Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačního inženýrství

Bakalářská práce

Vývoj webové aplikace online konfigurátoru únikových her

Iveta Čakovská

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Iveta Čakovská

Systémové inženýrství a informatika
Informatika

Název práce

Vývoj webové aplikace online konfigurátoru únikových her

Název anglicky

Web application development for the online configurator of escape games

---

Cíle práce

Hlavní cíl bakalářské práce je vývoj webové aplikace s využitím jazyka PHP. Jedná se o návrh a tvorbu webové aplikace online konfigurátoru únikových her. Aplikace bude vycházet z klasických únikových her, kde bude možnost navolit si další možnosti hry.
Dalším cílem je popsat postup a využité technologie pro splnění hlavního cíle.

Metodika

Práce je rozdělena na dvě části, teoretickou a praktickou.

Teoretická část se bude zabývat popisem jazyka PHP, programovacími jazyky a ostatními programy využitých při tvorbě, které budou vycházet z odborné literatury.

Praktická část spočívá v návrhu a tvorbě webové aplikace. Back-end bude vytvořen v jazyce PHP a front-end v HTML a CSS. Aplikace bude vytvořena pomocí PHP frameworku Nette lokálním vývojem a následně popsán postup.

Doporučený rozsah práce

35-40 stran

Klíčová slova

PHP, Nette, webová aplikace, HTML, CSS

Doporučené zdroje informací

Hopkins, C. PHP okamžitě. Brno : Computer Press, 2014. ISBN 978-80-251-4196-0
VRÁNA, J. 1001 tipů a triků pro PHP. Brno : Computer Press, 2010. ISBN 978-80-251-2940-1
WELLING,L.THOMPSON, L. Mistrovství PHP a MySQL. Brno : Computer Press, 2017. ISBN 978-80-251-4892-1

Předběžný termín obhajoby

2020/21 LS – PEF

Vedoucí práce

Ing. Jiří Brožek, Ph.D.

Garantující pracoviště

Katedra informačního inženýrství

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelikán, Ph.D.

Vedoucí katedry

Elektronicky schváleno dne 23. 2. 2021

Ing. Martin Pelikán, Ph.D.

Děkan

V Praze dne 02. 03. 2021

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci "Vývoj webové aplikace online konfigurátoru únikových her" jsem vypracovala 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 autorka uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušila autorská práva třetích osob.

V Praze dne 15.3.2021 _____

Poděkování

Ráda bych touto cestou poděkovala Ing. Jiřímu Brožkovi, Ph.D. za cenné rady, trpělivost a čas, který mi věnoval při realizaci této práce.

Vývoj webové aplikace online konfigurátoru únikových her

Abstrakt

Tato bakalářská práce se zabývá vývojem webové aplikace online konfigurátoru únikových her v jazyce PHP frameworku Nette. Aplikace umožňuje nakonfigurovat si vlastní únikovou hru.

V teoretické části jsou popsány využité technologie a nástroje pro vývoj této webové aplikace. Praktická část popisuje analýzu společností nabízející únikové hry dle přání zákazníka. Dále popisuje byznysové požadavky, specifikaci požadavků, návrh a samotný vývoj aplikace. Zde byly popsány nejdůležitější části zdrojového kódu Admin modulu, Front modulu a samotné konfigurace. K závěru bylo provedeno testování a návrhy na další vylepšení aplikace.

Klíčová slova: PHP, Nette, MySQL, Webová aplikace, Úniková hra, Docker, CSS, HTML

Web application development for the online configurator of escape games

Abstract

This bachelor thesis deals with the web application development for the online configurator of escape games written in PHP language for Nette Framework. The application allows to set own escape game.

The theoretical part describes the technologies and tools used to develop this web application. The practical part is concerned with the analysis of companies offering escape games according to the customer's wishes. Further it describes business requirements, specific requirements, design and development of the application itself. Important parts of the source code Admin module, Front module and configuration were described here. Finally, testing and suggestions for further improvement of the application were proved.

Keywords: PHP, Nette, MySQL, Web application, Escape game, Docker, CSS, HTML

Obsah

Seznam obrázků

Seznam drátových modelů

Seznam zdrojových kódů

1 Úvod

Únikové hry jsou fyzické hry založené na logice, kde je hlavním úkolem uniknout z uzavřené místnosti pomocí stop, vyřešením různých hádanek a úkolů ve stanoveném čase. Únikové hry vyhledávají především lidé, kteří řeší rádi rychle logické úkoly.

Inspirací pro tuto bakalářskou práci mi byla záliba v hraní únikových her. Doposud jsem na internetu nenalezla možnost volného nakonfigurování únikové hry dle mých představ s následnou možností fyzicky se hry zúčastnit. Běžně je možno hru vytvořit na přání pouze v případě kontaktujeme společnost, která hry na přání nabízí. Tento způsob může některé zákazníky odradit od zdlouhavé komunikace s danou společností poskytující únikové hry na přání.

Postupu při nastavení únikové hry, kterou jsem se zabývala v bakalářské práci, dají zákazníci dle mého názoru přednost před zdlouhavou komunikací se společností poskytující únikové hry na přání. Již při pohledu na uživatelské rozhraní bude mít budoucí účastník hry možnost zapojit svou fantazii a zvolit si kombinace žánru, tématu, či postavy, které jsou mu blízké.

V této práci je popsán návrh a následná implementace webové aplikace, která umožní nakonfigurovat únikovou hru podle představ uživatele, bez nutnosti kontaktování dané společnosti.

2 Cíl práce a metodika

2.1 Cíl práce

Hlavní cíl bakalářské práce je vývoj webové aplikace s využitím jazyka PHP. Jedná se o návrh a tvorbu webové aplikace online konfigurátoru únikových her. Aplikace bude vycházet z klasických únikových her, kde bude možnost navolit si další možnosti hry.
Dalším cílem je popsat postup a využité technologie pro splnění hlavního cíle.

2.2 Metodika

Práce je rozdělena na dvě části, teoretickou a praktickou.

Teoretická část se bude zabývat popisem jazyka PHP, programovacími jazyky a ostatními programy využitých při tvorbě, které budou vycházet z odborné literatury.

Praktická část spočívá v návrhu a tvorbě webové aplikace. Back-end bude vytvořen v jazyce PHP a front-end v HTML a CSS. Aplikace bude vytvořena pomocí PHP frameworku Nette lokálním vývojem a následně popsán postup.

Východiskem webové aplikace bude online konfigurátor únikových her, který bude rozdělen na Admin modul pro administraci a Front modul pro uživatele. V tomto konfigurátoru bude možnost navolit si vlastní únikovou hru z výběrů dle představ zákazníka.

3 Teoretická východiska

3.1 Základní pojmy

3.1.1 Webová aplikace

Webová aplikace označuje aplikaci, u které není zapotřebí instalace, ale je třeba internetové připojení a webový prohlížeč, jelikož běží na straně serveru. Rozdíl mezi webovou aplikací a klasickou webovou stránkou spočívá především v tom, že webová aplikace dokáže plnit složitější úlohy a využívá databáze. Nejznámějšími příklady webové aplikace je kupříkladu Facebook, Gmail, GoogleDocs atd. Dalšími výhodami je například, že uživatel není povinen nic aktualizovat, protože aktualizace probíhá na straně serveru. Uchovávání a zálohování dat je prováděno na serveru, čímž je umožněn přístup odkudkoliv. Co se týče bezpečnostních rizik, lze je snížit výběrem kvalitního poskytovatele nebo tím, že aplikace poběží pouze na lokální síti určité organizace. (1)

3.1.2 Backend

Backend znamená tu část webové stránky, aplikace, která je uživateli skryta. Výraz backend je složen ze dvou anglických slov „back“ a „end“, jedná se tedy o jakési pozadí stránek, vysvětluje otázku, jak všechno funguje. Backend je na straně serveru. Opak backendu je frontend, díky kterému se uživatelé dostanou do nepřímého kontaktu právě s backendovou částí aplikace. Frontend bude vysvětlen v kapitole 3.1.3. Frontend. Mezi hlavní úkoly patří uspořádávání dat a zajišťování správné funkčnosti na straně klienta. Pod tuto část spadá také psaní API, tvorba knihoven nebo práce se systémovými komponentami bez uživatelských rozhraní. Data, která jsou vygenerována v backendu jsou následně předána do frontendu a vyobrazena uživateli. Procesy v backendové části jsou například zpracování příchozího požadavku na webovou stránku, spouštění skriptů k vygenerování HTML, práce s databází, šifrování a dešifrování dat. (2) (3)

Mezi nejčastější backend programovací jazyky patří PHP, C++, Java, Python, JavaScript a Node.js. (2)

3.1.3 **Frontend**

Frontend představuje tu část webové stránky, aplikace, která s uživatelem přímo interaguje, a jíž rovněž vidí. Zjednodušeně se dá říct, že frontend představuje vzhled a označuje se také jako „klientská strana“ aplikace. Do frontendu patří vše, co uživatel vidí na webové stránce či aplikaci, jako jsou například barvy, obrázky, tabulky, navigační menu atd. Vše, co se zobrazí při otevření prohlížeče, implementuje frontend vývojář, ať už to je design, chování dané aplikace, struktura anebo obsah. Hlavními cíli frontendu jsou odezva a výkon - je potřeba zajistit správné fungování webu na všech zařízeních. Mezi frontend programovací jazyky patří především HTML, CSS a JavaScript. (2)

3.1.4 **OOP**

Objektově orientované programování je paradigma opírající se o koncept objektů a tříd. Využívá se pro strukturování softwarového programu do jednoduchých, opakovaně použitelných částí kódu, které slouží k vytváření jednotlivých instancí objektů. (4)

Objekt je samostatná entita obsahující data a funkčnost. Třída je množina objektů, která má určité vlastnosti, nedefinuje konkrétní objekty dané třídy, ale pouze udává, jaké vlastnosti bude mít objekt té třídy. Instance je konkrétní objekt určité třídy. Třídy mohou obsahovat funkce, kterým se říká metody dostupné pouze pro objekty daného typu. Tyto funkce jsou definovány ve třídě a provádějí některé akce užitečné pro tento konkrétní typ objektu. (4)

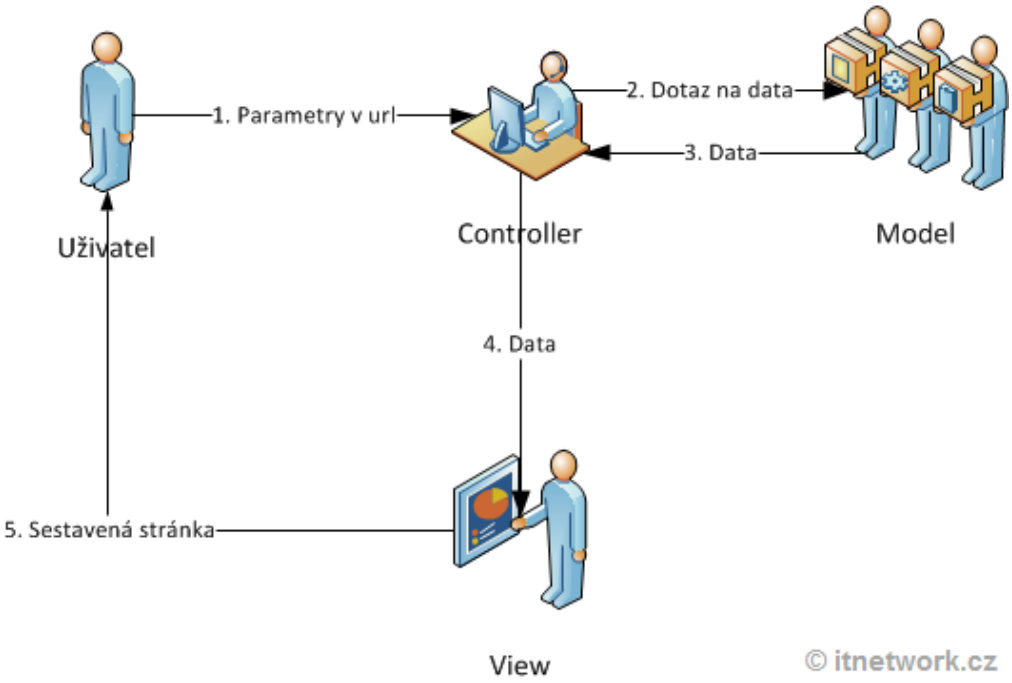
Mezi OOP programovací jazyky patří třeba JavaScript, C++, Java nebo Python. (4)

3.1.5 **MVC**

MVC je zkratka pro Model-view-controller. MVC je softwarová architektura založena na principu rozdělení datového modelu aplikace, uživatelského rozhraní a řídící logiky do oddělených částí, aby úprava jednoho z uvedených měla minimální vliv na ostatní. MVC architektura bude rovněž představena u frameworku Nette v kapitole 3.2.1. Nette. (5) (6)

Model (Model) představuje veškerou logiku související s daty. Ať už to jsou data, která jsou přenášena mezi komponentami View a Controller, nebo jakákoliv jiná data související s byznysovou logikou. Tato logika má na starost získávání dat, výpočty, určení zda je něco

pravda nebo nepravda atd. View (Pohled)představuje zobrazení výstupu uživateli. View je vlastně zobrazovač výstupu, který obsahuje minimální množství logiky. Controller (Kontroler) je jakýsi prostředník mezi uživatelem, modelem a viewem. Spojuje celý systém dohromady, a zároveň propojuje další komponenty. (5) (6)



Obrázek 1- MVC architektura (zdroj:https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor)

MVC odděluje pohledy a modely vytvořením protokolu pro přihlášení, nebo oznámení mezi nimi. Pohled má povinnost zajistit, aby jeho vzhled odrážel stav modelu. Pokud se data modelu změní, závisející pohledy dostanou upozornění, čímž vznikne příležitost pro pohled, aby se mohl sám aktualizovat. Díky tomuto přístupu se umožní připojit více pohledů k modelu. Je možné vytvořit nový pohled na model bez potřeby přepsání. Hlavní vztahy MVC jsou dány návrhovými vzory Observer, Composite a Strategy. (7)

3.2 PHP

Programovací skriptovací jazyk PHP se využívá především pro programování dynamických webových aplikací. Byl vytvořen Rasmusem Lerdofem v jazyce C v roce 1995 se záměrem sledování jeho online životopisu a osobních údajů, proto byl původní název „Personal Home Page“, nyní však zkratka PHP znamená „Hypertext Preprocessor“. Jedná

se o jazyk pro webové stránky zabudovaný do HTML, díky čemuž lze kód PHP vložit do HTML webové stránky. (8) (9) (10)

Po přístupu na stránku je kód PHP zkontrolován serverem dané stránky, výstup funkcí PHP se obvykle vrací jako HTML kód, který je prohlížeč schopen přečíst. Uživatelé nemohou na stránce zobrazit kód PHP z toho důvodu, že se transformuje při načtení stránky do HTML. Tato transformace zajišťuje dostatečnou bezpečnost pro přístup k databázím a dalším zabezpečeným informacím. Ovšem díky řadě dalších funkcí umožňuje tento jazyk webovým vývojářům snadno a rychle psát dynamicky generované stránky. Nejvýznamnější funkce PHP je v podporování široké škály databází. Pro vývoj webových aplikací se tento jazyk považuje jako nejoblíbenější. (8) (9)

Syntaxe PHP přebrala většinu z jazyků C, Javy a Perlu. Velikost psaní písmen nemá v jazyce PHP jednotný řád. Přesto se proměnné a funkce píší převážně s malým písmenem na začátku. Pro lepší čitelnost se zkratky, které jsou součástí identifikátoru, píší s malým písmenem. Velká písmena na začátku se využívají pro psaní názvů tříd. Pro uzavření kódu v PHP jsou použitelné čtyři způsoby: <?php ?>, <? ? >, <% %>, <script language="php"> </script>. (11)

3.2.1 **Nette**

Frameworky ulehčují programátorovi operace, které se opakují, a nemusí psát stejný kód několikrát. Navíc díky frameworku se práce stává přehlednější, jelikož je rozdělena do několika sekcí. Nette je postaveno na MVC architektuře a jedná se objektově orientované programování. (12)

Nette je plnohodnotný framework PHP, který je vytvořen sadou znovupoužitelných balíčků. Tyto balíčky se dají využívat nezávisle na zbytku frameworku, přičemž se instalují pomocí Composeru. Zmíněný Composer je nástroj pro správu závislostí v PHP, který umožňuje vyjmenovat knihovny, na kterých projekt závisí. Ty se budou automaticky instalovat a aktualizovat. (13) (14) (15)

Na následujícím obrázku je adresářová struktura frameworku Nette, sandbox je v tomto případě kostra webové aplikace. Aplikace v Nette je postavena na komponentách tří typů, jež se dělí o tři základní úlohy. Prvním typem je presenter, který má na starost řízení, přes který komunikuje s uživatelem. Model je druhý typ, který obsahuje logiku aplikace, např. práci s databází. Posledním typem jsou Pohledy neboli templates. Jde o šablony s HTML kódem, jež mají na starost výstup. Životní cyklus je následující: Uživatel předá požadavek routeru, který podle URL adresy rozpozná požadavek uživatele a zavolá mu příslušný presenter. Presenter podle parametru zjistí požadavek, který si získá potřebný model. Model z databáze získá potřebná data, která předá zpět presenteru, ty se následně předají šabloně, jež obsahuje HTML stránku pro daný požadavek. Data se vloží do Latte značek v šabloně, která pošle výsledný HTML výstup presenteru a ten je zašle uživateli. Následně se uživateli zobrazí HTML stránka se splněným požadavkem. (12) (16)

```
sandbox/
├── app/                    ← adresář s aplikací
│   ├── Model/              ← třídy modelové vrstvy
│   ├── Presenters/         ← presentery a šablony
│   │   ├── HomepagePresenter.php  ← třída presenteru Homepage
│   │   └── templates/      ← adresář se šablonami
│   │       ├── @layout.latte ← šablona layoutu
│   │       └── Homepage/     ← šablony presenteru Homepage
│   │           └── default.latte  ← šablona akce 'default'
│   ├── Router/             ← konfigurace URL adres
│   └── Bootstrap.php       ← zaváděcí třída Bootstrap
├── bin/                    ← skripty spouštěné z příkazové řádky
├── config/                 ← konfigurační soubory
│   ├── common.neon
│   └── local.neon
├── log/                    ← logované chyby
├── temp/                   ← dočasné soubory, cache, …
├── vendor/                 ← knihovny instalované Composerem
│   ├── ...
│   └── autoload.php        ← autoloading všech nainstalovaných balíčků
├── www/                    ← veřejný adresář neboli document-root projektu
│   ├── images/             ← další adresáře, třeba pro obrázky
│   ├── .htaccess           ← pravidla mod_rewrite
│   └── index.php           ← prvotní soubor, kterým se aplikace spouští
└── .htaccess               ← zakazuje přístup do všech adresářů krom www
```

Obrázek 2 Adresářová struktura Nette (zdroj: https://doc.nette.org/cs/3.1/application)

Nette používá šablonovací systém Latte s podporou HTML, který především zamezuje zranitelnosti webové aplikace před XSS. XSS je zkratkou pro Cross-Site-Scripting, jde o útok, který je založen na principu vložení kódu do klasického HTML. Jedná

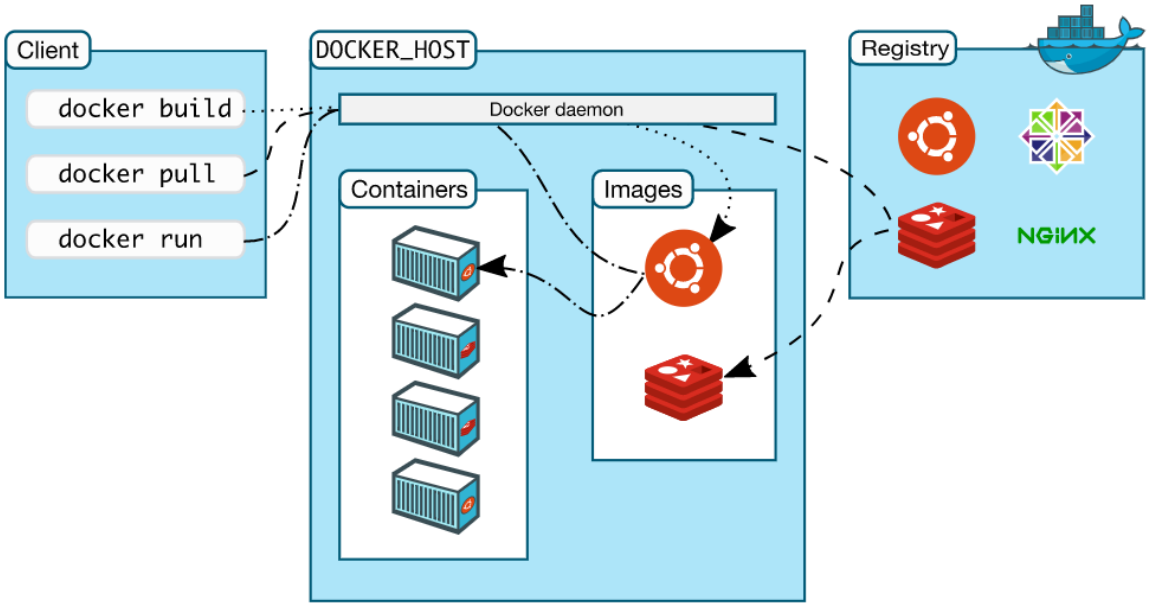
se o nejvíce zabezpečený šablonovací systém pro PHP, z něhož také vychází syntaxe. Po každé změně ve zdrojovém souboru se automaticky vygeneruje šablona, jejíž změny jsou okamžitě vidět na výstupu. (16) (17)

Tracy je nástroj využívaný pro ladění kódu PHP. Hlavní přínos spočívá v rychlém odhalováním a logováním chyb, vypisováním proměnných, sledování paměťových nároků, měření času skriptů a databázových dotazů. Tracy neboli laděnka se přidá do kódu aplikace. Při chybě se zobrazí konkrétní část zdrojového kódu se zvýrazněným řádkem, kde navíc vypíše, o jakou chybu se jedná. (18)

3.3 **Docker**

Docker je otevřený software využívající kontejnery pro rozběhnutí různých aplikací pomocí lokálního vývoje. Architektura Dockeru je založena na modelu klient-server, klient komunikuje s Docker daemonem, který buduje, spouští a distribuuje kontejnery. Deamon je trvalý proces na pozadí spravující kontejnery u jednoho hostitele. Také spravuje objekty, jako jsou obrazy, kontejnery, sítě a úložiště. Rozhraní REST API je využíváno při komunikaci klienta s daemonem Dockeru. Komunikace probíhá přes UNIX socket nebo síťové rozhraní. Specifikací Dockeru je schopnost zabalit a rozběhnout aplikaci ve volně izolovaném prostředí, které se nazývá kontejner. Tento software zjednodušuje vývojářům životní cyklus vývoje. Následující Obrázek č.3 představuje architekturu softwaru Docker. (19) (20)

Obraz (image) je pouze čitelná šablona s instrukcemi pro vytvoření kontejneru. Pro vytvoření vlastního obrazu se vytvoří Dockerfile s jednoduchou syntaxí, která definuje kroky potřebné k vytvoření obrazu a jeho spuštění. Příkazem docker image build se vytvoří obraz z Dockerfile. Každá instrukce v Dockerfile si vytvoří vrstu v obrazu. Co dělá obraz rychlým, malým a lehkým v porovnání s ostatními virtualizačními technologiemi je to, že pokud se změní Dockerfile a přebuduje se obraz, tak se změní pouze vrstvy, které byly přebudovány. Je možné vytvářet buď vlastní obrazy, nebo použít již vytvořené a publikované ostatními ve veřejném registru Docker Hub. (19)



Obrázek 3 - Architektura Dockeru (zdroj:https://docs.docker.com/get-started/overview/)

Docker kontejner je spustitelná instance obrazu. Kontejner je abstrakce na vrstvě aplikace, která zabalí kód spolu se závislostí dohromady. Jeden hostitel může provozovat současně několik kontejnerů. Výhodou oněch kontejnerů je skutečnost, že zvládnou více aplikací, a oproti virtuálním počítačům zabírají méně místa. Nadefinování kontejneru je prováděno pomocí jeho obrazu a konfiguracemi, které se navolí při vytváření nebo spouštění. (19) (20)

3.4 MySQL

MySQL je otevřený systém pro správu relačních databází (RDBMS). Pro práci s daty v těchto databázích je využíván jazyk SQL, který je zkratkou pro Structured Query Language. Jedná se o strukturovaný dotazovací jazyk, využívající se pro přidávání, odebírání a další úpravy informací v databázi. Tento jazyk je k dispozici v databázových systémech MySQL, PostgreSQL, Microsoft SQL Server, Oracle a mnoha dalších. Mezi standardní příkazy SQL patří například ADD, DROP, UPDATE, INSERT, které lze využít společně s MySQL. Každá databáze obsahuje tabulky, přičemž každou tabulku tvoří sloupce a řádky. Každý jednotlivý záznam má předem nadefinovaný datový typ. (21) (22) (23)

I MySQL je založen na síťové architektuře klient-server. V tomto případě je MySQL na straně serveru. Klient může zadat požadavek konkrétního příkazu SQL, na který serverová aplikace zodpoví požadovanou informací a zobrazí se na straně klienta. Dynamické webové stránky generují obsah z databáze při načítání každé jednotlivé webové stránky. (21) (22)

Mezi hlavní přednosti MySQL patří kupříkladu rychlý výkon, snadnost použití, využitelnost jak ve Windows, tak v dalších unixových operačních systémech. MySQL nabízí vysoký výkon za poměrně nízkou cenu. Pod licencí otevřeného softwaru je zdarma, pro komerční účely je však již vyžadován poplatek. (23)

Pro správu MySQL přes web slouží softwarový nástroj zdarma phpMyAdmin, který byl napsán v jazyce PHP. Jedná se o webovou aplikaci, která uživatelům umožňuje spravovat jejich databáze pomocí webového grafického rozhraní. Tato aplikace podporuje mnoho funkcí MySQL vyhledávání, odstraňování, kopírování, přejmenovávání a mnoho dalších. Přes phpMyAdmin se dají také například exportovat data do formátů CSV, SQL, XML, PDF. (24)

PostgreSQL, Microsoft SQL Server a Oracle patří mezi hlavní konkurenty MySQL, přesto však MySQL nabízí mnoho předností, díky kterým je stále nejčastější a nejoblíbenější volbou pro výběr databázového systému. (23)

3.5 **HTML**

HTML je zkratka pro „Hypertext Markup Language“, jedná se o značkovací jazyk, který je využívaný k vytváření webových stránek. Ačkoliv se webové stránky staly modernějšími a interaktivnějšími, značkovací jazyk HTML 5 se zjednodušil tak, že je zapotřebí méně kódu oproti předešlým verzím HTML 4 nebo XHTML 1.0. Hlavní důvod je ten, že při formátování většiny prvků HTML 5 spoléhá na kaskádové šablony stylů nebo JavaScript. (25)

Správný formát zahrnuje značky s elementy <html>,<head> a <body>. Hlavička HTML dokumentu <head> obsahuje většinou název stránky, metadata a odkazované soubory. Samotný obsah stránky se vloží mezi značky <body>. Mezi značkami s elementem <meta> je obsažen způsob určení kódování českého jazyka na stránce. MS Office programy

využívají kódování windows-1250, zatímco kódování utf-8 se využívá v databázi MySQL. Při tvorbě webových stránek se používá především utf-8, neboť umožňuje rozsáhlejší výběr znaků než windows-1250. (26) (25)

3.6 CSS

Zkratka CSS se užívá pro Cascading Style Sheets, jak už napovídá samotný název, je tvořen pomocí kaskádových stylů. Programovací jazyk CSS určuje vzhled webové stránky (barvy, obrázky, rozmístění) v HTML souboru. Bez onoho HTML souboru je sám o sobě nepoužitelný, je tedy potřeba, aby byl propojen právě s HTML souborem. CSS se dá propojit například i s XML. (27) (28)

Pomocí CSS je možno místo definování každé tabulky, každého bloku textu v HTML stránky definovat pouze jednou. Usnadňuje změnu stylů na několika stránkách zároveň. Pokud bude kupříkladu zapotřebí změnit barvu písma pro několik stránek webu, a budou-li odkazovat na stejnou šablonu stylů, provede se tato změna v dané šabloně, čímž se změní barva na všech stránkách webu. Díky CSS mají vývojáři větší přehled o vzhledu webové stránky než za využití HTML. (27) (28)

Syntaxe CSS má určitá pravidla, definující určité skupiny stylů, které se zobrazí na webové stránce. Vnořené styly se definují pomocí párového elementu <style> umístěným v sekci <head>. Externí stylování se ukládá do samostatného souboru s příponou css, které se připojí nepárovým elementem <link>. Tento element má dva atributy - prvním z nich je rel, který udává typ stylu, druhým pak href, jenž odkazuje na soubor s příponou css. K jednomu souboru css je možno připojit více webových stránek, čímž se zajistí jednotné zobrazení. (26)

3.7 JavaScript

JavaScript je skriptovací programovací jazyk využívaný především při vývoji webu. Byl vyvinut společností Netscape za účelem přidávání dynamických a interaktivních prvků na webové stránky. Syntaxe JavaScriptu je podobná programovacímu jazyku C, je založena

na ECMAScript, která je vyvinuta společností Sun Microsystems. JavaScript funguje na straně klienta (návštěvníka webové stránky), ne na straně serveru, tudíž je zpracováván webovým prohlížečem. Načtením webové stránky lze funkci JavaScriptu spustit, aniž by byla zapotřebí komunikace se serverem a před odesláním může zkontrolovat, zda jsou například vyplněna všechna požadovaná pole ve webovém formuláři. Tím je schopna vytvořit chybovou hlášku dříve, než se na server přenesou jakékoliv informace. (29) (30)

U skriptovacích jazyků jako je například PHP je i u JavaScriptového kódu možnost jej vložit kamkoliv do HTML webové stránky. Na rozdíl od HTML zůstane plně viditelný ve zdroji webové stránky, HTML dokáže zobrazit pouze výstup kódu na straně serveru. Rovněž lze kód vložit do stránky načtením ze samostatného souboru s příponou js.. Psaní příkazů je omezeno určitými pravidly, například každý příkaz musí končit středníkem. (29) (26) (30)

Pomocí značek <script> lze funkce JavaScriptu zavolat při určitých událostech uživatele jako jsou například onClick, onMouseUp, onKeyUp atd. Atribut onClick se provede tehdy, klikne-li se na část stránky, která je nadefinována určitým elementem, myší. Pro pokročilejší přidávání dynamických prvků na webových stránkách se využívají knihovny JavaScriptu jako je jQuery, pro základní funkce na straně klienta se pak používá právě JavaScript. (30) (26)

4 Vlastní práce

4.1 Analýza společností nabízející únikové hry

Pro analýzu webových stránek a aplikací nabízející únikové hry, byly vybrány 3 společnosti, jež nabízejí předem připravená témata únikové hry, a dále únikové hry dle individuálního výběru.

4.1.1 Escape Zone

První společnost k analyzování je Escape Zone (dostupná z adresy https://escapezone.cz/), která byla vyhledána pod heslem „úniková hra na přání“. Tato společnost nabízí především mobilní boxy, mobilní místnosti a únikové hry na přání. U mobilních místností je možnost vybrat si z předem připravených témat hry. Jedná se o přenosné hry, které dovezou k zákazníkovi společně se všemi rekvizitami, kde bude následně hra probíhat. Nikde nebyl nalezen podrobnější popis o přesném fungování této služby. U všech těchto služeb, které tato společnost nabízí, probíhá objednávka pouze přes kontaktní formulář.

4.1.2 The Chamber

Společnost The Chamber (dostupná z adresy https://thechamber.cz/hrynamiru/) byla vyhledána pod heslem „úniková hra na míru“. Na rozdíl od první analyzované společnosti je zde možnost u předem připravených témat objednávat přes rezervační systém. Tato společnost taktéž nabízí „únikové hry na míru“ pro větší počet lidí, například pro firmy. U této služby je potřeba kontaktovat danou společnost na telefonním čísle nebo e-mailové adrese.

4.1.3 The Room

Poslední analyzovanou společností je The Room (dostupná z adresy https://www.theroom.cz/teambuilding-hry/), která byla vyhledána pod heslem „úniková hra na míru“. The Room se zaměřuje především na firemní akce a teambuldingy. V nabídce se nacházelo pouze jedno připravené téma, a to „Privátní Escape café&bar IMAGINATORIUM“, přičemž se jedná se o teambuildingový program. Všechny služby jsou dostupné prostřednictvím individuálního výběru po provedení nezávazné objednávky na dané e-mailové adrese nebo telefonním čísle.

4.1.4 **Shrnutí**

Při analýze společností, které nabízejí únikové hry, jsem neobjevila žádnou společnost, která by poskytovala přímo online konfigurátor vlastní únikové hry. Ve všech analyzovaných společnostech je pouze možnost zvolení únikové hry na přání po kontaktování dané společnosti. Všechny společnosti nabízí hry na přání pro firmy a větší počet lidí.

4.2 Byznysové požadavky

Pokud by tato aplikace byla využita společnostmi nabízející únikové hry na přání, ušetřilo by to mnohdy zdlouhavou komunikaci mezi zákazníkem a firmou přes e-mail nebo telefonní číslo. Většinu zákazníků tento způsob kontaktování může odradit, navíc ne všichni zákazníci mají představu, co vše lze v únikové hře navolit. Právě proto je tento konfigurátor vhodný především pro urychlení komunikace.

4.3 Specifikace požadavků

1. Úvodní obrazovka s informacemi jak aplikace funguje (UseCase a Scénář „Úvod“)
2. Jednoduchý a přehledný vzhled aplikace (UseCase a Scénář „Aplikace“)
3. Konfigurace základního nastavení hry (UseCase a Scénář „Základní nastavení“)
4. Konfigurace žánrů, postavy a upřesnění tématu hry (UseCase a Scénář „Žánr“, „Postava“ a „Téma“)
5. Progress bar, který umožňuje zpětně upravovat konfiguraci
6. Shrnutí a možnost objednání únikové hry (UseCase a Scénář „Košík“)
7. Ceník a FAQ
8. Rozdělení na Admin modul a Front modul (UseCase a Scénář „Admin- Přihlášení“)
9. Přidání proměnných prvků do konfigurátoru (UseCase a Scénář „Admin- Přidat město“ a „Admin- seznam měst“)
10. Zobrazení veškerých objednávek (UseCase a Scénář „ Admin – Objednávky“ a „Admin- Kompletní objednávka“)

4.4 Návrh

Při vývoji webové aplikace je důležité si nejprve řádně promyslet její výsledný vzhled, a rovněž funkčnost, již by měla plnit. V tomto případě bylo zásadní správně kroky v konfigurátoru rozvrhnout a rozlečnit tak, aby byly pro uživatele srozumitelné. Na ukázku je zde popsán první a třetí krok konfigurátoru, zbylé dva jsou pak uvedeny v příloze.

4.4.1 UseCase, Scénář a drátový model

V této kapitole je představen návrh webové aplikace, který obsahuje UseCase (případ užití), Scénář a k nim příslušný drátový model. UseCase a Scénář „Aplikace“ a „Nakonfigurovat“ se vztahuje ke všem částem ve Front modulu, proto bude popsán pouze jednou.

4.4.1.1 Úvod

Drátový model 1 - Úvod (zdroj: autorka)

UseCase „Úvod“ - Uživatel očekává zobrazení úvodní stránky s informacemi, o jakou webovou aplikaci se jedná, a co od ní může očekávat.

Scénář „Úvod“ - Systém zobrazí úvodní stránku, která obsahuje úvodní informace o webové aplikaci.

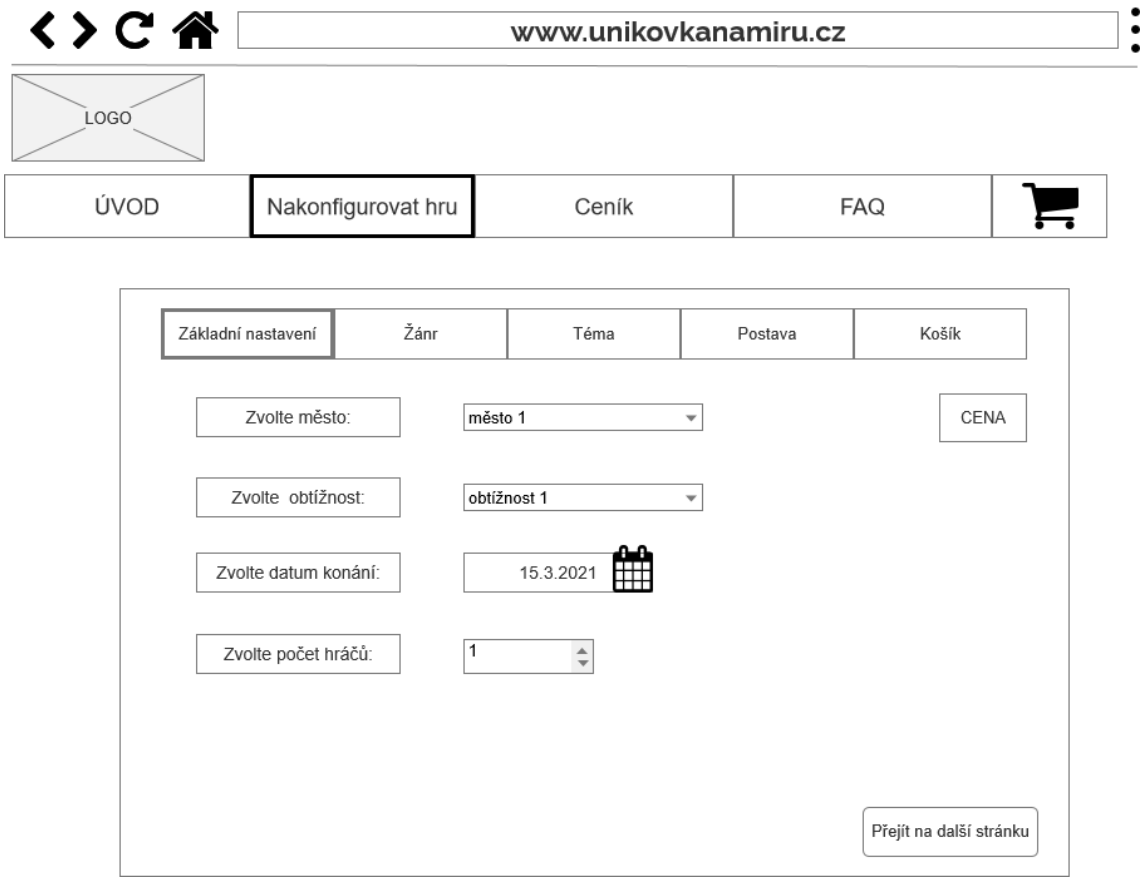
UseCase „Aplikace“ - Uživatel očekává zobrazení webové aplikace a možnost jednoduchého a intuitivního ovládání aplikace, rovněž zobrazení hlavního navigačního panelu.

Scénář „Aplikace“ - Systém zobrazí webovou aplikaci, která se skládá z navigačního panelu, jenž obsahuje:

- Úvod
- Nakonfigurovat hru
- Ceník
- FAQ
- Košík (logo)

Systém zobrazí nad navigačním panelem logo aplikace.

4.4.1.2 Nakonfigurovat hru – Základní nastavení



Drátový model 2 - Nakonfigurovat hru - Základní nastavení (zdroj: autorka)

UseCase „Nakonfigurovat hru“ - Po kliknutí na „Nakonfigurovat hru“ na navigačním panelu uživatel očekává zobrazení s možností vlastního nakonfigurování únikové hry, u prvního kroku pak tlačítko s možností přejít na další krok. Dále očekává progress bar, přes který bude mít lepší přehled o konfiguraci a informaci, v jakém kroku se nachází.

Scénář „ Nakonfigurovat hru“ – Po zvolení „Nakonfigurovat hru“ z hlavního panelu systém zobrazí první krok konfigurátoru. Vpravo dole se zobrazí tlačítko „Přejít na další stránku“, které odkazuje na další krok, v tomto případě na výběr žánru. Také systém zobrazí progress bar se všemi kroky konfiguratoru pro lepší přehlednost a informovanost.

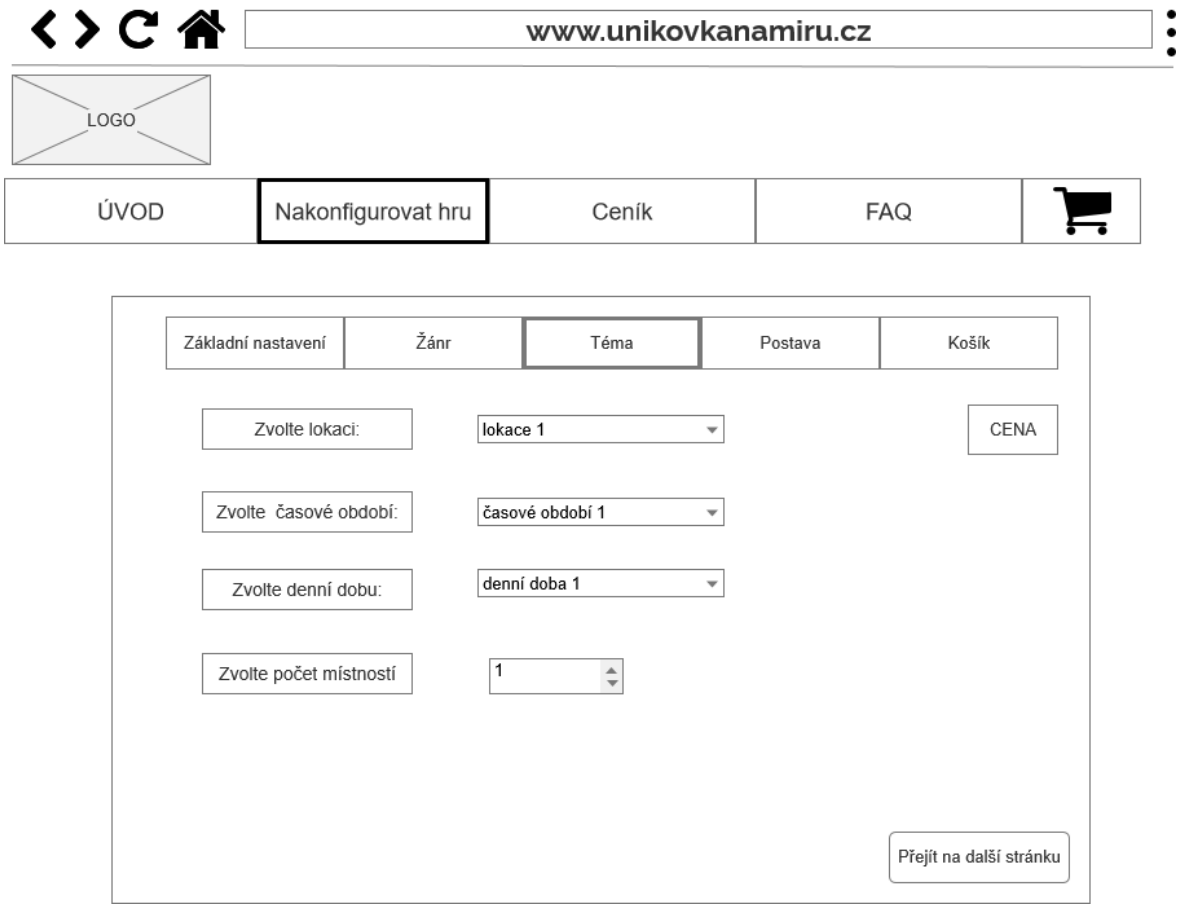
UseCase „Nakonfigurovat- Základní nastavení“- Uživatel očekává zobrazení prvního kroku konfigurátoru, kde bude na výběr z několika možností. Od základního nastavení uživatel očekává výběr města, obtížnosti, data konání a počet hráčů v jeho týmu. V neposlední řadě uživatel očekává informaci o průběžné ceně.

Scénář „Nakonfigurovat- Základní nastavení“ - Systém zobrazí první krok konfigurátoru „Základní nastavení“, kde si zákazník zvolí:

- Město
- Obtížnost
- Datum konání
- Počet hráčů

Výběr města zobrazí select box s několika možnostmi, kde bude možné únikovou hru hrát. U obtížnosti systém zobrazí select box s výběrem náročnosti hry. Po kliknutí na ikonu kalendáře systém zobrazí datepicker pro zvolení data konání. Dále systém zobrazí pole pro vyplnění počtu hráčů.

4.4.1.3 Nakonfigurovat hru - Téma



Drátový model 3 - Nakonfigurovat hru - Téma (zdroj: autorka)

UseCase „Nakonfigurovat – Téma“ - V sekci téma uživatel očekává navolení několika prvků, které budou definovat jeho vlastní téma únikové hry. Poté uživatel očekává volbu lokace, časového období, denní doby a počtu místností.

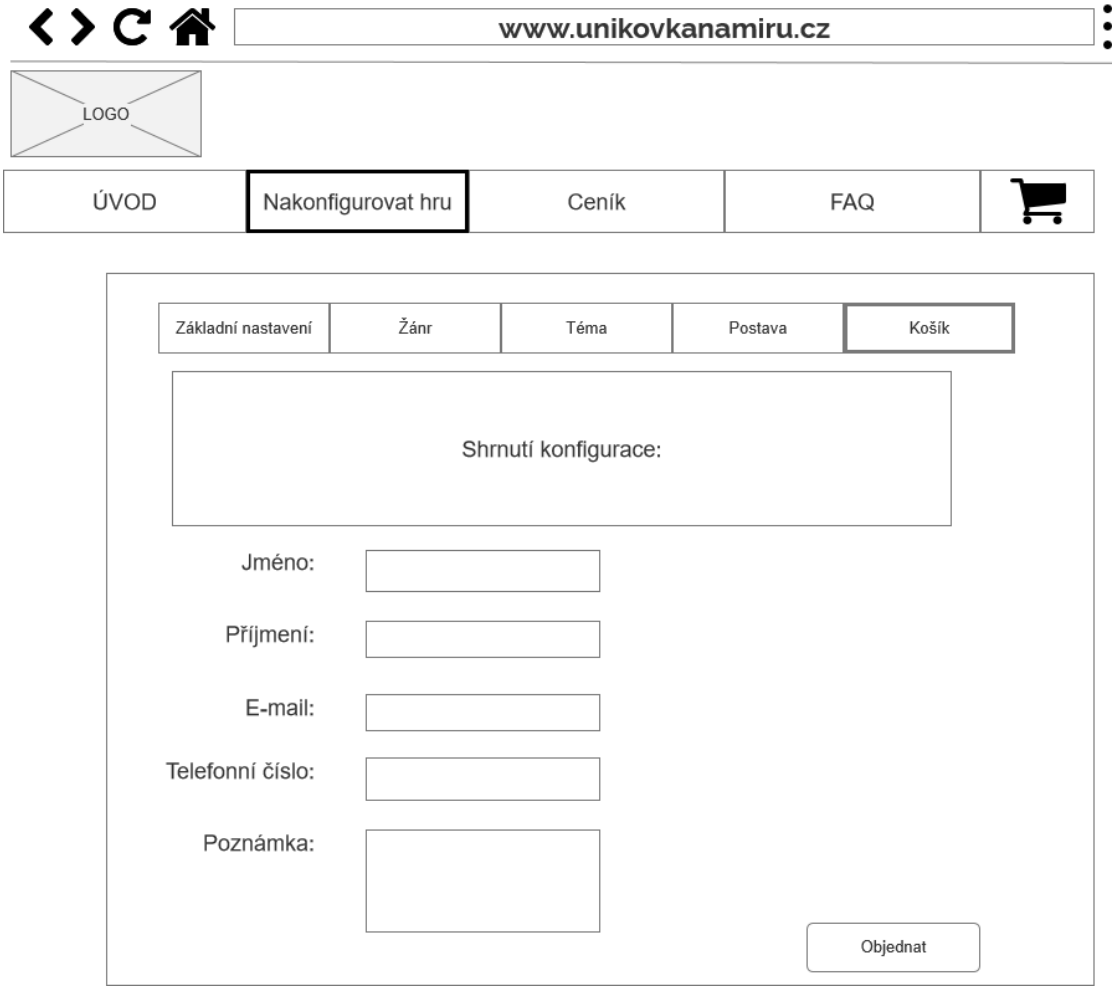
Scénář „Nakonfigurovat- Téma“ - Systém zobrazí třetí krok konfigurátoru, kde bude možnost volby čtyř prvků:

- Lokace
- Časové období
- Denní doba
- Počet místností

U výběru lokace systém zobrazí select box s lokacemi, kde se bude úniková hra odehrávat. U časového období systém zobrazí select box s výběrem časového období, z jaké doby se

bude úniková hra odehrávat. U denní doby systém zobrazí výběr denní doby se select boxem, který určí, zda se úniková hra bude odehrávat v noci nebo přes den. Poslední prvek je počet místností, u něhož systém zobrazí pole pro vyplnění, kde si uživatel zvolí, kolik místností si přeje zahrnout do své únikové hry.

4.4.1.4 Nakonfigurovat hru - Košík



Drátový model 4 - Nakonfigurovat hru - Košík (zdroj: autorka)

UseCase „ Košík“ - Při kliknutí v hlavním menu na ikonu košíku uživatel očekává prázdný košík, který odkazuje na nakonfigurovaní hry. Po vyplnění všech povinných polí v konfigurátoru uživatel očekává zobrazení celkové shrnutí s cenou v části shrnutí. Následně očekává možnost změny výběru přes progress bar.
Při nespokojenosti s výběrem uživatel očekává pole pro poznámku případné změny, která

není v nabídce. Pro odeslání objednávky uživatel očekává formulář pro vyplnění údajů a tlačítko „objednat“.

Scénář „ Košík“ - Při nekompletně nakonfigurované hře po kliknutí na košík v hlavním menu systém zobrazí informaci o nutnosti projít všemi kroky konfigurátoru. Po nakonfigurovaní všech povinných prvků systém zobrazí shrnutí celého výběru nakonfigurované hry:

- Základní nastavení
- Žánr
- Postavu
- Počet místností
- Cenu

Pro změnu výběru systém zobrazí progress bar, přes který je možnost změnit určitý výběr. Pro vyplnění údajů o zákazníkovi systém zobrazí pod shrnutím formulář s tlačítkem „Objednat“.

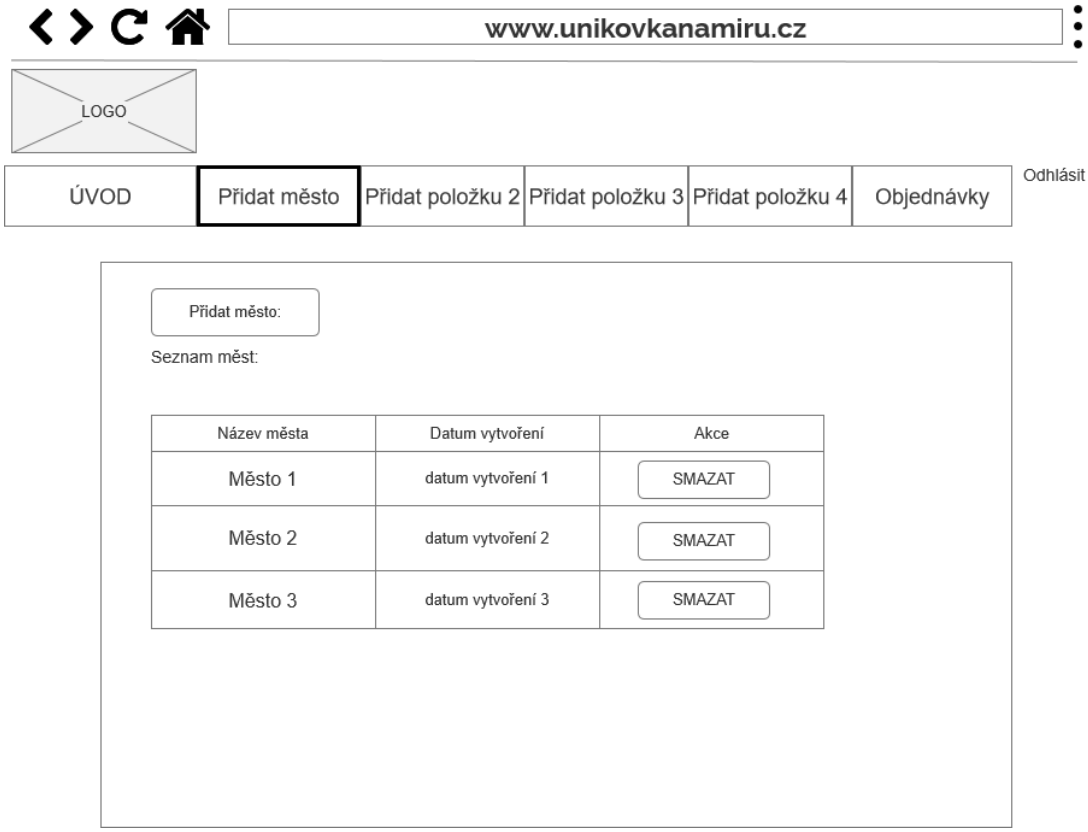
4.4.1.5 Admin - Přihlášení

Drátový model 5 - Admin - Přihlášení (zdroj: autorka)

UseCase „Admin – Přihlášení„ - Pro vstup do Admin modulu uživatel očekává formulář pro zadání přihlašovacího jména a hesla s tlačítkem „ Přihlásit“.

Scénář „Admin- Přihlášení“ - Pro zadání jména a hesla Admin uživatele systém zobrazí formulář pro přihlášení s tlačítkem „Přihlásit“.

4.4.1.6 Admin - Přidat město

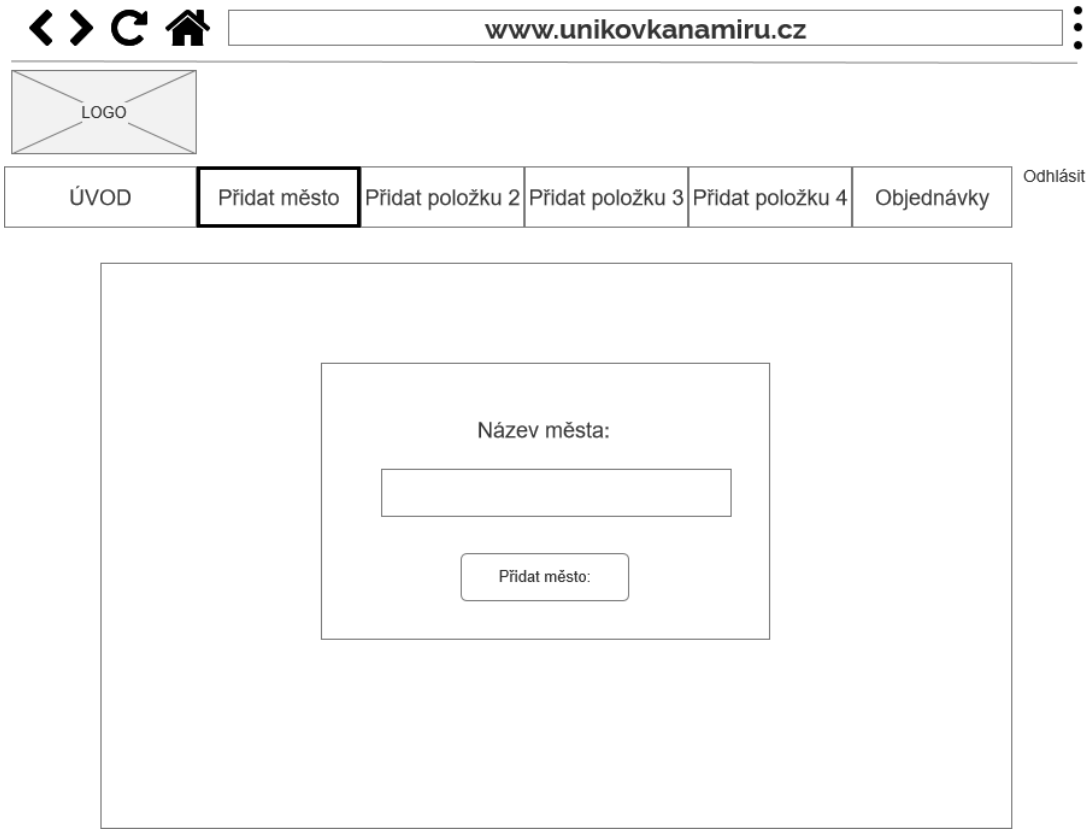


Drátový model 6 - Admin - Seznam měst (zdroj: autorka)

UseCase „ Admin – Seznam měst“ - Po přihlášení do Admin modulu uživatel očekává menu s editací konkrétních položek. Po kliknutí na „Přidat město“ v hlavním menu uživatel očekává seznam aktuálních měst, které bude možno editovat. Dále uživatel očekává možnost přidání města.

Scénář „Admin-Seznam měst“ - Po přihlášení do Admin modulu a kliknutí na „Přidat město“ v hlavním menu systém zobrazí seznam měst s datem vytvoření, které jsou již přidané

v databázi. Systém také zobrazí tlačítko „Smazat“ pro smazání daného města. Pro přidání nového města systém zobrazí tlačítko „ Přidat Město“.

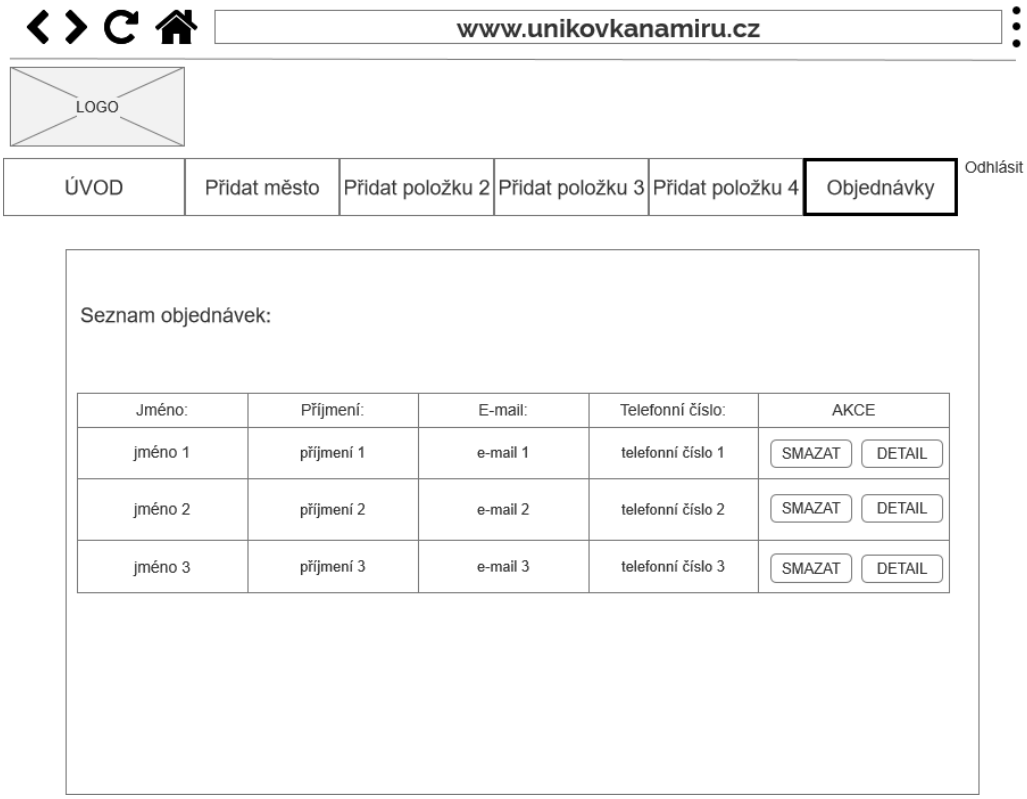


Drátový model 7 - Admin - Přidat město (zdroj: autorka)

UseCase „ Admin – Přidat město“ - Po kliknutí na tlačítko „Přidat město“ uživatel očekává pole pro přidání nového města s tlačítkem „Přidat město“.

Scénář „ Admin – Přidat město“ - Po přesměrování z předchozí stránky na přidání konkrétního města systém zobrazí formulář pro vyplnění názvu města s tlačítkem „ Přidat město“.

4.4.1.7 Admin - Objednávky



Drátový model 8 - Admin - Seznam objednávek (zdroj: autorka)

UseCase „Admin- Seznam objednávek“ - Po kliknutí na hlavním panelu na záložku „Objednávky“ uživatel očekává seznam zákazníků v systému, kteří si hru již nakonfigurovali. Pro smazání objednávky uživatel očekává tlačítko „ Smazat“. Dále uživatel očekává tlačítko „Detail“ se zobrazením všech informací o zákazníkovi a o hře, kterou si objednal.

Scénář „ Admin- Seznam objednávek“ - Po kliknutí na „Objednávky“ v hlavním panelu systém zobrazí seznam objednávek a rovněž základní údaje o zákaznících. Systém také zobrazí tlačítko „ Smazat“ pro odstranění objednávky a tlačítko „Detail“ pro informaci o kompletní objednávce zákazníka.

Drátový model 9 - Admin - Objednávka (zdroj: autorka)

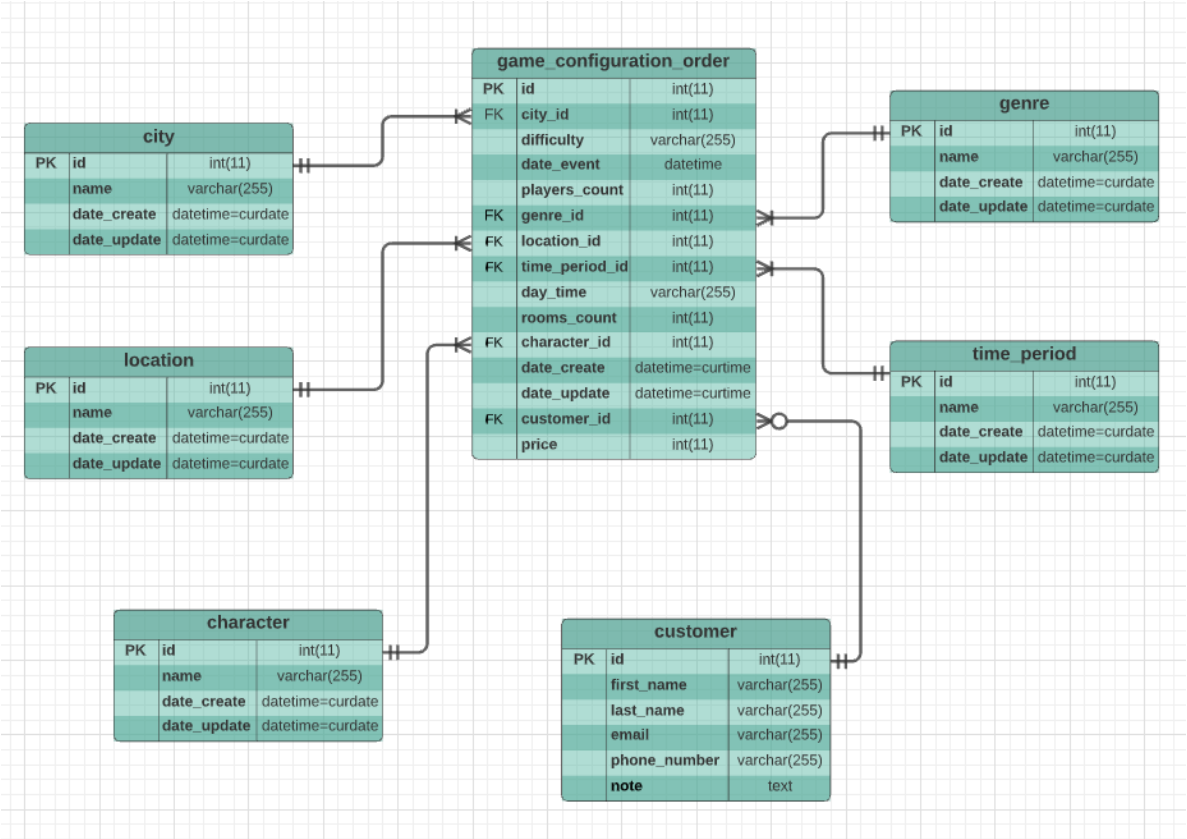
UseCase „Admin – Objednávka“ - Po kliknutí na „Detail“ u konkrétního zákazníka uživatel očekává zobrazení kompletní objednávky daného zákazníka s jeho nakonfigurovanou hrou. Pro zobrazení seznamu objednávek uživatel očekává tlačítko „Zpět na objednávky“.

Scénář „Admin- Objednávka“ - Po přesměrování při kliknutí na tlačítko „Detail“ u konkrétního zákazníka systém zobrazí všechny údaje o zákazníkovi, pod kterým bude zobrazeno jeho shrnutí nakonfigurované hry. Pro přesměrování zpět na objednávky systém zobrazí tlačítko „ Zpět na objednávky“.

4.4.2 **Databáze**

Databáze byla použita ze serveru muj.cloud.cz poskytující webhosting. Pro připojení databáze jsou zapotřebí údaje, které jsou zaznamenány ve složce config v souboru local.neon,

jedná se o databázový systém MySQL.



Obrázek 4 - Entity Relationship Diagram - Databáze (zdroj: autorka)

Databáze se skládá ze sedmi tabulek, které jsou provázány pomocí primárních a cizích klíčů. Hlavní tabulka je game_configuration_order, do které se ukládají všechny ostatní tabulky, tudíž obsahuje všechna data samotného nakonfigurování hry.

Tabulky city, genre, location, time_period a character obsahují prvky, které se přidávají do databáze pomocí Admin modulu.

Tabulka customer zahrnuje hlavní údaje o zákazníkovi pro vytvoření objednávky. Id zákazníka se následně propíše do tabulky game_configuration_order, která obsahuje všechny zvolené možnosti zákazníka v online konfigurátoru.

4.5 Implementace webové aplikace

Webová aplikace vytvořená v této bakalářské práci je online konfigurátorem únikových her. Byla vytvořena ve vývojovém prostředí PhpStormu v programovacím jazyce PHP pomocí frameworku Nette. Aplikace byla rovněž rozdělena na Admin modul a Front modul. Pro větší přehlednost je tvořena pomocí komponent, services, templates, presenterů a modelů.

Při lokálním vývoji je zapotřebí nejprve spustit systém Docker a následně Docker kontejner, a to přes terminál v PhpStormu za pomoci příkazu docker-compose up. Po zadání do prohlížeče localhost s odpovídajícím portem za dvojtečkou (v tomto případě localhost:90), se následně zobrazí výstup webové aplikace. Odpovídající port je zobrazen v souboru docker-compose.yml.

Pro ulehčení UI byla využita již vytvořená volně dostupná šablona a následně upravena z webové stránky https://www.free-css.com/free-css-templates/page258/loxury. (31)

4.5.1 Admin modul

Pro přístup do Admin modulu je potřeba přihlášení, po zadání localhost:90/admin do prohlížeče se objeví formulář, jenž ověří správnost jména a hesla. Údaje pro přihlášení jsou uvedeny ve složce config v konfiguračním souboru common.neon. Po ověření správnosti údajů se přesměruje na stránku do prostředí správce, kde je umožněno přidávat nejdůležitější proměnné prvky do konfigurátoru, které se následně propíší do samotných prvků v Front modulu přes databázi. Admin modul umožňuje přidávání a odstraňování pěti hlavních prvků, které bude možné v budoucnu upravovat - města, žánru, postavy, lokace a časového období. V Admin modulu v záložce Objednávky se zobrazí seznam zákazníků vytvořen přes datagrid. Přes tento seznam je možno zobrazit kompletní nakonfigurovanou hru zákazníka.

V následujících zdrojových kódech bude popsána komponenta přidávání města.

4.5.1.1 Přidat město

Následující zdrojové kódy popisují komponentu CityForm, pomocí které se přidávají města přes Admin modul do Front modulu do prvního kroku základního nastavení konfigurátoru.

Třída CityForm dědí Control, který je potomek třídy Nette\Application\UI\Control. Metoda construct je vždy automaticky provolána v moment inicializování třídy, v tomto případě CityForm.

```
class CityForm extends Control
{

   /**
    * @var FormFactory
    */
   private $formFactory;
   /**
    * @var City
    */
   private $cityServices;


   public function __construct(FormFactory $formFactory, City $cityServices)
   {

      $this->formFactory = $formFactory;
      $this->cityServices = $cityServices;
   }
```

Zdrojový kód 1 - CityForm - metoda construct

Pomocí metody render je formulář vygenerován do html struktury, který je přes šablonovací systém latte vykreslen na výstupu pro přidání města.

```
public function render(): void
{
   $template = $this->template;

   $template->setFile(__DIR__ .
'/../../Presenters/templates/components/cityForm.latte');
   $template->render();

}
```

Zdrojový kód 2 - CityForm - metoda render

Pomocí metody createComponentForm v CityPresenter dojde k vytvoření formuláře dle předlohy komponenty. Komponenta je samostatný balíček, při kliknutí na submit, což je tlačítko na přidání města na samotném výstupu se provolá metoda formSuccess CityForm. Zde se předává název přidaného města do cityServices, kde dojde ke zpracování, případné validaci dat.

```php
public function formSuccess(Form $form): void
{
    /**
     * @var ArrayHash $values
     */
    $values = $form->getValues();

    try {
        $this->cityServices->formSuccess($values);

        $this->presenter->flashMessage('Město bylo přidáno', 'success');

        $this->presenter->redirect('City:list');

    } catch (\RuntimeException $e) {
        $this->presenter->flashMessage('Někde se stala chyba!', 'danger');
    }
}
```

Zdrojový kód 3 - CityForm - metoda formSuccess

Pokud bylo město přidáno zobrazí se flash message s potvrzující hláškou, v opačném případě se vyhodí výjimka, že někde nastala chyba. Při úspěšném přidání se název města předá do modelu, kde se ukládá do databáze.

```php
protected $tableName = 'city';

/**
 * @param string $name
 *
 * @return bool
 */
public function create(string $name): bool
{
    $res = $this->db->table($this->tableName)->insert([
        'name' => $name,
    ]);

    return ($res) ? true : false;
}
```

Zdrojový kód 4 - City model - metoda create

Do modelu City se předá proměnná typu string, která obsahuje název města, kde následně dojde k uložení dat do databáze. Název města je povinný pro přidání do databázové tabulky city, id města je autoincrement

```php
/**
     * @param string $id
     *
     * @return bool
     */
    public function deleteById(string $id): bool
    {
        $rowDeletedCount = $this->db->table($this->tableName)
            ->where('id', $id)
            ->delete();

        return $rowDeletedCount > 0;
    }
```

Zdrojový kód 5 - City model - metoda deleteById

Pomocí metody deleteById dochází k mazání záznamu města na úrovni databáze podle předaného id. Proměnná $rowDeletedCount obsahuje počet smazaných záznamů v databázi. V případě, že počet smazaných záznamů bude větší než nula, dojde k úspěšnému vymazání města z databáze.

Metoda createComponentForm vytváří formulář pro přidání města na úrovni presenteru, kde se onen formulář provolá. Zde se přidá pole pro zadání města, které se zobrazí na výstupu.

```php
public function createComponentForm(): Form
{
    $form = $this->formFactory->create();

    $form->addText('name', 'Přidat město')
        ->setRequired('Název města je povinný údaj!');

    $form->onSuccess[] = [$this, 'formSuccess'];

    return $form;
}
```

Zdrojový kód 6 - CityForm - metoda createComponentForm

4.5.2 **Front Modul**

Front modul obsahuje výstup aplikace, který vidí běžný uživatel. Pod záložkou Nakonfigurovat se zobrazí první krok konfigurátoru, jenž bude popsán v následující kapitole.

Dalším krokem konfigurátoru je výběr žánru únikové hry, který se přidává pomocí Admin modulu.

Třetím krokem je pak zvolení tématu hry. Tato sekce obsahuje výběr lokace, kde si zákazník přeje, aby se jeho úniková hra konala. Dále se ve třetím kroku zvolí časové období např: středověk, husité, 2.světová válka. Další možností je výběr denní doby - zde si zákazník nastaví, aby se jeho úniková hra konala v noci nebo přes den. Poslední volbou ve třetím kroku konfigurátoru je počet místností, kde každá přidaná místnost navíc přičítá k celkově ceně předem nadefinovanou částku.

Náplní čtvrtého kroku je výběr postavy, přičemž se určí, za jakou postavu si přeje zákazník nebo celý tým hrát. Postava se též přidává pomocí Admin modulu.

Pro finální objednávku je nutno vyplnit v košíku údaje o zákazníkovi. Košík bude popsán ve zdrojových kódech níže v kapitole 4.5.3.2. Košík.

4.5.2.1 Základní nastavení

V následujících zdrojových kódech bude popsán první krok konfigurace „Základního nastavení“. První volba obsahuje město, které je možno přidat přes Admin modul záložku Přidat Město. V této sekci si zákazník vybere město, ve kterém si přeje, aby jeho úniková hra probíhala. Města jsou přidána přes stránku localhost:90/admin/city/create. Přidávání měst bylo popsáno v předešlé kapitole 4.5.1.1. Přidat město.

Následující zdrojový kód zobrazuje metodu getCityIdList vytvořenou v services BasicSettingsForm, přes kterou se z Admin modulu získají data do Front modulu. Proměnná datového typu pole $cityList nabývá hodnotu z databázové tabulky city, která propisuje názvy města do pole formuláře typu select box.

```php
/**
 * @return array
 */
public function getCityIdList(): array
{
   $cityList = [];

   $cities = $this->cityModel->getList();
   foreach ($cities as $city) {
      $cityList[$city->id] = $city->name;
   }

   return $cityList;
}
```

Zdrojový kód 7 - BasicSettingsForm - getCityIdList

První krok konfigurace Základního nastavení obsahuje také volbu obtížnosti. Vzhledem k tomu, že obtížnost není prvek, který obsahuje mnoho možností, je proto nadefinován pomocí konstanty a není možno jej editovat přes Admin modul. Na obtížnost je vytvořen value object, díky kterému se připisuje cena, a to dle druhu obtížnosti.

```php
class Difficulty
{
   public const EASY = 'pro začátečníky';
   public const MEDIUM = 'pro pokročilé';
   public const HARD = 'pro mistry';

   public const LEVELS = [
      self::EASY,
      self::MEDIUM,
      self::HARD,
   ];

   public const PRICES = [
      self::EASY => 1200,
      self::MEDIUM => 1500,
      self::HARD => 1800,
   ];
```

Zdrojový kód 8 - Difficulty - value object

Na následujícím zdrojovém kódu se provádí výpočet ceny, která se odvíjí od zvolení druhu obtížnosti v prvním kroku. Je vytvořena pomocí JavaScriptu a implementována do basicSettingsForm.latte. Po kliknutí do pole s výběrem obtížnosti se vyvolá akce, která vypočte cenu dle aktuálně zvolené obtížnosti.

```
<script>
    document.querySelector(".difficulty").addEventListener("click", (e)
=> {
        let difficultyId = document.querySelector(".difficulty").value;
        let priceDifficulty = {$priceDifficulty};

        document.querySelector(".price-configuration").textContent =
priceDifficulty[difficultyId];
    });
</script>
```

Zdrojový kód 9 - basicSettingsForm - výpočet ceny

V druhém kroku konfigurátoru „Žánr“ už je proměnná zanesena přímo do šablony genreForm.latte, kde se předává až do třetího kroku „Téma“, v němž se cena načítá dle zvoleného počtu místností.

```
<div class="text-right">
    <h5>Cena: <span class="price-configuration">{$priceDifficulty}</span>
Kč</h5>
</div>
```

Zdrojový kód 10 - genreForm.latte - výpočet ceny

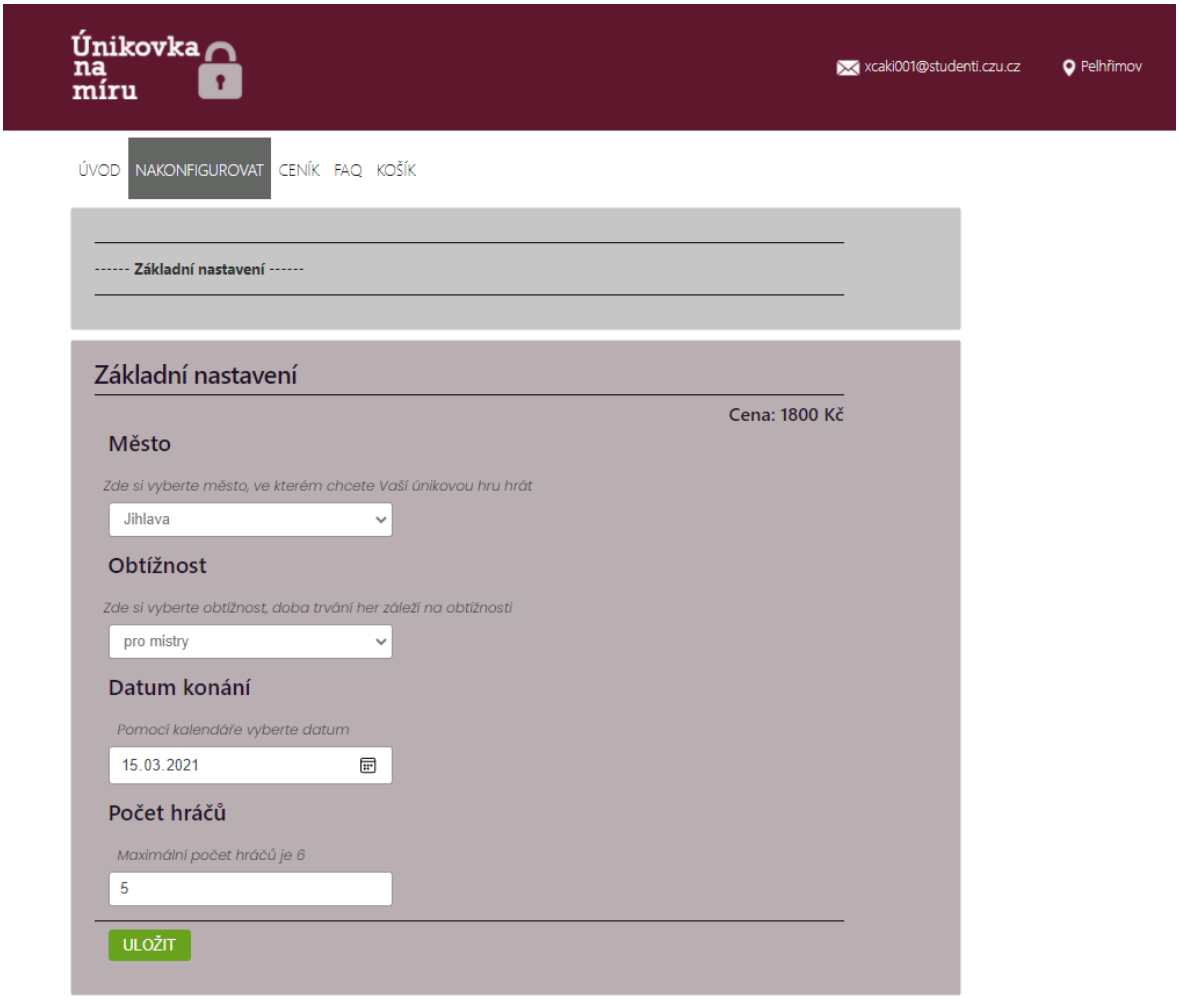
Další volbou v prvním kroku konfigurátoru „Základní nastavení“ je výběr data konání, který je nadefinován přes metodu setHtmlType, jenž je vlastností Nette. Zde se vybere datum konání únikové hry pomocí datepickeru, který je omezen pouze na výběr v budoucím čase.

Počet hráčů se též nemění jako obtížnost, tudíž není potřeba, aby byl v Admin modulu přidán k editování. Je zde pouze omezení minima a maxima povolených hodnot, které je možno do pole formuláře zadat.

```php
public function createComponentForm(): Form
{
   $form = $this->formFactory->create();

   $cityList = $this->basicSettingsServices->getCityIdList();

   $form->addSelect('cityId', 'Zvol město město', $cityList)
   ->setRequired();
   $form->addSelect('difficulty', 'Zvolte obtížnost', Difficulty::LEVELS)
      ->setRequired();
   $form->addText('date', 'Datum')
      ->setHtmlType('Date')
      ->setRequired('Vyplňte prosím datum konání.');
   $form->addInteger('playersCount', 'Počet hráčů')
      ->addRule(Form::MIN, 'Vyberte počet hráčů v rozmezí 1-6', 1)
      ->addRule(Form::MAX, 'Vyberte počet hráčů v rozmezí 1-6', 6)
      ->setRequired('Určete prosím počet hráčů.');


   if ($defaults = $this->basicSettingsServices->getDefault($this-
>gameConfiguration)) {
      $form->setDefaults($defaults);
   }


   $form->onSuccess[] = [$this, 'formSuccess'];

   return $form;
}
```

Zdrojový kód 11 - BasicSettingsForm - metoda createComponentForm

Následující obrázek ukazuje výstup prvního kroku konfigurátoru základního nastavení, který byl vykreslen přes šablonovací systém Latte. Po vyplnění všech polí a kliknutí na tlačítko uložit se konfigurátor přesměruje na další krok, jímž je výběr žánru.



Obrázek 5 - Front modul - Základní nastavení (zdroj: autorka)

4.5.3 Konfigurace

4.5.3.1 Základní nastavení

Na ukázku bude popsána konfigurace prvního kroku „Základní nastavení“ navazující na předchozí kapitolu 4.5.2.1 Základní nastavení. Následující zdrojový kód popisuje GameConfiguration.php, což je value object, který slouží k naplnění dat. Tento objekt obsahuje všechna data, jež se vyplňují v konfigurátoru. GameConfiguration.php nekomunikuje s databází, ale předává se v dalších krocích pomocí metody getSession.

```php
class GameConfiguration
{
    private $city;
    private $difficulty;
    private $dateEvent;
    private $playersCount;
    private $genre;
    private $location;
    private $timePeriod;
    private $dayTime;
    private $roomsCount;
    private $character;
    private $customer;
    private $price;
    private $priceDifficulty;
    private $priceRoomsCount;

    public const PRICE_PER_ROOM = 100;

    /**
     * @return mixed
     */
    public function getCity()
    {
        return $this->city;
    }

    /**
     * @param mixed $city
     */
    public function setCity($city): void
    {
        $this->city = $city;
    }
```

Zdrojový kód 12 - GameConfiguration - value object

Zdrojový kód uveden níže se nachází v komponentě BasicSettingsForm. Metoda formSuccess předá hodnoty navoleného města, obtížnosti, data konání a počtu hráčů z formuláře prvního kroku „Základní nastavení“ do basicSettingsServices, kde se z těchto hodnot vytvoří objekt gameconfiguration, který se vrátí a uloží se zde do session.

```
public function formSuccess(Form $form): void
{
   /**
    * @var ArrayHash $values
    */
   $values = $form->getValues();
   try {
      $gameConfiguration = $this->basicSettingsServices-
>formSuccess($values);

      $basicSettings = $this->presenter->getSession('BasicSettings');
      $basicSettings->gameConfiguration = $gameConfiguration;

      $this->presenter->redirect('Genre:create');
```

Zdrojový kód 13 - BasicSettings - metoda formSuccess

Po předání hodnot do services BasicSettings metody formSuccess se zkontrolují předané hodnoty nastavená cena a datum. V případě úspěšné validace dat dojde k inicializaci objektu gameConfiguration, do které se nastaví zvalidované hodnoty. Poté se po nastavení hodnot celý objekt gameConfiguration vrátí zpět do metody formSuccess v komponentě BasicSettingsForm, kde se uloží do session.

```
public function formSuccess(ArrayHash $values): GameConfiguration
{
   $city = $this->cityModel->getByIdAsArray($values['cityId']);
   $priceDifficulty = Difficulty::getPriceById($values['difficulty']);

   if ($priceDifficulty === null) {
      throw new \RuntimeException('Price cant be null');
   }

   try {
      $date = new \DateTime($values['date']);

      if ((new \DateTime())->getTimestamp() > $date->getTimestamp()) {
         throw new \RuntimeException('Date must be in future', 1);
      }

   } catch (\RuntimeException $e) {
      throw $e;
   }

   $gameConfiguration = new GameConfiguration();
   $gameConfiguration->setCity($city);
   $gameConfiguration-
>setDifficulty(Difficulty::getLevel($values['difficulty']));
   $gameConfiguration->setPriceDifficulty($priceDifficulty);
   $gameConfiguration->setDateEvent($values['date']);
   $gameConfiguration->setPlayersCount($values['playersCount']);

   return $gameConfiguration;
}
```

Zdrojový kód 14 - BasicSettings services - metoda formSuccess

4.5.3.2 Košík

Finálním krokem v online konfigurátoru je „Košík“, který obsahuje shrnutí navolených možností. Ve shrnutí jsou obsaženy všechny kroky ze Základního nastavení, Žánru, Téma a Postavy. Pro změnu volby určitého výběru je možné se vrátit přes progress bar, který následně uloží změnu na úrovni value objectu GameConfiguration.php a objeví se změněný ve shrnutí. Metoda getDefault ze services Cart zajistí, že při návratu na předchozí krok se nevymažou kroky následující, které už byly nakonfigurovány.

```
public function getDefault(?GameConfiguration $gameConfiguration = null): ?array
{
   if ($gameConfiguration === null) {
      return null;
   }
   if (!$gameConfiguration->getCity() ||
      !$gameConfiguration->getDifficulty() ||
      !$gameConfiguration->getDateEvent() ||
      !$gameConfiguration->getPlayersCount()) {

      return null;
   }

   return [

      'cityId' => $gameConfiguration->getCity()['id'],
      'difficulty' => Difficulty::getIdByName($gameConfiguration-
>getDifficulty()),
      'date' => $gameConfiguration->getDateEvent(),
      'playersCount' => $gameConfiguration->getPlayersCount(),
   ];
}
```

Zdrojový kód 15 - Cart services - metoda getDefault

4.5.3.3 Ukládání do databáze

Po vyplnění všech čtyř kroků konfigurátoru a následném vyplnění údajů v košíku u objednávky nakonfigurované hry se objekt gameConfiguration předá do metody create objektu GameConfigurationOrder a dojde k uložení všech nakonfigurovaných dat do databázové tabulky game_configuration_order.

```php
class GameConfigurationOrder extends Base
{
    protected $tableName = 'game_configuration_order';

    /**
     * @param GameConfiguration $gameConfiguration
     *
     * @return bool
     */
    public function create(GameConfiguration $gameConfiguration): ActiveRow
    {
        $res = $this->db->table($this->tableName)->insert([
            'city_id' => $gameConfiguration->getCity()['id'],
            'difficulty' => $gameConfiguration->getDifficulty(),
            'date_event' => $gameConfiguration->getDateEvent(),
            'players_count' => $gameConfiguration->getPlayersCount(),
            'genre_id' => $gameConfiguration->getGenre()['id'],
            'location_id' => $gameConfiguration->getLocation()['id'],
            'time_period_id' => $gameConfiguration->getTimePeriod()['id'],
            'day_time' => $gameConfiguration->getDayTime(),
            'rooms_count' => $gameConfiguration->getRoomsCount(),
            'character_id' => $gameConfiguration->getCharacter()['id'],
            'customer_id' => $gameConfiguration->getCustomer()['id'],
            'price' => $gameConfiguration->getPrice(),

        ]);

        return $res;
    }
```

Zdrojový kód 16 - GameConfigurationOrder model - metoda create

4.5.3.4 Problém při implementaci

Při implementaci se vyskytl problém s ukládáním všech navolených prvků do databáze. Metoda getSession nebyla schopna udržet vazby dependency injection, což mělo za následek, že se do databáze neuložila všechna potřebná data. Z toho důvodu byl vytvořen value object GameConfiguration.php, který tento problém vyřešil.

V Nette Frameworku je na rozdíl od čistého PHP datové úložiště session rozděleno pomocí sekcí, kde každá jednotka využívá vlastní sekci a díky tomu nemůže dojít k problému s názvy. (32)

4.5.4 **Webhosting**

Po lokálním vývoji přes localhost pomocí systému Docker byla založena doména www.unikovkanamiru.cz přes službu nabízející webhosting muj.cloud. Aplikace byla

následně nahrána přes FTP připojení na tuto doménu. Přes tento účet byla také vytvořena a připojena databáze.

4.6 Testování

Webová aplikace byla pravidelně testována v průběhu lokální vývoje, při kterém byl především využit ladící nástroj Tracy. Po nahrání webové aplikace pomocí FTP připojení na doménu www.unikovkanamiru.cz bylo potřeba otestovat funkčnost a srozumitelnost pro uživatele.

Mezi nejdůležitější části k otestování přes Admin modul bylo mazání dat z databáze a přehlednost webové aplikace. Subjektům byly zadány tyto úkoly:

1. Přihlášení do Admin modulu
2. Smazat konkrétní město
3. Najít kompletně nakonfigurovanou hru daného zákazníka
4. Smazat jeho objednávku
5. Přidat postavu
6. Odhlášení

U žádných z těchto úkolů se aplikace ani subjekty nesetkaly s žádným větším problémem. Pouze u úkolu 2 jeden subjekt chvíli váhal, kde dané město smazat. Po následné kontrole ve Front modulu test databáze proběhl také v pořádku.

Ve Front modulu bylo nejdůležitější otestovat, zda je aplikace přehledná, funkční, kroky konfigurátoru smysluplně rozdělené a zda se ukládají do databáze. Subjektům byly zadány tyto úkoly:

1. Nakonfigurovat první dva kroky konfigurátoru
2. Přejít do košíku
3. Přejít na ceník
4. Projít celý konfigurátor
5. Z košíku změnit vybraný žánr a vrátit se zpět
6. Odeslat objednávku

Výsledky testování Front modulu proběhly také bez problémů. Po splnění 1. úkolu aplikace správně v košíku odkázala na povinnost splnění všech kroků v konfigurátoru. U 5. úkolu

opět jeden subjekt chvíli váhal jak změnit vybraný žánr. Po kontrole v Admin modulu 6. úkol proběhl také úspěšně. Pro všechny byl konfigurátor srozumitelný, někteří by ocenili přidání obrázků například k výběru města.

5 Výsledky a diskuse

V této práci byla vytvořena funkční webová aplikace online konfigurátoru únikových her, která umožňuje nakonfigurovat si vlastní hru podle představ zákazníka. Tato aplikace byla také nahrána na www.unikovkanamiru.cz. Níže jsou sepsány návrhy na vylepšení od autorky práce a od testovaných subjektů.

V dalším vývoji by bylo vhodné přidat k editaci do Admin modulu všechny prvky uvedené v konfigurátoru. Tímto krokem by se do budoucna značně ulehčila práce na straně backendu například v případě změny denní doby. Přes Admin modul by se mohla přidávat časová okna s volnou kapacitou. Dalším vylepšením by byla úprava menu, které by bylo rozbalovací a členěné na jednotlivé kroky jako jsou přidat město, seznam měst atd. Pro další vývoj by u objednávek bylo potřeba zakomponovat, které objednávky již proběhly, případně rozdělení například podle měst. Admin modul by se také mohl rozšířit na přidávání uživatelských účtů pro přístup do Adminu.

Ve Front modulu při testování bylo subjekty zmíněno, že by ocenili lepší vzhled samotného konfigurátoru jako například přidání obrázků k výběru města. Tento krok by se dal vylepšit u téměř všech prvků v konfigurátoru. Následně by se mohlo přidat grafické vykreslení, které by naznačovalo, jak nakonfigurovaná úniková hra bude následně vypadat. Také další prvky konfigurátoru, například postavy, by se daly rozšířit tak, že na výběr by bylo více postav pro jeden tým. Dalším vylepšením by byla možnost registrace a přihlašování zákazníků, případně založení věrnostních účtů.

6 Závěr

Hlavním cílem této bakalářské práce byl vývoj webové aplikace online konfigurátoru únikových her, který usnadňuje komunikaci mezi zákazníkem a společností nabízející únikové hry na přání.

V teoretické části byly popsány nástroje a technologie využité pro implementaci této webové aplikace. Práce představila pojmy webová aplikace, backend a frontend, objektově orientované programování, MVC návrhový vzor. Dále se zabývala programovacím jazykem PHP a frameworkem Nette, pomocí kterého byla tato aplikace vytvořena. Také byl představen software Docker, přes který probíhal lokální vývoj této webové aplikace. Rovněž se práce zabývala databázovým systémem MySQL využitém při této práci. V neposlední řadě byla pozornost věnována HTML,CSS a JavaScriptem využívanými při tvorbě vzhledu webových stránek nebo aplikací.

V praktické části byla nejprve provedena analýza webových stránek nabízející únikové hry na přání. Dále byly vyspecifikovány byznysové požadavky a specifikace požadavků. Poté byl vytvořen návrh webové aplikace pomocí UseCase a Scénářů, drátových modelů a datové struktury. Následně byl popsán vývoj webové aplikace, který byl rozdělen na Admin modul, Front modul a konfiguraci. Na závěr byla aplikace po nahrání na www.unikovkanamiru.cz otestována. Po vyhodnocení testování byly popsány a následně představeny možnosti a návrhy pro další vylepšení této webové aplikace. Tyto návrhy byly sepsány ve výsledcích a diskuzi.

7 Seznam použitých zdrojů

1. Webová aplikace. *Management Mania.* [Online] [Citace: 1. 2. 2021.]. Dostupné z: https://managementmania.com/cs/webova-aplikace-web-application
2. Frontend vs Backend. *GeeksforGeeks.* [Online] [Citace: 1. 12. 2020.]. Dostupné z: https://www.geeksforgeeks.org/frontend-vs-backend/
3. Backend. *TechTerms.* [Online] [Citace: 1. 12. 2020.]. Dostupné z: https://techterms.com/definition/backend
4. What is Objected Oriented Progamming? OOP explained in Depth. *Educative.* [Online] [Citace: 22. 2. 2021.]. Dostupné z: https://www.educative.io/blog/object-oriented-programming
5. MVC Framework - Introduction. *Tutorialspoint.* [Online] [Citace: 1. 12. 2020.]. Dostupné z: https://www.tutorialspoint.com/mvc_framework/mvc_framework_introduction.htm
6. Architektura MVC. *Jak psát PHP?* [Online] [Citace: 1. 12. 2020.]. Dostupné z: http://jakpsatphp.cz/MVC/
7. MVC architektura. *ITnetwork.cz.* [Online] [Citace: 5. 12. 2020.]. Dostupné z: https://www.itnetwork.cz/navrh/mvc-architektura-navrhovy-vzor
8. PHP. *TechTerms.* [Online] [Citace: 1. 12. 2020.]. Dostupné z: https://techterms.com/definition/php
9. PHP History. *NuSphere.* [Online] [Citace: 1. 12. 2020.]. Dostupné z: http://www.nusphere.com/php/php_history.htm
10. History of PHP. *PHP.* [Online] [Citace: 1. 12. 2020.]. Dostupné z: https://www.php.net/manual/en/history.php.php
11. VRÁNA, Jakub. *1001 tipů a triků pro PHP.* Brno : Computer Press, 2010. ISBN 978-80-251-2940-1.
12. Lekce 1 - Úvod do Nette frameworku pro PHP. *ITnetwork.cz.* [Online] [Citace: 2. 2. 2021.]. Dostupné z: https://www.itnetwork.cz/php/nette/zaklady/uvod-do-php-frameworku-nette
13. Composer. *Nette.* [Online] [Citace: 2. 2. 2021.]. Dostupné z: https://doc.nette.org/cs/3.1/composer

14. Přehled balíčků Nette. *Nette.* [Online] [Citace: 2. 2. 2021.]. Dostupné z: https://nette.org/cs/packages
15. Proč používat nette? *Nette.* [Online] [Citace: 2. 2. 2021.]. Dostupné z: https://doc.nette.org/cs/3.1/why-use-nette
16. Jak fungují aplikace? *Nette.* [Online] [Citace: 2. 2. 2021.]. Dostupné z: https://doc.nette.org/cs/3.1/application
17. Začíname s Latte. *Latte.* [Online] [Citace: 2. 2. 2021.]. Dostupné z: https://latte.nette.org/cs/guide
18. Začínáme s Tracy. *Tracy.* [Online] [Citace: 1. 2. 2021.]. Dostupné z: https://tracy.nette.org/cs/guide
19. Docker overview. *Docker Documentation.* [Online] [Citace: 25. 2. 2021.]. Dostupné z: https://docs.docker.com/get-started/overview/
20. Docker frequently asked questions (FAQ). *Docker Documentation.* [Online] [Citace: 25. 2. 2021.]. Dostupné z: https://docs.docker.com/engine/faq/#what-does-docker-technology-add-to-just-plain-lxc
21. MySQL. *TechTerms.* [Online] [Citace: 5. 12. 2020.]. Dostupné z: https://techterms.com/definition/mysql
22. MySQL. *TechTarget.* [Online] [Citace: 7. 2. 2021.]. Dostupné z: https://searchoracle.techtarget.com/definition/MySQL
23. WELLING, Luke a Laura THOMSON. *Mistrovství PHP a MySQL.* [překl.] Ondřej BAŠE. Brno : Computer Press, 2017. ISBN 978-80-251-4892-1.
24. About. *phpMyAdmin.* [Online] [Citace: 25. 2. 2021.]. Dostupné z: https://www.phpmyadmin.net/
25. HTML. *TechTerms.* [Online] [Citace: 5. 12. 2020.]. Dostupné z: https://techterms.com/definition/html
26. LAURENČÍK, Marek. *Tvorba www stránek v HTML a CSS.* Praha : Grada Publishing, 2019. ISBN 978-80-271-2241-7.
27. What is CSS? *MDN Web Docs.* [Online] [Citace: 7. 12. 2020.]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps/What_is_CSS
28. Co je CSS? *PĚSTUJEME WEB.* [Online] [Citace: 7. 12. 2020.]. Dostupné z: http://www.pestujemeweb.cz/obsah/css/co-je-css.php
29. JavaScript. *TechTerms.* [Online] [Citace: 5. 12. 2020.]. Dostupné z: https://techterms.com/definition/javascript

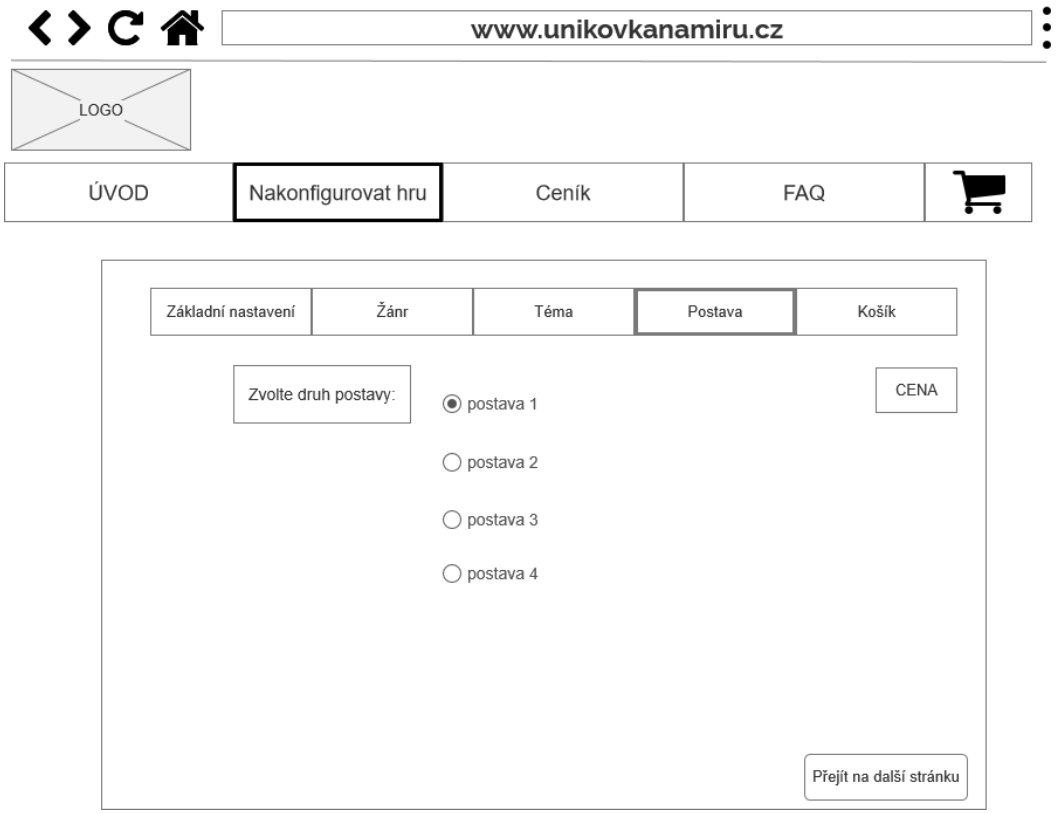
30. What is JavaScript? *MDN Web Docs.* [Online] [Citace: 5. 2. 2021.]. Dostupné z: https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript
31. Loxury Free Template. *Free CSS.* [Online] [Citace: 1. 10. 2020.]. Dostupné z: https://www.free-css.com/free-css-templates/page258/loxury
32. Sessions. *Nette.* [Online] [Citace: 2. 25. 2021.]. Dostupné z: https://doc.nette.org/cs/3.1/sessions

8 Přílohy

8.1 Drátové modely

8.1.1 Drátový model Žánru

8.1.2 **Drátový model Postavy**



8.2 **Seznam příloh na CD**

Kompletní zdrojový kód webové aplikace online konfigurátoru únikových her.