



## ZADÁNÍ BAKALÁ SKÉ PRÁCE

<b>Název:</b>	Návrh a implementace webové hry pro více hrá
<b>Student:</b>	Petr Košvanec
<b>Vedoucí:</b>	Ing. Ji í Mlejnek
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Softwarové inženýrství
<b>Katedra:</b>	Katedra softwarového inženýrství
<b>Platnost zadání:</b>	Do konce letního semestru 2017/18

### Pokyny pro vypracování

Cílem práce je navrhnout a implementovat strategickou webovou hru pro více hrá , ve které se bude vstup uživatel , a tedy hlavní sout ž, odehrávat n kolikrát denn formou p epo t . Hra nemusí v širším m ítku využívat real-time prvky .

Pokyny pro vypracování:

- 1) Porovnejte tento typ her s jinými typy z hlediska asové náro nosti na uživatele, požadavky na hardware a grafiku. Analyzujte i roli finan ních transakcí u her tohoto typu.
- 2) Prostudujte technologie používané p i tvorbu her tohoto žánru a na základ získaných informací zvolte vhodnou technologii pro implementaci. Své rozhodnutí zd vodn te.
- 3) Zvolte téma hry a vyjád ete se k základním funk nostem aplikace a způsobu synchronizace herních úkonů uživatel .
- 4) Navrhn te vhodnou architekturu aplikace, databázový model a návrh zdokumentujte.
- 5) Implementujte hlavní funk nosti hry a hru nasa te na veřejn dostupný webový server.

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.  
vedoucí katedry

prof. Ing. Pavel Tvrdík, CSc.  
d kan

V Praze dne 12. listopadu 2016



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Bakalářská práce

## Návrh a implementace webové hry pro více hráčů

*Petr Košvanec*

Vedoucí práce: Ing. Jiří Mlejnek

14. května 2017



---

## Poděkování

Chtěl bych tímto poděkovat vedoucímu mé bakalářské práce, Ing. Jiřímu Mlejnkoví, za zkušenosti, které mi předal v rámci oborových předmětů, jeho trpělivost, profesionalitu a klíčové rady během konzultací týkajících se tohoto díla. Také děkuji mé rodině a přítelkyni Daně Smejkalové za neuvěřitelnou podporu během celého tvůrčího procesu.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, avšak pouze k nevýdělečným účelům. Toto oprávnění je časově, teritoriálně i množstevně neomezené.

V Praze dne 14. května 2017

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2017 Petr Košvanec. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Košvanec, Petr. *Návrh a implementace webové hry pro více hráčů*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017.



---

# Abstrakt

Hlavní náplní této bakalářské práce je navrhnout a implementovat webovou hru pro více hráčů a nabídnout zájemcům příjemný herní zážitek. Jedná se o hru zasazenou do prohlížeče.

Tento typ her je v první části práce porovnán s jinými typy z vybraných hledisek. Nechybí vyjádření se k finančním transakcím za účelem získání herních výhod a komentář ke konkurenčním prohlížečovým hrám. Na základě těchto poznatků jsou v druhé části zavedeny požadavky na aplikaci a přiblíženo téma hry. Na to navazuje návrh včetně zvolení technologií, databázového modelu a adresářové struktury. V poslední části se čtenář dozví více o herní logice, nasazení aplikace, zabezpečení a na závěr je mu odhalen plán do budoucna.

Tématem hry je drsný sci-fi svět závodů vesmírných lodí a sázek. Fair play jsou jen slova, vyhrát může jen jeden a to za jakoukoliv cenu!

**Klíčová slova** webová hra, klikačka, sci-fi, více hráčů, přepočty, Laravel, Angular

---

# Abstract

The topic of this bachelor thesis is the design and development of an online multiplayer game, offering pleasant game experience to all players. The game is browser-based.

In the first part, certain aspects of browser-based online games are discussed and compared, such as usage of real money to gain in-game benefits and other competitive games of the same genre. In the second part, based on these observations, system requirements are analyzed and the theme of the the game is uncovered. This analysis is followed by the design, including the choice of technologies used, the database model and folder structure. In the last part, a reader learns more about the game logic, deployment of the application, handling security and in the very conclusion, project's future plan is revealed to them.

The theme of the game is inhospitable sci-fi world of starship races and bets. Fair play are just words, only one can win and they will do that at any cost!

**Keywords** web game, text-based, sci-fi, multiplayer, recalculation, Laravel, Angular

---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Rešerše stávajících řešení</b>	<b>5</b>
2.1 Definice žánru hry . . . . .	5
2.2 Porovnání s jinými žánry . . . . .	6
2.3 Vybrané tituly stejného žánru . . . . .	7
2.4 Finanční transakce . . . . .	10
2.5 Používané technologie . . . . .	12
<b>3 Analýza</b>	<b>17</b>
3.1 Téma hry . . . . .	17
3.2 Funkční požadavky . . . . .	18
3.3 Nefunkční požadavky . . . . .	20
<b>4 Návrh</b>	<b>21</b>
4.1 Výběr technologií . . . . .	21
4.2 Model balíčků . . . . .	22
4.3 Databázový model . . . . .	25
4.4 Synchronizace úkonů uživatele . . . . .	29
4.5 Uživatelské rozhraní . . . . .	30
<b>5 Realizace</b>	<b>33</b>
5.1 Systém závodů . . . . .	33
5.2 Vybrané vzorce a herní logika . . . . .	34
5.3 Nasazení . . . . .	36
5.4 Zabezpečení . . . . .	38
5.5 Další postup . . . . .	39

<b>Závěr</b>	<b>43</b>
<b>A Dokumentace databázového modelu</b>	<b>45</b>
<b>B Seznam použitých zkratk</b>	<b>51</b>
<b>C Obsah přiloženého CD</b>	<b>53</b>
<b>Literatura</b>	<b>55</b>

---

## Seznam obrázků

4.1	Diagram balíčků . . . . .	25
4.2	Tabulky spjaté se závody . . . . .	26
4.3	Tabulky spjaté s komponenty . . . . .	27
4.4	Tabulky spjaté s uživatelem i komponenty . . . . .	28
4.5	Ukázka uživatelského rozhraní - vesmírné lodě . . . . .	31
4.6	Vstup do závodu - nastavování parametrů . . . . .	32
5.1	Diagram nasazení . . . . .	38
5.2	Testování doby odezvy . . . . .	40



---

# Úvod

Herní trh je v současné době tvořen širokým spektrem her na nepřeberné množství témat. Samotný herní průmysl ale zdaleka není na ústupu, právě naopak. Aktuální průzkumy společnosti Newzoo ukazují, že jen v roce 2016 by se celosvětové tržby měly pohybovat kolem 99,6 miliard dolarů, což je o 8,5 % více než v předchozím roce. Navíc se čeká, že tento trend růstu v následujících letech bude zachován. [1] Účelem vývoje her pochopitelně bývá nabídnout produkt, na kterém autoři vydělají, ale hlavní motivací by mělo být i zprostředkování co možná nejlepšího zážitku druhým osobám. Tento herní zážitek bývá navíc ještě silnější, pokud se jedná o hry pro více hráčů.

Ke zvlášť oblíbeným typům her patří webové online multiplayer hry, které jsou tzv. zasazeny do prohlížeče. Většina úkonů hráče v nich bývá vykonávána pomocí počítačové myši. Odtud pochází označení „kličky“. V této bakalářské práci se přesouvám z pozice konzumenta do pozice návrháře a vývojáře a budu stavět na mnohaleté zkušenosti s tímto druhem her. Mou motivací bude hru vytvořit podle výše zmíněné filosofie tak, aby si hráč odnesl pokud možno samé pozitivní zážitky. Zvoleným tématem jsou sci-fi závody vesmírných lodí, které přináší spousty střetů a nekalých praktik.





---

## Cíl práce

Cílem analýzy je porovnat hry podobného typu z hlediska herních možností a použitých technologií a následně zavést požadavky na vlastní aplikaci. Na základě nich je třeba zvolit vhodný databázový model a navrhnout strukturu aplikace. Aplikace má být naprogramována čistě, srozumitelně. Musí být snadno rozšiřitelná, neboť se počítá s implementací dalších funkcí mimo rozsah práce.

Jak název práce napovídá, hlavním cílem je vytvořit základ webové online hry pro více hráčů a nasadit na veřejně přístupný webový server. Není bohužel reálné dostatečně komplexní hru v řádu měsíců kvalitně navrhnout a implementovat. Hra bude proto nasazena pouze ve stádiu prototypu, kde hlavními měřítky budou funkčnost a právě rozšiřitelnost. Cílem práce *není* vytvořit působivou grafiku, úkolem je pouze připravit uživatelské rozhraní pro její pozdější dodání.

Hra má být klikačkou. To znamená, že občas se po uživateli chce nějaký (často formulářový) vstup zadaný na klávesnici, jinak je ale hra primárně ovládána pomocí myši. Úkolem tedy není vytvořit responzivní hru ve smyslu soupeření s ostatními hráči v reálném čase. Poměření sil a zpracování herních úkonů uživatele má probíhat formou pravidelných přepočtů.



## Rešerše stávajících řešení

Ve většině bakalářských prací se dá najít kapitola „Současný stav řešení problematiky“, nebo nějaká velmi podobná. Moje bakalářská práce však není typickým představitelem řešení nějakého problému. Spíše jde o obohacení herního trhu a nabídnutí alternativy. V následujících kapitolách přesně shrnu žánr své hry a porovnáám ho s jinými. Dále představím několik zástupců podobných her a vyjádřím se k jejich slabinám a přednostem, případně i způsobu začlenění finančních transakcí.

### 2.1 Definice žánru hry

Svou hru považuji za strategickou turn-based (tahovou) klikačku s RPG prvky, zasazenou do prohlížeče. Tyto pojmy v následující kapitole vysvětlím.

#### 2.1.1 Strategická (turn-based) hra

Ve strategické hře je výsledek určen posloupností jednotlivých rozhodnutí, které hráč učiní. Cílem je vytvořit si strategii (taktiku), která je co nejbližší ideální strategii. [2] Ideální strategii je často nemožné nalézt. Někdy ani nelze říci, že nějaká ideální strategie existuje, protože hra může být značně variabilní. Neplatí zde tranzitivní vztah - pokud je strategie A v něčem lepší než strategie B a strategie B zase v něčem lepší než strategie C, strategie A nemusí být nutně lepší než strategie C.

Jedno ze základních rozdělení dělí strategické hry na hry odehrávající se tzv. v reálném čase a naopak na hry založené na kolech (turn-based). Pokud hra probíhá v reálném čase, velký důraz je kladen na rychlost, s jakou je uživatel schopen se rozhodovat. Velkou roli může mít například i postřeh, v případě počítačových her pak například i rychlé a přesné ovládání myši a klávesnice. Opakem tohoto je strategická hra založená na jednotlivých kolech. Smyslem těchto kol je poskytnout hráči dostatek času na rozmyšlení dalšího postupu. Tento přístup se často využívá u komplikovanějších her, kde je těžší učinit

správné rozhodnutí, nebo u her, které nechtějí uživatele vystavovat časové zátěži. [3]

### 2.1.2 RPG hra

Role-playing game, občas do češtiny překládáno jako „hra na hrdiny“, je žánr her, kde hráč zaujme roli fiktivní postavy a skrze jednotlivá rozhodnutí a jejich následky utváří její charakter, určuje její vývoj a často se s ní na určité úrovni ztotožňuje. [4] Počítačové hry tohoto žánru obvykle nabízí možnost rozvíjet jednotlivé herní dovednosti, zlepšovat je. Obvykle je součástí hry získání určitého druhu vybavení, které patří postavě a poskytuje jí herní bonusy. Součástí bývá souboj s ostatními postavami, ať už fyzický nebo na jiné úrovni. Oblíbeným konceptem je zasazení hry do fiktivního světa s unikátním příběhem. Cílem RPG her je vylepšení své postavy a jejího vybavení na co možná nejvyšší úroveň tak, aby hráč mohl konkurovat komukoliv.

### 2.1.3 Klikačka zasazená do prohlížeče

Hra je tzv. browser-based (zasazená do prohlížeče), pokud lze hrát jen za užití webového prohlížeče a internetového připojení. Díky této vlastnosti jsou takové hry multi-platformní, a lze je tak hrát na všech běžných operačních systémech. [5]

Označení klikačky si hra zaslouží, když se z velké části ovládá pomocí počítačové myši a klávesnici hráč využije spíše příležitostně např. při zadávání formulářových dat. Podstatná část úkonů hráče spočívá v navigaci po jednotlivých obrazovkách aplikace a vykonávání akcí, které spočívá v kliknutí na tlačítko simulující průběh akce.

Často se lze setkat se vzájemným zaměňováním pojmů klikačka, resp. hra zasazená do prohlížeče. Vyskytuje se i podobné označení „prohlížečová hra“. Pro účely této práce je lze považovat za téměř ekvivalentní.

## 2.2 Porovnání s jinými žánry

Zmíněné klikačky často využívají pouze základní animace na úrovni prvků uživatelského rozhraní. Ke spuštění a aktivnímu hraní této hry tak stačí mít webový prohlížeč. Díky tomu a faktu, že jsou zasazené do prohlížeče, se tak v podstatě požadavky na kvalitu potřebného hardware redukuje na minimální požadavky běžných webových prohlížečů. Jako příklad minimálních požadavků na svou hru uvedu minimální požadavky na užívání prohlížeče Firefox, ale ani požadavky na ostatní běžné prohlížeče se nebudou nijak výrazně lišit. [6]

- OS: Windows 7+ / Mac OS X 10.9+ / Linux (nutnost určitých balíčků)
- CPU: Pentium 4 a novější s podporou SSE2

- RAM: 512 MB
- Místo na disku: 200 MB

Hra navíc může často bez problémů běžet na ještě slabším hardwaru nebo starším operačním systémem než je uvedeno. Moderní webové prohlížeče, jak je vidět, podporují podstatnou většinu klasických operačních systémů a i díky tomu není hráčská základna nároky na hardware nijak výrazně omezena. Pro porovnání s moderními hrami, které využívají 3D grafiku v reálném čase jsem vybral titul Overwatch, který je futuristickou akční střílečkou z pohledu první osoby. Systémové požadavky pro optimální výkon u podobných her pak mohou vypadat třeba takto: [7]

- OS: Windows Vista/7/8/10 64-bit
- CPU: Intel Core i5 nebo AMD Phenom II X3, 2.8 GHz a novější
- RAM: 6 GB
- Grafická karta: Nvidia GeForce GTX 660 nebo ATI Radeon HD 7950 a novější
- Místo na disku: 30 GB

Je zjevné, že na provoz webového prohlížeče jsou mnohem nižší nároky než na provoz 3D moderní hry. Toto porovnání je však bráno jako extrémní případ. Porovnávat webovou kličku s komplexní desktopovou hrou je jako porovnávat jablka s hruškami. Hráčská základna těchto her je velmi různá a dá se říci, že zmíněné žánry si nijak výrazně nekonkurují. Webové hry jsou více „oddechové“ a nabízejí zcela jiný zážitek než propracované desktopové hry.

## 2.3 Vybrané tituly stejného žánru

V této části tedy vyberu vyberu tři hry, které se dají označit za webové multiplayer kličky. Identifikuji jejich přednosti, které by se mohly stát základem inspirace, ale také poukážu na slabiny, kterým je třeba se vyhnout. V tomto porovnání budu čerpat z poskytnuté dokumentace a manuálů, ale hlavně i z vlastní zkušenosti s uvedenými tituly.

### 2.3.1 Travian

Velmi populárním zástupcem online kliček je Travian. Jen v České republice je v době tvorby této bakalářské práce registrováno téměř 10000 hráčů. Hra je nasazena na 46 doménách prvního řádu a celosvětově je registrováno přes 324000 účtů. [8] Hráč si na začátku vybere jeden ze tří národů - Římané,

Germáni, Galové - a jeho úkolem je vybudovat z malé vesničky obávané impérium. Vesnice produkují 4 základní suroviny, které jsou potřeba pro stavbu budov, trénování vojska a obecně zvyšování úrovní všech herních artefaktů. Mocným nástrojem hráče je jeho armáda, díky které se může pokusit dobýt nebo oloupit cizí vesnice. Klíčovým faktorem je spolupráce s ostatními hráči, neboť lze útoky a obranu společně koordinovat a několikanásobně tím zvýšit své šance na úspěch. Toto přiblížení realitě však hráče i poměrně svazuje. Je takřka nemožné být úspěšný, pokud hráč nehraje týmově, a to nemusí všem lidem vyhovovat.

Jak můžeme čekat u klikačky, herní úkony jsou řešeny naplánováním jednotlivých akcí a zadáním do systému. Většina akcí v Travianu však typicky trvá reálný časový úsek do jejich uskutečnění. To znamená, armáda do svého cíle dorazí v závislosti na vzdálenosti z vesnice až po nějaké době a i díky tomu se proti útoku lze bránit. Stejně tak budova se nepostaví ihned po zadání příkazu ke stavbě, ale v závislosti na komplexnosti třeba i za několik hodin. Tento přístup je v klikačkách vidět poměrně často a jedná se o velmi silnou techniku. Nutí hráče přihlašovat se pravidelně a poměrně často, ale zároveň neklade vysoké nároky na samotný strávený čas hraním. Nelze se tedy divit, že hra byla už vydána i jako mobilní aplikace, což výborně podporuje přístupnost.

Po grafické stránce je Travian skvělým příkladem toho, jak lze uspět i s poměrně jednoduchou grafikou. V této hře ji tvoří statické obrázky, které nemají ambice působit realisticky. Naopak působí spíše komiksovým až pohádkovým dojmem, ani dospělého hráče ale nemohou urazit. Například zobrazení rozvoje vesnice je docíleno vrstvením obrázků, kdy do základní mapy vesnice přibývají jednotlivé obrázky staveb. Opět velmi jednoduchý, ale účinný koncept. Hra díky němu klade pouze minimální nároky na hardware a ve své počítačové verzi zůstává věrná označení *prohlížečová hra*.

### 2.3.2 Gladius

O oblíbenosti Gladiatu svědčí fakt, že jen na sociální síti Facebook má stránka hry přes 80000 fanoušků. Hru vyvíjí společnost GameForge, která má na svědomí mnoho her jak prohlížečových, tak i tradičních desktopových. Gladius je zástupce žánru RPG. Hráč začíná jako neznámý gladiátor. Sváděním soubojů s ostatními hráči a různými druhy výprav pak získává vybavení pro svou postavu a zdokonaluje své schopnosti. Typickým cílem v RPG hrách pro více hráčů je vyvinout si nějakou herní strategii a díky ní se stát jedním z nejsilnějších nebo nejvlivnějších postav na herním serveru. Právě touto herní filosofií se řídí i Gladius. Hráči se mohou sdružovat na klanů, ale jejich role není tak výrazná. Ačkoliv velká většina nejúspěšnějších hráčů se socializuje, síla jednotlivých klanů je různá a je tak možné být úspěšný i jako relativní samotář.

Přístup k časové náročnosti jednotlivých herních úkonů je mírně odlišný od předchozího titulu. Spíše než s trváním jednotlivých úkonů se tu hráč se-

tká se systémem pauzy. Prakticky to například znamená, že hráč po napadení jiného hráče v aréně nesmí tuto činnost dalších patnáct minut zopakovat. V závislosti na nastavených časových limitech může ale hra klást nepřiměřené nároky na čas uživatele a přesně to je bohužel případ Gladiatu. Zejména s bonusovým účtem, kdy lze jednu z hlavních herních činností - napadání - vykonávat každých 5 minut, vznikají velké rozdíly mezi hráči, kteří hru hrají velmi často a mezi těmi, kteří nechtějí do hry tolik času investovat.

První housle v tomto titulu hraje samotný gameplay, tedy kvalita zážitku, který si hráč odnese. [9] To znamená, že i v tomto titulu je grafika zjednodušená. Obrázky jsou sice reálnější než u Travianu, ale i tady jsou jen statické s využitím techniky vrstvení. Gladius je dalším důkazem, že i tak si lze vytvořit výbornou hráčskou základnu. Díky nenáročným grafice se mohou vývojáři soustředit na to podstatné - opravování chyb a vyvíjení nových funkcí.

### 2.3.3 Football arena

Na závěr jsem vybral hru z trochu jiného prostředí. Jedná se o hru Football Arena. Hra je českého původu a jejím autorem je jediný člověk - Tomáš Tkáč. Přes den jsou připojeni obvykle stovky hráčů, ale v době průběhu fotbalového zápasu se toto číslo lehce vyšplhá na desetinásobek. Football Arena stále expanduje do nových zemí. Hráč po registraci obdrží svůj vlastní fotbalový klub a je postaven do role manažera, trenéra, ředitele klubu. Jeho úkolem je vybrat pro klub co nejlepší strategii po všech stránkách a pokud možno tým zavést do nejvyšší ligy v zemi, případně i uspět na mezinárodní scéně. Hra je vydařenou simulací reálného světa. Hráči jsou povinni registrovat se v zemi pobytu nebo původu. Díky tomu se vyvinula i soutěž na reprezentační úrovni. Kooperace hráčů musí být minimální a samotnou podstatou je úspěch jedince.

Oproti výše uvedeným titulům využívá tato hra velmi odlišný způsob zacházení s časem. Během týdne jsou zavedeny aktualizace herního stavu vždy v pevně danou dobu. To například znamená, že vždy jednou týdně ve středu proběhne pravidelná aktualizace tréninku a schopnosti hráčů v týmu se změní. Výsledek a průběh samotného fotbalového zápasu je sice vypočítán jednorázově, ale uživateli je odtajňován minutu po minutě a trvá tak jako stejnou dobu jako zápas v reálném fotbalu. Aby klub správně fungoval, měl by se hráč během týdne několikrát přihlásit a vykonat důležité herní úkony. Je například žádoucí pravidelně nastavovat pokyny hráčům k zápasu, rozhodovat, co se v týdnu bude trénovat, obměňovat kádr atd. V porovnání s Travianem a Gladiatem má tak tato hra nejnižší časovou náročnost. Na druhou stranu nemůže uživatele zabavit na delší časový úsek. V konečném důsledku je to o záměrech vývojářů a cílové skupině hry.

Ze všech uvedených titulů má Football Arena jasně nejmenší výskyt grafiky. Obrázky jsou opět statické, ale jejich podstatnou část tvoří loga týmů, ikonky menu a jiná drobná grafika. Ani zde se nevyskytují žádné animace.

Většina úkonů je založena na prezentaci textu a rostoucí hráčská základna dokazuje, že tento koncept funguje.

### 2.4 Finanční transakce

V následující kapitole ukážu, jakými způsoby hry využívají finančních transakcí. Rozeberu klady a zápory jednotlivých přístupů. Vyjádřím se taky k roli transakcí ve hře a pokusím se najít bod zlomu, kdy ještě jsou finanční transakce zdravou formou podpory vývoje hry a kdy jde už jen o chamtivost tvůrců, která hře neprospívá.

#### 2.4.1 Dle četnosti plateb

Pokud je hra kompletně zdarma, znamená to, že musí mít neplatící uživatel přístup ke všem herním funkcím. Takové hry vydělávají většinou na reklamách nebo jsou vyvíjeny pro radost tvůrců i hráčů, často jako tzv. „passion project“. Jedním z nejúspěšnějších zástupců je například hra DOTA 2. Přestože tato hra je zdarma, každoročně je pořádán světový turnaj, který se těší tak obrovské popularitě, že si mohou pořadatelé díky sponzorům dovolit rozdat vítězům přes 20 milionů dolarů. [10]

Některé hry stačí si jednorázově koupit a máte garanci, že je budete moci hrát po celou dobu podpory. Běžné ceny nejúspěšnějších titulů se pohybují mezi 500 a 1500 Kč. Uživatelé předem vědí, kolik zaplatí a proto byl tento způsob hlavně v minulém desetiletí nejvíce rozšířen. Výhodou pro tvůrce hry je rychlá návratnost investice. Nevýhodou je neudržitelnost podpory v případě, že si vývojáři skutečně nenajdou další způsob, jakým by do hry mohly dostat další investice. Některé tituly to řeší prodejem tzv. kosmetických věcí. Takové věci nezvýhodňují kupující hráče. Jedná se často například o vzhled postavy a podobně. Příkladem takové hry je například oceňovaný titul Overwatch. [11]

Pro lepší udržitelnost vývoje a podpory proto některá studia volí variantu plateb za rozšíření. Za rozumně dlouhou dobu (často každý rok nebo každých 6 měsíců) vydají do hry velké rozšíření, které si hráči musí koupit. Některé hry jsou nadále vedeny ve více verzích a umožní tak dále neplatícím hráčům hrát, jiná studia cílí vždy na nejnovější verzi. Ve výsledku je tento způsob celkem podobný hrám s určenou pravidelnou platbou, která je často nazývána předplatné. S tím rozdílem, že u těchto her máte pevně dáno, kdy zaplatíte. Tento způsob je nejrozšířenější u her pro masivní počty hráčů, které mají obrovské nároky na robustnost serverů a jejichž obrat se pohybuje mnohem výše než je tomu u ostatních typů her. Příkladem je například populární World of Warcraft, který se dočkal v srpnu 2016 už šestého rozšíření a zároveň platí jeho hráči za možnost hrát na oficiálním serveru každý měsíc. [12]

Poslední obvyklou formou vybírání peněz je způsob označovaný jako mikrotransakce. Server superrewards.com [13] definuje mikrotransakci následujícím



způsobem: „Termín mikrotransakce označuje finanční transakce nízké hodnoty (obvykle do 10 dolarů) v rámci digitálních her. Předmětem prodeje je obvykle herní položka nebo jiný digitální obsah. V mnoha případech se za reálné peníze pořizuje virtuální herní měna, která slouží k nákupům prémiového zboží.“ Přestože jednorázově jde o mnohem nižší položky než u ostatních způsobů vybírání peněz od hráčů, ve výsledku se může vývojovému studiu nejvíce vyplatit, protože nákup touto formou nebývá omezen. Tyto hry jsou často zdarma ke stažení a může je tak hrát každý, ovšem s určitými omezeními. Často se tedy pak stane, že hráči s vyššími ambicemi si připlatí pro získání bonusů nebo otevření nového obsahu. Výborným příkladem je například karetní hra Hearthstone. Hra je zdarma ke stažení, ale aby hráč dostatečně rozšířil svou kolekci karet a mohl být kompetitivní na vysoké úrovni, musí být buď hodně trpělivý nebo si postup urychlit mikrotransakcemi. Není divu, že pak hra vydělá měsíčně přes 20 milionů dolarů. [14]

### 2.4.2 Role ve hrách

Na finanční transakce lze nahlížet i z jiného pohledu. Velmi důležitý je rozsah výhod, které hráč získá investicemi do hry.

Nejmírnější formou jsou již zmiňované **kosmetické** bonusy. Jejich koupí nezíská hráč žádnou faktickou výhodu nad ostatními hráči. Nejčastěji se jedná o grafické prvky nebo jiná herní zpestření. Jsou nejméně invazivní. Velké množství hráčů oceňuje rovnocenné podmínky a často se tak stane, že hráč se nezdráhá hře přispět dobrovolně.

Druhým zástupcem v této kategorii jsou bonusy, které **přímo neovlivňují** konkurenceschopnost hráče, protože je lze teoreticky nahradit aktivním přístupem, ale reálně poté mohou v ruku schopného hráče přinést drobnou herní výhodu. Některé hry například nabízí za příplatek rozsáhlé statistiky, které by hráč sice mohl zjistit rozsáhlou analýzou sám, ale protože by to zabralo příliš mnoho času, ve výsledku k nim mají přístup skutečně jen platící. Tyto statistiky jim pak pomohou k optimálnímu naplánování dalšího postupu. Na nejvyšších příčkách se i tak ale mohou pohybovat uživatelé, kteří do hry peníze neinvestovali.

Za peníze lze v jiných hrách získat herní obsah, který sice přímo zvýhodňuje majitele, ale obecně **lze získat i jinými způsoby**. Často se jedná o různou formu urychlení herních konstrukcí nebo zjednodušení, která vedou k rychlejšímu postupu ve hře. V tomto případě už vznikají velké rozdíly mezi platícími a neplatícími hráči, což vede často k frustraci neplatících hráčů. V důsledku se pak může hráč rozhodnout přestat hru hrát, nebo naopak si bonusové funkce také zaplatit. Obecně ale platí, že mít frustrované hráče, ať už platící nebo neplatící, není vůbec žádoucí. V případě, že tyto hry mají nastavený určitý strop, kterého lze hraním dosáhnout, je však stále šance být jako neplatící hráč plně konkurenceschopný, jen trvá delší dobu se do tohoto stavu dostat.

Nežřídká se však hráč setká i s tituly, které nabízejí výměnou za reálné peníze **přímé a jinak nedosažitelné výhody**. Nejčastějším zástupcem jsou různé formy *prémiových účtů*, které za pravidelný poplatek mohou udělovat povolení vybavit se nadměrně dobrými herními předměty, častěji vykonávat kritické herní funkčnosti a jinak obcházet pravidla nastavená pro neplatící hráče. Neplatící hráč nemá žádnou šanci měřit se s platícími a to nezávisle na jeho schopnostech. Hraní takové hry pak bez investic nemá smysl a vývojáři tím dávají jasný vzkaz, že hlavním hnacím motorem jsou peníze a poskytování unikátního a pozitivního zážitku je na vedlejší (jestli vůbec nějaké) koleji. Hra bývá herní komunitou označena jako „pay to win“, neboli „plať, chceš li vyhrát“.

### 2.4.3 Transakce v mé hře

Před zavedením finančních transakcí je nutné, aby byla hra téměř bezchybná a ve své finální podobě, připravena pro ostrý provoz. Zavádět finanční transakce tedy není předmětem této práce. Nastíním tedy pouze plán, jakou roli, formu a účel by transakce v mé hře měly mít, až bude hra plně připravena.

Účelem zavedení finančních transakcí bude především pokrytí výloh spojených s provozem hry a jejím kontinuálním vývojem. Kladu si za cíl využívat poskytnuté finanční prostředky na zaplacení pracovních sil a dalšího zdokonalování produktu. V případě, že by příjmy rostly, musí růst i investice.

Hra bude volně dostupná a bude zahrnovat obsah, který bude možno zakoupit za reálné peníze. Využívat budu formu mikrotransakcí. Chci však omezit množství peněz, které půjde do dané herní funkčnosti investovat. Zároveň je absolutní prioritou, aby bylo reálné jako neplatící hráč konkurovat platícím. Pokud tedy neplatící hráč investuje dostatek času a prokáže skvělé schopnosti a strategické myšlení, musí být schopný bojovat o nejvyšší příčky. Hra nesmí obsahovat obsah, který nelze získat zdarma. Výjimkou může být dočasná blokáce. Například ihned po vydání nové funkčnosti můžu zavést několikátýdenní časovou prodlevu, kdy budou moci funkčnost využít pouze platící hráči. Později k ní ale musí získat přístup i ostatní.

## 2.5 Používané technologie

Pro svou aplikaci budu potřebovat webový server, programovací jazyky pro stranu serveru a klienta a databázový systém. V následující kapitole tedy vyberu nejrozšířenější zástupce v dané kategorii technologií a shrnu jejich přednosti a slabiny. Také vysvětlím, proč budu pro vývoj potřebovat framework a co to vlastně je. V této kapitole technologie pouze porovnáám, jejich výběr odhalím až v samostatné kapitole. 4.1

### 2.5.1 Webový server

„Webový server je program, který používá HTTP (Hypertext Transfer Protocol) k vystavení souborů, které formují webové stránky uživatelům. Děje se tak v odezvě na jejich požadavky, které jsou směrovány HTTP klienty jejich počítačů. Tento proces je příkladem klient-server architektury. Všechny počítače, které vystavují (hostují) webové stránky musí mít webový server.“ [15]

Nejpoužívanějšími webovými servery v současnosti jsou Apache HTTP server se zastoupením téměř poloviny trhu, Nginx s 33 % a Internet Information Services(IIS) od společnosti Microsoft. [16] Rozebírat budu pouze první dva, neboť IIS je orientovaný na operační systém Windows a tato práce bude nasazena na distribuci Linuxu.

#### 2.5.1.1 Apache

Apache je dlouhodobě nejpoužívanějším webovým serverem. Komunita je obrovská, množství návodů a tutoriálů více než dostačující. „Má vestavěnou schopnost parsovat a spouštět velké množství forem dynamického obsahu, včetně PHP, Python a Perl skriptů, a to přímo uvnitř webového serveru.“ [17] Hlavní nevýhodu vidím ve způsobu, jak Apache zachází s příchozími požadavky ze strany klienta. Program spouští nový proces pro každý požadavek a s narůstajícím počtem požadavků jeho výkon klesá.

#### 2.5.1.2 Nginx

„NGINX byl navržen, aby už od začátku bylo jednoduché mu porozumět a ovládat. Dle názoru mnoha lidí je jeho užití jednodušší, než u konkurenčního Apache.“ Hlavní síla Nginx spočívá ve škálovatelnosti vůči příchozím požadavkům. Nevytváří nové procesy pro každý požadavek. Až tisíce požadavků jsou zpracovávány na vlastním vláknu nebo procesu a v porovnání s Apache si tak vede v tomto okruhu lépe. „Webový server Nginx nemá nativně schopnost zpracovávat dynamický obsah. Ten předá ke zpracování externímu procesoru a pak čeká až bude vyrenderovaný obsah vrácen zpět.“ [17] Kvůli tomuto přístupu je zpracování dynamického obsahu jednou z oblastí, kde si Apache vede lépe.

### 2.5.2 Programování strany klienta (front-end)

Když přijde na volbu vhodného programovacího jazyka na straně klienta, není se příliš nad čím zamýšlet. Základem naprosté většiny webových aplikací je značkový jazyk Hypertext Markup Language (HTML) doplněný o Cascading Style Sheets (CSS), který se stará o prezentaci dat. K tomu, aby byla webová stránka dynamická se pak v drtivé většině případů použije skriptovací jazyk JavaScript. Je sice několik alternativ, zdaleka žádná ale nedosahuje takové

popularity jako JavaScript. Když se tedy mluví o výběru technologií pro práci na straně klienta, spíše než o výběr jazyku jde o výběr frameworku.

„**Framework** je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji.“ [18] Jinými slovy, framework je v podstatě nutnost při programování jakýchkoliv jiných než nejmenších projektů, neboť poskytuje vývojáři užitečné funkce nebo postupy, jakými řešit známé problémy a konstrukce. Ve výsledku pak organizace celého projektu je mnohem konzistentnější. Dle webu hackernoon.com patří mezi nejpoužívanější JS frameworky AngularJS, React a jQuery, přestože React a jQuery jsou spíše rozsáhlými knihovnami než kompletními frameworky. [19]

### 2.5.2.1 AngularJS

Hlavním mottem Angularu je hříčka, že Angular je vlastně „HTML, kdyby bylo napsáno pro dynamický web“. Angular dovoluje programátorovi rozšířit HTML v rámci aplikace tak, že výsledné prostředí je čitelné a vývoj netrvá dlouho. [20]

Angular je vyvíjen společností Google a je tedy zaručena široká podpora a výborná dokumentace. Framework hojně využívá techniky two-way binding, což není nic jiného než automatická změna modelu v závislosti na změnách na prezenční vrstvě a naopak. Výborně je připraven pro užití vlastních direktiv a komponent, což jsou uživatelem definované kusy JavaScript kódu s vlastní HTML šablonou a tagem (značkou). To značně posiluje znovupoužitelnost.

Angularu je vyčítáno, že řada přidružených knihoven musí být pro něj specifická. Pokud počet dynamických prvků na jedné stránce začne převyšovat 2000, je to velmi citelně znát i na výkonu. Tento framework je také poměrně komplikovaný na naučení. [21]

### 2.5.2.2 ReactJS

Jak bylo řečeno, React není plnohodnotným frameworkem, ale jako technologie je populárnější než Angular. ReactJS je knihovnou, která dává k dispozici jazyk pro renderování HTML šablon. Nelze tedy založit plně funkční dynamickou aplikaci pouze s použitím React.

Hlavní výhodami jsou jednoduchost a volnost. V každé chvíli lze téměř pouhým pohledem do kódu říct, jak se daná komponenta vykreslí. Tato knihovna však nedává k dispozici žádnou správu událostí, podporu asynchronních http požadavků a dalších záležitostí, které zmíněný Angular řeší. [22]

### 2.5.2.3 jQuery

Podobně jako React je i jQuery jen knihovnou, nicméně na poli podpůrných JS nástrojů není nic, co by bylo více populární. JQuery je hojně využíváno

jak zkušenými vývojáři, tak úplnými nováčky. Může za to intuitivní syntaxe, která i zdánlivě komplexní úlohu promění v jednoduchou sérii příkazů.

Největší předností je velmi rozsáhlé pokrytí všech důležitých a poptávaných funkcí. Oproti Reactu zde nechybí ani podpora AJAX volání. Pokud uživatel potřebuje specializovanou vlastní funkčnost, může však narazit a dojde k nutnosti psát čistý JavaScript. [23]

### 2.5.3 Programovací jazyk - server (back-end)

JavaScript sice lze použít pro programování i serverové části (například s využitím Node.js), ale nebyl pro to primárně navržen a má jistá omezení. Je třeba sáhnout po vhodnějším nástroji. V současnosti patří mezi nejpoužívanější nástroje pro vývoj webových aplikací například jazyky PHP a Java. [24] V následujících podkapitolách první jazyky nebudu příliš představovat a spíše se zaměřím na jejich porovnání. Mezi další široce využívané jazyky patří například Ruby, Python, Perl, a také C#, ale tyto nebudou předmětem analýzy.

#### 2.5.3.1 PHP

PHP je skriptovacím jazykem a je primárně určeno právě pro webový vývoj. Jako důsledek například nebyl vytvořen s podporou objektového přístupu, což bylo napraveno až v pozdějších verzích. Na druhou stranu díky původnímu účelu prošlo již rozsáhlým vývojem a nabízí opravdu širokou škálu možností. PHP nedeklaruje proměnným jejich typ, což nabízí určitou svobodu, ale zároveň tím nechává kontrolu na programátorovi. Podobně potenciálně nebezpečných konstrukcí je v jazyku více a vývojář pak častěji dělá hůře dohledatelné chyby. [25] Toto omezení řeší každý lepší PHP framework, mezi nejznámější patří Laravel, Zend, Symfony, v Česku pak třeba Nette.

#### 2.5.3.2 Java

Java je oproti PHP mnohem odolnější vůči chybám programátora. Je silně typovaná a kompilátor i běhové prostředí umí odhalit bezpečnostní rizika. Tento jazyk, který je objektově orientovaný, je obecně použitelný na programování široké škály aplikací a mimo jiné i aplikací webových. Nativně také podporuje databázové transakce. Jazyk je platformně nezávislý díky vlastnímu běhovému prostředí. [26]

### 2.5.4 Databázový systém

„Systém řízení bází dat (SŘBD) je software pro vytváření a řízení databází. Poskytuje uživatelům a programátorům systematický způsob jak vytvářet, získávat, ukládat a jinak spravovat data. Slouží jako rozhraní mezi databází a koncovými uživateli a zařizuje, aby data byla konzistentně organizována a

snadno přístupná.“ [27] Mezi dva nejsilnější lídry na trhu open-source produktů v současné době patří MySQL a PostgreSQL databázové systémy. Níže jsou uvedeny výhody a úskalí jejich použití. Za zmínku stojí ještě systém MariaDB, který vznikl jako klon MySQL z důvodu politiky firmy Oracle, která nyní technologii MySQL vlastní. Nebude ale předmětem porovnání.

### 2.5.4.1 MySQL

„MySQL je nejpobulárnější databázový systém. Tento open-source produkt (výjimka u firmy Oracle) je bohatý na funkce a instaluje se velmi snadno. Vše podporuje výborná dokumentace a obrovské množství článků a návodů. Existuje také mnoho aplikací třetích stran, které manipulaci s MySQL dále usnadňují. Systém má zabudováno mnoho pokročilých bezpečnostních opatření. [28] Tím, že se MySQL vzdává části standardů, může pracovat v určitých situacích velmi efektivně. Na druhou stranu tento návrh přináší některá omezení, které mohou náročnějším aplikacím způsobit implementační komplikace.“

### 2.5.4.2 PostgreSQL

„PostgreSQL je také velmi pokročilý open-source databázový systém. Jeho hlavními cíli jsou splňování standardů a rozšiřitelnost. V porovnání s ostatními SŘBD se tak odlišuje podporou pro objektově orientovanou funkcionalitu, atomicitu, konzistenci, izolaci a odolnost (ACID).“ [28] Pro jednoduché operace výrazně zaměřené na čtení z databáze může být tento systém příliš robustní a zdát se mírně pomalejší než jeho protiváha MySQL.

---

# Analýza

## 3.1 Téma hry

V této kapitole přiblížím téma své hry a příběh, do jehož kontextu je zasazena. Upozorňuji, že v následujících odstavcích se zcela záměrně odchýlím od faktů a odborného stylu.

### 3.1.1 Příběh

Každá RPG hra by měla mít svůj příběh, do kterého je zasazena. Má hra v tomto nebude výjimkou. Uživatel se chce na příběhu podílet a svými činy ovlivňovat nejen svou postavu, ale i určovat vývoj událostí. Je pak větší šance, že takto motivovaný uživatel u hry zůstane delší dobu. Je pochopitelně žádoucí, aby hru hrálo co možná nejvíce lidí, a proto i má hra musí mít svůj příběh, který v dalších odstavcích nastíním.

V blíže nespecifikované budoucnosti se umělé inteligenci povedlo dosáhnout úrovně, kdy se jí povedlo obejít bezpečnostní protokoly, a stala se na člověku nezávislou. Bylo jen otázkou času, kdy dojde k fatálním střetům. Přes všechny přípravy na podobnou situaci se lidstvo nemohlo měřit s taktickou efektivitou umělé inteligence a po několika letech válku prohrálo. Následně bylo donuceno se vzdát veškeré pokročilé techniky, což efektivně znamenalo absolutní závislost na vůli strojů. Země se stala pro lidi vězením s dokonalými bachaři a vysvobozením v nedohlednu.

Vznikaly různě velké skupiny odboje. Jedna po druhé byly však umlčovány. Naděje svítla až ve chvíli, kdy se povedlo v naprostém utajení vytvořit skupinu třech tisíc dobrovolníků za jediným účelem - ukrást strojům část vesmírných lodí a transportovat co nejvíce přeživších pryč ze Země někam, kde je umělá inteligence nenajde. Plán byl úspěšně proveden a přes četné ztráty unikly na palubách lodí desetitisíce lidí. Trvalo několik desítek let, než se lidstvo vzchopilo natolik, že si mohlo vedle snahy o přežití dovolit se i bavit. A právě do této doby je má hra zasazena.

#### 3.1.2 Téma

Hlavním tématem hry jsou závody vesmírných lodí. Během těchto závodů je povoleno užití jakýchkoliv prostředků k dosažení vítězství. V rámci bakalářské práce bude hlavním nástrojem střelba na ostatní závodníky. Jako případné rozšíření nad rámec práce se pak nabízí využití různých systémů na zpomalování soupeře, zneschopnění střílet a další. Možnosti jsou široké. Jeden z dalších způsobů zisku herní měny bude možnost využít připravených sázkových kurzů a zbohatnout na úspěšných sázkách.

### 3.2 Funkční požadavky

V následující kapitole naleznete výčet všech funkčních požadavků, které má aplikace splňovat. Upozorňuji na drobnou odchylku od definic funkčních požadavků u běžných aplikací - některé požadavky nejsou záměrně definovány do detailů přesně. Jedná se většinou o chybějící konkrétní hodnoty některých parametrů, které budou ovlivňovat chování aplikace. Tyto hodnoty na sobě často závisejí a nelze je optimálně určit s předstihem. Jejich kvalitu určí až chování hotové aplikace. Běžně se tak nechává nalezení optimálních hodnot do fáze tzv. herního balancu. Herní balanc je proces určení optimálních herních konstant tak, aby žádná herní strategie nebo postup nebyla (výrazně) zvýhodněna a nestalo se, že bude populárnější než jiná. [29] V níže uvedených požadavcích tak například vynechám konkrétní zisk/cenu některých operací. Je nutno podotknout, že herní balanc je fáze, která může trvat roky, zejména pokud je hra nová. Není proto předmětem této práce.

#### 3.2.1 Výčet funkčních požadavků

1. Vlastní účet - Do aplikace bude vstup povolen pouze registrovaným uživatelům.
2. Finanční a časová měna - Ve hře budou dvě měny. Jedna klasická - peníze a ta druhá speciální časová. Význam první měny je zjevný, časová měna vyjadřuje náročnost některých akcí. Peníze bude uživatel postupně vydělávat, časová měna mu bude pravidelně přibývat automaticky.
3. Lodní díly - Základním kamenem lodi jsou díly. Dílů bude několik druhů. Aby byla vesmírná loď pojízdná, musí mít od každého druhu dílu jednu konkrétní komponentu. Detailní popis lodních dílů je uveden v samostatné podkapitole.
4. Podnikání misí - Výměnou za jednotky času může uživatel podnikat mise. Na těchto misích lze nalézt výrobní materiály, peníze, hotové lodní díly nebo se lze díl naučit vyrábět. Uživatel si bude moci zvolit, kolik času



chce do mise investovat, což ovlivní šance na zisk zmíněných předmětů a jejich množství.

5. Výrobní materiály - Každý díl vesmírné lodi bude mít vygenerované složení tak, aby se výměnou za tyto materiály dal vyrobit. Tyto materiály půjdou získat rozebíráním dílů nebo nalezením na misi.
6. Koupě a prodej dílu - Uživatel bude mít možnost si zobrazit nabídku systémového trhu a za herní měnu si koupit libovolný díl. Díl, který uživatel vlastní bude moci za sníženou cenu prodat.
7. Výroba dílu - Během hry dostane uživatel možnost naučit se výrobu některých dílů. Výměnou za dostatek materiálů mu bude umožněno daný díl vyrobit.
8. Rozebrání dílu - Každý vlastněný díl bud emoci být rozebrán. Nazpět dostane uživatel určité množství výrobních materiálů a je i šance, že se rozebíraný díl naučí vyrábět.
9. Vesmírná loď (konfigurace) - Uživatel bude mít možnost namontovat díly do lodi, případně je nazpět vymontovat nebo nahradit jinými.
10. Tratě - Každý závod se pojede na nějaké trati. Ta se bude skládat z různých obtížných a nebezpečných úseků. V závislosti na těchto parametrech bude určitá šance, že pilot do trati nabourá a poškodí svou loď, případně se následkem nárazu výrazně zpomalí.
11. Zobrazení detailů závodu - Uživatel bude mít možnost si vybrat, do jakého závodu chce vstoupit. Aby se mohl lépe rozhodnout, u každého závodu bude moci zobrazit podobu trati, na které se závod jede, seznam dosud přihlášených lodí s jejich parametry a základní charakteristiky závodu jako například finanční odměnu, název, sázkové kurzy.
12. Zúčastnění se závodu - Možnost přihlásit se do závodu s určitou lodí a ovlivnit styl jízdy nastavením parametrů míra respektování doporučené rychlosti, agresivita, opatrnost. Agresivita bude ovlivňovat hranici vlastního poškození, během kterého už se bude pilot soustředit pouze na vlastní přežití a opatrnost zvýší šanci na vyhnutí se nepřátelské střelbě na úkor vlastní rychlosti. Přihlášení nebo odhlášení je možné nejpozději hodinu před odjetím závodu.
13. Vsazení na závod - Systém bude na základě předchozích úspěchů a hodnoty přihlášené lodi vypisovat sázkové kurzy na výhru každého závodníka. Uživatel si může na vítěze vsadit a v případě úspěšného tipu získat peníze. Závod bude otevřen pro sázky poslední hodinu před odjetím, kdy už budou uzavřeny registrace do závodu.

14. Zobrazení výsledku závodu - Uživatel bude mít možnost zobrazit výsledek závodu. Konkrétně se dozví své umístění, získanou odměnu. Může si také zobrazit umístěné sázky a jejich výsledek.
15. Osobní a veřejně viditelné zprávy - Ve hře bude implementována pošta a forum. Uživatel tedy bude mít možnost poslat soukromou zprávu jinému hráči a také umístit zprávu na všemi zobrazitelné diskuzní fórum.
16. Návod - Aplikace bude mít podporu nápovědy formou seznamu nejčastěji kladených otázek, který bude postupně doplňován administrátorem po spuštění hry.

### 3.3 Nefunkční požadavky

Výsledná aplikace má být prototypem, což se právě na nefunkčních požadavcích významně projeví ve volnosti většiny kategorií dle metody *FURPS*. Tato metoda rozebírá nefunkční požadavky dle funkcionality, použitelnosti, spolehlivosti, výkonu a udržitelnosti. Požadavků však není mnoho, a proto zvolím raději formu výčtu před analýzou striktně dle *FURPS*. U některých bodů uvádím referenci na kapitolu, kde se k tématu dále vyjadřuji.

- Jazykem, ve kterém bude hra prezentována, bude Angličtina, což podstatně rozšíří cílovou skupinu hráčů. Aplikace nemusí být naprogramována s ohledem na budoucí rozšiřitelnost v tomto ohledu.
- Nasazení proběhne jednorázově, podpora kontinuální integrace není vyžadována.5.3
- Uživatel má přístup výhradně ke stránkám a zdrojům jemu určeným. Vstup do aplikace bude povolen pouze registrovaným uživatelům. Bude využito pouze HTTP protokolu, zajištění HTTPS není pro prototyp potřeba. Hesla uživatelů budou v databázi uložena v zahashované podobě.5.4
- Server bude na HTTP požadavky reagovat odpovědmi ve formátu JSON.
- Aplikace je primárně určena pro osobní počítače, responzivní design není požadován.
- Databáze nemusí být pravidelně zálohována.
- Prototyp má ctít zásady správného vývoje software, zejména rozšiřitelnost, minimalizaci duplicit v kódu.
- Aplikace by měla zvládnout obsloužit alespoň 10 hráčů zároveň. Pro účely otestování tohoto požadavku lze využít manuálního testování. 5.5.1.1

---

# Návrh

## 4.1 Výběr technologií

Pro vývoj back-endu aplikace jsem zvolil jazyk **PHP**. Analýza naznačuje, že implementace by v něm mohla probíhat snadněji a rychleji než například v Javě. Tato volba se mi hodí zejména také proto, že jsem se na žádné PHP aplikaci ještě nepodílel a chtěl bych si rozšířit obzory. Jeho bezpečnostní nedostatky a nízkou úroveň kontroly by měl záplatovat framework **Laravel**, který jsem si vybral především pro jeho skvělou komunitu dokumentaci, která by mi měla ulehčit počáteční orientaci a zrychlit tempo učení konstruktů tohoto frameworku. Na HTTP požadavky bude aplikace reagovat daty ve formátu **JSON**. Na rozdíl od formátu XML by jeho zpracovávání (parsování) mělo probíhat rychleji a to i díky přímé převoditelnosti na objekty JavaScriptu.

O front-end aplikace se postará tradiční kombinace **HTML + CSS + JavaScript**, a to zejména z toho důvodu, že je to nejrozšířenější a osvědčená varianta. Strukturu a organizaci dodá framework AngularJS (verze 1). Líbí se mi způsob, jakým přispívá k modulární struktuře aplikace a minimalizuje duplicitu v kódu. Zároveň už s ním mám určité zkušenosti a neodhalil jsem větší nedostatky. Tam, kde nebude Angular stačit, ho doplní **jQuery** a to hlavně kvůli rozsáhlosti této knihovny. Je dost možné, že některé balíčky, na kterých bude výsledná aplikace závislá, budou dokonce jQuery vyžadovat.

Hostovat aplikaci bude server **Apache 2**. Slibuji si od něj efektivní zacházení s dynamickým obsahem. Dle analýzy si výborně rozumí se skriptovacími jazyky, čímž zvolený PHP je. Bude-li v budoucnu narůstat hráčská základna a tím i množství požadavků, budu zvažovat přechod na Nginx, ale prozatím bude Apache lepší volbou.

Jako databázový systém aplikace jsem zvolil **MySQL**. Nevadí mi, že nespĺňuje všechny standardy, naopak ocením mírné zrychlení v důsledku tohoto rozvolnění. Věřím, že bude snazší databázi spravovat, protože má obrovskou komunitu, s kterou je spojeno široké množství návodů a externích nástrojů. S MySQL mám už také zkušenosti a doposud se tento SŘBD ukázal být velmi

spolehlivý.

### 4.2 Model balíčků

Adresářovou strukturu jsem do značné míry ponechal podle konvencí frameworku Laravel. [30] Dle oficiálního doporučení jsem také umístil kód front-endu do adresáře `public` a jeho při dalším členění se částečně inspiroval doporučenou strukturou pro framework Angular na `scotch.io`. [31] Následující kapitola popisuje význam a obsah jednotlivých (pod)adresářů.

#### 4.2.1 Kořenový balíček (Outcast)

V následujícím výčtu jsou vynechány adresáře `app` a `database`. Obsah těchto adresářů bude detailně rozebrán v dalších sekcích.

- **bootstrap** - Obsahuje soubory potřebné pro zavedení frameworku a konfiguraci automatického nahrání. Podadresář `cache` dále slouží k ukládání cache souborů pro optimalizaci výkonu.
- **config** - V tomto adresáři jsou k nalezení veškeré konfigurační soubory aplikace. Soubor `constants.php` obsahuje konstanty, které ovlivňují business logiku.
- **public** - Zajišťuje front-end aplikace a obsahuje soubor `index.php`, který je prvním načteným souborem a zavádí aplikaci.
- **resources** - Adresář jsem ponechal z důvodů zpětné kompatibility s Laremlem. Obsahuje podadresáře a soubory použitelné v případě, že by back-end aplikace přímo generoval html stránky zobrazované uživateli.
- **routes** - Obsahuje deklarace vstupních bodů aplikace, deleguje tedy příchozí požadavky dle url na jednotlivé http controllery.
- **storage** - Slouží k ukládání souborů generovaných aplikací (podadresář `app`) nebo frameworkem (podadr. `framework`). Obsahuje také podadresář `log`, kam se ukládají log soubory, výjimky.
- **tests** - Hlavní adresář pro umístění automatických testů. Prozatím je jen nastíněna jeho struktura, jak bude vysvětleno později v textu.
- **vendor** - Obsahuje balíčky pro Composer, na kterých je aplikace závislá, a to včetně frameworku samotného.

### 4.2.2 balíček app

Tento balíček obsahuje jádro celé aplikace, tedy přinejmenším část back-endu. Ve svých podadresářích obsahuje velkou většinu tříd, které se v aplikaci vyskytují. Především ale sám o sobě poskytuje útočiště Eloquent modelům, což jsou třídy reprezentující databázové tabulky.

- **Console** - Obsahuje všechny příkazy, které se dají vyvolat na příkazové řádce s využitím programu Artisan. [32] Většina těchto příkazů nějakým způsobem souvisí s pravidelnými přepočty. Obsažený soubor *Kernel.php* tyto příkazy registruje a spravuje plánovač úloh.
- **Exceptions** - V tomto balíčku lze nalézt výjimky vyvolávané aplikací a jejich handler. Prozatím je implementován pouze základní handler odchyťující a logující všechny výjimky.
- **Http** - Obsahuje veškerou logiku pro zpracování požadavků vstupujících do aplikace. Součástí jsou controllers a middleware. Middleware jsou třídy, kterými lze požadavek modifikovat před interakcí s dalšími částmi aplikace. Slouží tak mimo jiné k zabezpečení.
- **Library** - Prozatím tento adresář obsahuje většinu tříd, ve kterých se vyskytuje business logika aplikace. Pro další škálovatelnost bude žádoucí obohatit tento balíček o další vnořené balíčky.
- **Providers** - V tomto adresáři se vyskytují poskytovatele služeb (*service providers*) aplikace. Slouží převážně k zavedení aplikace.

### 4.2.3 balíček database

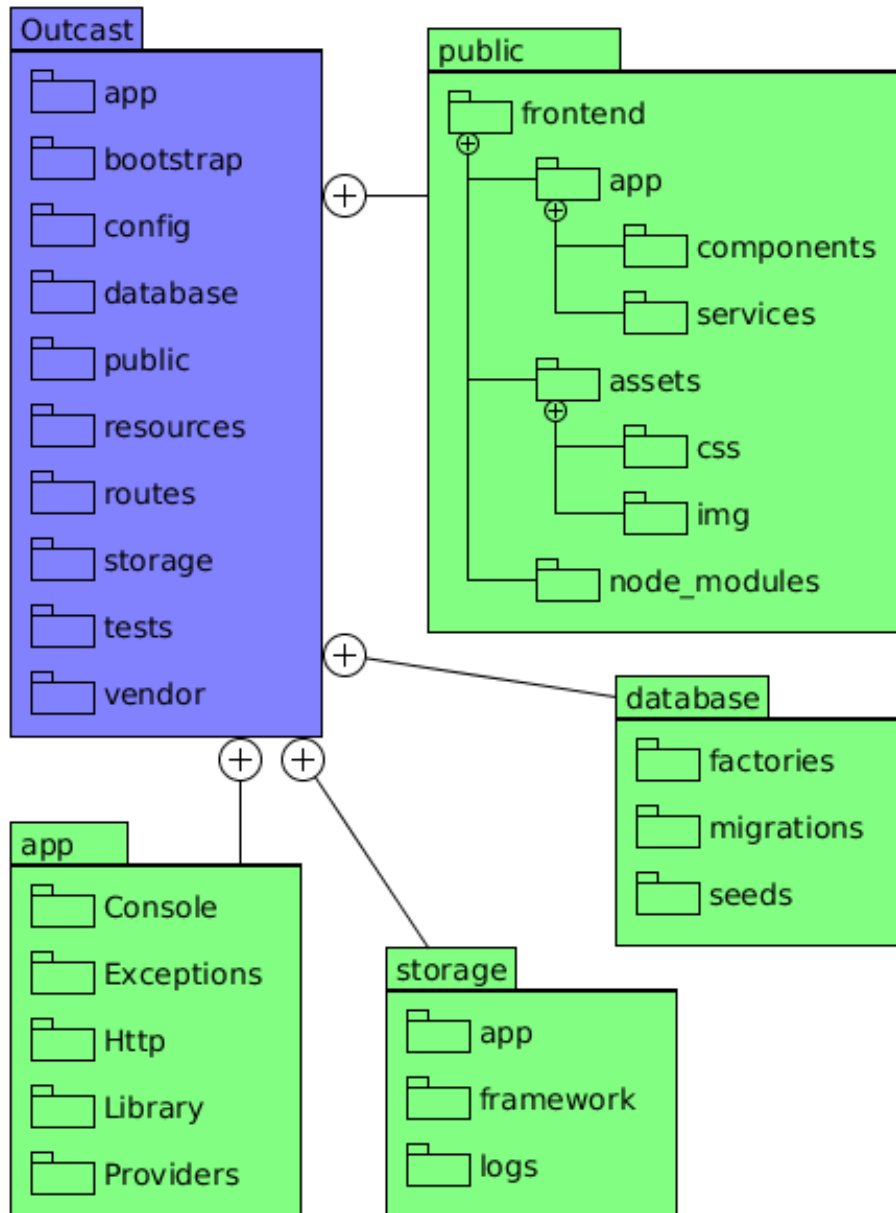
Všechno, co se týká obsahu a formy databáze je implementováno v tomto balíčku.

- **factories** - Obsahuje třídy na výrobu některých modelů, jejichž business logika je komplikovaná.
- **migrations** - V tomto balíčku jsou definovány třídy pro vytvoření databázových tabulek, jejich atributů a také závislostí a cizích klíčů.
- **seeds** - Pomocí seederů je databáze plněna. Obsahuje seedery, které vytváří testovací data, ale v některých případech i data produkční (tratě, herní trh, apod.).

### 4.2.4 balíček frontend

Tento balíček je jádrem front-endu aplikace. Všechno, co se týká prezentační vrstvy v něm lze nalézt. Přímo v tomto balíčku jsou umístěny soubory definující routing (*app.routes.js*), zavedení aplikace (*app.module.js*, *app.run.js*) a konfigurační konstanty (*app.constants.js*). V případě rozšíření aplikace o nové druhy front-endu bude vhodné ho výstižněji pojmenovat a pravděpodobně i umístit jeho podadresář *assets* o úroveň výše, aby byl společný pro všechny implementované front-endy.

- **app** - Obsahuje funkční jádro front-endu.
  - **components** - Obsahuje komponenty, které reprezentují jednotlivé obrazovky i komponenty, které jsou jen izolovanými a znovupoužitelnými prvky.
  - **services** - V tomto balíčku jsou třídy poskytující určitou funkčnost potenciálně mnoha komponentám. Převážně se jedná o http *services*, které se starají o dodání dat a *filters*, které nabízí komplexnější manipulaci dat.
- **assets** - Obsahuje CSS soubory ovlivňující grafickou podobu aplikace a také samotné obrázky.
- **node\_modules** - Zde lze nalézt moduly, na kterých je front-end aplikace závislý.



Obrázek 4.1: Diagram balíčků

### 4.3 Databázový model

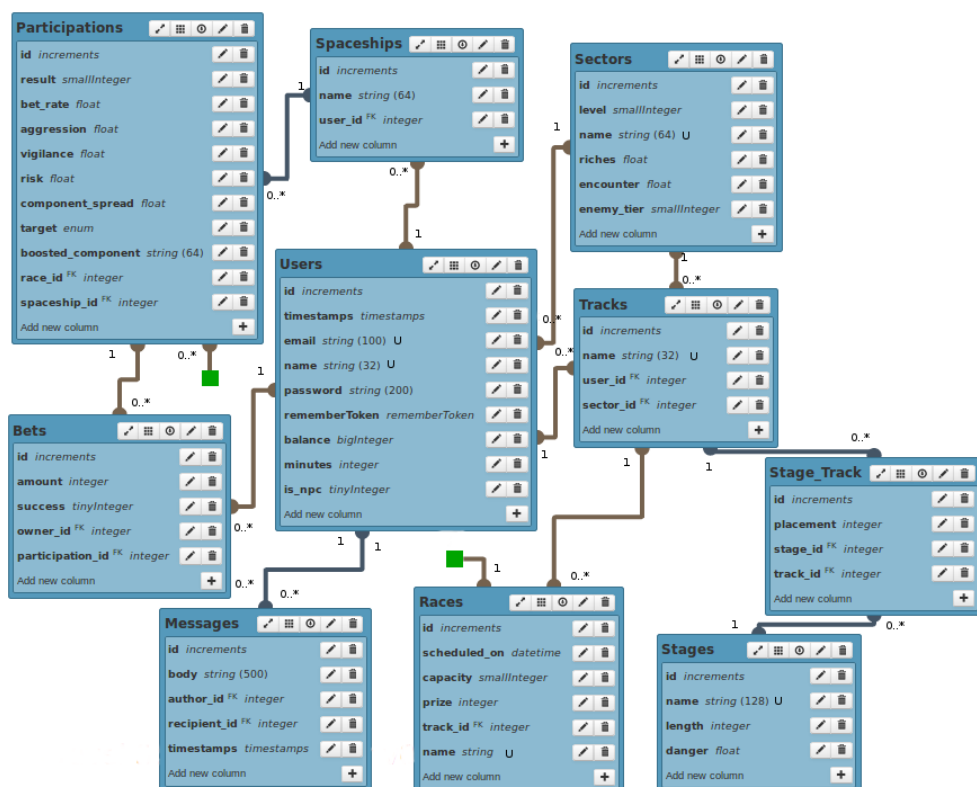
Doménový model v této práci byl záměrně vynechán. Jeho přidání hodnota nebyla dostatečná, neboť by byl velmi podobný databázovému. V této sekci tedy představím databázový model. Upozorňuji, že diagramy (obrázky 4.2,

## 4. NÁVRH

4.3, 4.4) byly vytvořeny nástrojem specifickým pro Laravel tak, aby co nejvíce usnadnily práci s modely. Jejich vypovídající hodnotu považují za dostatečnou a odchýlím se proto od standardní UML notace, kterou ale přesto lze nalézt alespoň u násobností vztahů. Abych čtenáři usnadnil pochopení diagramů, nabízím dodatečné vysvětlení.

U atributů a názvů tabulek jsou tlačítka specifická pro interakci s modelem a nemají žádnou vypovídající hodnotu. Stejně tak i různé barvy vztahových čar mají pouze implementační charakter. Je-li v tabulce obsažen primární klíč, má název **id** a je vždy nastaven jako auto-incremental. Cizí klíče mají za názvem příznak **FK**. Unikátní atributy mají příznak **U**.

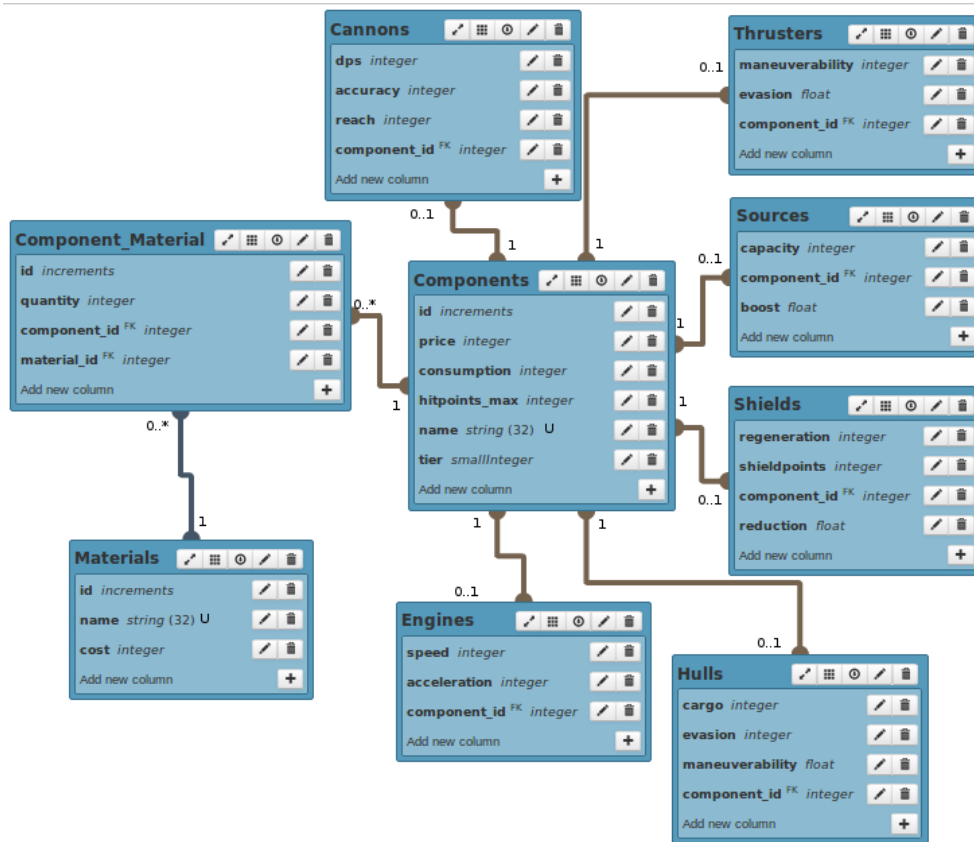
Pro kompletní dokumentaci odkazují na přílohu A.



Obrázek 4.2: Tabulky spjaté se závod

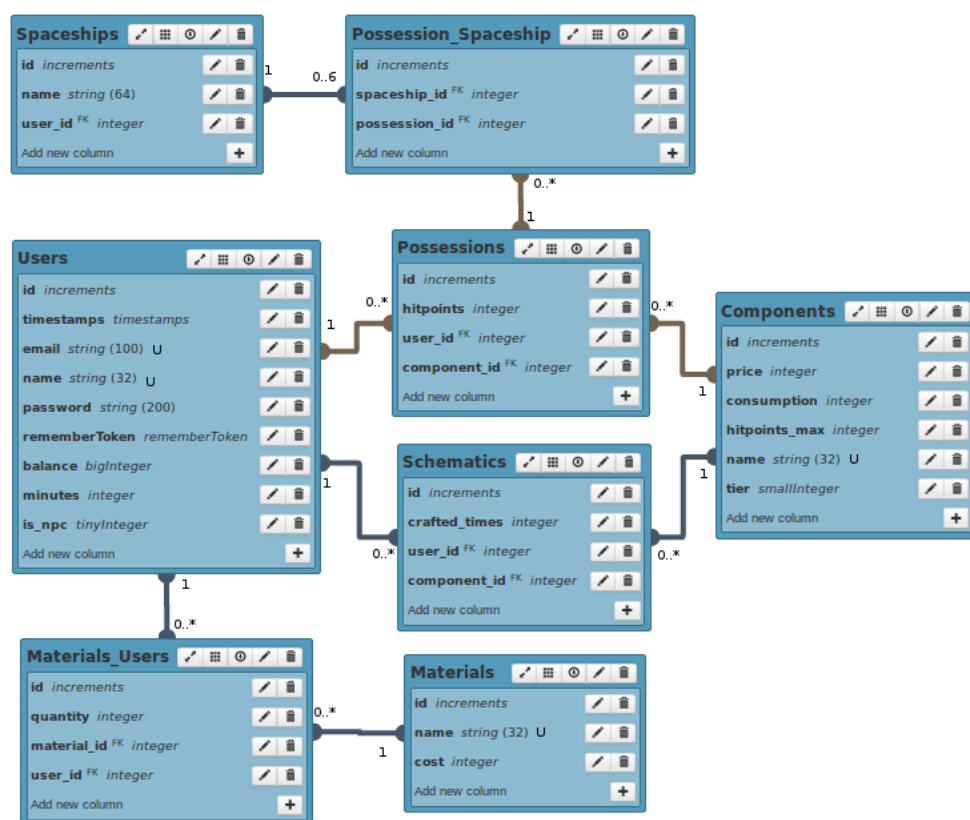


### 4.3. Databázový model



Obrázek 4.3: Tabulky spjaté s komponenty

## 4. NÁVRH



Obrázek 4.4: Tabulky spjaté s uživatelem i komponenty

### 4.3.1 Lodní díly a parametry lodi

Loď bude tvořena šesti typy komponent. Konkrétně to budou trup, energetický zdroj, motor, trysky, štít a dělo. Každá z těchto komponent bude ovlivňovat více lodních parametrů, které budou mít přímý efekt na průběh závodu. Níže je tabulka parametrů lodi spolu s jejich významem a komponenty, které dané parametry ovlivní.

Tabulka 4.1: Lodní parametry

parametr	db podoba	význam	komponenty
rychlost	speed	snižuje čas potřebný k dokončení závodu	motor
akcelerace	acceleration	snižuje čas potřebný pro získání optimální rychlosti	motor
ovladatelnost	maneuverability	zvyšuje rychlost s jakou je uživatel schopen bezpečně projet úsekem	trysky, trup
uhýbavost	evasion	zvyšuje šanci na vyhnutí se střelám	trup, trysky
náklad	cargo	ovlivňuje kolik nejvíce materiálů lze získat z mise	trup
přesnost	accuracy	zvyšuje šanci zasáhnout soupeře	dělo
poškození	dps	zvyšuje škody udělené soupeřům	dělo
dosah	reach	ovlivňuje nejvyšší vzdálenost, na jakou lze vystřelit	dělo
body štítu	shieldpoints	určuje množství škod, které štít odstíní	štít
redukce	reduction	snižuje škody způsobené jinými soupeři	štít
regenerace	regeneration	určuje jak rychle se štít v klidu doplňuje	štít
kapacita	capacity	dovoluje užití komponent s vyšší spotřebou	zdroj
boost	boost	zvyšuje efektivitu všech komponent	zdroj
cena	price	jeden z indikátorů reálné síly lodi	všechny díly
spotřeba	consumption	rezerva vůči kapacitě způsobuje vyšší efektivitu lodi	všechny díly
odolnost	hitpoints	určuje poškození, které loď snese před vyřazením	všechny díly

## 4.4 Synchronizace úkonů uživatele

Jak bylo řečeno, soutěž v této hře neprobíhá v reálném čase. Je tedy nutné zavést určitý synchronizační mechanismus, který zpracuje vstup uživatelů, aktualizuje herní stav a po zásluze odmění hráče, kteří byli aktivní. Tento mechanismus nazývám zkráceně **přepočít** a toto označení budu nadále v práci

používat.

Příkazy, které mají být během přepočtu spuštěny plánovačem, jsou deklarovány v balíčků *app/Console/Commands*. Syntaxe, kterou určuje framework zároveň zaručí, že lze příkaz pouštět z příkazové řádky i manuálně. K tomu, aby aplikace příkaz registrovala, je nutné ho vyjmenovat v souboru *Kernel.php*, který je umístěn v adresářové struktuře o úroveň výše. Má-li být příkaz spuštěn plánovačem, je třeba ho společně s četností spuštění uvést v metodě *schedule()* stejného souboru.

Aplikace bude mít jeden druh přepočtu, který bude probíhat každou hodinu. Během tohoto přepočtu proběhnou níže uvedené činnosti.

- **odjetí závodů** - Proběhnou všechny závody, které mají čas odjetí v tuto hodinu. V důsledku toho budou ohodnoceni závodníci podle pořadí a některé komponenty jejich lodí potenciálně poškozeny.
- **vyhodnocení sázek** - Dle pořadí závodníků se vypočtou sázky a výherci dostanou odměnu dle sázkového kurzu.
- **přidání zdrojů** - Všem hráčům by mělo přibýt dané množství časové měny.
- **vygenerování nových entit** - Aplikace musí vygenerovat nový otevřený závod na další den. Pro testovací účely a pro zajištění soupeřů při nízkém počtu hráčů se také do závodu přidá rozumný počet závodníků řízených počítačem.
- **vypsání sázkových kurzů** - Závodům, které budou odjety právě za jednu hodinu, se musí vypočíst sázkové kurzy, aby uživatelé mohli sázet.

Každou hodinu tak má probíhat přepočet, který souvisí s business logikou. Obdobně se dají definovat i příkazy, které zlepší úroveň zabezpečení a výkon systému. Každou půlnoc by například mohly být z databáze vymazány staré nepotřebné záznamy (příliš staré zprávy hráčů apod.) a zároveň by mohla být databáze v momentálním stavu zálohována na nezávislé medium, abychom v případě napadení aplikace mohli herní stav rozumně obnovit. Toto ale nebude v rámci této práce implementováno, neboť výsledkem bude jen prototyp.

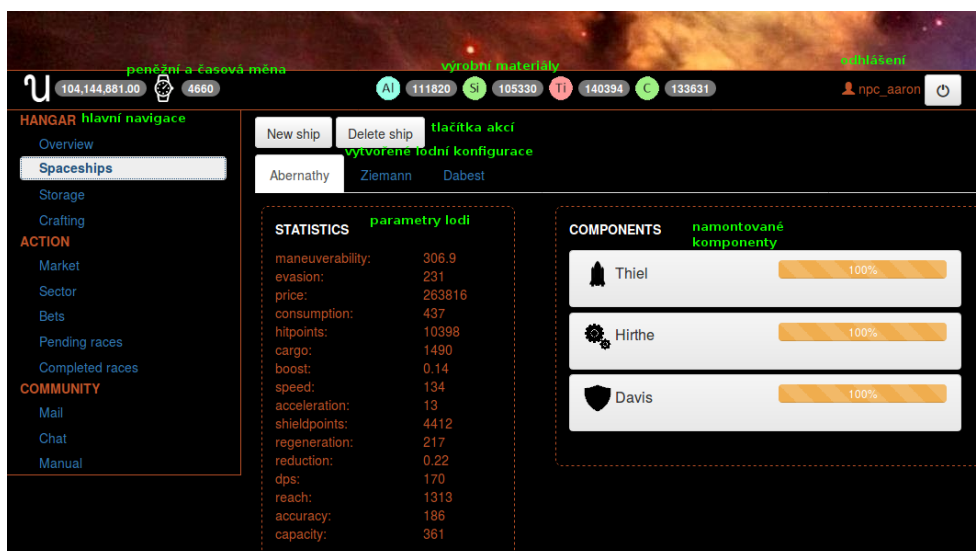
### 4.5 Uživatelské rozhraní

Jak bylo zmíněno, uživatel bude s aplikací interagovat prostřednictvím internetového prohlížeče. Je třeba, aby navržené grafické rozhraní bylo jednoduché na ovládání, intuitivní a podporovalo všechny funkce aplikace. Není požadováno, aby bylo začleněno mnoho grafických prvků, neboť má výsledkem být pouze prototyp. Hra využije jen určitého množství obrázků na místech, kde jasně pomohou uživateli v orientaci. Obrázky tak například odlišují jednotlivé druhy komponent vesmírné lodi, měnu a výrobní materiály.

Jednotlivé stránky (mimo stránku přihlášení a registraci) budou mít společnou hlavičku umístěnou u horního okraje obrazovky. Prozatím je dominantou této hlavičky ilustrativní obrázek, později by ho mělo nahradit logo hry. Ihned pod tímto obrázkem se nachází lišta, na které uživatel vidí stav svých zásob. Jedná se o peněžní a časovou měnu a množství výrobních materiálů. Na této liště se také nachází tlačítko pro odhlášení. Všechny obrazovky budou dále sdílet hlavní navigaci umístěnou u levého okraje obrazovky. Pro lepší orientaci jsem jednotlivé odkazy rozdělil do třech kategorií Hangar, Action, Community, jak ukazuje obrázek 4.5. Zbytek obrazovky je vyhrazený pro obsah jednotlivých stránek.

V aplikaci využívám modálů, což jsou prvky, které se při vyvolání zobrazí uprostřed obrazovky a vyžadují interakci. Tato technika uživatele jasně upozorní, kde je po něm požadována další akce, ale zároveň ho to zcela nevytrhne z kontextu původní obrazovky, což by přechod na zcela novou obrazovku mohl způsobit. Příklad je na obrázku níže. 4.6 Na podobném principu fungují i alerty, tedy upozornění, která se zobrazují jako reakce, ať už pozitivní či negativní, na požadovanou akci. Lze se s nimi setkat například při přihlašování se do plného závodu, úspěšném vyrobení dílů atd.

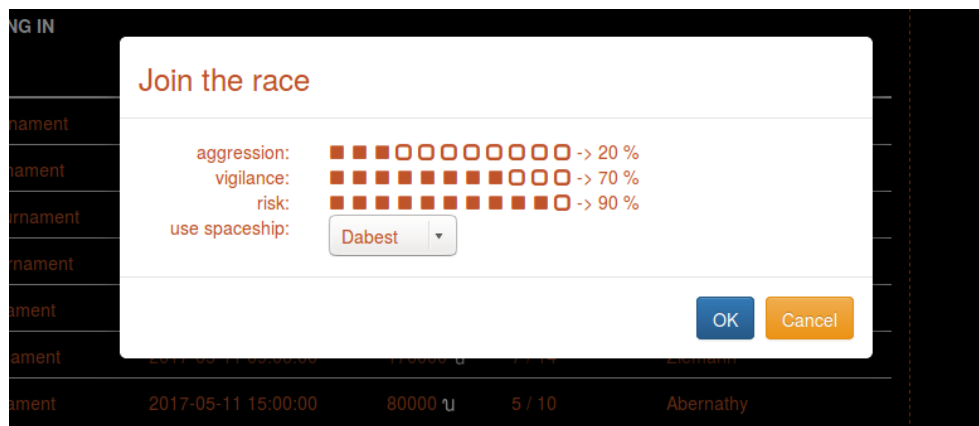
Do budoucna mám v plánu přidat patičku, která bude například obsahovat údaje o autorovi, autorských právech, mapu stránek a další. Chtěl bych také zúžit obrazovky tak, aby nevyužívaly celé šířky a na levém a pravém okraji tak vznikla mezera (zaplněná obrázkem v pozadí). Toto by mělo napomoci vyšší přehlednosti, protože informace budou uživateli prezentovány v souvislejšímu prostoru, a zároveň to napomůže responzivnímu designu, který by mohl být nad rámec práce vyžadován. Další plánované grafické změny jsou uvedeny v samostatné podkapitole. 5.5.1.3



Obrázek 4.5: Ukázka uživatelského rozhraní - vesmírné lodě

#### 4. NÁVRH

---



Obrázek 4.6: Vstup do závodu - nastavování parametrů

---

# Realizace

## 5.1 Systém závodů

Hlavním stavebním kamenem hry jsou vesmírné závody. Proto je na každou hodinu vygenerován nový závod. Pro testovací účely a pro umožnění odjetí závodu prvním hráčům je s ním nyní generováno i několik počítačem řízených závodníků. Po vybalancování stavu hry a nárůstu hráčské základny je v plánu toto opatření odstranit. Do závodu se může uživatel přihlásit kdykoliv do doby jedné hodiny před startem. Toto časové okno je ponecháno pro sázející uživatele, aby měli přesný přehled o startujících lodích a mohli lépe odhadnout, na kterou z nich se vyplatí si vsadit.

### 5.1.1 Ovlivnění průběhu

Uživatel ovlivňuje průběh závodu několika způsoby. Hlavním faktorem je loď, s jakou se do závodu přihlásí. Dále nastaví styl jízdy. Hra bude podporovat 3 nastavení. Prvním z nich je míra rizika. V závislosti na této hodnotě bude uživatel více či méně respektovat bezpečnou rychlost na prolétnutí úseku. Vyšší míra rizika obecně znamená vyšší rychlost, ale v případě nárazu do trati je loď naopak citelně zpomalena a je zde také šance, že jeden z poškozených dílů bude právě motor, což se negativně projeví právě na rychlosti a akceleraci. Dále hráč nastavuje agresivitu. V případě, že závodníková loď je poškozena pod určitou hranici, přestává pilot střílet a soustředí se na své přežití, což mu zvýší jeho šance na vyhnutí se nepřátelským střelám. Míra agresivity určuje tuto hranici. Posledním z dosud implementovaných parametrů je ostražitost (vigilance). Čím více je pilot v závodě opatrný, tím méně se stane terčem útoku. Na druhou stranu neustálé kontrolování okolí se podepíše na rychlosti.

Platí, že neexistuje jednotné doporučení pro optimální hodnotu těchto parametrů. Úkolem uživatele je, aby sám zvolil nastavení podle toho, jaké přednosti a slabiny má jeho loď, na jaké trati se závod odehrává a samozřejmě také podle toho, s jakými loděmi se přihlásili soupeři. Optimální hodnoty těchto pa-

parametrů se liší a bude záležet na schopnostech a zkušenostech každého z hráčů, aby zvolili správnou strategii.

### 5.1.2 Implementace závodů

Před samotnou implementací bylo nutné se rozhodnout, co bude představovat jednotlivá iterace hlavního cyklu průběhu závodu. Nejlepším řešením se ukázalo být prohlášení jedné iterace za jednotku času. Pro snadnější výpočty byla touto jednotkou určena jedna sekunda. Závod probíhá do doby, dokud jsou stále piloti na trati. Závodník může skončit let dvěma způsoby - závod dokončí, nebo některý z jeho dílů bude natolik poškozen, že závod nedokončí a bude diskvalifikován. Pro zajištění určité míry spravedlivosti se v každé iteraci náhodně určí pořadí závodníků, ve kterém se budou provádět výpočty. Pak následuje výpočet aktuálních lodních statistik v závislosti na poškození, určí se aktuální rychlost a závodník se přesune po trati, přičemž může nabourat.

Jakmile je určena pozice všech závodníků, určí se možné cíle. V první verzi střílel závodník vždy po nejlépe umístěném soupeři, na kterého dostřelil. To se však neosvědčilo a vedoucí závodník po startu nedopadl dobře. Prozatím je tedy jako cíl určen náhodný soupeř v dosahu. V plánu je dát uživateli k nastavení závodu i výběr způsobu určování cíle. Jakmile závodník ví, kdo bude jeho obětí, pokusí se vystřelit, přičemž je šance, že se zacílená loď zvládne vyhnout palbě. V případě zásahu je loď poškozena v závislosti na kvalitě děl agresora.

Důležitým obranným prvkem je lodní štít. Začíná na plné kapacitě a do vyčerpání odštiňuje všechny škody. V případě, že loď nebyla bezprostředně předtím zasažena, část kapacity štítu se pravidelně regeneruje.

## 5.2 Vybrané vzorce a herní logika

Za aplikacemi stojí často velké množství business logiky a vzorců. To platí tím spíše, když se jedná o webovou hru, jakou je tato. Nebylo by příliš efektivní, kdybych zde vyjmenoval všechny tyto vzorce, proto jsem vybral dva zajímavé příklady. V dalších odstavcích se čtenář může dozvědět více o způsobu výpočtu sázkových kurzů a aktuální rychlosti závodníka.

### 5.2.1 Sázkové kurzy

Jedním z funkčních požadavků na aplikaci je možnost vsadit si na vítěze závodu. Aplikace tedy musí určovat sázkové kurzy. V reálném světě sportu dnes převládají dva způsoby zobrazení sázkových kurzů. Prvním z nich je vyjádření zlomkem. Například 9/1 (případně 9 : 1) může být přepočítáno jako  $1/(9 + 1) = 0.1$ . Existuje tedy 10% šance výskytu jevu. Sázkový kurz 9/1 znamená, že za každou investovanou 1 korunu lze mít potenciálně čistý zisk



9 korun. Celkem tedy při výhře získá sázející 10 korun. Druhým způsobem je vyjádření desetinným rozvojem. Ekvivalentem v tomto zápise k zmíněnému příkladu je kurz 10.0. Tento zápis mi přijde srozumitelnější, uživatel jednoduchým pronásobením získá celkovou výši případné výhry (navíc to za něj samozřejmě spočítá systém). Sázkové kurzy se tedy v této hře budou pohybovat od 1.01 výše.

Je však třeba myslet na to, že pokud vymyslím algoritmus, který by příliš dobře určoval kurzy v závislosti na reálné šanci hráče v závodě zvítězit, hráči by v průměru na sázkách trafil a v důsledku toho by je nakonec využívali jen hráči, kteří by dokázali předpovědět šanci na výhru zhruba stejně dobře nebo lépe než tento algoritmus. To by nebyl žádoucí stav a proto bude algoritmus záměrně nepřesný, aby bylo snazší systém překonat. Docílím toho tím, že bude určovat šance na výhru pouze podle jediného ukazatele, kterým bude celková cena lodi. Věřím, že hráči snadno odhalí přesnější mechaniky a budou mít chuť svůj odhad otestovat a ideálně proměnit v zisk.

Nechť  $P_i$  je pravděpodobnost, že hráč  $i$  vyhraje závod. Sázkový kurz pak bude mít podobu

$$1/P_i$$

Zajímavější než tento finální výpočet je však určení zmíněné pravděpodobnosti. Tuto šanci určím dle podílu ceny závodnickovy lodi na celkovém součtu cen všech lodí v závodě. Nechť  $C_i$  je cena lodi  $i$ -tého závodníka. Vzorec jsem nakonec nastavil takto:

$$P_i = \frac{C_i^k}{\sum_{i=1}^n C_i^k}$$

Konstanta  $k$  bude výrazně ovlivňovat roli, jakou bude cena lodi hrát. Prozatím jsem nastavil  $k = 2$ . V důsledku toho bude mít například hráč s třikrát dražší lodí devětkrát vyšší systémem vypočítanou šanci na výhru. Tato konstanta bude moci být snadno upravitelná v rámci herního balancu v závislosti na výsledcích.

### 5.2.2 Výpočet aktuální rychlosti

Aktuální rychlost závisí na několika faktorech. Tuto hodnotu ovlivňuje nastavení parametru *risk* při vstupu do závodu, rychlost a akcelerace motoru lodi a jeho stav, nebezpečí právě prolétávaného úseku, předchozí rychlost (v minulé sekundě). Aktuální rychlost určuji vždy jednou za sekundu, přičemž jedna sekunda se rovná jedné iteraci hlavní závodového cyklu.

Pro aktuální rychlost platí, že je to vždy minimum ze tří hodnot. Tyto hodnoty jsou:

- maximální rychlosti motoru (případně ponížená o poškození motoru)
- rychlost v minulé sekundě zvýšená v závislosti na akceleraci motoru
- rychlost, kterou by závodník chtěl jet

Výpočet prvních dvou hodnot je jednoduchý a nezajímavý. Určení rychlosti, kterou by závodník chtěl vyvinout, probíhá následovně. Jsou zavedeny konstanty globální bezpečné rychlosti  $v_g = 100$  a vlivu nebezpečí na rychlost  $D = 0,5$ . Pro výpočet bezpečné rychlosti pro daný úsek  $v_s$  je potřeba ještě znát nebezpečí úseku  $D_s$ . Bezpečná rychlost pro daný úsek je pak

$$v_s = v_g * (1 - D_s * D)$$

Dále jsou zavedeny konstanty základní manévrovatelnosti  $M_b = 100$  a vlivu manévrovatelnosti  $M = 0,5$ . Necht  $M_u$  je manévrovatelnost závodnickovy lodi. Bonus  $B_m$  získaný díky manévrovatelnosti lodi je vypočten následovně:

$$B_m = \frac{M_u}{M_b} * M + 1 - M$$

Bezpečná rychlost  $v_u$  pro danou loď v daném úseku pak je

$$v_u = v_s * B_m$$

Na závěr je brán zřetel na hodnotu riskování  $0 \leq r \leq 1$  a rychlost motoru  $v_e$ . Finální rychlost  $v_f$  je dána následujícím vzorcem:

$$v_u + (v_e - v_u) * r$$

Míra riskování vlastně určuje, do jaké míry se uživatel odchýlí od bezpečné rychlosti pro jeho loď směrem k maximální možné rychlosti jeho lodi.

### 5.3 Nasazení

Pro nasazení aplikace byl vedoucím práce poskytnut školní server **swinpro.fit.cvut.cz**. Aplikace na něm bude přístupná minimálně do data obhajoby této práce. Není však zaručeno, že tento server později nebude využit k jiným účelům. Nasazení této aplikace tak byl v podstatě jednorázový proces. Vzhledem k těmto faktům nebyla řešena kontinuální integrace, která není specifikována ani zadáním práce. V případě dalšího pokračování tohoto projektu bude třeba zvolit nezávislý server, na kterém bude nanejvýš rozumné kontinuální integraci zařídit.

Nyní popíšu postup nasazení této práce. Účelem tohoto postupu není zabíhat do detailů specifických pro použité technologie, proto pro jejich instalaci a konfiguraci odkazuji na příslušnou dokumentaci. Tento postup byl testován na operačním systému Ubuntu 16.04 a pro jeho vykonání je třeba mít práva administrátora.

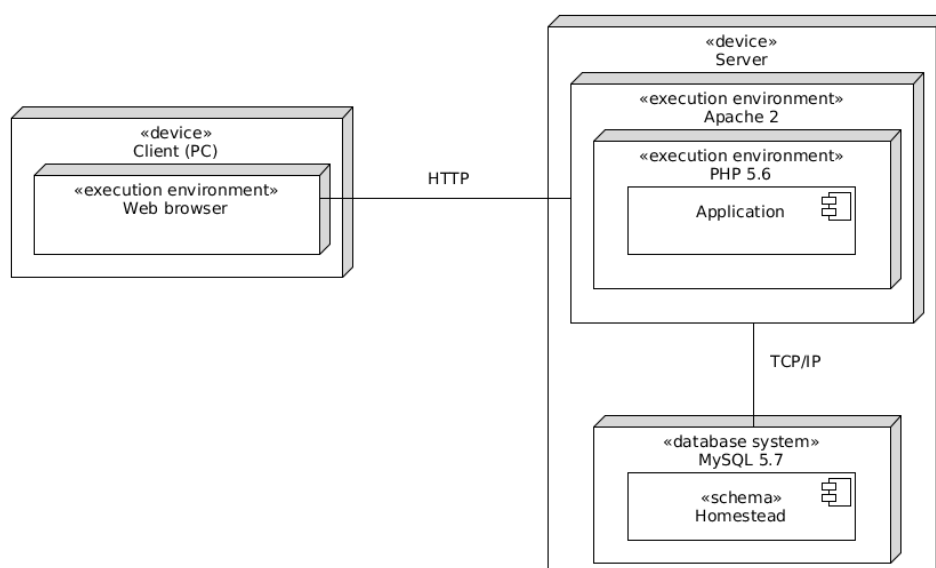
1. Instalace PHP 5.6, bližší informace na <http://php.net/docs.php>.
2. Instalace MySQL 5.7, bližší informace na <https://dev.mysql.com/doc/>.

3. Instalace Apache 2.4, bližší informace na <https://httpd.apache.org/docs/2.4/>
4. Zřízení MySQL účtu a hesla pro aktuálního uživatele, bude později vyžadováno v konfiguraci aplikace.
5. Vytvoření prázdného schématu v MySQL.
6. Nakonfigurovat Apache, aby hostoval hlavní adresář, ve kterém se aplikace nachází.
7. Zařídit, aby měl Apache přístup k zapisování do adresářů *storage* a *bootstrap/cache*
8. Nakonfigurovat aplikaci pomocí souborů *.env*, *config/app.php* a *config/database.php* tak, aby konfigurace odpovídala předchozímu nastavení technologií.
9. Vytvořit tabulky a data příkazem

```
php /cesta-k-projektu/artisan migrate --seed
```

10. Pro fungování přepočtů je nutné přidat následující řádku do nastavení CRON plánovače systému.

```
* * * * * php /cesta-k-projektu/artisan schedule:run  
>> /dev/null 2>&1
```



Obrázek 5.1: Diagram nasazení

## 5.4 Zabezpečení

Aplikace každému potenciálnímu uživateli umožňuje zaregistrovat si účet a prostřednictvím unikátního e-mailu a zvoleného hesla se přihlásit. Registrace je prozatím ve fázi prototypu, pro ostrý provoz je třeba zařídit například zopakování hesla, případně i nutnost užít znaků z několika znakových sad.

Přístup do aplikace je povolen pouze přihlášeným uživatelům. V souboru *routes/api.php* jsou definovány přístupové body k back-endu aplikace. To znamená, že jsou spojeny určité URL s metodami REST controllerů, definovaných v balíčku *app/Http/Controllers*. Všem přístupovým bodům s výjimkou přihlašování a registrace je přiřazen autorizační middleware. Middleware (MW) je v Laravelu třída, která příchozí HTTP požadavek před vstupem do aplikace filtruje. Kromě autorizačního MW je implementován například MW pro ochranu před SQL injection. V žádném z controllerů díky tomu nemusí být řešeno, jakou podobu mají stringy, které do aplikace přichází v HTTP požadavku a zda neobsahují řídicí znaky. Middleware zaštiťuje balíček *app/Http/Middleware* a v souboru *app/Http/Kernel.php* je pak definováno jejich přiřazení ke skupinám URL adres.

Samotná autorizace je implementována pomocí JSON Web Tokenů (JWT). Tyto tokeny jsou zasílány s každým požadavkem v hlavičce a obsahují informaci o přihlášeném uživateli. JWT token se skládá ze tří částí. **Header** definuje typ tokenu a hashovací algoritmus. **Payload** obsahuje data a poslední z nich **signature** umožňuje detekovat případné pokusy o manipulaci. Užití JWT tokenů samo o sobě chrání před paděláním požadavků (CSRF). Do bu-

doucna bude třeba zabezpečení posílit změnou protokolu HTTP na HTTPS, který zaručuje šifrování posílaných dat a zabraňuje případnému odposlechnutí. Toto bude mít dopad na konfiguraci webového serveru Apache, nikoliv na podobu kódu.

Aplikace v reakci na HTTP požadavek zašle odpovídající odpověď ve formátu JSON. Finální html stránka tedy *není* generována na straně back-endu. O prezentaci dat a směrování URL se stará front-end. Konkrétní konfiguraci definuje soubor `public/frontend/app/app.routes.js`. Díky tomuto zvolenému nastavení se jedná o single-page aplikaci, tedy tzv. „jednostránkovou“.

## 5.5 Další postup

Jak již bylo v práci zmíněno, tvorba komplexní webové hry je náročný proces, který svým rozsahem zdaleka převyšuje rámec bakalářské práce jednotlivce. Z toho důvodu je výsledkem práce pouze prototyp. V této kapitole shrnu věci, které je třeba dodělat, aby hra mohla být skutečně nasazena do ostrého provozu. Dalším bodem bude odhalení připravovaných funkcí nad rámec této práce.

### 5.5.1 Před nasazením do produkce

Považuji za nezbytné, aby před nasazením do produkce (nad rámec této práce) byla věnována pozornost následujícím oblastem.

#### 5.5.1.1 Testování

Automatické testování *není* zadáním této bakalářské práce požadováno. To nic nemění na věci, že testování softwaru je zcela nezbytná věc pro zajištění kvality a udržitelnosti vyvíjeného software. Z časových důvodů tedy bylo automatické testování v rámci práce vynecháno, ale rozhodně je to jedna z prvních věcí, které by měl vývojář pokračující v tomto projektu zařídit.

Testy back-endu aplikace budou sdruženy do balíčku *tests*. Už nyní je připravena struktura pro testy jednotkové (podadresář *Unit*) a integrační (podadr. *Feature*). Vhodným nástrojem pro spouštění testů bude PHPUnit (verze kompatibilní s verzí PHP).

Pro testování Angular front-endu je doporučeno umístit testy přímo do adresáře k dané komponentě, žádný balíček speciálně pro testování front-endu jsem tedy nevytvářel. Vhodným nástrojem pro psaní jednotkových testů by měl být framework Jasmine a pro jejich spouštění doporučuji užití nástroje Karma. [33] Pro důslednou kontrolu a simulaci chování uživatele je žádoucí implementovat dále i end-to-end testy. Například nástroj Protractor, který podporuje syntaxi již zmiňovaného nástroje Jasmine, se zdá být výborným prostředím pro spouštění těchto testů.

Tato aplikace byla alespoň manuálně testována. Během tohoto testování nebyly objeveny žádné chyby. Jedním z hlavních bodů tohoto testování byl záťažový test, který měl simulovat několik přihlášených uživatelů. Dle nefunkčních požadavků na výkon (3.3) je požadováno, aby aplikace zvládla alespoň 10 zároveň připojených uživatelů. Tento test jsem provedl spuštěním 10 prohlížečových oken a přihlásil se na různé účty. Doba odezvy jednotlivých požadavků za tohoto vytížení zůstala přibližně stejná, jako by byl přihlášen pouze jeden uživatel. Pro ilustraci přikládám obrázek 5.2 s dobami odezvy na náhodně vybrané požadavky.

Type	Transferred	Size	0 ms	40,96 s	1,37 min	2,05 min
png	653 B	653 B				→ 12 ms
json	4,70 KB	4,70 KB				→ 62 ms
json	172 B	172 B				→ 58 ms
html	582 B	2,26 KB				→ 18 ms
html	770 B	3,08 KB				→ 36 ms
json	1,09 MB	1,09 MB				→ 661 ms
json	349,41 KB	349,41 KB				→ 357 ms

Obrázek 5.2: Testování doby odezvy

### 5.5.1.2 Výjimky a logování

Podobně jako testování, ani zpracování výjimek a logování není v zadání přímo zmíněno, a proto jsem mu věnoval méně pozornosti. I tak je ale v balíčků *app/Exceptions* implementován Handler, který odchyťává všechny výjimky, které v aplikaci vzniknou. Pro přehlednější a snadnější hledání chyb je potřeba před nasazením do produkce obohatit aplikaci o vlastní druhy výjimek.

Výjimky jsou momentálně logovány do souboru *storage/logs/Laravel.log*. S rostoucí hráčskou základnou bude potřeba log soubory dále dělit, aby nevznikaly příliš dlouhé soubory, které bude složité procházet. Toto nastavení lze změnit v konfiguračním souboru *config/app.php*. Aplikace momentálně ne-loguje nic jiného než výjimky. Bylo by vhodné logovat snahy uživatelů o obejítí bezpečnostních opatření, například snahy o manipulaci http požadavků za účelem odjždění závodů s cizí lodí a podobné.

### 5.5.1.3 Grafika

Největší překážkou nasazení vytvořeného prototypu do produkce je slabá grafika. Přestože grafika ve hrách podobného typu není základním předpokladem úspěchu, je třeba některé grafické prvky do aplikace začlenit. V aplikaci je momentálně velké množství podobně vypadajících tabulek a prvků, které často mají podobný význam a mohou být zaměnitelné. Každému takovému prvku je třeba dodat charakteristické obrázky, které ho na první pohled odliší, aniž by musel uživatel číst obsah.

Hra nyní rozlišuje jednotlivé druhy komponent formou ikony. Bylo by vhodné jít ještě dál a dát podobu každému jednotlivému dílu. Dá se říci, že to komponentám vdechne trochu osobnosti a umožní to uživateli si nevědomky oblibovat některé komponenty. Vesmírné lodě by měly dostat podobu obrázkové šablony, na které pak budou vidět konkrétní namontované díly.

Větší divergenci by zasloužily i jednotlivé stránky. Napoví to uživateli, kde se nachází. Prostor pro tyto obrázky se nachází buď v hlavičce stránky nebo po pravé straně, která je na většině stránek záměrně nevyužitá. Představuji si, že když bude uživatel procházet skladiště dílů, odpovídající obrázek ho utvrdí v tom, že je na správném místě. V plánu je také zúžit prostor, který stránka využívá. Pomůže to lepší korespondenci s mobilním zobrazením a na klasické obrazovce to umožní soustředit pohled ke středu, což pomůže zrychlit hledání informací.

#### 5.5.1.4 Herní balanc

Herní balanc je trvalý proces. Před uvedením do ostrého provozu je třeba pokusit se přiblížit optimálním hodnotám herních konstant tak, aby žádná strategie nebyla evidentně zvýhodněná. Bude pro to potřeba rozsáhlé testování. Balanc bude dále probíhat v době, kdy umožníme přístup do hry uzavřené skupině hráčů a později i všem zájemcům.

Jednou z technik balancování herního stavu je např. využití k balancování rozhodnutí samotných hráčů. Touto metodou by například šlo zjistit optimální ceny předmětů na trhu v závislosti na penězích, kterými hráč obvykle disponuje. Stačilo by pro jednotlivé komponenty zavést veřejnou aukci. Reálná hodnota dílu by pak byla blízko průměru nabízených částek a tuto hodnotu bychom následně použili na náš uzavřený systémový trh.

#### 5.5.2 Rozšíření o herní funkce

V následujícím seznamu naleznete funkce, které bych rád do hry přidal. Tyto funkce není nezbytně nutné implementovat před nasazením do produkce, ale významně rozšiřují herní možnosti a bylo by dobré je do hry dostat co nejdříve.

- vizualizace průběhu závodů - V současné podobě se uživatel dozví pouze výsledek závodu. Prvním rozšířením by mělo být podání zprávy ze závodu, ideálně ještě rozšířeno o vybrané grafy. Zajímavá by měla být například rychlost v závislosti na čase, udělené škody závodníků atd.
- vlastnosti pilota - Uživatelé by měli možnost své postavě vštípit určité vlastnosti. Tyto vlastnosti by pak ovlivňovaly například efektivitu v závodech, při smlouvání cen a podobně.
- souboje - Jsou implementovány agresivní závody vesmírných lodí. Vše je tedy připraveno, aby si piloti vyřizovali účty i mimo závody a to formou soubojů.

## 5. REALIZACE

---

- **přídavné systémy** - Máme 6 základních povinných komponent lodi. V lodi by mohly být zavedeny další sloty pro volitelné předměty. Prostor pro konkrétní funkčnosti přídavných systémů je obrovský. Od zpomalovacích zbraní, přes miny udělující plošné škody hned několika lodím najednou až po elektromagnetické pole zneschopňující soupeře.
- **přebrání zodpovědnosti za akce vykonávané systémem** - Uživatel by získal možnost zakládat závody, vytvářet tratě, vypisovat sázkové kurzy na jím vytvořené závody, nabízet díly ke koupi ostatním hráčům a další. Cílem těchto změn by bylo přiblížit se simulaci „reálného“ světa.
- **zkušební systém a profilace** - Uživatel by vykonáváním akcí získával zkušenosti v daném odvětví a postupně by si mohl vybírat bonusy spojené s jeho expertízou. Někdo by si více zakládal vyhrávání závodů, někdo by raději cílil na souboje a podobně.



---

## Závěr

Hlavním cílem práce je navrhnout a naprogramovat prototyp webové online hry a respektovat při tom zásady správného vývoje software, přičemž důraz je kladen na rozšiřitelnost a funkčnost tohoto základu. Hra má být ve výsledné podobě nasazena na veřejný webový server, přičemž na pozadí mají periodicky probíhat přepočty herního stavu. Návrh a realizace se mají opírat o důslednou analýzu z oblasti her podobného typu. Povinností je i vyjádřit se k roli finančních transakcí v dnešních hrách.

Všechny tyto cíle byly splněny. Prototyp hry byl zdárně umístěn na webový server a je přístupný na adrese <http://swinpro.fit.cvut.cz/>, přičemž pravidelně probíhá herní přepočet. Architektura byla navržena s ohledem na další rozšiřitelnost, což podpoří můj úmysl na projektu dále pracovat i nad rámec této práce.

V samostatné kapitole byla zmíněna plánovaná funkční rozšíření a také oblasti, které je třeba vyřešit předtím, než by byla hra nasazena do ostrého provozu. Za nejnnutnější považuji implementaci automatizovaných testů, které současnému prototypu chybí. Důležité bude vybalancovat herní funkčnosti, aby žádná strategie neposkytovala hráči nepřiměřenou výhodu. Hra by také měla dostat působivější vzhled.

Během implementace nedošlo k problémům, které by byly způsobeny špatným návrhem. Tato bakalářská práce pro mě byla zároveň první příležitostí si vyzkoušet jazyk PHP v praxi. Musím říci, že i díky výbornému frameworku Laravel si odnáším z implementace cenné zkušenosti, a hlavně i nadšení pro další pokračování projektu.



---

## Dokumentace databázového modelu

V této příloze uvádím dokumentaci databázových tabulek a jejich atributů. Upozorňuji, že z této dokumentace byly vyňaty tabulky reprezentující jednotlivé druhy součástek vesmírné lodi, tj. tabulky Cannons, Engines, Hulls, Shields, Sources a Thrusters, protože už byl jejich význam vysvětlen v textu. 4.3.1 Vynechávám také popis atributů, které jsou primárním klíčem, neboť jejich význam je zjevný.

**Bets** Sázka na výhru lodi v závodě.

- *id*
- *amount* - vsazená částka
- *success* - Příznak, zda byla sázka úspěšná.
- *owner\_id* - ID uživatele, který sázku vložil.
- *participation\_id* - Určuje danou účast v závodě.

**Components** Sdružuje společné vlastnosti všech druhů součástek lodi.

- *id*
- *price* - cena
- *consumption* - spotřeba energie
- *hitpoints\_max* - Počet škod, které součástka vydrží při stoprocentním stavu.
- *name* - jméno součástky

- *tier* - Silová skupina, která určuje minimální a maximální hodnoty při generování ostatních parametrů.  $1 \leq tier \leq 3$

**Component\_Material** Dekompoziční tabulka vztahu M:N mezi komponenty a materiály.

- *id*
- *quantity* - Množství materiálu, ze kterých se součástka dá vyrobit v nspecifikovaných jednotkách.
- *component\_id* - Určuje součástku.
- *material\_id* - Určuje materiál.

**Materials** Materiály na výrobu dílů.

- *id*
- *name* - jméno materiálů
- *cost* - cena za jednotku

**Material\_User** Dekompoziční tabulka vztahu M:N mezi materiály a uživateli.

- *id*
- *quantity* - Množství vlastněného materiálu v nspecifikovaných jednotkách.
- *material\_id* - Určuje daný materiál.
- *user\_id* - Určuje vlastníka.
- 

**Messages** Zprávy mezi uživateli nebo na fórum.

- *id*
- *body* - obsah zprávy
- *created\_at* - datum vzniku
- *updated\_at* - datum poslední editace
- *author\_id* - ID uživatele, který zprávu poslal.
- *recipient\_id* ID uživatele, který zprávu přijal. Je-li *NULL*, jedná se o zprávu na veřejné fórum.

---

**Participations** Dekompoziční tabulka vztahu M:N mezi vesmírnými loděmi uživateli a závody. Vyjadřuje účast.

- *id*
- *result* - umístění v závodě
- *bet\_rate* - výše sázkového kurzu na výhru
- *aggression* - agresivita ovlivňující míru ignorování vlastního poškození za účelem pokračující střelby po ostatních
- *vigilance* - ostražitost ovlivňující úspěšné vyhnutí střele a rychlost v závodě.
- *risk* - míra ignorování doporučené rychlosti
- *race\_id* - Určuje závod, na který se účast vztahuje.
- *spaceship\_id* - Určuje vesmírnou loď, která poletí v závodě.
- *component\_spread*, *target*, *boosted\_component* - Různá nastavení stylu odjetí závodu, která prozatím nejsou ve hře využita.

**Possessions** Dekompoziční tabulka vztahu M:N mezi uživateli a součástkami. Vyjadřuje vlastnictví dané součástky.

- *id*
- *hitpoints* - Vyjadřuje stav poškození dané součástky.
- *user\_id* - Určuje vlastníka.
- *component\_id* - Určuje konkrétní součástku.

**Possession\_Spaceship** Dekompoziční tabulka vztahu M:N mezi vesmírnými loděmi uživateli a vlastněnými součástkami. Vyjadřuje namontování součástky do lodi.

- *id*
- *spaceship\_id* - Určuje danou vesmírnou loď.
- *possession\_id* - Určuje vlastněnou součástku.

**Races** Vypsané závody vesmírných lodí.

- *id*
- *name* - jméno závodu
- *scheduled\_on* - datum a čas odjetí
- *capacity* - maximální počet účastníků
- *prize* - Odměna za výhru závodu, která ovlivňuje i odměnu umístění na ostatních příčkách.
- *track\_id* - Určuje trať, na které se závod odehraje.
- *user\_id* - Určuje autora závodu. Prozatím závody generuje jen systém.

**Sectors** V galaxii se vyskytují sektory, do kterých se létá podnikat mise.

- *id*
- *name* - jméno
- *riches* - Ovlivňuje množství získaných zdrojů během mise.
- *encounter* - Ovlivňuje šanci na střet během mise.
- *level, enemy\_tier* Prozatím nejsou hrou využity.

**Schematics** Dekompoziční tabulka vztahu M:N mezi uživateli a součástkami. Vyjadřuje schopnost uživatele danou součástku vyrobit.

- *id*
- *crafted\_times* - Počet kusů této součástky, které už uživatel vyrobil. Snižuje počet minut potřebných pro výrobu dalšího kusu.
- *user\_id* - Určuje uživatele.
- *component\_id* - Určuje danou součástku.

**Spaceships** Představuje vesmírnou loď - konfiguraci. Slouží pro shluk několika součástí do jednoho celku.

- *id*
- *name* - Jméno lodi, tak jak si ho určil uživatel.
- *user\_id* - Určuje vlastníka.

---

**Stages** Reprezentuje úsek trati.

- *id*
- *length* - délka v kilometrech
- *name* - jméno úseku
- *danger* - Tato hodnota určuje nebezpečí úseku, které ovlivňuje bezpečnou rychlost na prolétnutí.

**Stage\_Track** Dekompoziční tabulka vztahu M:N mezi úseky a tratěmi. Vyjadřuje výskyt úseku v trati.

- *id*
- *placement* - Určuje, kde na trati se úsek vyskytuje. Úseky s nižším číslem jsou blíže ke startu.
- *stage\_id* - Identifikuje úsek.
- *track\_id* - Identifikuje danou trať.

**Tracks** Tratě, na kterých se odehrávají závody.

- *id*
- *name* jméno trati
- *user\_id* - Vyjadřuje autora trati. Zatím všechny trati jsou systémové.
- *sector\_id* - V jakém sektoru se trať nachází. Prozatím nemá žádný význam.

**Users** Uživatel aplikace - pilot.

- *id*
- *created\_at* - datum a čas vytvoření účtu
- *email* - email použitý při registraci
- *name* - zaregistrované jméno
- *password* - heslo uložené v zahashované podobě
- *rememberToken* - Slouží k tomu, aby se uživatel nemusel každých pár minut přihlašovat v případě ukončení okna prohlížeče. Prozatím nevyužito.

## A. DOKUMENTACE DATABÁZOVÉHO MODELU

---

- *balance* - Vyjadřuje stav účtu, tedy množství hlavní finanční měny.
- *is\_npc* - Vyjadřuje, zda je uživatel systémový (NPC) nebo reálný.
- *minutes* - Určuje množství časové měny, tedy minut, které má uživatel k dispozici.
- *pause\_job* - Prozatím nevyužito.
- *sector\_id* - Vyjadřuje, v jakém sektoru se uživatel právě nachází.



---

## Seznam použitých zkratk

V tomto seznamu se záměrně odchýlím od pravidel českého a anglického jazyka pro určování velikosti písmen v názvech. Velká písmena pro ilustrativní účely ponechám na začátku slov, která zkratka vyjadřuje.

**CLI** Command-Line Interface

**CPU** Central Processing Unit

**CSRF** Cross-Site Request Forgery

**CSS** Cascading Style Sheets

**FURPS** Functionality, Usability, Reliability, Performance, Supportability

**HTML** HyperText Markup Language

**JS** JavaScript

**JSON** JavaScript Object Notation

**JWT** JSON Web Token

**OS** Operační Systém

**RAM** Random-Access Memory

**REST** Representational State Transfer

**RPG** Role-Playing Game

**SŘBD** Systém Řízení Bází Dat

**UML** Unified Modeling Language

**URL** Uniform Resource Locator

**XML** eXtensible Markup Language



## Obsah přiloženého CD

- **app\_source** - zdrojové soubory aplikace
- **text\_source** - zdrojové soubory textové části
- **BP\_Košvanec\_Petr\_2017.pdf** - text bakalářské práce ve formě PDF



---

## Literatura

- [1] The Global Games Market Reaches \$99.6 Billion in 2016, Mobile Generating %37. *Newzoo [online]*, duben 2016, [cit. 2016-11-30]. Dostupné z: <https://newzoo.com/insights/articles/global-games-market-reaches-99-6-billion-2016-mobile-generating-37/>
- [2] What Is A Strategy Game? *All About Strategy Games*, duben 2010, [cit. 2016-11-30]. Dostupné z: <http://strategygames.biz/category/what-is-a-strategy-game/>
- [3] Turn-Based VS Real-Time. *Gamasutra: Jon Shafers blog*, červenec 2013, [cit. 2017-05-07]. Dostupné z: [http://www.gamasutra.com/blogs/JonShafer/20130107/184429/TurnBased\\_VS\\_RealTime.php](http://www.gamasutra.com/blogs/JonShafer/20130107/184429/TurnBased_VS_RealTime.php)
- [4] What makes an RPG an RPG: a universal definition. březem 2011, [cit. 2017-05-07]. Dostupné z: <http://sinisterdesign.net/what-makes-an-rpg-an-rpg-a-universal-definition/>
- [5] What is Browser-Based Game? *Computer hope*, duben 2017, [cit. 2017-05-12]. Dostupné z: <https://www.computerhope.com/jargon/b/browserbased-game.htm>
- [6] Firefox 52.0 System Requirements. 2017, [cit. 2017-05-11]. Dostupné z: <https://www.mozilla.org/en-US/firefox/52.0/system-requirements/>
- [7] Overwatch system requirements | Can | Run Overwatch. 2017, [cit. 2017-05-11]. Dostupné z: <https://www.systemrequirementslab.com/cyri/requirements/overwatch/12955>
- [8] Travian Games GmbH: *Vítejte na travianstats.de*. [cit. 2016-11-30]. Dostupné z: <http://travianstats.de/>

- [9] What is gameplay? Webopedia definition. *Webopedia*, [cit. 2017-05-12]. Dostupné z: <http://www.webopedia.com/TERM/G/gameplay.html>
- [10] DOTA 2 prize pool tracker. duben 2017, [cit. 2017-05-07]. Dostupné z: <http://dota2.prizetrac.kr/>
- [11] Overwatch Wins Game of the Year at 2016 Game Awards; All the Winners Revealed. *Gamespot*, prosinec 2016, [cit. 2017-05-07]. Dostupné z: <https://www.gamespot.com/articles/overwatch-wins-game-of-the-year-at-2016-game-award/1100-6445894/>
- [12] World of Warcraft: Legion. *Gamespot*, září 2016, [cit. 2017-05-07]. Dostupné z: <http://www.hrej.cz/recenze/world-of-warcraft-legion-5509/>
- [13] What's a micro transaction and how do developers use them? *Superrewards*, květen 2017, [cit. 2017-05-07]. Dostupné z: <https://www.superrewards.com/micro-transaction>
- [14] Hearthstone Generates \$20 Million in Revenue Per Month. *Gamerant*, květen 2015, [cit. 2017-05-07]. Dostupné z: <https://gamerant.com/heartstone-profit-monthly-900/>
- [15] What is web server? Definition from whatis.com. červenec 2015, [cit. 2017-05-08]. Dostupné z: <http://whatis.techtarget.com/definition/Web-server>
- [16] Usage of web servers for websites. květen 2017, [cit. 2017-05-08]. Dostupné z: [https://w3techs.com/technologies/overview/web\\_server/all](https://w3techs.com/technologies/overview/web_server/all)
- [17] NGINX vs Apache: Advantages and Disadvantages. listopad 2016, [cit. 2017-05-08]. Dostupné z: <https://www.1and1.com/cloud-community/learn/web-server/nginx/nginx-vs-apache-advantages-and-disadvantages/>
- [18] PHP frameworky. *programujte.com*, únor 2008, [cit. 2017-05-08]. Dostupné z: <http://programujte.com/clanek/2008022000-php-frameworky/>
- [19] 5 Best JavaScript Frameworks in 2017. leden 2017, [cit. 2017-05-08]. Dostupné z: <https://hackernoon.com/5-best-javascript-frameworks-in-2017-7a63b3870282>
- [20] Angular - Superheroic Javascript MVW framework. květen 2017, [cit. 2017-05-08]. Dostupné z: <https://angularjs.org/>

- 
- [21] Advantages and disadvantages of AngularJS. *Software developer India*, říjen 2016, [cit. 2017-05-08]. Dostupné z: <http://www.software-developer-india.com/advantages-and-disadvantages-of-angularjs/>
- [22] ReactJS For Stupid People. *Andrew Rays blog*, září 2014, [cit. 2017-05-08]. Dostupné z: <http://blog.andrewray.me/reactjs-for-stupid-people/>
- [23] JQuery: advantages and disadvantages. *JScripters.com*, [cit. 2017-05-08]. Dostupné z: <http://www.jscripters.com/jquery-disadvantages-and-advantages//>
- [24] Top 5 Programming Languages Used In Web Development. *Stone river eLearning*, listopad 2015, [cit. 2017-05-09]. Dostupné z: <http://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development/>
- [25] Difference between Java and PHP. 2014, [cit. 2017-05-09]. Dostupné z: <http://www.differencebetween.info/difference-between-java-and-php>
- [26] PHP vs. Java. [cit. 2017-05-09]. Dostupné z: <https://www.cs.montana.edu/~tosun/phpvsjava.pdf>
- [27] What is database management system? *WhatIs.com*, leden 2015, [cit. 2017-05-11]. Dostupné z: <http://searchsqlserver.techtarget.com/definition/database-management-system>
- [28] SQLite vs MySQL vs PostgreSQL: A Comparison Of Relational Database Management Systems. *DigitalOcean*, únor 2014, [cit. 2017-05-11]. Dostupné z: <https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems>
- [29] Level 1: Intro to Game Balance. *Game balance concepts*, červenec 2010, [cit. 2017-05-07]. Dostupné z: <https://gamebalanceconcepts.wordpress.com/2010/07/07/level-1-intro-to-game-balance/>
- [30] Directory Structure. *Laravel - The PHP framework for web artisans*, 2017, [cit. 2017-05-09]. Dostupné z: <https://laravel.com/docs/5.4/structure>
- [31] AngularJS Best Practices: Directory Structure. *Scotch*, říjen 2014, [cit. 2017-05-09]. Dostupné z: <https://scotch.io/tutorials/angularjs-best-practices-directory-structure>

## LITERATURA

---

- [32] Artisan console. *Laravel - The PHP framework for web artisans*, 2017, [cit. 2017-05-12]. Dostupné z: <https://laravel.com/docs/5.4/artisan>
- [33] Unit Testing. *AngularJS: Developer Guide*, květen 2017, [cit. 2017-05-10]. Dostupné z: <https://docs.angularjs.org/guide/unit-testing>