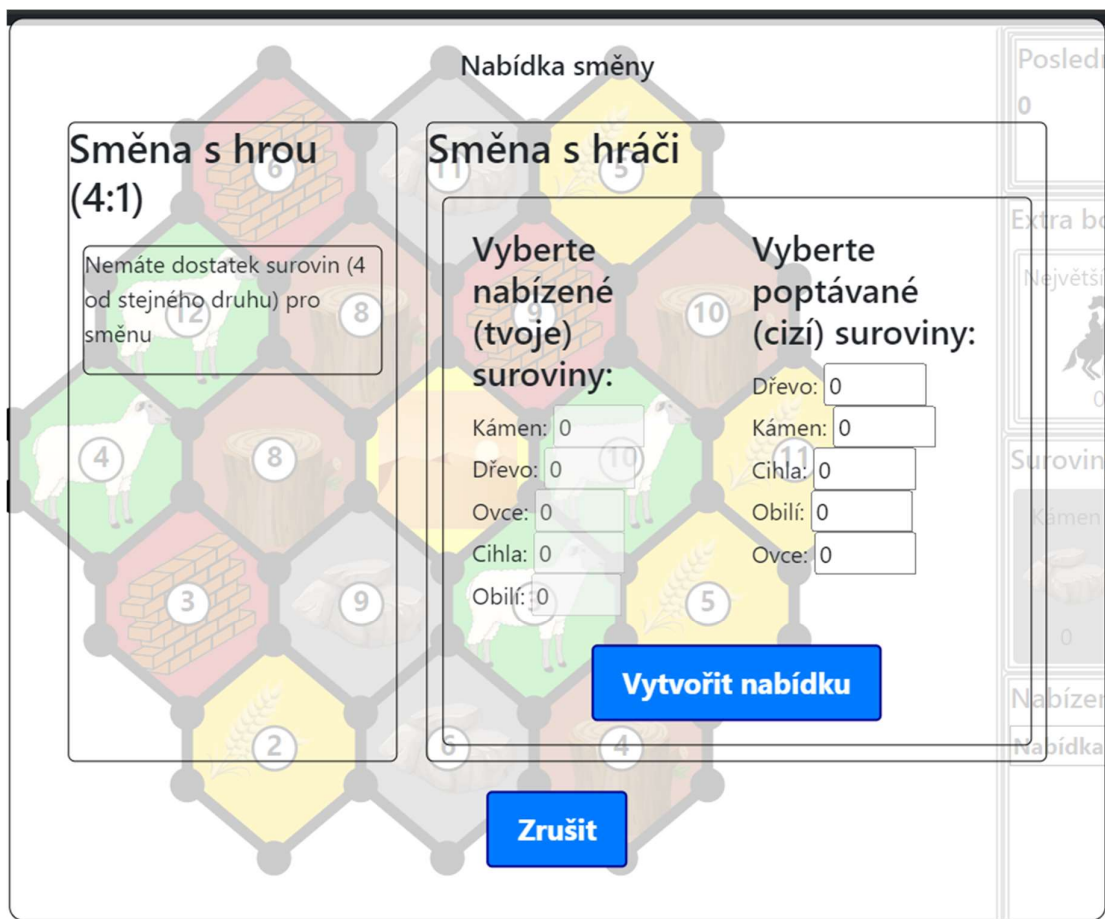
Obrázek 6 Stránka s průběhem hry



Bakalářská práce Václav Steinhauser

V průběhu hry může hráč ještě vytvářet směny v následujícím okně.

Obrázek 7 Nabídka směny



5.1.2 Uživatelská příručka

Hra je inspirovaná osadníky z katanu, takže většinu pravidel má stejných, ale v některých se přece jen liší, takže si je ukážeme v této kapitole.

Hlavní částí stránky sloužící ke komunikaci se serverem je sekce „Instrukce“ do té se vypisují instrukce pro hráče, nebo chybové hlášky při rozpoznání nesprávného chování. Celý levý panel slouží k vytváření nových aktivit a tlačítka na něm jsou aktivní jen pro aktuálního hráče, ale i to jen za určitých podmínek. Pravý panel pak ukazuje informace o stavu hry.

Speciální sekci je sekce „Nabízené směny“, ta jediná není omezena na aktuálního hráče. Pokud se hráč rozhodne že chce s někým vyměnit své karty, tak se všem ostatním hráčům zobrazí řádek tabulky, ve kterém jsou v sloupcích nabízené suroviny a jejich počty, vyžadované suroviny a jejich počty a uživatelské jméno nabízejícího hráče. Libovolný hráč může na toto tlačítko kliknout, to pak všem hráčům zmizí a dvěma hráčům, kteří uzavřeli

obchod se vymění suroviny. Nabídka směny se sama smaže až příště přijde na tah nabízející hráč, protože všichni hráči byli na tahu, nabídku viděli a nesouhlasili.

V momentě, kdy se připojí aspoň dva hráči se prvním hráči aktivuje tlačítko „Začít hru“. Hra začne po stisknutí tohoto tlačítka, nebo připojením maximálního počtu hráčů, což jsou 4. Tím začíná fáze rozestavování počátečních vesnic. Aby první hráč neměl výhodu rozestavují se vesnice dvě, a to v pořadí, že nejdřív postaví první vesnici a u ní cestu hráči od prvního k poslednímu a druhou vesnici s cestou staví jako první poslední hráč a končí první. Po dokončení rozestavování přichází na tah první hráč. Hráč na tahu má podle instrukcí následující možnosti:

- "Vyber místo pro stavbu vesnice." – Od hráče se očekává kliknutí na rozcestí, na kterém proběhne stavba vesnice, která má určitá pravidla. Žádná vesnice nesmí být postavena na rozcestí vzdáleném jen jednu cestu od jiné vesnice nebo města. Dále musí být k němu postavená cesta. Dále nesmí být políčko vlastněné jiným hráčem, což nastane, pokud daný hráč postaví cestu k němu a následně i navazující cestu dál. Na desce jsou vlastněné cesty a rozcestí vybarvené barvou hráče.
- "Vyber místo pro stavbu města." – Stavba města má pravidla jednodušší a je možná jen na místě, kde hráč má postavenou vesnici, která tím zaniká.
- "Vyber místo pro stavbu cesty." – I stavba cesty má svoje pravidla a to, že cesta nesmí být vlastněná jiným hráčem a musí navazovat na cestu nebo vesnici vlastněnou daným hráčem. Hráč, který má nejdelší souvislou cestu (ale minimálně délky 4) získá 2 výherní body v bodové kartě „Nejdelší cesta“.
- "Vyber políčko, kam se má přesunout zloděj." – Pokud při hodu kostkami padne číslo 7, nebo hráč zahraje akční kartu „Rytíř“ dostane tuto instrukci, po které se očekává kliknutí na herní pole, na které se přesune ikonka zloděje. Tím se políčko stane neaktivním až do přesunutí zloděje pryč.
- "Vyber hráče (vpravo nahoře), který sousedí s políčkem, kam přijde zloděj a chceš mu sebrat surovinu." – po přesunutí zloděje si hráč vezme jednu náhodnou surovinu od jednoho hráče, který má postavenou vesnici nebo město na okrajích tohoto pole. Tato instrukce očekává kliknutí na tabulku s hráčem (a jeho body), kterému chce surovinu vzít. Pokud cílový hráč nějaké karty má, pak se jedna náhodná přesune hráči, který je na tahu. Pokud žádné karty nemá tak má aktuální hráč smůlu.

- "Vyber surovinu, kterou chceš získat." a "Vyber druhou surovinu, kterou chceš získat." – tyto dvě instrukce očekávají kliknutí na kartičku se surovinami a pak se provede akce, podle toho, jaká aktivita hráče tuto instrukci vyvolala.
- "Vyber akci" – Tato instrukce se zobrazí, pokud hráč nemá žádnou vynucenou aktivitu a může si vybrat z akčního panelu.

Tlačítka na akčním panelu se zobrazují podle toho, jestli má hráč dost prostředků k jejich zahrání. Tlačítka v sekci „Nákup“ závisí na počtu surovin a dostupnosti. Akčních karet je ve hře omezené množství a stavby mají omezení, že někdy v danou chvíli není možné stavbu nikam umístit. Po kliknutí se hráči strhnou suroviny a pokud se jednalo o stavbu, tak se objeví instrukce k jejímu umístění, nebo pokud se jednalo o akční kartu tak se objeví v seznamu akčních karet. Akční karty a jejich počty jsou následující:

- 15x „Rytíř“ – Slouží k přesunutí zloděje a vzetí náhodné karty od jiného hráče. Dále hráč, který zahrál největší množství těchto karet (ale aspoň 3) získá 2 vítězné body, které se mu zobrazí v bodové kartě „Největší vojsko“.
- 2x „Stavba cest“ – Hráč, který tuto kartu zahraje získá zdarma dvě cesty, které ale musí hned postavit.
- 2x „Vynález“ – Po zahrání si hráč může dvakrát vybrat surovinu, od které dostane pokaždé jeden kus.
- 2x „Monopol“ – Když hráč zahraje tuto kartu, je po něm také vyžádáno vybrání suroviny, ale pak všem ostatním hráčům vezme všechny karty s touto surovinou.
- 5x „Výherní bod“ – Tato karta je speciální, protože je to bodová karta zamíchaná mezi akční karty. Tato karta se nijak nehraje a při jejím nákupu je rovnou přidán jeden výherní bod do bodové karty „Akční body“

Hráč si může vybrat v jakém tahu akční kartu zahraje, ale pak musí všechny její efekty využít okamžitě a nejdou odložit. Kdykoliv během tahu ale může vytvořit směnu. Směny mají dvě možnosti, buď okamžitá směna 4 karet jedné suroviny za jednu kartu libovolné suroviny, nebo vytvoří nabídku ostatním hráčům. Pokud hráč nabízené suroviny utratí, tak se směna na konci jeho tahu automaticky zruší.

Po ukončení tahu hráče aplikace nasimuluje hod dvou kostek, následně součet vypíše hráčům do panelu „Posední hod“. Pokud je součet 7, tak vytvoří instrukci pro přesun zloděje pro dalšího hráče a nic dalšího se neděje. Pokud padne jiné číslo tak se podívána políčka s tímto číslem a každému městu které s políčkem sousedí dá dvě karty suroviny z políčka.

Vesnice, které s políčkem sousedí dostanou kartu se surovinou jednu. Pokud je na políčku zloděj tak ale z daného políčka nikdo nic nedostane.

Cílem hry je získat 10 vítězných bodů, které hráč získá z bodových karet, ale také dostane dva body za každé postavené město a jeden bod za každou vesnici. Jakmile jeden z hráčů dosáhne této hranice tak na konci jeho tahu se hra označí za ukončenou a žádný hráč už nemůže provádět žádné akce.

5.1.3 Porovnání s podobnými aplikacemi

Podobná aplikace je například na webu <https://blazorgames.net/> který poskytuje několik klasických tahových her pro hráče. Rychlost a hratelnost této aplikace s cizí aplikací byla srovnatelná. Velká výhoda této aplikace spočívá v neztracení herního postupu při odpojení nebo obnovení stránky.

6 Závěr

V této práci zaměřené na návrh a implementaci webové aplikace pro adaptaci deskové hry inspirované hrou Osadníci z Katanu bylo hlavním cílem vytvoření prostředí umožňujícího hráčům hrát tuto deskovou hru online a ukládat stav rozehraných her. Práce byla rozdělena do teoretické a praktické části.

V rámci teoretické části bylo provedeno studium existujících informačních zdrojů týkajících se deskových her a webových technologií, které poskytlo potřebné znalosti pro návrh a implementaci aplikace.

Praktická část byla zaměřena na konkrétní realizaci aplikace s využitím získaných poznatků a metodiky vývoje. Byl proveden návrh a implementace klíčových komponent aplikace, včetně databáze pro ukládání her, jádra aplikace obsahujícího pravidla hry a prostředí v ASP.NET doplněného o SignalR sloužícího ke komunikaci s ostatními částmi.

Po nasazení a testování aplikace byly zváženy možnosti optimalizace a rozšiřování. Využitý byl programovací jazyk C# s frameworky ASP.NET a Entity Framework s napojením na databázový systém MariaDB.

Celkově lze říct, že tato práce představuje komplexní přístup k vytvoření online platformy pro hraní deskových her s důrazem na uživatelskou přívětivost, efektivitu a rozšiřitelnost.

7 Seznam použitých zdrojů

CATAN, 2020. Herní pravidla hry Osadníci z Katanu. Online [PDF]. Catan.com. Dostupné z: https://www.catan.com/sites/default/files/2021-06/catan_base_rules_2020_200707.pdf. [cit. 2024-03-13].

CATAN, 2024. About Us. Online. CATAN. Dostupné z: <https://www.catan.com/about-us>. [cit. 2024-03-13].

DOCKER INC., 2024. Docker overview. Online. Docker Documentation. Dostupné z: <https://docs.docker.com/get-started/overview/>. [cit. 2024-03-13].

FREEMAN, Adam, 2021. Pro ASP.NET Core Identity: Under the Hood with Authentication and Authorization in ASP.NET Core 5 and 6 Applications. Apress. ISBN 978-1-4842-6858-2.

HOUSER, Pavel, 2017. AlphaZero: Obrovský triumf umělé inteligence v šachách. Sciencemag.cz. 7. 12. 2017. Dostupné také z: <https://sciencemag.cz/alphazero-obrovsky-triumf-umele-inteligence-v-sachach/>.

Hrajeme.cz, 2024. Online. Dostupné z: <https://www.hrajeme.cz/hry/11111/>. [cit. 2024-03-13].

MARIADB, 2024. MariaDB in brief. Online. MariaDB.org. Dostupné z: <https://mariadb.org/en/>. [cit. 2024-03-13].

MARIADB, 2024. MariaDB Server Documentation. Online. MariaDB KnowledgeBase. Dostupné z: <https://mariadb.com/kb/en/documentation/>. [cit. 2024-03-13].

MDN, 2024. JavaScript. Online. MDN Web Docs. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/JavaScript>. [cit. 2024-03-13].

MICROSOFT, 2022. Introduction to C#. Online. Microsoft Learn. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/csharp/language-reference/language-specification/introduction>. [cit. 2024-03-13].

MICROSOFT, 2023. Overview of .NET Framework. Online. Microsoft Learn. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/framework/get-started/overview>. [cit. 2024-03-13].

MICROSOFT, 2023. Overview of ASP.NET Core MVC. Online. Microsoft Learn. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/mvc/overview>. [cit. 2024-03-13].

MICROSOFT, 2023. Overview of ASP.NET Core. Online. Microsoft Learn. Dostupné z: <https://learn.microsoft.com/en-us/aspnet/core/introduction-to-aspnet-core>. [cit. 2024-03-13].

MICROSOFT, 2024. Introduction to .NET. Online. Microsoft Learn. Dostupné z: <https://learn.microsoft.com/en-us/dotnet/core/introduction>. [cit. 2024-03-13].

NAYYERI, Keyvan a WHITE, Darren, 2014. Pro ASP.NET SignalR. Apress. ISBN 978-1430263197.

PÍSEK, Slavoj, 2003. ASP.NET - začínáme programovat: podrobný průvodce začínajícího uživatele. Praha: Grada. ISBN 80-247-0526-5.

SOLIS, Daniel a SCHROTENBOER, Cal, 2018. Illustrated C# 7. Apress. ISBN 978-1-4842-3288-0.

VEMULA, Remi, 2017. Real-Time Web Application Development. Apress. ISBN 978-1-4842-3269-9.

VOGEL, Eric, 2021. Beginning Entity Framework Core 5: From novice to Professional. Apress. ISBN 978-1-4842-6882-7.

ZAPLETAL, Miloš, 1991. Velká kniha deskových her. Praha: Mladá fronta. ISBN 80-204-0188-1.

8 Seznam obrázků

Obrázek 1 Úvodní stránka nepřihlášeného uživatele.....	31
Obrázek 2 Stránka s registrací	31
Obrázek 3 Přihlašovací stránka.....	32
Obrázek 4 Úvodní obrazovka přihlášeného uživatele	32
Obrázek 5 Připojení se ke hře	33
Obrázek 6 Stránka s průběhem hry	33
Obrázek 7 Nabídka směny	34

Přílohy

Příloha 1 Zabalené řešení Visual Studia doplněné o konfigurační soubory k vytvoření Docker kontejneru s databází potřebnou pro běh aplikace.