

**Česká zemědělská univerzita v Praze**

**Provozně ekonomická fakulta**

**Katedra informačního inženýrství**



**Diplomová práce**

**Vývoj herní manažerské aplikace**

**Bc. Nastassia Paulavets**

**© 2021 ČZU v Praze**

# ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE

Provozně ekonomická fakulta

## ZADÁNÍ DIPLOMOVÉ PRÁCE

Bc. Nastassia Paulavets

Systémové inženýrství a informatika  
Informatika

Název práce

**Vývoj herní manažerské aplikace**

Název anglicky

**Development of game management application**

---

### Cíle práce

Cílem práce je navrhnout, implementovat a otestovat webovou manažerskou marketingovou hru v prostředí jazyka Python. Hra je určena pro více osob a pro moderátora tak, aby se mohla hrát na učebnách PEF.

### Metodika

V první části práce bude přehled použité teorie, nástrojů a technik, v druhé části práce bude vlastní projekt. Prodávaná a nakupovaná služba bude pro jednoduchost jen jednoho druhu a za jednotkovou cenu. Hra bude mít konfigurovatelnou výplatní matici a také možnost nakupovat a nabízet různé množství. Na programovou realizaci budou použity JavaScript, Python, HTML, CSS a standardy softwarového inženýrství, především UML.

**Doporučený rozsah práce**

60 – 100 stran

**Klíčová slova**

Aplikace, Python, hra, management, Piškvorky.

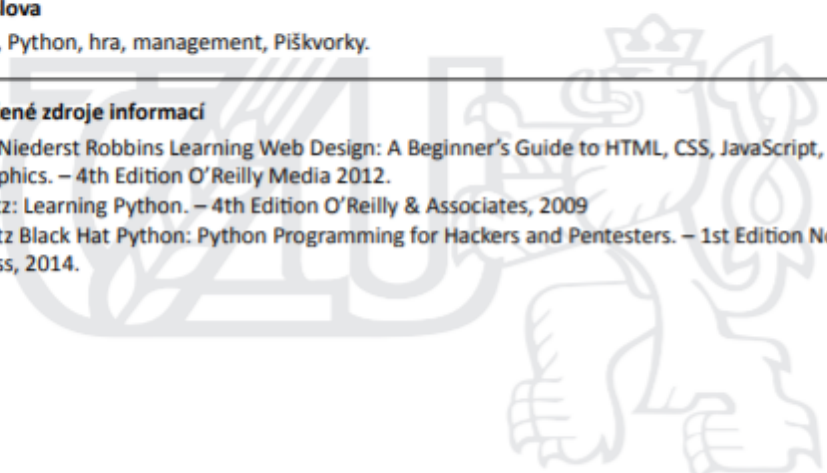
---

**Doporučené zdroje informací**

Jennifer Niederst Robbins Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. – 4th Edition O'Reilly Media 2012.

Mark Lutz: Learning Python. – 4th Edition O'Reilly & Associates, 2009

ustin Seitz Black Hat Python: Python Programming for Hackers and Pentesters. – 1st Edition No Starch Press, 2014.



---

**Předběžný termín obhajoby**

2019/20 LS – PEF

**Vedoucí práce**

doc. Ing. Vojtěch Merunka, Ph.D.

**Garantující pracoviště**

Katedra informačního inženýrství

---

Elektronicky schváleno dne 11. 3. 2020

**Ing. Martin Pelikán, Ph.D.**

Vedoucí katedry

---

Elektronicky schváleno dne 11. 3. 2020

**Ing. Martin Pelikán, Ph.D.**

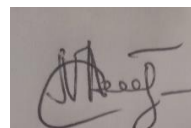
Děkan

V Praze dne 26. 03. 2021

## Čestné prohlášení

Prohlašuji, že svou diplomovou práci "Vývoj herní manažerské aplikace" jsem vypracovala samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu použitých zdrojů na konci práce. Jako autorka uvedené diplomové práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 31.3.2021

A square image containing a handwritten signature in black ink on a light-colored background. The signature is stylized and appears to be 'M. Hájek'.

## **Poděkování**

Ráda bych touto cestou poděkovala doc. Ing. Vojtěchu Merunkovi, Ph.D. za velmi užitečnou metodickou pomoc a cenné rady při zpracování diplomové práce

# Vývoj herní manažerské aplikace

## Abstrakt

Diplomová práce se zabývá návrhem nové manažerské hry. V teoretické části jsou přehled použité teorie, nástrojů a technik, v praktické části práce je vlastní projekt. Prodávaná a nakupovaná služba jsou pro jednoduchost jen jednoho druhu a za jednotkovou cenu. Hra má konfigurovatelnou výplatní matici a také možnost nakupovat a nabízet různé množství. V praktické části se práce věnuje samotnému návrhu. Jsou v ní specifikovány cíle a požadavky na softwarovou aplikaci i ekonomický model hry a následně je popsán způsob jejich realizace. Návrh hry se soustředí zejména na technické zpracování hry, na specifikování konkrétních algoritmů, kterými se model hry řídí, a na implikace vztahů mezi proměnnými v modelu.

**Klíčová slova:** Aplikace, Python, hra, manažerská, hráč, HonestyFraud, Django, jQuery, Bootstrap, prodej, nákup, podvod, CSS.

# Development of game management application

## Abstract

The thesis deals with the design of a new management game. In the theoretical part there is an overview of the theory, tools and techniques used, in the practical part of the work there is a project of its own. The sold and purchased service is for simplicity of only one kind and at a unit price. The game has a configurable payment matrix, as well as the ability to buy and offer different amounts. In the practical part, the work is devoted to the design itself. It specifies the goals and requirements for the software application as well as the economic model of the game, and then describes how to implement them. The design of the game focuses mainly on the technical processing of the game, on the specification of the specific algorithms that govern the game model, and on the implications of the relationships between the variables in the model.

**Keywords:** Application, Python, game, management, player, HonestyFraud, Django, jQuery, Bootstrap, sale, purchase, scam, CSS.

# Obsah

<b>1 Úvod .....</b>	<b>10</b>
<b>2 Cíl práce a metodika .....</b>	<b>11</b>
2.1 Cíl práce .....	11
2.2 Metodika.....	11
<b>3 Teoretická východiska .....</b>	<b>12</b>
3.1 Přístupy pro vývoj aplikace .....	12
3.2 Etapy vývoje aplikace .....	13
<b>4 Vlastní práce Aplikaci HonestyFraud.....</b>	<b>30</b>
4.1 Popis technické zadání pro vývoj herní aplikace.....	30
4.2 Vývoj aplikace .....	31
4.3 Návod k instalaci hry .....	37
<b>5 Výsledky a diskuse .....</b>	<b>51</b>
<b>6 Závěr .....</b>	<b>57</b>
<b>7 Seznam použitých zdrojů .....</b>	<b>58</b>

## Seznam obrázků

Obrázek 1 – Datový model
Obrázek 2 – Bootstrap
Obrázek 3 - Struktura Bootstrapu
Obrázek 4 – Příklad Bootstrapu
Obrázek 5 – Příklad 1 jQuery
Obrázek 6 – Příklad 2 jQuery
Obrázek 7 – Příklad 3 jQuery
Obrázek 8 – Příklad 4 jQuery
Obrázek 9 – Příklad 5 jQuery
Obrázek 10 – Příklad 6 jQuery
Obrázek 11 – Struktura celé databáze
Obrázek 12 – Popis tabulek
Obrázek 13 – Graf UML
Obrázek 14 – Krok 2
Obrázek 15 – Krok 3
Obrázek 16 – Krok 5
Obrázek 17 – Krok 6
Obrázek 18 – Krok 7
Obrázek 19 – Krok 8
Obrázek 20 – Krok 9
Obrázek 21 – Krok 10
Obrázek 22 – Krok 11
Obrázek 23 – Krok 12
Obrázek 24 – Krok 13



Obrázek 25 – Krok 14  
Obrázek 26 – Krok 15  
Obrázek 27 – Krok 16  
Obrázek 28 – Krok 17  
Obrázek 29 – Krok 18  
Obrázek 30 – Krok 19  
Obrázek 31 – Krok 20  
Obrázek 32 – Krok 21  
Obrázek 33 – Krok 22  
Obrázek 34 – Krok 23  
Obrázek 35 – Administrátor  
Obrázek 36 – Chat  
Obrázek 37 – Hra mezi hráči  
Obrázek 38 – Obchod  
Obrázek 39 – Transakce  
Obrázek 40 – Turnaj  
Obrázek 41 – Body  
Obrázek 42 – Transakce a body  
Obrázek 43 – Co vidí hráči

### **Seznam tabulek**

Tabulka – Roydělení bodů po dokončení transakce

# 1 Úvod

Hra je napsána na základě teorie her. Teorie her je matematická metoda učení optimálních strategií ve hrách. Hra je chápána jako proces, který zahrnuje dvě a více stran, které vedou boj za realizaci svých zájmů. Každá ze stran má svůj cíl a používá nějakou strategii, která může vést k vítězství nebo ztrátě — v závislosti na chování ostatních hráčů. Teorie her pomáhá vybrat nejlepší strategie s přihlédnutím k představám o ostatních účastnících, jejich zdrojích a jejich možných činnostech

V teorii her jsou hry formálně definované pojmy. Hra obsahuje hráče, jejich možné tahy (nebo akce nebo strategie) a funkci, odkazující na zisk každého hráče v závislosti na provedených tahů. V literatuře hry jsou psány jedním ze dvou následujících způsobů.

Normální forma hry je většinou reprezentována maticí, která zobrazuje hráče, jejich možné strategie a možné výhry (viz tabulka vpravo). Obecněji může být reprezentována funkcí, která přiřadí zisk každému hráči na základě dané kombinace pohybů. V příkladu vpravo jsou dva hráči. Úkolem prvního hráče je vybrat řádek, úkolem druhého je vybrat sloupec. Každý hráč má dvě možnosti. Zisk je zaznamenán uvnitř matice, první číslo určuje zisk pro Hráče 1, druhé určuje zisk pro hráče 2. Takže pokud první hráč zvolí A a druhý X, zisk prvního hráče je 4 a zisk druhého hráče je 3.

U her v normální formě se předpokládá, že si hráči zvolí pohyby současně, nebo alespoň nevědí, jaký tah si soupeř zvolil. Pokud mohou hráči znát pohyby soupeře, hra je prezentována převážně v rozsáhlé formě.

Rozsáhlá forma hry se používá k formalizaci her, ve kterých hraje roli pořadí pohybů. Hry jsou prezentovány ve formě stromů (viz obrázek vlevo). Každý uzel zde představuje místo, kde si některý z hráčů zvolí pohyb. Hráč, který vybere tah, je určen číslem napsaným vedle uzlu. Žebra představují možné pohyby hráče. Zisk pro jednotlivé hráče je uveden v listu stromu.

Ve hře na obrázku jsou dva hráči. Hráč 1 vybírá první a má na výběr buď f, nebo u. Hráč 2 vidí tah Hráče 1 a poté vybere buď A, nebo R. předpokládejme, že hráč 1 vybere u a hráč 2 vybere A. Potom zisk prvního hráče je roven 8 a zisk druhého hráče je 2.

Rozsáhlá forma může také ukázat situaci, kdy si hráči vybírají pohyby současně, stejně jako hry s neúplnými informacemi. Pokud hráč neví, ve kterém z několika států se nachází, kolem těchto stavů je nakreslen kruh.

## **2 Cíl práce a metodika**

### **2.1 Cíl práce**

Cílem práce je navrhnout, implementovat a otestovat webovou manažerskou marketingovou hru v prostředí jazyka Python. Hra je určena pro více osob a pro moderátora tak, aby se mohla hrát na učebnách PEF.

### **2.2 Metodika**

V první části práce bude přehled použité teorie, nástrojů a technik, v druhé části práce bude vlastní projekt. Prodávaná a nakupovaná služba bude pro jednoduchost jen jednoho druhu a za jednotkovou cenu. Hra bude mít konfigurovatelnou výplatní matici a také možnost nakupovat a nabízet různé množství. Na programovou realizaci budou použity JavaScript, Python, HTML, CSS a standardy softwarového inženýství, především UML.

## 3 Teoretická východiska

### 3.1 Přístupy pro vývoj aplikace

Vývoj je podrobný projekt s výzkumem a realizací: cíl, úkoly, koncepce, komunikace, Architektura, hodnocení zdrojů.

**Rozlišujte následující přístupy:**

1. Nativní (platformově závislý) přístup. Vývoj aplikace v "rodném" jazyce pro každou platformu. Například, tak jsou vyvinuty ty aplikace, které jdou "šité" do dispension-Galerie, Fotoaparát, poznámky, zprávy.
2. Multiplatformní přístup. Aplikace je vyvíjena pouze jednou, předpokládá se nasazení na různých platformách.
3. Hybridní přístup. Je kombinací vlastností nativního a multiplatformního vývoje. Pokud jde o řešení určitých úkolů, efektivním řešením bude kombinace dvou předchozích přístupů.

Na základě výše uvedeného může dojít k názoru, že multiplatformní aplikace bude úspěšná na všech platformách, včetně těch nepopulárních. K dosažení požadovaných cílů však bude zapotřebí dodatečný kód podpory různých platform, jak je popisováno v (Darie 2005) . Přístup založený na platformě je široce rozšířený, ale přichází s řadou obtíží při možném přenosu Softwarového produktu na alternativní platformy. V tomto případě existují standardy a doporučení, pravidla a pokyny pro použití značkového stylu, které pomáhají dát prvky rozhraní "obvyklé" umístění a vzhled. Jazykové prostředí, ve kterém je vyvíjena aplikace závislá na platformě, má obvykle potřebné nástroje pro implementaci známého rozhraní. Naproti tomu proces vytváření multiplatformní aplikace pomocí webových technologií může vyžadovat značné náklady na zdroje. Řešení tohoto problému pomáhá tzv. rámce-sady knihoven a hotových softwarových modulů, které umožňují rychlejší a snadnější vytváření struktury aplikace. Různé multiplatformní (frameworky umožňují implementovat rozhraní cílové platformy, nicméně výkon při použití tohoto přístupu bude nižší, než při přímém použití платформенно-dependentní mechanismy.

Aplikace vyvinutá pro konkrétní softwarovou platformu má obvykle přímý přístup k hlavním možnostem cílového zařízení a jeho softwarových služeb. Při navrhování aplikace s více platformami je potenciál projektu omezen na možnosti použitého rámce. Často multiplatformní framework nemá schopnost využívat všechny funkce každé z podporovaných platform a poskytuje softwarové rozhraní pro nejběžnější funkce. Taková sada příležitostí je pro většinu projektů dostatečná.

**Výhody platformově závislého přístupu lze přičíst:**

1. Vysoký výkon aplikace vzhledem k nedostatku intermediálních multiplatformní rozhraní.

2. Obvykle je k dispozici širší škála možností zařízení.
3. Úzká integrace s uživatelským rozhraním cílové platformy.

Hlavní nevýhodou přístupu závislého na platformě je poměrně vysoká namáhavost přenosu softwaru na alternativní platformy.

**Multiplatformní přístup má následující výhody:**

1. Nízká náročnost přenosu softwaru na nové a perspektivní platformy.
2. Jednoduchost doprovodu Softwarového produktu vzhledem k použití jednotných přístupů platí pro různé platformy.

**Nevýhody multiplatformního přístupu zahrnují:**

1. Poměrně vysoká náročnost vývoje vzhledem k potřebě zajistit provozuschopnost na každé z podporovaných platform.
2. Snížení výkonu konečného produktu kvůli použití intermediálních programových rozhraní platformy abstrakce.

V důsledku analýzy různých přístupů k vývoji pro implementaci klient-server herní aplikace byl vybrán multiplatformní přístup.

### **3.2 Etapy vývoje aplikace**

Navrhování uživatelského rozhraní je důležitou součástí celého projektu. Ve fázi návrhu je snadnější formulovat požadavky a omezení na konečný softwarový produkt. Identifikace chyb ve fázi návrhu snižuje náklady na testování a ladění aplikace. Ve fázi návrhu budeme schopni zjistit potřeby cílového publika, co je nejdůležitější a co ne. Budeme schopni analyzovat a navrhnout chování uživatelů na webu. Díky návrhu je možné s nejlepším efektem sdílet práci na internetovém zdroji, jak je popisováno v (Heilmann 2007).

**Proces práce na projektu se skládá z následujících fází:**

1. Stanovení cílů a cílů projektu.
2. Vývoj designu.
3. Sazba.
4. Programování.
5. Testování.

## 6. Běh a doprovod.

Herní aplikace implementuje způsob interakce uživatelů na herní úrovni, ukazuje úroveň jejich logického myšlení.

**Na základě výsledků analýzy přístupů k rozvoji proces realizace tohoto projektu byl rozdělen do následujících fází:**

### 1. *Počáteční fáze.*

- 1.1. Určení cílů aplikace.
- 1.2. Definice cílové publikum.

### 2. *Upřesnění.*

- 2.1. Formulace požadavků na funkčnost Softwarového produktu.
- 2.2. Výběr programovacího jazyka a nástrojů pro vývoj serveru.

### 3. *Vybudování.*

- 3.1. Systém kontroly verzí.
- 3.2. Vývoj návrhu klientských komponent a navrhování rozhraní klient-server interakce.
- 3.3. Implementace serverových komponent a rozhraní klient-server interakce. Knihovny třetích stran.

### 4. *Zavedení.*

- 4.1. Testování.
- 4.2. Aprobace metodou znaleckého posudku.

Účel aplikace. V této fázi je nutné co nejpřesněji určit cíle vytvořeného projektu.

Cílové publikum. V této fázi je určena cílová skupina vytvořené aplikace. To bude přímo záviset na potřebné funkčnosti a designu projektu.

Funkční projekt. Jsou definovány možnosti vytvořeného projektu, základní funkce. Výběr programovacího jazyka pro webový server. Výběr jazyka přímo závisí na zadaných úkolech.

PHP je programovací jazyk spustitelný na straně webového serveru. Dostatečně flexibilní a výkonný, což vysvětluje velkou popularitu a využití v projektech jakéhokoli rozsahu, jak je popisováno v (Ustin 2014).

### **Výhody:**

1. Volný software (PHP Licence).
2. Jednoduché zvládnutí ve všech fázích.
3. Podpora od velké komunity uživatelů a vývojářů.

4. Rozvinutá podpora databází.
5. Dostupnost obrovského množství knihoven a rozšíření jazyka.
6. Možnost použití v izolovaném prostředí.
7. Sweep téměř na jakémkoli serveru.
8. Portován do mnoha hardwarových platforem a OS.

#### **Nedostatky:**

1. Není vhodný pro vytváření desktopových aplikací nebo součástí systému.
2. Mít slabé prostředky pro práci s výjimkami.
3. Globální možnosti konfigurace ovlivňují základní syntaxi jazyka, což ztěžuje nastavení serveru a odvíjení aplikací.
4. Bezpečnostní problémy mají webové aplikace napsané v PHP.

#### **Python.**

Podle (Lutz 2009), používá se jako interpretovaný jazyk pro skripty různých druhů přiřazení. Syntaxe jazyka je disciplinární a efektivní, což výrazně usnadňuje programátorům spolupracovat na kódu. Python – мультипарадигмальный язык: umožňuje kombinovat procesní přístup k psaní kódu objektově - orientovaný a funkční.

#### **Výhody:**

1. Snadno se učí, zejména-v počátečních fázích.
2. Otevřený vývoj.
3. Syntaxe s jeho vlastnostmi stimuluje psaní dobře čitelného kódu.
4. Poskytování nástroje dynamické sémantiky a rychlého prototypování.
5. Mít velkou komunitu, která je dobře naladěná vůči nováčkům.
5. Obrovské množství užitečných knihoven a rozšíření jazyka se snadným používáním ve svých projektech je velmi jednotný importní mechanismus.
7. Absolutně všechno je objekt ve smyslu OOP, přičemž programátor není nucen objektovým přístupem.

#### **Nedostatky:**

1. Nepříliš pohodlná podpora multithreadingu.

2. Není to tak velké množství kvalitních softwarových projektů ve srovnání s jinými univerzálními programovacími jazyky, jako je Java.
3. Původní omezení finančních prostředků na práci s databázemi.

Java je přísně typizovaný objektově orientovaný programovací jazyk.

#### **Výhody:**

1. Vysílání programu na Java byte-kód v jazyce Java, který je spuštěn virtuální stroj Java – plná nezávislost byte - kódu od operačního systému a hardwaru, který umožňuje spouštět Java aplikace na jakémkoli zařízení, pro které existuje virtuální stroj.
2. Flexibilní bezpečnostní systém-provádění programu je plně řízeno virtuálním strojem.

#### **Nedostatky:**

1. Poměrně velké množství spotřebované paměti.
2. Nízká produktivita v některých případech.

System kontrol verzi. Potreba softwaru pro usnadneni prace s menicimi se informacemi je velmi vysokaa. Pomernoe castaa je situace, kdy pri praci na projektu je potreba upravit kod, ale zaroven neztratit funkcnii variantu, která je v soucasne době. Zpravidla se vytvoří novaa složka, kde se produkt zkopíruje a již se v něm pracuje. Po nějaké době se takových složek může stát mnoho, zejména pokud na projektu bude pracovat několik lidí. Tento způsob práce se zcela ospravedlní, pokud nakonec získáte jen několik složek-kopii projektu. Ale, když vývoj trvá delší dobu, pokud je potřeba časté úpravy, zlepšení, testování nových funkcí, které nakonec mohou přijít vhod, a budete muset vzdát takové možnosti, pak systém pro správu verzí, jen je potřeba, jak je popisováno v (Ustin 2014).

#### **Především je zapotřebí v následujících případech:**

1. Archivace a restaurování. Při dlouhodobém vývoji je archivován určitý počet stavů projektu. Pak každá z nich má vždy možnost obnovit.
2. Synchronizace práce týmu. Je velmi důležité, pokud vývoj není veden jednou osobou, mít představu o stavu projektu všem členům týmu. Při běžné výměně souborů mezi sebou je velká pravděpodobnost, že první přetrpí změnu druhého. Služba pro správu verzí umožňuje " sloučit " změny z různých stavů projektu (jeho kopii) do jednoho bez rizika, že kód někoho jiného bude přepsán.
3. Ukládání historie vývoje. Často je nutné znát nejen projekt v jeho konečném stavu, ale i to, jak se vyvíjel v průběhu celého vývoje.



4. Zrušení změn. V případě, kdy úpravy kódu vedly k tomu, že část projektu nebo celý projekt přestal fungovat správně, musí být stav projektu vrácen zpět na pracovní verzi. Opravte vše ručně dlouho a ne spolehlivě-můžete udělat ještě více chyb.
5. Pomocí systému kontroly verzí můžete rychle a automaticky vrátit zpět do počátečních stavů projektu.
6. Alternativní / experimentální implementace. Můžete pracovat na různých implementacích jedné části projektu, abyste zjistili, který z experimentů bude lepší. V takových případech systém kontroly verzí umožní sledovat obě implementace a poté pohodlně "sloučit" jednu z nich do pracovního projektu.

Systémy kontroly verzí jsou rozděleny do dvou skupin: centralizované a distribuované.

### **Centralizované systémy.**

Tyto systémy jsou aplikace typu klient-server. Repozitář projektu je uložen v jediné instanci na serveru, ke kterému přistupujete pomocí speciální klientské aplikace. Oblíbeným příkladem takového systému je Subversion.

Výhodou centralizovaného systému je povědomí všech vývojářů o tom, co každý z nich dělá a v jaké fázi vývoje je nyní.

Závažnou nevýhodou centralizovaného systému je jediný bod selhání v souvislosti s využitím centralizovaného serveru. V případě selhání poslední práce se službou kontroly verze nebude možné, stejně jako výměna s ostatními vývojáři. Pokud dojde k selhání pevného disku, na kterém je uložena centrální databáze a nejsou k dispozici žádné zálohy, bude celá historie projektu ztracena, s výjimkou místních jednorázových snímků úložiště.

### **Distribuované systémy.**

Umožňuje uložit kopii úložiště od každého vývojáře. Můžete si vybrat centrální úložiště, kde budou shromažďovány změny z místních, s ním budou synchronizovány místní úložiště. Při práci s distribuovaným systémem, uživatelé mají potřeby jsou synchronizovány s centrálním úložištěm, ale pracují přímo s místní kopií.

Velkou výhodou je, že developer je při práci na projektu zcela autonomní. Zvyšuje se také spolehlivost. To je způsobeno tím, že při ztrátě dat na serveru, je možné zkopírovat vlastní úložiště na jiný server a pokračovat v práci. Koneckonců, každá kopie centrálního úložiště představuje úplnou zálohu všech dat. Běžný distribuovaný systém kontroly verzí lze nazvat GIT.

**Vývoj designu.** Podle (Robbins 2012), aplikace, která není intuitivní a obtížná pro vizuální vnímání, nepřispívá k získání velkého množství srdcí. Při navrhování designu je důležité vzít v úvahu uživatelské zkušenosti. Je třeba vytvořit takový model, který bude snadno pochopit a zvládnout celé cílové publikum, stejně jako nezpůsobí odmítnutí a neochotu znovu použít aplikaci.

Implementace rozhraní klient-server interakce. Pro vytváření klientské části webových aplikací se obvykle používají: HTML, CSS, JavaScript (nebo VBScript), PHP. Klientské prostředí (například prohlížeč Google Chrome) je přední hranou aplikace. V tomto prostředí, zobrazí se stránky aplikace. Objekty tohoto prostředí jsou proto

povinny být schopny manipulovat se stránkami. K tomu je potřeba klientský jazyk – JavaScript.

## Django.

Django je zdaleka jedním z nejpopulárnějších webových rámců Pythonu. Jeho slogan "rámování pro perfekcionisty s deadlineem" nejlépe odráží jeho filozofii. V něm je z krabice vše potřebné pro rychlý a efektivní rozvoj. Hned po instalaci budete mít svůj standardizovaný motor, ORM pro práci s databází, migrace, nástroje pro ověřování, ukládání a šifrování hesla, uživatelské stránky správce, seznamy, nástroje pro práci s formami a tak dále. Všechny tyto komponenty dokonale spolupracují a umožňují vám vytvářet své projekty v co nejkratším čase.

Django je rámec open source webových aplikací napsaný v Pythonu, který se volně drží architektury Model-View-Controller. Původně určený pro správu několika zpravodajských stránek světové společnosti v Lawrence, Kansas, byl později veřejně vydán pod licencí BSD open source v červnu 2005. Snímek byl pojmenován po jazzovém kytaristovi Django Reinhardtovi.

Jedná se o další z mnoha webových rámců, jako je Ruby on rails. Ve srovnání s Ruby on rails však nabízí zajímavou možnost automatického vytváření správy projektu, která je dynamicky generována podle datového modelu.

```
1 from django.http import
  HttpResponse
2
3 def hello(request):
4     return
  HttpResponse("Hello world")
```

Obrázek 1 – Datový model

Hlavním úkolem django je snadné vytváření komplexních, databázově řízených webových aplikací. Je zaměřena na opakovatelnost a zaměnitelnost komponent, rychlý vývoj, vždy v duchu "suchého" (neopakujte se) - neopakujte se.

Django je rámec, který zanechává většinu řešení za aplikačním programátorem, je možné použít libovolnou podmnožinu nabízených funkcí. Vhodná sada funkcí může být také použita pro vývoj desktopových aplikací, jak je popisováno v (Ustin 2014)

Projekt je chápán jako nejvyšší úroveň v Django, to je charakterizováno souborem `конфигурационным settings.py` podle názvu je patrné, že je to normální modul v Pythonu. Tento konfigurační soubor je společný pro všechny aplikace v Django. Tento soubor lze vytvořit ručně nebo pomocí skriptu. Díky schopnosti vytvářet struktury projektů a aplikací pomocí jednoduchého skriptu není nutné používat složitější integrované vývojové prostředí (IDE). Programátor je vhodný pro práci s běžným textovým editorem s podporou zvýraznění syntaxe.

**Standardní postup pro vytvoření projektu skriptem po instalaci knihovny Django do místní instalace Pythonu je následující:**

1. Kopírovat skript `django-admin.py` od místní instalace Pythonu až po katalog s projekty.
2. Vytvoření projektu z příkazového řádku pomocí příkazu: `python django-admin.py startproject (název projektu)`.

**Ve druhé fázi se v aktuálním pracovním adresáři vytvoří adresář s názvem projektu. Adresář bude obsahovat následující soubory:**

*manage.py* -scénář, který je zodpovědný za řízení projektu.

**Mezi jeho hlavní úkoly patří:**

- vytvoření základní struktury nové aplikace,
- vytvoření schématu databáze,
- spuštění testovacího webového serveru,
- spuštění modulárních testů.
- `settings.py` -konfigurační skript sdílený pro všechny aplikace v projektu obsahuje zejména,
  - nastavení připojení k databázi,
  - nastavení připojení k serveru SMTP pro snadné odesílání e-mailů,
  - Konfigurace middleware (diskutováno dále v části Middleware),
  - nastavení časového pásma a Locale aplikace,
  - registrace žádostí v projektu,
  - `urls.py` - globální konfigurace adresy URL. V tomto souboru jsou adresy URL přeměrovány na konkrétní zobrazení,
  - `__init__.py` - adresář projektu je také Pythonem díky tomuto souboru.

Konfigurační soubor je generován plně *комментированным*, takže i úplný začátečník v práci s touto knihovnou bude pochopitelně význam jednotlivých konfiguračních klíčů. V tomto okamžiku je příkaz pro spuštění testovací server je již k dispozici, tak, že `python` lze vyzkoušet `manage.py runserver` a po přístupu na URL-adrese `http://localhost:8000` můžete již vidět úvodní stránku. Můžete začít vytvářet aplikaci pouze tehdy, pokud jsme již vytvořili projekt. Aplikace v Django jsou vždy součástí projektu, jedna aplikace může být součástí několika projektů. Je však důležité, aby interpret Pythonu našel registrovaný modul aplikace. Aplikace jsou tedy samostatné součásti systému, které mohou mít závislosti mezi sebou. Django v těchto případech nepředstavuje žádnou magii na pozadí, ale pouze podporuje to, co Python již zná ve výchozím nastavení. Struktura nové aplikace je vytvořena po spuštění: `python manage.py startapp`, jak je popisováno v (Darie 2005). Při spuštění tohoto skriptu se v aktuálním pracovním adresáři vytvoří adresář názvů aplikací s následující strukturou:

- `views.py` -obsahuje jednotlivé funkce zobrazení.
- `urls.py` -obsahuje mapování adres URL s individuálním zastoupením.
- `models.py` - obsahuje popis datového modelu aplikace.
- `tests.py` -obsahuje modulární testy.

- `__init__.py` -dělá aplikace Pythonu.

Každá žádost musí být zapsána v `settings.py` v položce `INSTALLED_APPS`. Poté můžete ovládat aplikaci pomocí skriptu `manage.py` a použijte všechny standardní funkce, které Django nabízí. V případě většího zájmu o bližší seznámení s jednotlivými soubory doporučujeme navštívit oficiální dokumentaci.

Žádost je prezentována v Django jako instance třídy `HttpRequest`. V tomto objektu jsou metadata o požadavku HTTP zpřístupněna prostřednictvím jeho atributů a metod. Stručný přehled atributů třídy `HttpRequest`:

- metoda je textový řetězec, který představuje metodu http dotazu.
- cesta-cesta na požadovanou stránku.
- Get-atributy odeslané na adresu URL.
- Post-atributy odeslané v těle dotazu.
- Dotaz-sdílení atributů.
- Cookies-soubory cookie odeslané na žádost http.
- Soubory-Soubory ke stažení.
- Meta-obsahuje všechny dostupné HTTP hlavičky.
- uživatel je objekt ověřeného uživatele.

Odpověď je instance třídy `HttpResponse`. Tento objekt má schopnost vyplnit všechny odpovědné hlavičky protokolu HTTP.

Mezi přijetím dotazu a zasláním odpovědi z určitého podání prochází dotaz řetězcem nazývaným middleware. V této knihovně se však pochopení middleware liší od obvyklého. Zde jsou třídy middleware chápány podobně jako filtr v Java Servlet API. Je zřejmé, že je to aplikace šablony návrhu řetězce odpovědnosti. Úkolem této konstrukční šablony je vyhnout se vázání odesílatele a příjemce zprávy pomocí řetězce objektů, které mohou reagovat. Django používá funkci prohlížení pro zachycení dotazů. Tato funkce je porovnána s určitou adresou URL. Standardní způsob je definovat pohledy jako jednotlivé funkce, ale díky flexibilitě Pythonu můžete použít pohledy srovnatelné s existující instancí objektu.

Po zpracování dotazu do funkce je vhodné použít vestavěný systém šablon. Šablona zde je obyčejný textový soubor, který díky speciálním značkám umožňuje resetovat text z modelu určeného pro vykreslování. Model je instance třídy `Context`, která je vytvořena z objektu `dictionary`. Tento vzorový systém je velmi flexibilní a umožňuje vám snadno psát vlastní rozšíření. Další jeho velkou výhodou je možnost dědičnosti vzorů. Toto dědictví umožňuje potomkovi znovu definovat nebo rozšířit oddíly definované v předkovi. Oddíly jsou definovány takto:

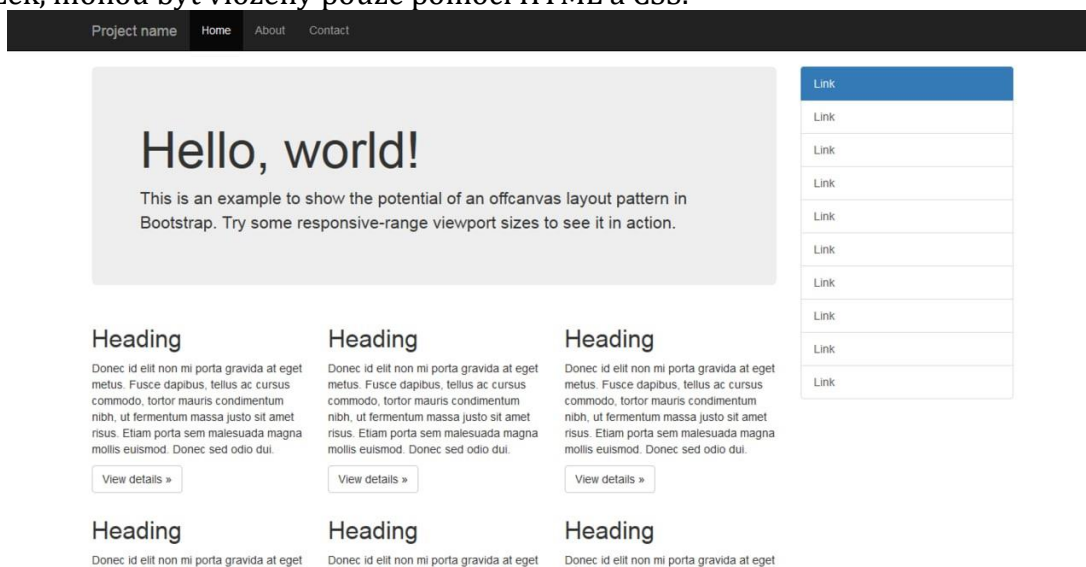
- začátek oddílu `{%block sectionname %}`
- konec sekce `{%endblock %}`

Dalším velmi silným dílem django je objektově relační zobrazení. Každý model je potomkem třídy `Model` z modulu `Django.db`. Jedná se o implementaci šablony návrhu `activeobject`. Každá třída má automaticky jedno pole a to je `id`. Pomocí tohoto pole identifikujete konkrétní objekt v databázi. Zde se používá dynamický princip Pythonu a za běhu se modely rozšiřují pomocí rozhraní API pro práci s metadaty. Další pole je definováno uživatelem, můžete použít jak Předdefinované typy databázových polí, tak realizovat své vlastní, jak je popisováno v (Ustin 2014).

## Bootstrap.

Bezplatná sada nástrojů pro vytváření webových stránek a webových aplikací. Obsahuje HTML a CSS šablony pro typografii, webové formuláře, tlačítka, štítky, navigační bloky a další komponenty webového rozhraní, včetně rozšíření JavaScriptu.

Bootstrap je bezplatná sada nástrojů kaskádových stylů s otevřeným zdrojovým kódem pro vytváření webových aplikací a webových aplikací. To zahrnuje šablony založené na HTML a CSS používané pro úpravu typografie, formuláře, tlačítka, navigační a jiné komponenty rozhraní, a další rozšíření programovacího jazyka JavaScript. Pro použití Bootstrap, budete potřebovat základní znalosti HTML a CSS, interaktivní prvky, jako jsou tlačítka, boxy, menu a další kompletně nainstalované a graficky zpracované položky, mohou být vloženy pouze pomocí HTML a CSS.



Obrázek 2 - Bootstrap

Slovo bootstrap původně znamenalo popruh na boty, strupli.

Podle (Hais 1997), v oblasti informačních technologií byl Bootstrap představen Markem Ottem a Jacobem Thorntonem jako rámcem, který udržuje konzistenci mezi vnitřními nástroji. Před bootstrap se mnoho knihoven používalo k vývoji rozhraní, což vedlo k nesouladu zdrojových kódů a vysoké údržbě. Vývojář Mark Otto řekl o tom: "Velmi malá skupina vývojářů se mnou měla za úkol navrhnout a postavit nový interní nástroj a viděla příležitost udělat něco většího. Prostřednictvím tohoto procesu jsme chtěli vytvořit něco mnohem důležitějšího než další nástroj pro vnitřní použití. O měsíc později jsme dokončili první verzi Bootstrap jako metodu dokumentování a sdílení společných návrhových šablon a přínosů ve společnosti

V srpnu 2011. byl Twitter Bootstrap propuštěn jako open source software a v únoru 2012. se stal nejoblíbenějším vývojářským projektem na Githubu.

Výhodou této sady nástrojů spočívá v tom, že se snadno zpracovává jakékoli uživatelské rozhraní ve webové aplikaci a neřeší, zda je, například, uživatelské rozhraní při správě back-end nebo front-end aplikací. Bootstrap je kompatibilní s nejnovější verzí všech hlavních prohlížečů a je elegantně přizpůsoben pro použití ve starších prohlížečích, jako je Internet Explorer 8. Od verze 2.0 podporuje také adaptivní design. To znamená, že rozložení stránek se dynamicky přizpůsobuje použitému zařízení (stolní počítač, Tablet, Mobilní telefon).

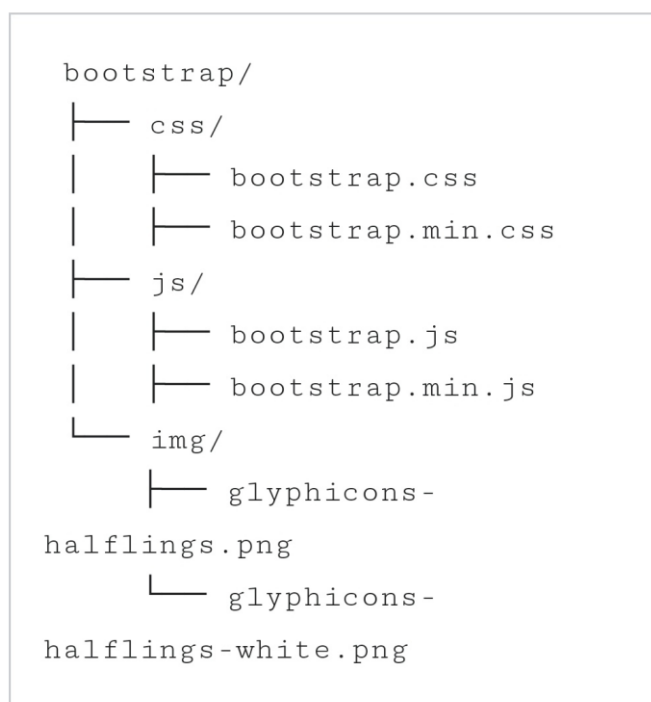
Bootstrap má otevřenou licenci a je k dispozici na Githubu. Vývojáři jsou vyzýváni, aby se zapojili do projektu a vytvořili svůj vlastní podíl na platformě. Členové komunity nedávno přeložili dokumentaci Bootstrap do mnoha různých jazyků, včetně čínštiny, španělštiny a ruštiny. Česká verze momentálně chybí.

Na internetu na stránce (Otto 2017) je, Bootstrap je modulární a většinou se skládá ze série méně kaskádových stylů (HTML stránky), které implementují různé součásti sad nástrojů. Komponenty Bootstrap jsou zahrnuty v tabulce stylů bootstrap.méně. Vývojáři mohou nastavit soubor počátečního stahování sami výběrem komponent, které chtějí použít pro svůj projekt. Nastavení jsou do jisté míry možné prostřednictvím centrální konfigurační tabulky stylů. Hlubší změny jsou možné s menším počtem deklarací. Použití jazyka stylů less umožňuje použití proměnných, funkcí, vnořených selektorů atd.

Hlavní formou bootstrap jsou kompilované soubory, které lze použít v jakémkoli webovém projektu. Bootstrap poskytuje kompilované CSS a js (bootstrap.\* ), stejně jako kompilované a minimalizované CSS a JS soubory (bootstrap.min.\*). Obrazové soubory jsou komprimovány pomocí aplikace ImageOptim, což je aplikace Mac pro kompresi souborů PNG. Twitter Bootstrap závisí na knihovně jQuery JavaScript.

Od verze 2.0, konfigurace Bootstrap také obsahuje speciální možnost "přizpůsobit" v dokumentaci. Kromě toho mohou vývojáři v případě potřeby zvolit formu potřebných komponent a jejich nastavení. Následně vygenerovaný balíček již obsahuje předem vytvořenou tabulku stylů CSS. Síťový systém a adaptivní webový design jsou standardně vybaveny rozložením mřížky o šířce 1170 Pixelů. Alternativně může vývojář použít proměnnou šířku rozvržení. Pro oba případy toolkit má čtyři možnosti vytvoření různých oprávnění a typy zařízení: mobilní telefony, portrétní a альбомные, tablety a počítače s nízkým a vysokým rozlišením. Každá možnost je nastavitelná šířka reproduktorů, jak je popisováno v (Urban 2018).

Po stažení a rozbalení souboru najdete tuto strukturu:



Obrázek 3 – Struktura Bootstrapu

Adresáře můžete přejmenovat pomocí stylů, skriptů a obrázků ve svém vlastním projektu (a odpovídajícím způsobem přizpůsobit odkazy na ně). Počínaje novějšími

verze Bootstrap zanechal cestu pro vlastní obrázky a od verze 4.0 také vektorovou grafiku pomocí vlastních písem.

## Úpravy stylů CSS

Bootstrap poskytuje sadu stylů, které poskytují základní definice stylů pro všechny klíčové části HTML. Poskytují jednotný, moderní vzhled formátování textu, tabulek a tvarových prvků.

## Opakovaně použitelné součásti úpravy

Kromě běžných prvků HTML obsahuje bootstrap i další běžně používané prvky uživatelského rozhraní. Sem patří tlačítka s pokročilými nastaveními a funkcemi (např. seskupení tlačítek nebo tlačítek s выпадающими меню, vytvoření navigační listů, horizontální nebo vertikální záložky, navigace, malá navigace, swap stránek, Atd.), štítky, pokročilé možnosti typografie, tvorba náhledů, výstražná upozornění a ukazatel průběhu

Na internetu na stránce (Ecma 2017) je, Bootstrap je k dispozici s několika komponentami JavaScriptu jako pluginy jQuery. Poskytují rozšířenou možnosti uživatelského rozhraní, jako jsou dialogová okna, pop-up tipy a tak Dále Se také rozšiřují funkčnost stávajících prvků uživatelského rozhraní, včetně, například, funkci automatického vyplňování polí formuláře zadání. Ve verzi 4.0 jsou podporovány následující komponenty:

- Upozornění (chybové zprávy, úspěch, varování, Informace),
- Ikona (typ textu oznámení: počet nepřečtených zpráv),
- Strouhanka (breadcrumb navigation),
- Tlačítka,
- Skupina tlačítek,
- Karta (štítky / karty),
- Kolotoč ("kolotoč" obrázků),
- Collapse (efekt komprese obsahu položky),
- Rozevírací seznamy (rozevírací seznamy z nabídky),
- Vstupní skupina,
- Jumbotron (největší položka webu),
- Skupina seznamů (stylizované seznamy),
- Média (styly zobrazení obrázků médií, videa, ...),
- Modální (modální dialogy / layovers),
- Navs (položky v navigačním panelu),
- Navbar (navigační lišta),
- Paginace,
- Popover (otevírání oken),
- Pokrok (zobrazuje průběh operace nebo procesu)
- Scrollspy (ukazatel na záhlaví, do kterého se aktuálně posouvá stránka),
- Karty (přepínatelné položky, z nichž jedna je aktuálně zobrazena),
- Pop-up tipy (bubble help).

Mnoho z těchto komponent implementovalo funkce JavaScriptu, pomocí kterých je lze ovládat. U některých z nich bootstrap také definuje události, na které mohou uživatelé "zavěsit" spuštění vlastního kódu (například při volání nebo zavření modálního okna).

## Úpravy grid systému

Bootstrap již zavedl takzvaný síťový systém s jednou z prvních verzí. Ta funguje ve spojení se zastávkovými body pro šířku okna. V síťovém systému mohou jednotlivé položky určit, jaký podíl na šířce nadřazeného prvku by měly zaujímat. To se provádí přidáním správné třídy ke správným prvkům.

```
<div class="row">
  <div class="col-lg-3 col-
md-4">
    Postranní panel
  </div>
  <div class="col-lg-9 col-
md-8">
    Hlavní panel
  </div>
</div>
```

Obrázek 4- Příklad Bootstrapu

Ve výše uvedeném příkladu jsou dva prvky v nadřazeném prvku (boční a hlavní panel) stylizovaný tak, že k velikosti (nebo rozsah šířky okna) lg (large) zabírají 3/12 (tj. 1/4) a 9/12 (tj. 3/4) šířky nadřazeného prvku, a pro velikost MD (medium) je tento poměr roven 4/12 a 8/12, tj. mezi 1/3 a 2/3. Pro velikosti SM (small) a XS (extra small) nebudou vedle sebe, ale pod sebou (implicitní chování). Jmenovatel těchto zlomků, 12, může být změněn v konfiguraci, ale kvůli jeho dobré dělitelnosti to obvykle není nutné. A zároveň přesnost dělení na dvanáct, stejně jako možnost rovnoměrné rozdělení prvků po celé šířce na polovinu, třetiny, čtvrtiny, шестые a двенадцатые zcela dostačující pro drtivou většinu případů, jak je popisováno v (Otto 2017).

## Nunjucks.

Další {{systém založený na tokenu}}, který poskytuje (stejně jako ostatní motory) logiku, provádění cyklů a schopnost spravovat proměnné. Kromě toho však šablony poskytují další prvky pro vytváření stránek, jako je dědění bloků, včetně dědění rozvržení, vlastních značek a maker. To vše je nejlepší pro CMS, kde je místo sérií "stavebních bloků".

Nunjucks je bohatý a výkonný jazyk šablon pro JavaScript vytvořený Mozillou, který všichni známe při práci s Firefoxem. Stručně řečeno, Nunjucks je bohatý a velmi pohodlný jak pro začátečníky, tak pro profesionály. Díky své lehké struktuře už víte, že



provedení nujuků bude rychlé a bezchybné. Tento nástroj je také flexibilní a rozšiřitelný pomocí vlastních filtrů a rozšíření, které si můžete představit dle svého přání. Nunjaki můžete použít v node.js nebo v jakémkoli moderním a populárním prohlížeči. Stránka Nunjucks obsahuje mnoho různých příkladů, které vám pomohou získat správný nápad.

## **jQuery - JavaScript.**

Zdroje jQuery jsou neobvykle výrazné. Tato knihovna umožňuje mnohem více při mnohem menším objemu kódu než v případě softwarových rozhraní DOM prohlížečů.

Metody jQuery se vztahují na všechny skupiny prvků. Standardní přístup nabízený v modelu DOM založený na akčním řetězci šablony "Select-repeat-change" již není vyžadován. Výsledkem je snížení počtu cyklů for V kódu, a tím i pravděpodobnost chyb v něm, jak je popisováno v (Robbins 2012).

Knihovna jQuery se vyrovnává s rozdíly v implementaci dom v různých prohlížečích (Cross-head problémy). Například bych neměl být znepokojen myšlenkou na pomocné funkce jednoho nebo druhého nástroje než notoricky známý Internet Explorer. Stačí formulovat jQuery vaše přání a knihovna sama zajistí kompatibilitu s jedním nebo druhým prohlížečem.

Knihovna jQuery má otevřený zdrojový kód. Pokud mi principy fungování nějakého nástroje nejsou zcela jasné nebo výsledný výsledek neodpovídá očekávanému výsledku, mohu se obrátit přímo na kód knihovny v jazyce JavaScript a v případě potřeby provést příslušné změny.

jQuery je knihovna JavaScript s širokou podporou prohlížeče, která zdůrazňuje interakci mezi JavaScriptem a HTML. Byl vydán Johnem Recigem v lednu 2006 v newyorském barcampu.

jQuery je bezplatný open source software pod licencí MIT.

Microsoft a Nokia oznámily, že uvedou jQuery na své platformy. Microsoft dodává tuto knihovnu současně s Visual Studio pro použití v jejich *ASP.NET* AJAX framework a *ASP.NET* MVC framework, zatímco Nokia integruje jquery ve své platformě web runtime.

Stejně jako CSS odděluje vlastnosti "Zobrazení" od struktury HTML, jQuery odděluje "chování" od struktury HTML. Například namísto přímého zadání události onclick přímo v souboru HTML button, spravovaná stránka jQuery nejprve najde vhodnou položku button a poté ji změní obslužný program událostí. Toto oddělení chování od struktury je také často označováno jako princip nenápadného JavaScriptu, jak je popisováno v (Chaffer 2007).

## **jQuery nabízí následující funkce:**

- Výběr prvků DOM pomocí Open crossbrauserového voliče Sizzle, odnože projektu jQuery
- Funkce pro zobrazení a změnu DOM (včetně podpory 1-3 a basic XPath)
- Události
- Manipulace s CSS
- Efekty a animace
- AJAX
- Škálovatelnost
- Nástroj-například informace o prohlížeči nebo funkce každého
- Pluginy JavaScriptu

Jedním z nejdůležitějších pojmů v jakémkoli kódu jQuery je takzvaná funkce "\$". "\$" je ve skutečnosti pseudonymem jmenného prostoru "jQuery".

jQuery nabízí funkci pro oříznutí řetězců. Tato funkce může být použita jako:

```
ret = " foo ";  
jQuery.trim(ret); // vrátí  
"foo"
```

Nebo také jako:

```
ret = " foo ";  
$.trim(ret);
```

Obrázek 5 – Příklad 1 jQuery

Tyto kódy jsou shodné. Použití „\$“ místo „jQuery“ je ad-hoc konvence a je považováno za rychlejší cestu pro přístup ke knihovně jQuery.

Pro vybrání všech odstavců s třídou „foo“ a pro přiřazení druhé třídy „bar“:

```
$( "p.foo" ).addClass( "bar" );
```

Obrázek 6 – Příklad 2 jQuery

Podle (Chaffer 2007), jQuery Pro spuštění funkce „mojefunkce“ ihned po načtení stránky (v žargonu jQuery: poté, co je zavolán manipulátor ready):

```
$( document ).ready( function() {  
    mojefunkce();  
});
```

Obrázek 7 – Příklad 3 jQuery

Toto je většinou používáno v kontextu jako:

```

$(document).ready(function() {
    // Obarvit pozadí lichých,
    resp. sudých řádků v tabulkách
    pomocí [[Kaskádové styly|CSS]]
    tříd lichyPruh, resp.
    sudyPruh.

    $('tr:odd').addClass("lichyPru
h");

    $('tr:even').addClass("sudyPru
h");
});

```

Obrázek 8 – Příklad 4 jQuery

jQuery většinou existuje jako jeden javascriptový soubor, obsahujícího všechny funkce pro DOM, Ajax, události a efekty. Do webové stránky může být vložen následovně:

```

<script
type="application/javascript"
src="/cesta/k/jquery.js">
</script>

```

Obrázek 9 – Příklad 5 jQuery

Nejnovější stabilní verze může být také načtena pomocí Google AJAX Libraries API. Tento způsob získávání knihovny má mnoho výhod včetně unifikovaného cachování a snížení odezvy, jak je popisováno v (Urban 2018). To může být provedeno následovně:

```
<script
type="application/javascript"
src="http://www.google.com/jsa
pi"></script>
<script
type="application/javascript">
google.load("jquery",
"1.3.2");
</script>
```

Obrázek 10 – Příklad 6 jQuery

## WebSocket.

Protokol WebSocket (standard RFC 6455) je určen pro řešení všech úkolů a zmírnění omezení výměny dat mezi prohlížečem a serverem.

To vám umožní předávat všechna data, na jakoukoli doménu, bezpečně a téměř bez zbytečného síťového provozu.

Chcete-li otevřít připojení, stačí vytvořit objekt WebSocket zadáním speciálního protokolu ws.:

```
var socket = new WebSocket("ws://javascript.ru/ws");
```

Objekt soketu má čtyři vlákna: jeden při extrakci dat a tři při změně stavu připojení.

WebSocket je počítačový komunikační protokol, který poskytuje plně duplexní (obousměrné) komunikační kanál po jednom TCP spojení. Protokol WebSocket byl standardizován společností IETF jako RFC 6455 v roce 2011 a rozhraní WebSocket API společnosti Web IDL bylo standardizováno konsorciem W3C.

Podle (Katz 2008), websocket je navržen takovým způsobem, že to může být realizován ve webových prohlížečích a na webových serverech, ale může být použit kdykoliv klientská a serverová aplikace. WebSocket je nezávislý protokol založený na protokolu TCP. Jeho jediný vztah k HTTP je, že jeho handshake (připojení) je interpretován HTTP serverem jako žádost o aktualizaci. Protokol WebSocket umožňuje komunikovat mezi prohlížečem a webovým serverem s nižšími režijními náklady, usnadňuje přenos dat v reálném čase ze serveru a na server. To je možné proto, že poskytuje standardizovaný způsob odesílání obsahu ze serveru do prohlížeče, aniž by museli dělat to na žádost klienta (jako je Ajax) a umožňuje posílat zprávy tam a zpět při udržování otevřeného spojení. Obousměrná komunikace mezi prohlížečem a serverem tak může probíhat. Toto spojení probíhá prostřednictvím portu TCP číslo 80 (nebo 443 v případě šifrovaných připojení TLS), což je výhoda pro prostředí, kde jsou připojení k internetu Neweb blokována firewallem. Stejně tak byla obousměrná komunikace mezi

prohlížečem a serverem dosažena nestandardními způsoby pomocí improvizovaných technologií, jako je comet.

Protokol WebSocket je v současné době podporována na většině hlavních prohlížečů, včetně Google Chrome, Microsoft Edge, Ms Internet Explorer, Firefox, Safari a opera. WebSocket však také vyžaduje podporu na straně serveru, nikoliv pouze klienta (prohlížeče) a jeho JavaScriptu.

Na rozdíl od http, WebSocket umožňuje obousměrnou komunikaci. Kromě toho WebSocket umožňuje toky zpráv založených na protokolu TCP. Vlastní TCP se zabývá bytovými proudy spíše než vlastním pojetím zpráv. Před příchodem WebSocket oboustranné spojení na portu 80 bylo dosažitelné pomocí kanálů, comet, ale jeho realizace byla netriviální a z důvodu režie protokolu TCP-квитирование a HTTP hlavičky nástrojem pro malé zprávy. Protokol WebSocket je zaměřen na řešení těchto problémů bez ohrožení bezpečnostních požadavků webu.

Specifikace WebSocket určuje Ws a WSS jako dvě nové sjednocené systémy identifikátorů zdroje (Uri), které se používají pro незашифрованных a možná, šifrované připojení. Na rozdíl od režimu názvu a fragmentu (# není podporováno), zbytek URI komponenty je definován pomocí obecné syntaxe Uri.

Pomocí prohlížeče vývojářských nástrojů mohou vývojáři sledovat vytváření připojení Websocketu, stejně jako jeho rámce.

Chráněné verze protokolu WebSocket je implementován ve Firefoxu 6, safari 6, Google Chrome 14, opera 12.10 a Internet Explorer 10. Podrobná zpráva o výsledcích testů protokolu naznačuje, že tyto prohlížeče odpovídají jeho konkrétním aspektům, jak je popisováno v (Urban 2018).

Starší, méně bezpečná verze protokolu byla implementována v opeře 11 a Safari 5, stejně jako v mobilní verzi safari v systému iOS 4.2. Prohlížeč BlackBerry v OS7 implementuje protokol WebSocket. Kvůli chybě návrhu byla zakázána ve Firefoxu 4 a 5, stejně jako opera 11.

Na internetu na stránce (Msdn 2013), je k dispozici je také přepínač příkazového řádku enable-WebSocket-over-SPDY pro Google Chrome, který vám umožní používat časně experimentální implementace WebSocket přes SPDY.

## **4 Vlastní práce Aplikaci HonestyFraud**

### **4.1 Popis technické zadání pro vývoj herní aplikace.**

#### **1. Obecné informace.**

1.1. Název: "HonestyFraud".

1.2. Oblast použití: všesměrová aplikace.

Je technický úkol se vztahuje na vývoj hry, klient-server aplikace, zaměřené na analýzu, syntézu, aktivní rozvoj znalostí, dovedností, osobních vlastností, ale také komunikaci mezi uživateli.

2. Důvody a účel vývoje.

2.1. Cílem vývoje je síťová počítačová herní aplikace, která využívá multiplatformní nástroje pro poskytování klient-serverové interakce.

2.2. Aplikace je vyvíjena z osobní iniciativy autora po dohodě s vedoucím maturitní práce, stejně jako v souladu s učebním plánem katedry.

3. Požadavky na vývoj produktů.

3.1. Hardwarové požadavky:

3.1.1. Místní počítač.

3.1.2. Prohlížeč.

3.1.3. Přístup k internetu.

3.2. Údaj o softwaru používaném k implementaci.

3.2.1. Místní počítač s operačním systémem Microsoft Windows 10.

3.2.2. Prohlížeč.

4. Požadavky na uživatelské rozhraní.

4.1. Intuitivní, bez nutnosti učení o vlastnictví aplikace.

4.2. Herní chat.

5. Složení a obsah prací na vytvoření aplikace.

5.1. Analýza požadavků na aplikaci.

5.1.1. Dynamické hrací pole.

5.1.2. Možnost zasílání zpráv mezi soupeři v herním chatu. Máme v něm systémové zprávy o tom, kdo vešel do místnosti a kdo vyšel ven.

5.1.3. Přítomnost intuitivního uživatelského rozhraní.

5.2. Návrh a vývoj aplikace.

5.2.1. Vypracování technické úlohy.

5.2.2. Vývoj struktury aplikace.

5.2.3. Vývoj rozhraní.

5.2.4. Implementace aplikace.

5.2.5. Psaní dokumentace.

5.2.6. Testování.

6. Pořadí odevzdání - přijetí produktu.

6.1. Průběžná kontrola-konec března 2021, objem-hlavní funkce, Kontrola-vedoucí.

## 4.2 Vývoj aplikace

### Vývoj aplikace se skládá z:

1. Vývoj databázové struktury.
2. Vývoj backend aplikace.
3. Vývoj klientské části.
4. Logika aplikace.

### Samotná aplikace se skládá z několika dílčích aplikací:

1. Home-obsahuje pomocný kód, funkční uživatelské tabulky, funkce pro registraci uživatele v aplikaci.
2. Game-podpisový turnaj, který je zodpovědný za samotný turnaj, stejně jako chat se zprávami.
3. Websocket-dílčí aplikace s generalizovanou funkcí pro interakci s webovými zásuvkami klientské části aplikace.

### Vývoj struktury database

Jako databáze aplikace používá SQLite databáze, nicméně Django může fungovat ze složitějších typů databází, jako je MySQL, PostgreSQL, atd.

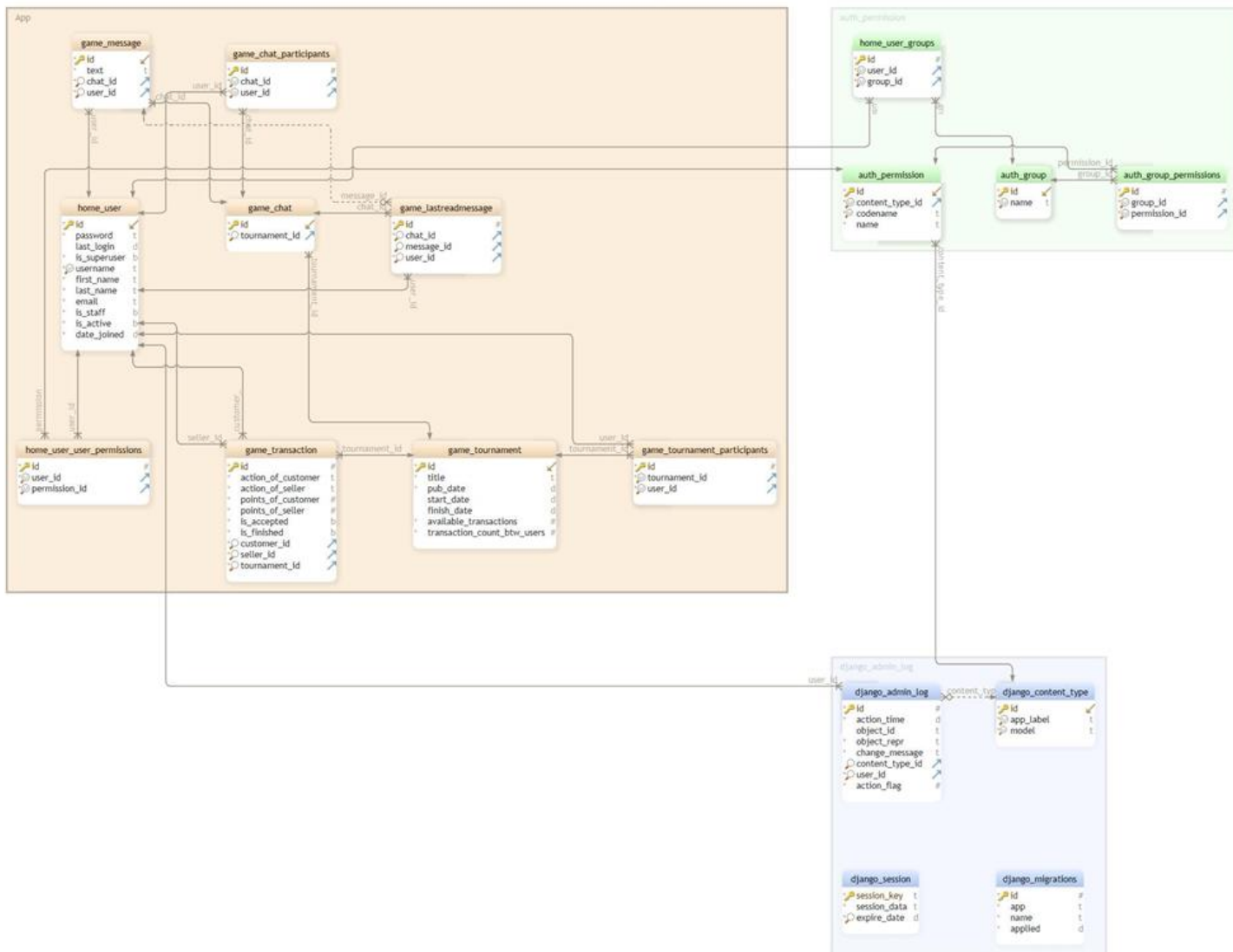
Obrázek ukazuje strukturu celé databáze.

Oddíly `auth_permission` a `django_admin_log` jsou součástí projektu Django a jsou poskytovány v `фреймворке` výchozí.

Sekce `App` je vytvořen v procesu vývoje aplikace. Obrázek ukazuje strukturu celé databáze.

Oddíly `auth_permission` a `django_admin_log` jsou součástí projektu Django a jsou poskytovány v `фреймворке` výchozí.

Sekce `App` je vytvořen v procesu vývoje aplikace.

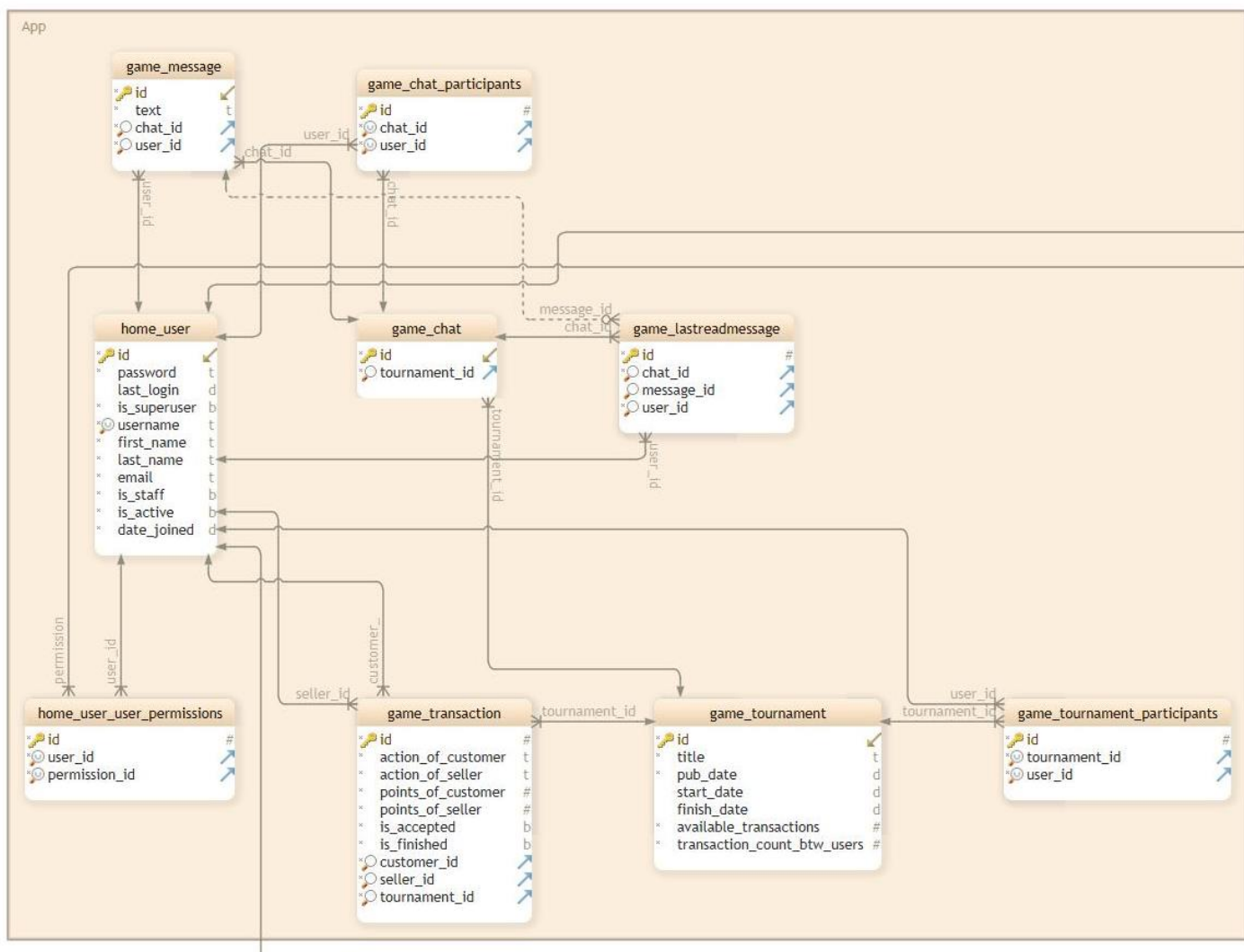


Obrázek 11 - Struktura celé databáze.

Oddíly **auth\_permission** a **django\_admin\_log** jsou součástí projektu Django a jsou poskytovány v fréjmworke výchozí.

Sekce **App** je vytvořen v procesu vývoje aplikace.





Obrázek 12 – Popis tabulek

### home\_user

Tabulka uživatele v dílčí aplikaci home:

- id – Primary key
- heslo-heslo
- last\_login - datum poslední autorizace
- username-přihlášení
- first\_name-jméno
- last\_name-příjmení
- email – email
- is\_staff-je zaměstnanec webu
- is\_active - je aktivní uživatel
- date\_joined – datum registrace

### game\_tournament

Tabulka turnajů:

- id – Primary key
- title-název turnaje
- pub\_date – datum vytvoření turnaje

- start\_date - datum zahájení turnaje
- finish\_date – datum dokončení turnaje
- available\_transactions - počet dostupných tranzitní pro jednoho uživatele
- transaction\_count\_btw\_users – počet transakcí mezi dvěma uživateli

### **game\_tournament\_participants**

Tabulka účastníků turnaje, organizuje mnoho spojení mezi tabulkami turnajů a uživatelů:

- id – Primary key
- tournament\_id-externí klíč k tabulce turnajů
- user\_id-externí klíč na tabulku uživatelů

### **game\_transaction**

Tabulka transakcí v turnaji:

- id – Primary key
- action\_of\_customer - výběr uživatele, skutečný nebo falešný nákup
- action\_of\_seller - výběr prodejce, skutečný nebo falešný prodej
- points\_of\_customer – brýle uživatele
- points\_of\_seller-brýle prodejce
- is\_accepted – transakce schválena oběma stranami
- is\_finished – transakce je u konce
- customer\_id-externí klíč na uživatelské tabulky, pole s kupujícím
- seller\_id-externí klíč na uživatelské tabulky, pole s prodejcem
- tournament\_id - externí klíč pro turnaj

### **home\_user\_user\_permissions**

- id – Primary key
- user\_id-externí klíč na tabulku uživatelů
- permission\_id-externí klíč na tabulku práv v rámečku django

### **game\_chat**

Tabulka herních chatů mezi uživateli:

- id – Primary key
- tournament\_id - externí klíč pro turnaj

### **game\_chat\_participants**

Tabulka účastníků chatu, organizuje mnoho spojení mezi tabulkami chatu a uživatelů:

- id – Primary key
- chat\_id-externí klíč k tabulce chat
- user\_id-externí klíč na tabulku uživatelů

### **game\_message**

Tabulka zpráv mezi uživateli v chatu:

- id – Primary key
- text zprávy
- chat\_id-externí klíč k tabulce chat
- user\_id-externí klíč na tabulku uživatelů

## **game\_lastreadmessage**

Tabulka posledního přečteného chatu pro zadaného uživatele. Umožňuje sledovat, která zpráva Naposledy přečetl konkrétní uživatel chatu:

- id – Primary key
- chat\_id-externí klíč k tabulce chat
- message\_id-externí klíč k tabulce zpráv
- user\_id-externí klíč na tabulku uživatelů

## **Vývoj backendu**

Funkce aplikace je postavena na standardním zpracování url, stejně jako zpracování webových soketů. Pro zpracování standardních adres URL se používá funkce Django se standardním směrovačem. Pro zpracování web-socket se používá Middleware, který určuje typ dotazu a odešle zprávu do příslušné zásuvky.

Při zpracování běžných stránek aplikace backend směruje žádost o příslušný pohled, který generuje stránku ze šablony.

Na stránkách, kde je inicializace webových zásuvek přítomna, je zaslána žádost o připojení k webu prostřednictvím zásuvky připojení.

Dále pomocí kódu JavaScriptu odešlete reakce na manipulaci s rozhraním a také se vrátí výsledná data.

Data jsou přijímána ve formátu JSON, po kterém jsou šablonovány pomocí knihovny Nunjucks, syntaxe šablon, která odpovídá syntaxi šablon Django.

Připojení k serveru prostřednictvím webových zásuvek se provádí v asynchronním režimu, který v současné verzi Django 3 se provádí pomocí standardních Python nástrojů.

## **Vývoj klientské části aplikace**

Django framework sám o sobě může být použit jak pro plnohodnotné vykreslování stránek, tak jako rozhraní API pro klientskou část. V tomto projektu jsou jednoduché stránky vykresleny pomocí backendu Django, zatímco pro realtime částí aplikace, Django se používá jako API pro interakci s webovými zásuvkami.

Součástí klienta je JavaScript Soubor pro realtime aplikace v okně turnaje, stejně jako na stránce statistiky turnaje. K implementaci této funkce byl napsán soubor *tournament.js*.

Pro implementaci interakce s webovou zásuvkou součástí aplikace. Používá se systém zpráv s nastavením typu zprávy.

### **K dispozici jsou následující typy zpráv:**

1. INIT-inicializace
2. UPDATE\_USER\_LIST - aktualizace seznamu uživatelů
3. OPEN\_CHAT - Otevření chatu
4. MESSAGE - odeslání zprávy
5. READ\_MESSAGE-zpráva přečtena
6. UPDATE\_TRANSACTION\_STATISTIC - aktualizace statistik transakcí
7. REQUEST\_TRANSACTION\_STATISTIC - dotaz statistiky transakcí
8. UPDATE\_TRADE\_PANEL - aktualizace obchodního panelu

9. SEND\_TRANSACTION\_ACTION-odkaz na vybranou akci transakce: skutečný nákup, falešný nákup, skutečný prodej, falešný prodej
10. SELL\_REQUEST - žádost o prodej
11. ACCEPT\_REQUEST - souhlas s transakcí
12. REJECT\_REQUEST-zrušení transakce
13. TRANSACTION\_FINISHED-transakce dokončena
14. UPDATE\_TOURNAMENT\_STATISTIC - aktualizace statistik turnaje
- 15 . Turnaj skončil
16. TOURNAMENT\_NOT\_STARTED-turnaj nezačal
17. TOURNAMENT\_STARTED-turnaj začal

## Logika aplikace

Na hlavní stránce aplikace je seznam turnajů. Když správce aplikace vytvoří turnaj, tento turnaj se objeví v seznamu turnajů.

Poté administrátor vstoupí do sekce statistiky a očekává, že uživatelé vstoupí do samotného turnaje. Poté, co všichni uživatelé vstoupili do turnaje, administrátor zahajuje turnaj. Poté se podle zprávy *TOURNAMENT\_STARTED* zobrazí žádost o všechny potřebné informace, jako je seznam uživatelů, aktualizace obchodního panelu a statistiky transakcí.

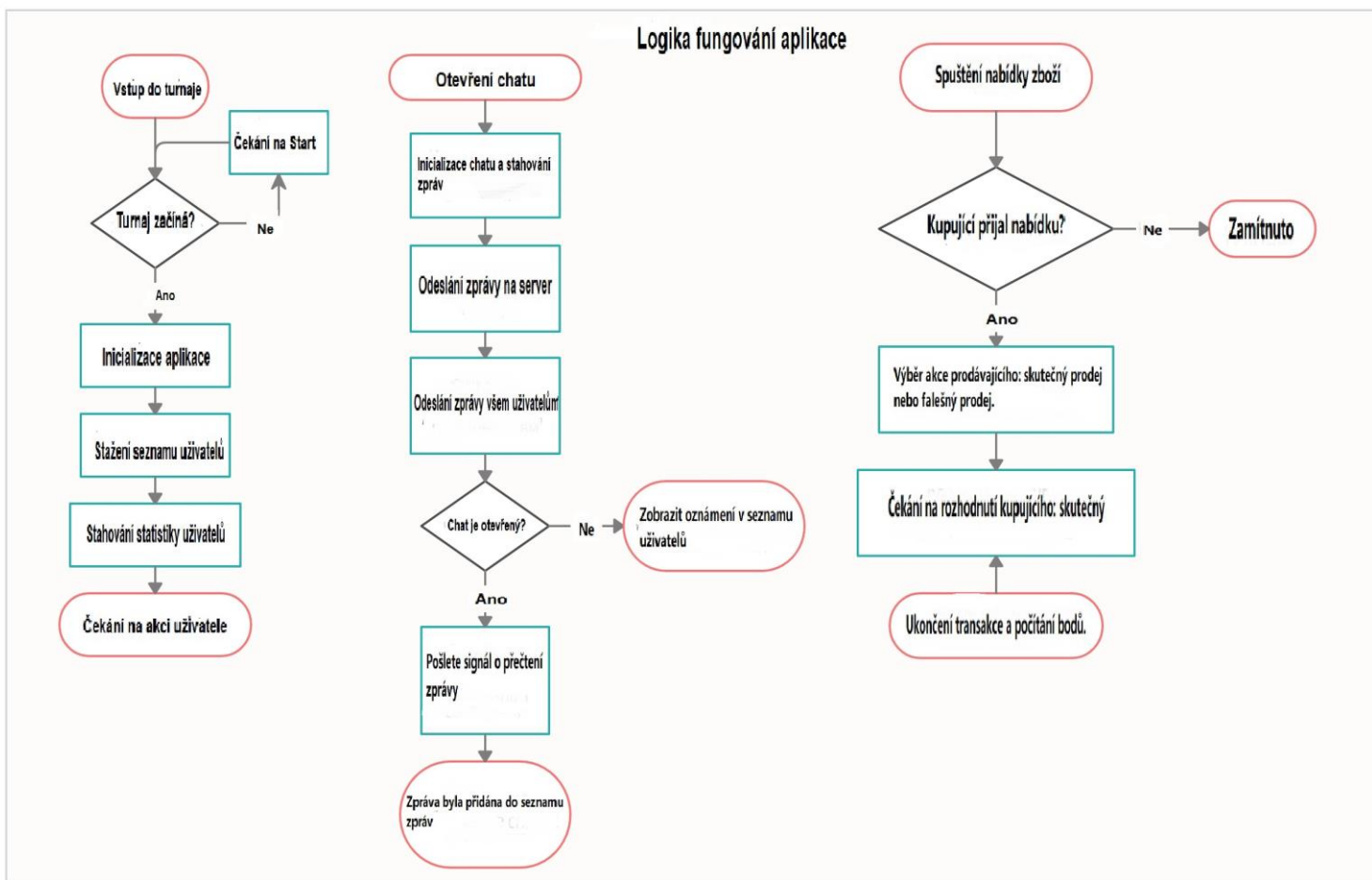
Každý uživatel má možnost sdílet zprávy s ostatními uživateli. Otevřením chatu dostane uživatel možnost zaslat nabídku na prodej zboží. Uživatel, který obdržel prodejní nabídku, má možnost si vybrat, přijmout nabídku nebo odmítnout. Jakmile je transakce přijata, každý uživatel provede akci. Kupující provede skutečný nákup, buď falešný, zatímco prodávající provádí skutečný prodej nebo falešný prodej. Když oba uživatelé zvolili akce, transakce je dokončena. Po dokončení transakce se přepočítávají statistiky transakcí. V případě, že všichni uživatelé provedli transakce, turnaj je dokončen. Po dokončení turnaje je redirekt všech uživatelů na stránku se statistikami.

V tomto případě má každý uživatel v turnaji maximální počet transakcí jak na jednoho uživatele, tak na počet transakcí mezi dvěma uživateli.

V případě, že počet dostupných transakcí u uživatele je u konce, pak uživatel může provádět transakce, ale zároveň má možnost využívat chat, aby se na možnost ovlivnit výběr dalších hráčů. Jak jim pomoci, tak i ublížit. Tato možnost umožňuje, aby do hry přispěl lidský faktor.

UML grafy logiky aplikace turnaje. Turnaj pokračuje, dokud počet transakcí pro všechny uživatele neskončí.

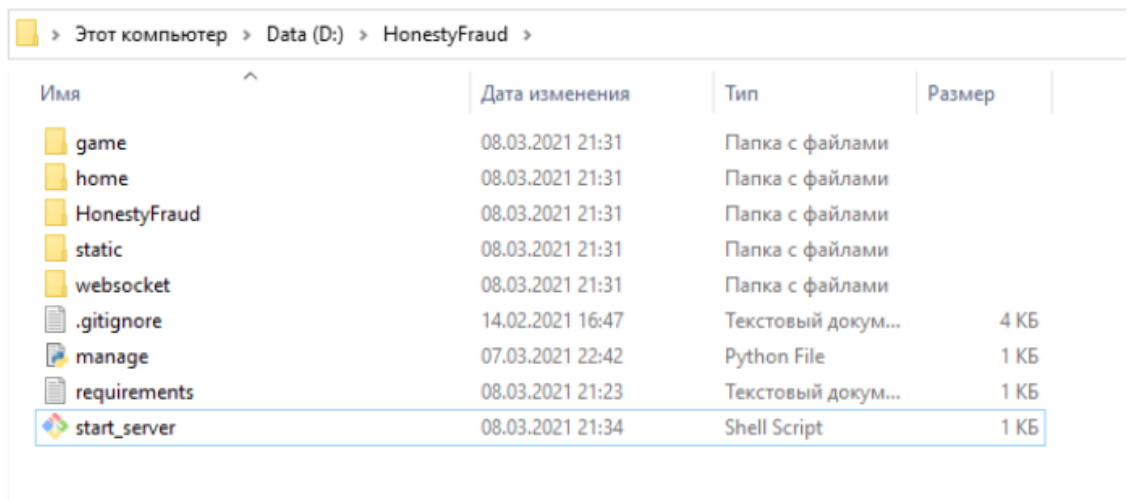
Učitel, stejně jako ostatní uživatelé mohou sledovat celkové statistiky turnaje na stránce Statistika. Aktualizace statistik se také provádí v reálném čase při provádění transakcí mezi uživateli.



Obrázek 13 – Graf UML

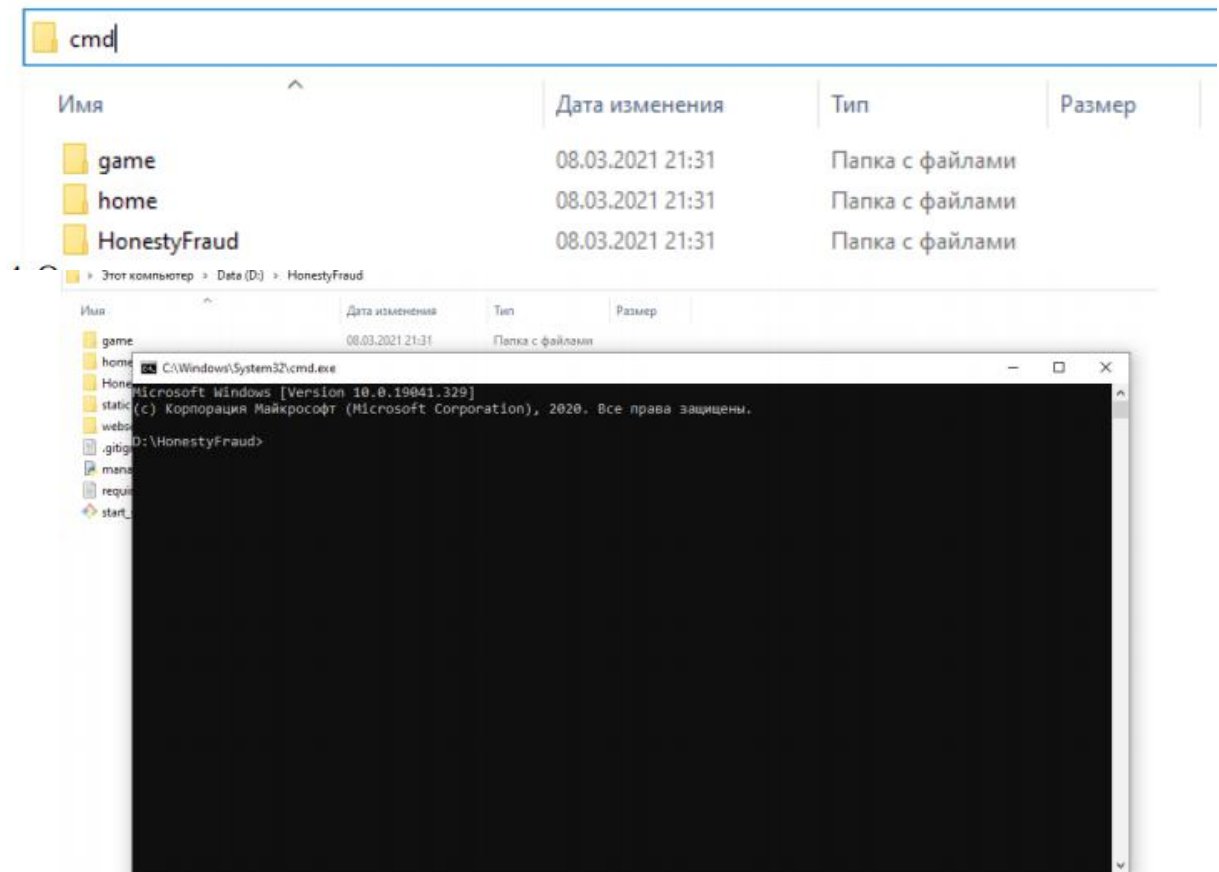
### 4.3 Návod k instalaci hry

1. Nainstalovat Python nejnovější verzi manuálu z následující stránky <https://www.c-sharpcorner.com/article/how-to-install-python-3-8-in-windows/>  
Odkaz na stránku pro stažení Pythonu <https://www.python.org/downloads/>
2. Zkopírujte Projekt do cílového adresáře, například do `D:\HonestyFraud`



Obrázek 14 – Krok 2

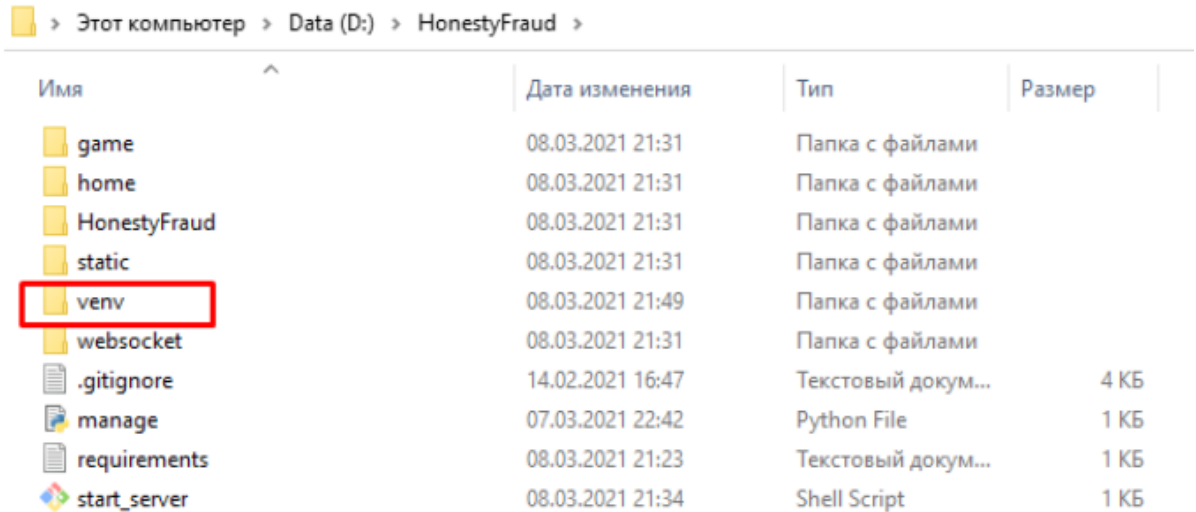
3. Otevřete příkazový řádek, Chcete-li to provést, zadejte do adresního řádku prohlížeče souborů příkaz **cmd**



Obrázek 15 – Krok 3

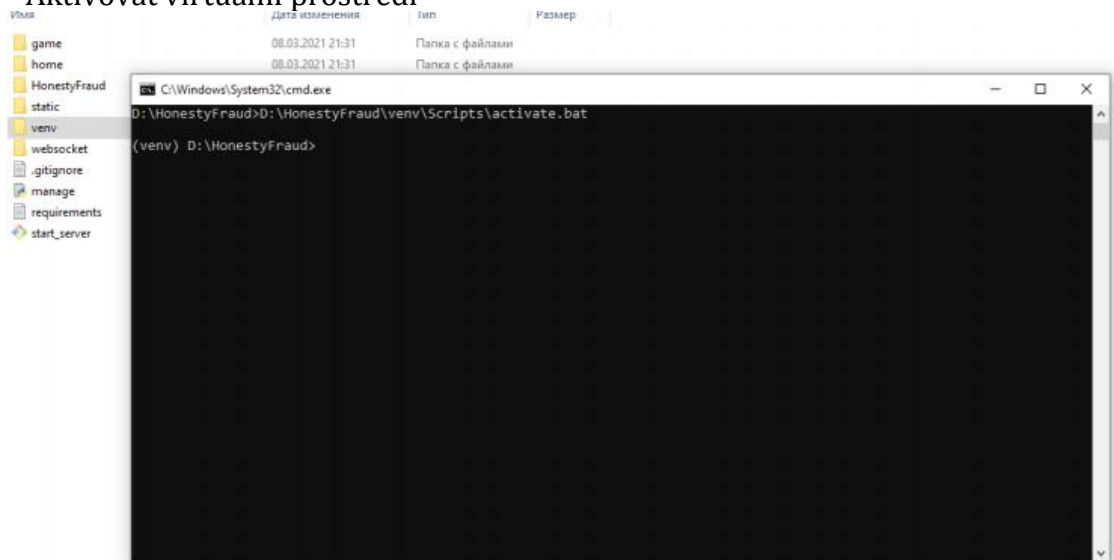
4. Vytvořit virtuální prostředí C pomocí následujícího příkazu  
**“C:\Users\NastassiaMori\AppData\Local\Programs\Python\Python39\python.exe -m venv D:\HonestyFraud\venv**  
 Důležitým bodem je použití správné cesty k souboru **python.exe**

5. Zobrazí se virtuální prostředí Pythonu pro instalaci knihoven



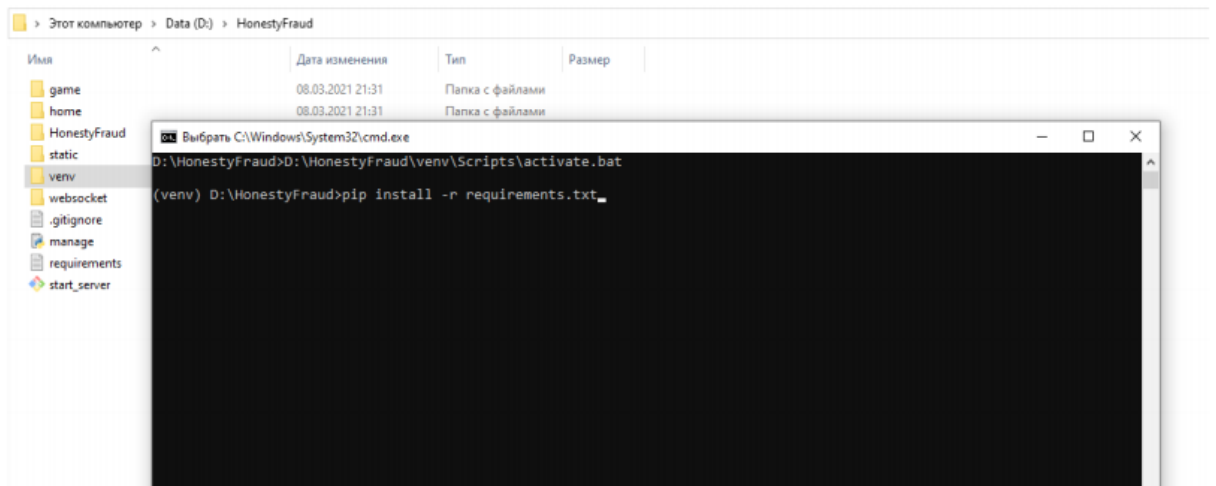
Obrázek 16 – Krok 5

6. Aktivovat virtuální prostředí



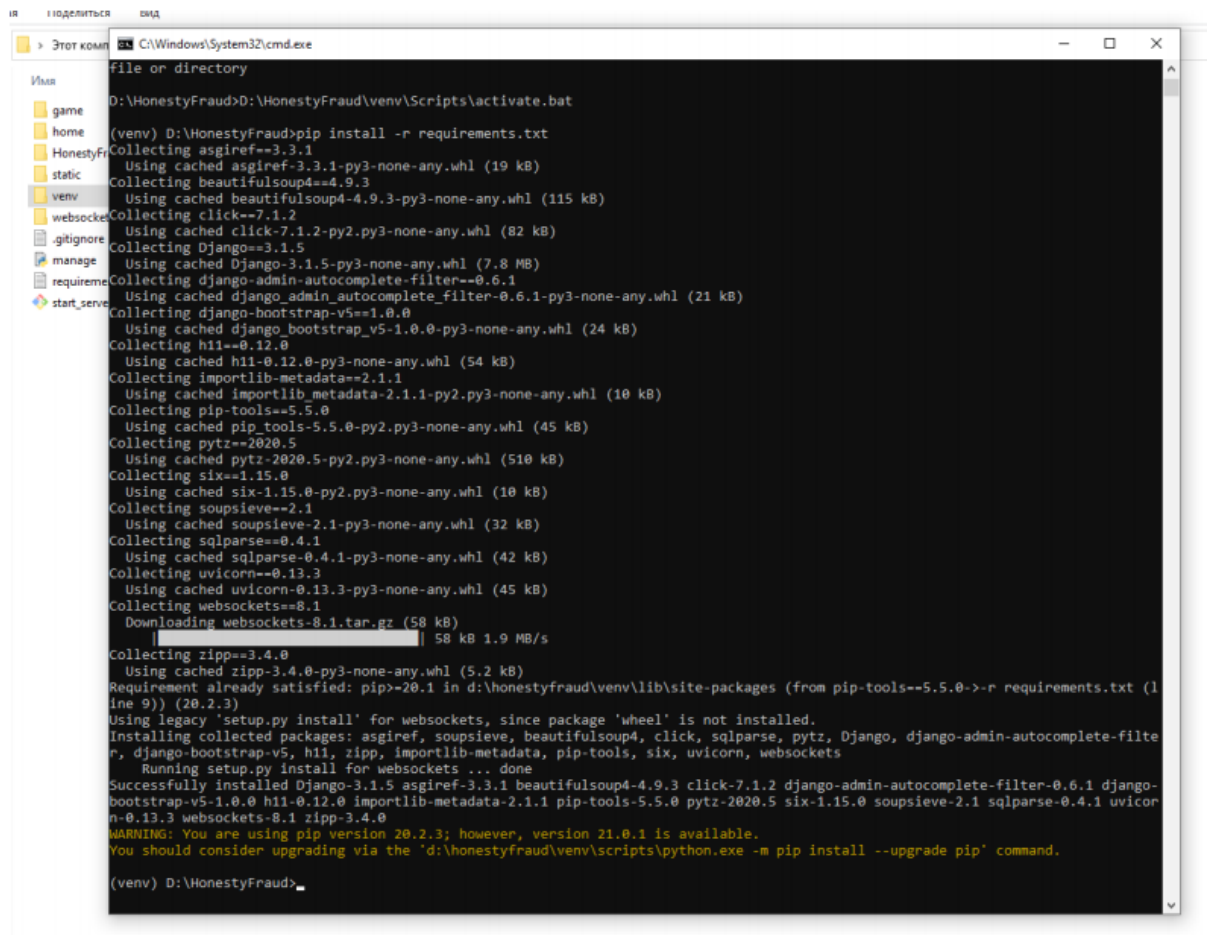
Obrázek 17 – Krok 6

7. Nastavit závislosti pomocí souboru requirements.txt  
Chcete-li to provést, použijte příkaz **pip install -r requirements.txt**



Obrázek 18– Krok 7

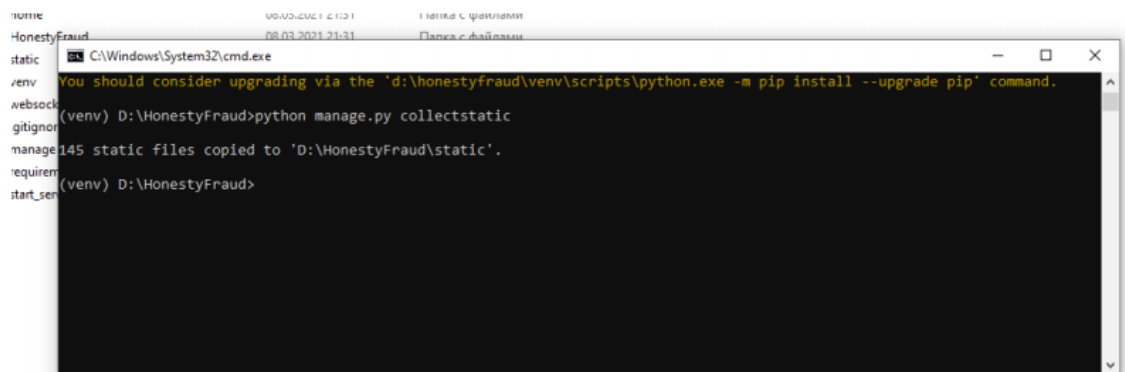
## 8. Výsledek



Obrázek 19 – Krok 8



9. Zkopírovat statické soubory pomocí příkazu **python manage.py collectstatic**

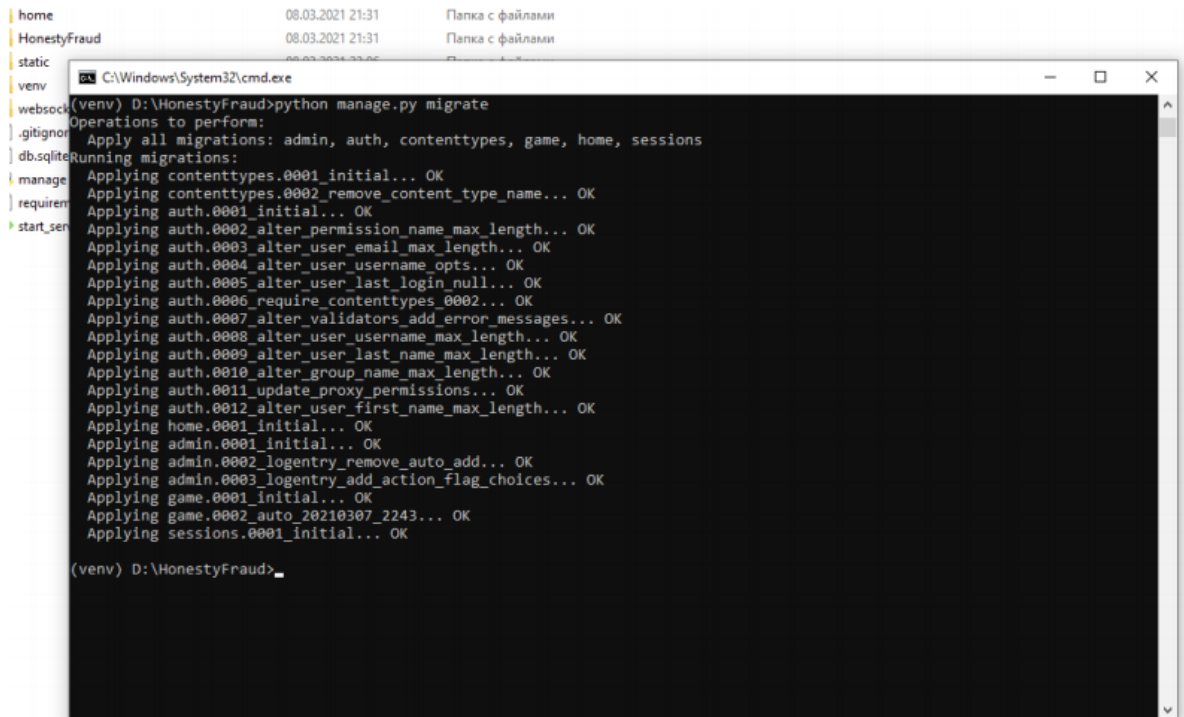


```
home
HonestyFraud
static
venv
websock
.gitignor
manage
requirem
start_ser

C:\Windows\System32\cmd.exe
You should consider upgrading via the 'd:\honestyfraud\venv\scripts\python.exe -m pip install --upgrade pip' command.
(venv) D:\HonestyFraud>python manage.py collectstatic
145 static files copied to 'D:\HonestyFraud\static'.
(venv) D:\HonestyFraud>
```

Obrázek 20 – Krok 9

10. Spustit migraci databáze **python manage.py migrate**

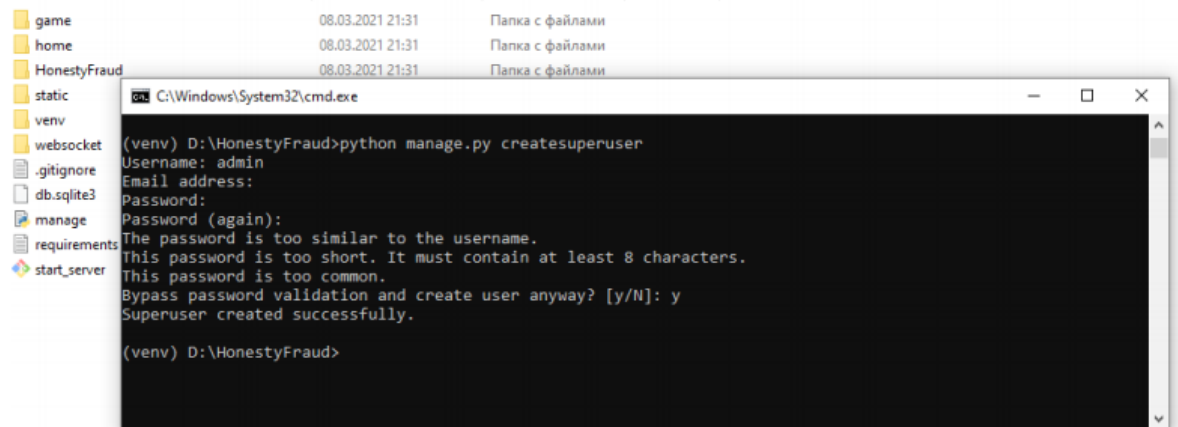


```
home
HonestyFraud
static
venv
websock
.gitignor
manage
requirem
start_ser

C:\Windows\System32\cmd.exe
(venv) D:\HonestyFraud>python manage.py migrate
Operations to perform:
Apply all migrations: admin, auth, contenttypes, game, home, sessions
Running migrations:
Applying contenttypes.0001_initial... OK
Applying contenttypes.0002_remove_content_type_name... OK
Applying auth.0001_initial... OK
Applying auth.0002_alter_permission_name_max_length... OK
Applying auth.0003_alter_user_email_max_length... OK
Applying auth.0004_alter_user_username_opts... OK
Applying auth.0005_alter_user_last_login_null... OK
Applying auth.0006_require_contenttypes_0002... OK
Applying auth.0007_alter_validators_add_error_messages... OK
Applying auth.0008_alter_user_username_max_length... OK
Applying auth.0009_alter_user_last_name_max_length... OK
Applying auth.0010_alter_group_name_max_length... OK
Applying auth.0011_update_proxy_permissions... OK
Applying auth.0012_alter_user_first_name_max_length... OK
Applying home.0001_initial... OK
Applying admin.0001_initial... OK
Applying admin.0002_logentry_remove_auto_add... OK
Applying admin.0003_logentry_add_action_flag_choices... OK
Applying game.0001_initial... OK
Applying game.0002_auto_20210307_2243... OK
Applying sessions.0001_initial... OK
(venv) D:\HonestyFraud>
```

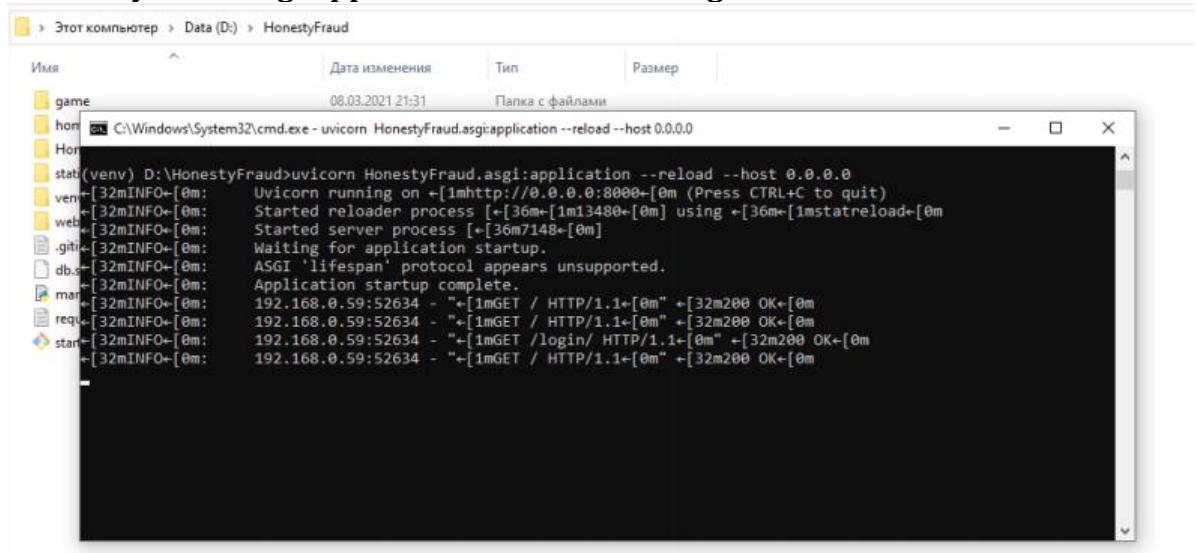
Obrázek 21– Krok 10

11. Vytvořit uživatele root pomocí příkazu **python manage.py createsuperuser**



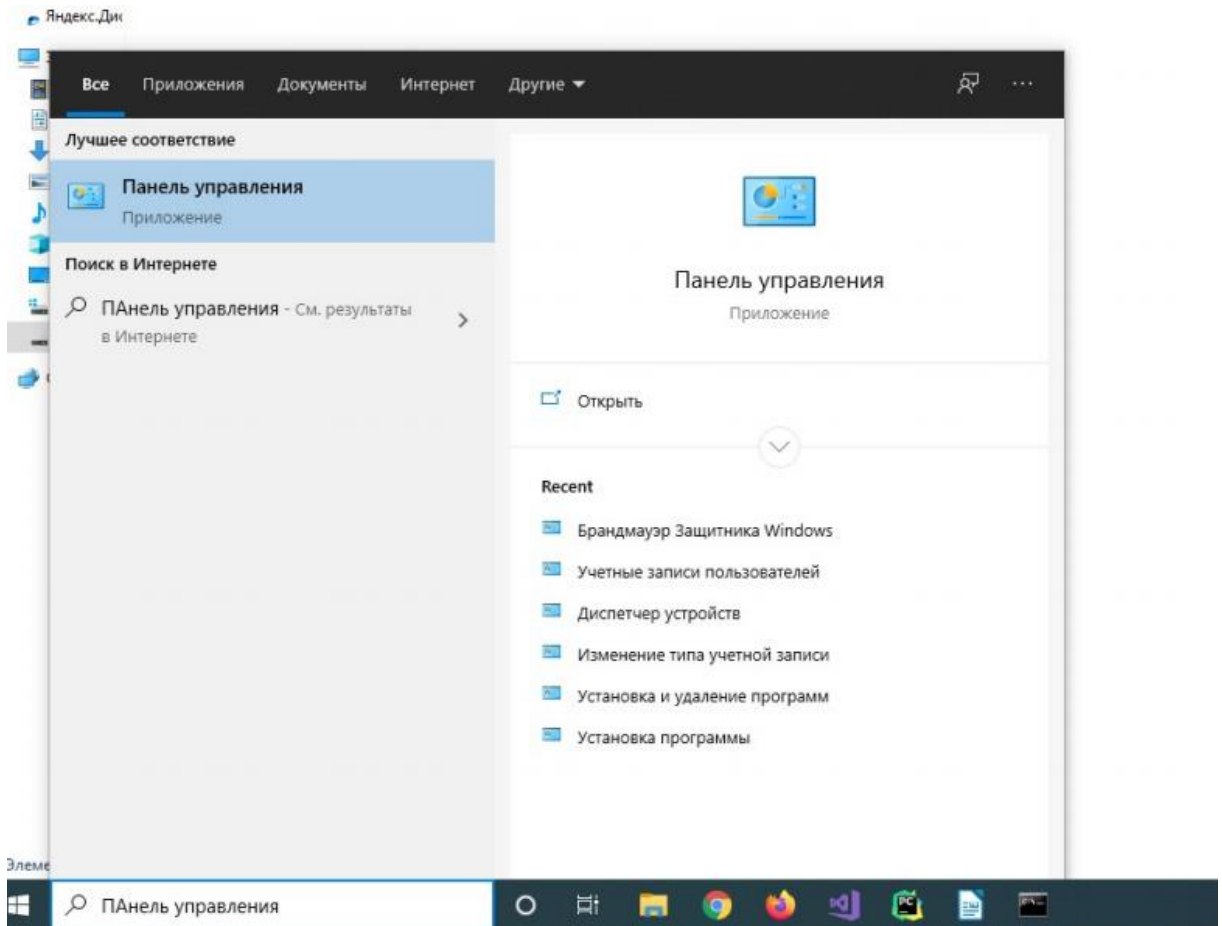
Obrázek 22 – Krok 11

12. Spustíte server následujícím příkazem  
**uvicorn HonestyFraud.asgi:application --reload --debug --host 0.0.0.0**

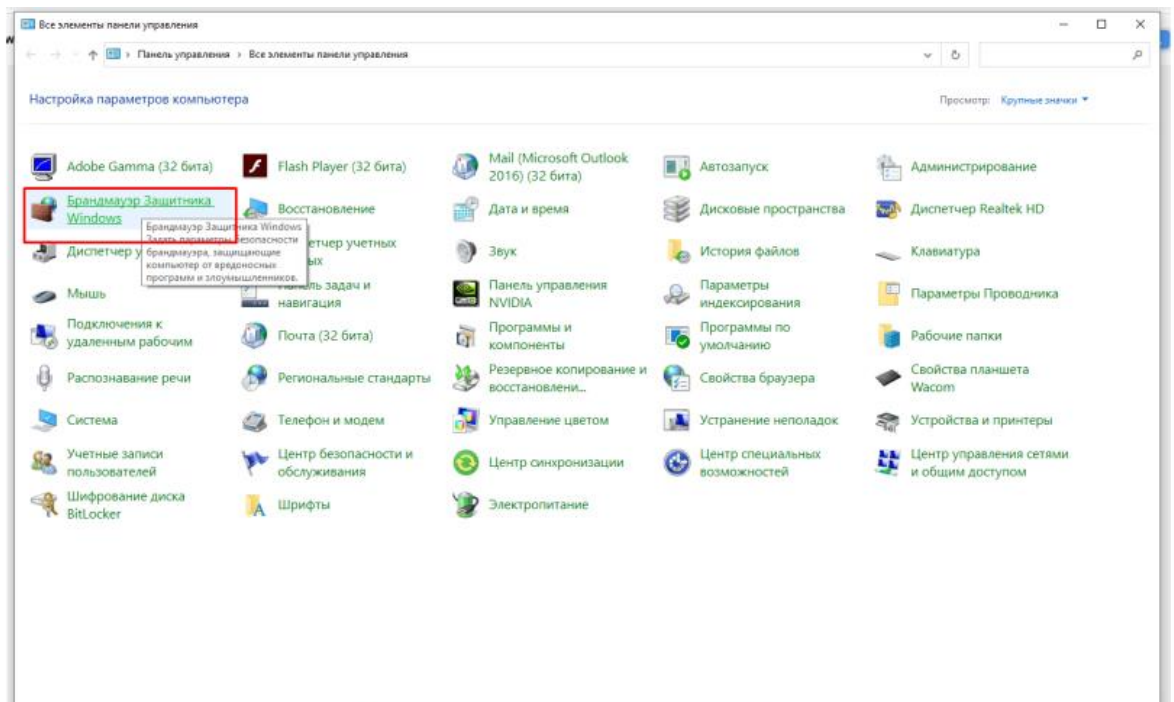


Obrázek 23 – Krok 12

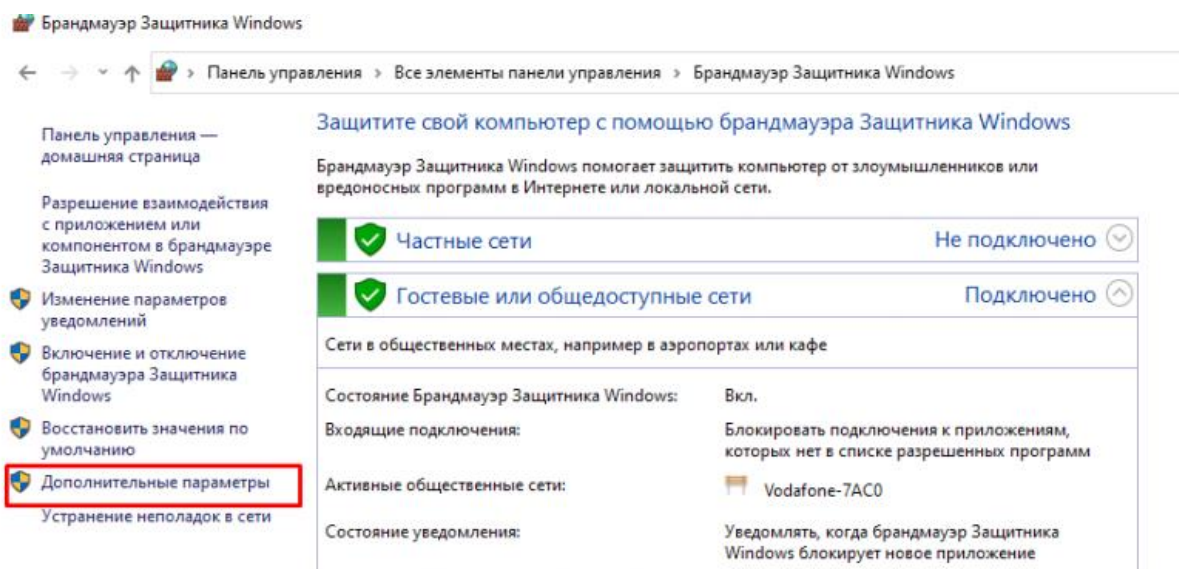
13. Dále musíte otevřít port 8000 v souborovém systému windows, jinak se nemůžete připojit k serveru zvenčí.



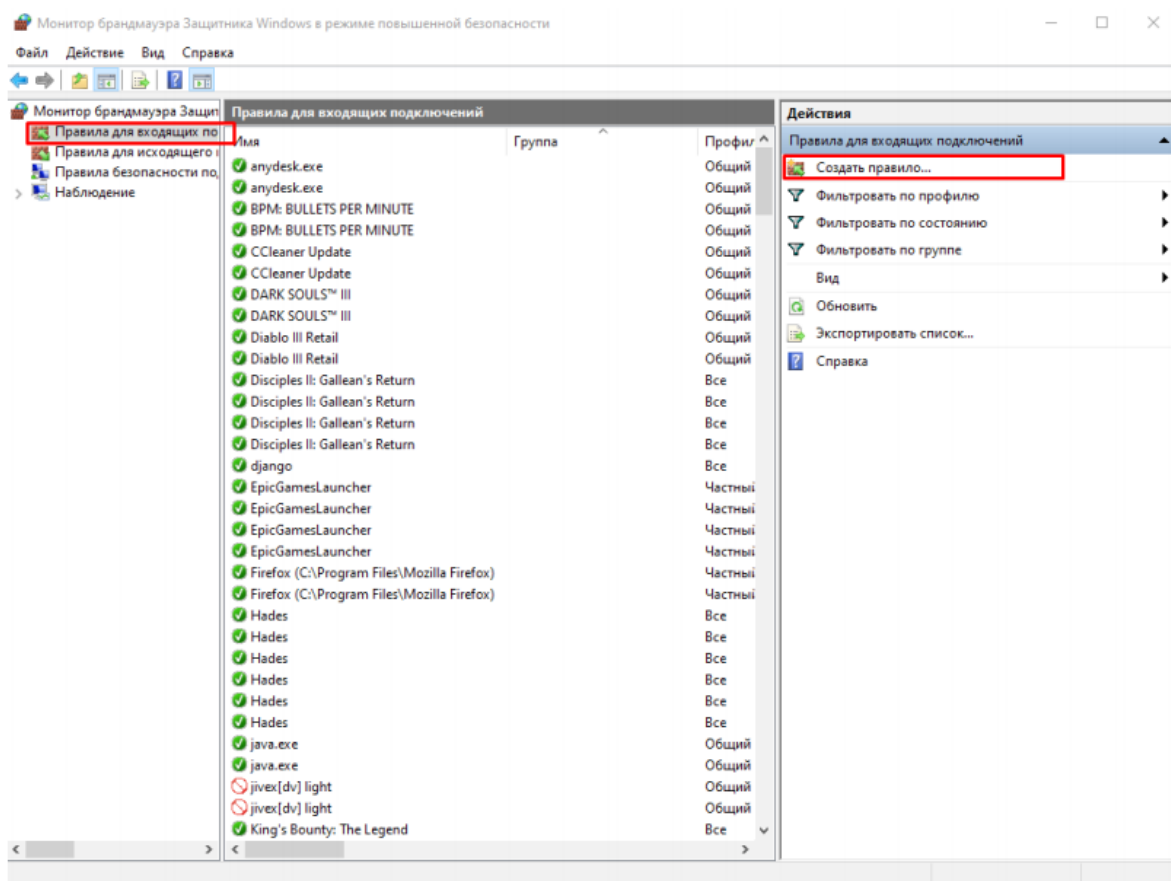
Obrázek 24 – Krok 13



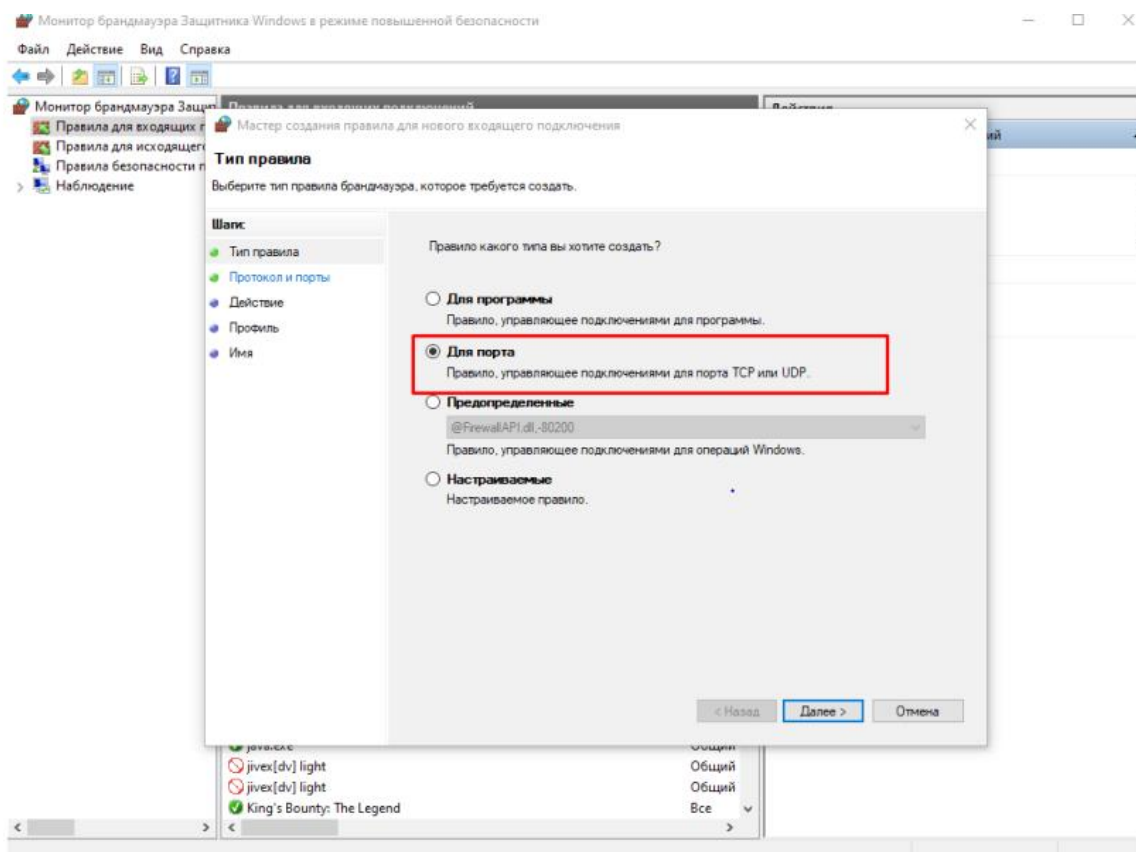
Obrázek 25 – Krok 14



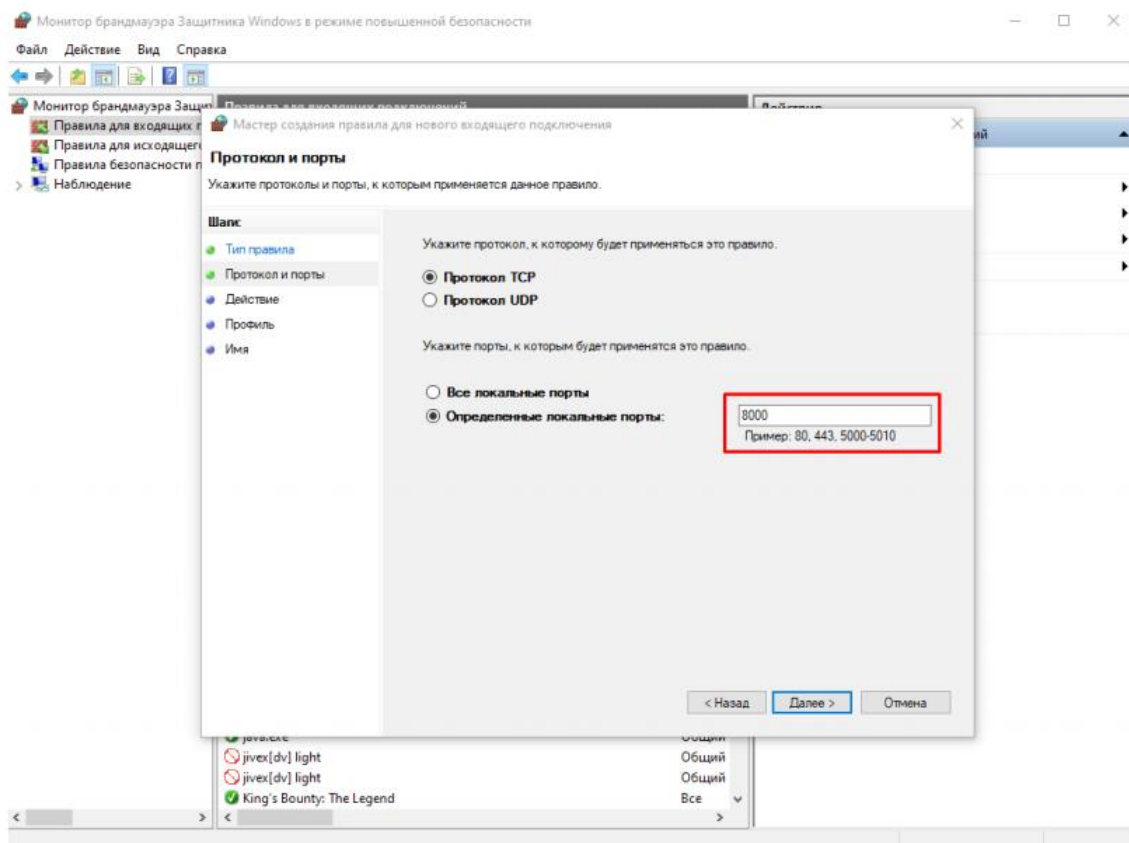
Obrázek 26 – Krok 15



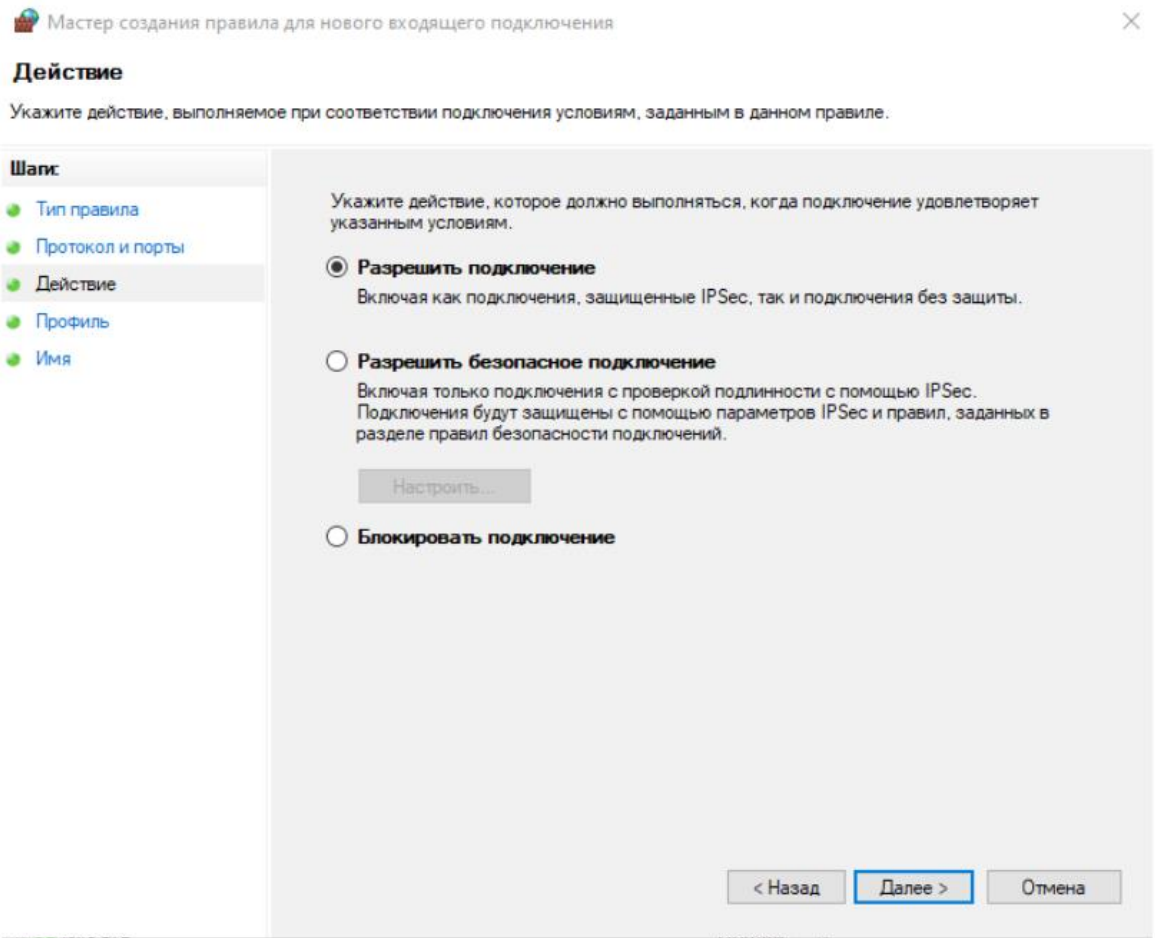
Obrázek 27 – Krok 16



Obrázek 28 – Krok 17



Obrázek 29 – Krok 18



Obrázek 30 – Krok 19

## Профиль

Укажите профили, к которым применяется это правило.

### Шаги

- Тип правила
- Протокол и порты
- Действие
- **Профиль**
- Имя

Для каких профилей применяется правило?

- Доменный**  
Применяется при подключении компьютера к домену своей организации.
- Частный**  
Применяется, когда компьютер подключен к частной сети, например дома или на работе.
- Публичный**  
Применяется при подключении компьютера к общественной сети.

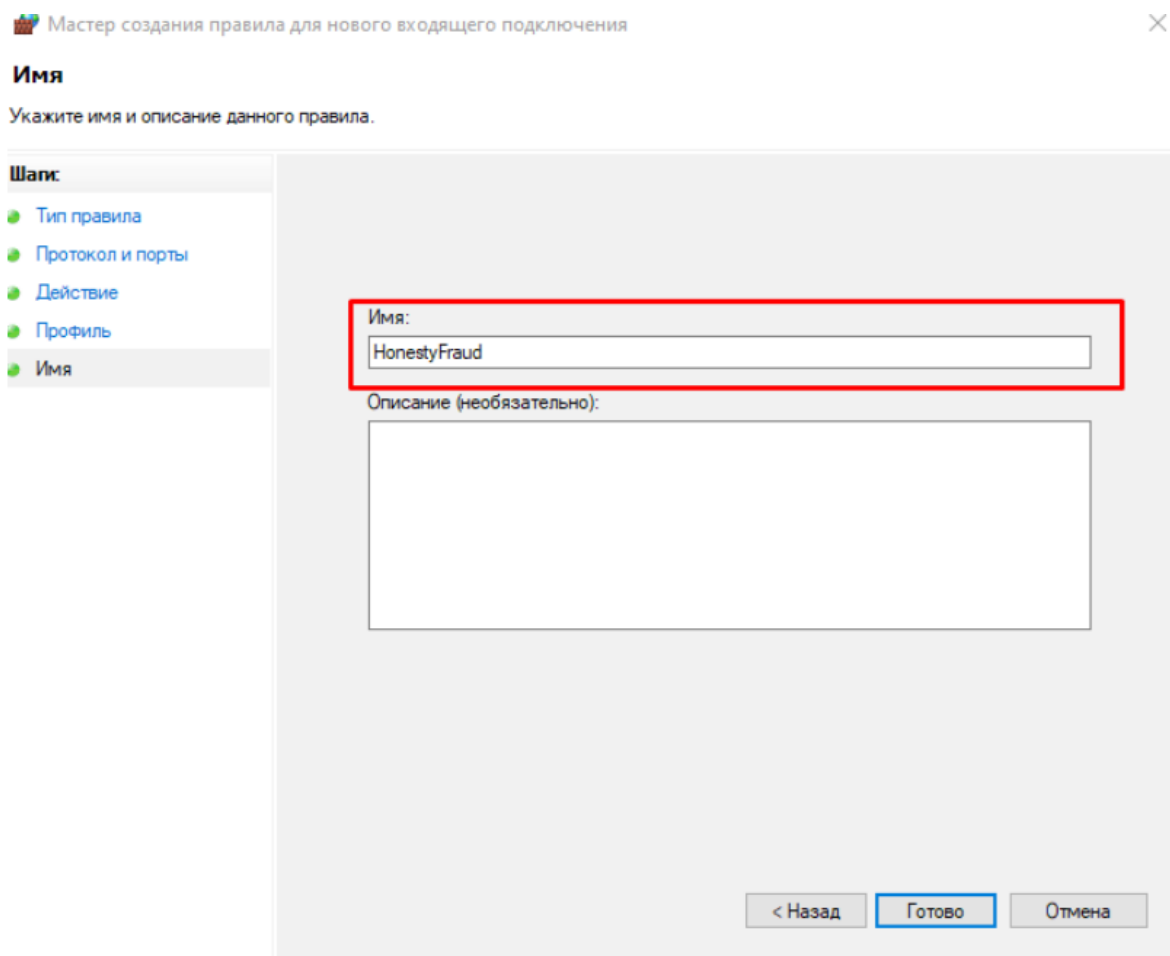
< Назад

Далее >

Отмена

Obrázek 31 – Krok 20





Obrázek 32 – Krok 21

14. Chcete-li vědět, na kterou adresu IP se připojit, musíte zkontrolovat nastavení sítě pomocí příkazu **ipconfig**

```
C:\Windows\System32\cmd.exe

Unknown adapter :

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Připojení k místní síti* 1:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Připojení k místní síti* 2:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Ethernet adapter Ethernet 3:

    Media State . . . . . : Media disconnected
    Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

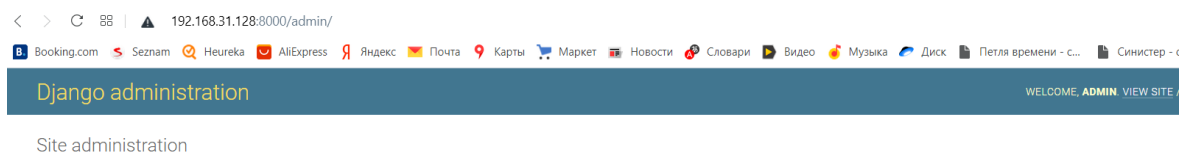
    Connection-specific DNS Suffix . :
    Link-local IPv6 Address . . . . . : fe80::c09d:6a2b:8b55:ff83%25
    IPv4 Address. . . . . : 192.168.31.128
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 192.168.31.1

D:\HonestyFraud 2>
```

Obrázek 33 – Krok 22

Na tomto obrázku IP adresa **192.168.31.128**

## 15. V prohlížeči se připojujeme k webu



Obrázek 34 – Krok 23

## 5 Výsledky a diskuse

### Popis procesu hry

Tato marketingová hra představuje možnost, kdy hráč musí získat nejvýhodnější dohodu pro sebe v podmínkách nákupu nebo prodeje zboží, stejně jako v podmínkách komunikace hráčů mezi sebou.

V tomto případě může hráč provést jak skutečný nákup nebo prodej, tak podvodný nákup nebo prodej, aniž by dal peníze nebo zboží jinému hráči. Hráč, který provádí обманную nákupů či prodeji dostane maximální skóre za transakci, pak tam jsou 2 body, v tom případě, pokud druhý hráč půjde na skutečný nákup a prodej zboží (tento hráč dostane 0 bodů).

The screenshot displays a web interface for managing game transactions and tournaments. On the left is a sidebar with navigation options: GAME (Chats, Messages, Tournaments, Transactions), HOME (Groups, Uživatelé), and buttons to '+ Přidat'. The main content area is titled 'transaction: přidat' and contains a form with the following fields: Tournament (dropdown '1'), Customer (dropdown 'user1'), Seller (dropdown 'user2'), Action of customer (dropdown 'Nevybraný'), Action of seller (dropdown 'Nevybraný'), Points of customer (input '1'), and Points of seller (input '1'). There are checkboxes for 'Is accepted' (unchecked) and 'Is finished' (checked). At the bottom of the form are three buttons: 'Uložit a přidat další položku', 'Uložit a pokračovat v úpravách', and 'ULOŽIT'.

tournament: přidat

Title:

Participants:

Available transactions:

Transaction count btw users:

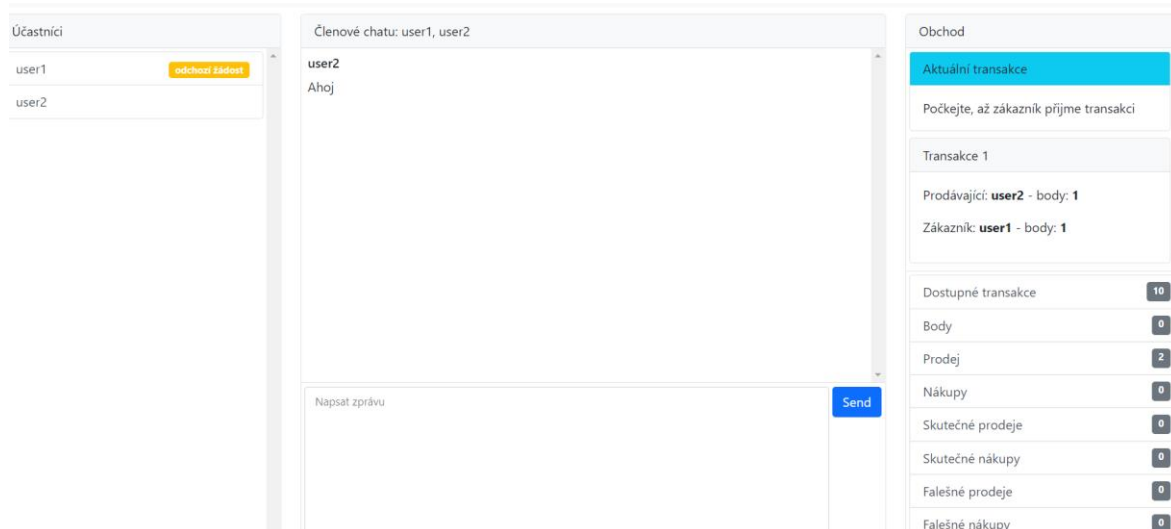
Pub date: -

Start date: -

Obrázek 35 – Administrátor

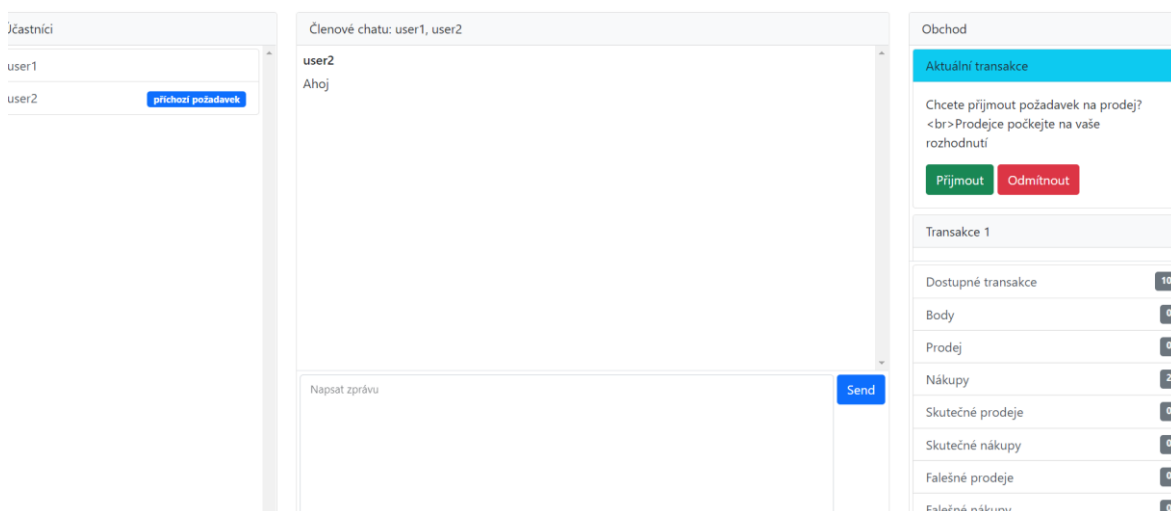
V tomto případě chat mezi hráči bude zjišťovat údaje o ostatních hráčů určit nejpravděpodobnější rozhodnutí jiného hráče, to znamená, že častěji provádí, skutečné akce nebo falešné. V tomto případě, pokud se hráči rozhodnou pro skutečný nákup a skutečný prodej, pak každý hráč dostane 1 bod. V tomto případě obchodování systém je

v rovnováze a na konci hry všichni hráči budou mít stejný počet bodů, pokud nějaký hráč se bude snažit podvádět ostatní hráče, pak to může vyhrát, a začít hrát v turnaji. Právě zde vstupuje do hry lidský faktor a možnosti komunikace mezi hráči prostřednictvím chatu.



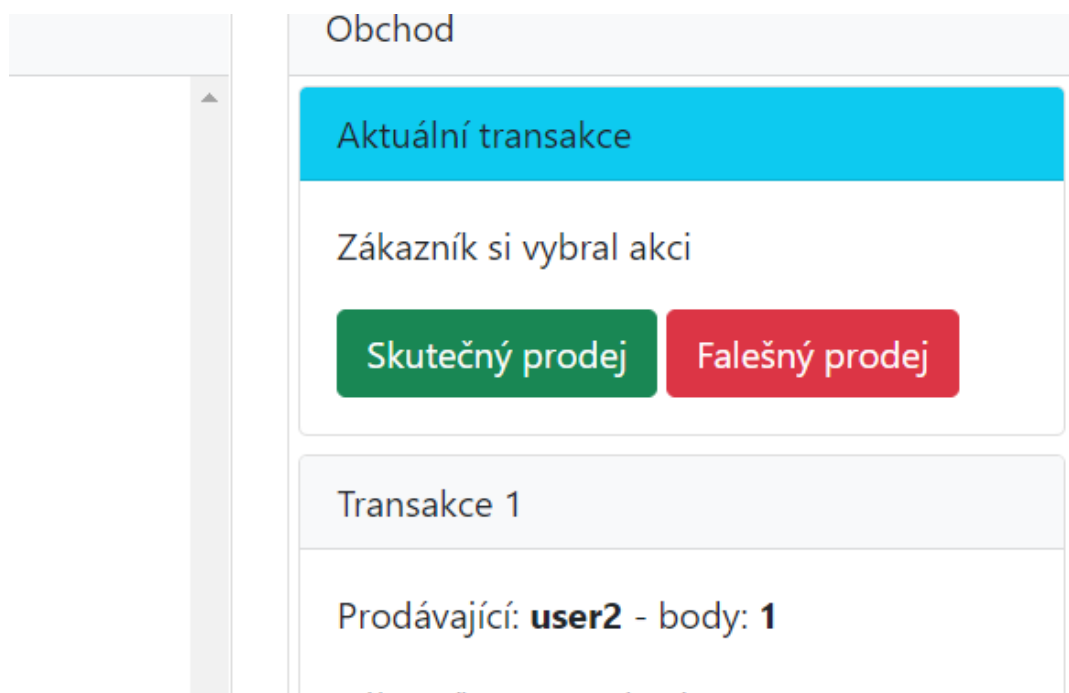
Obrázek 36 – Chat

Pokud některý hráč podvedl jiného hráče, druhý hráč může okamžitě začít informovat ostatní hráče o podvádění ze strany prvního hráče.



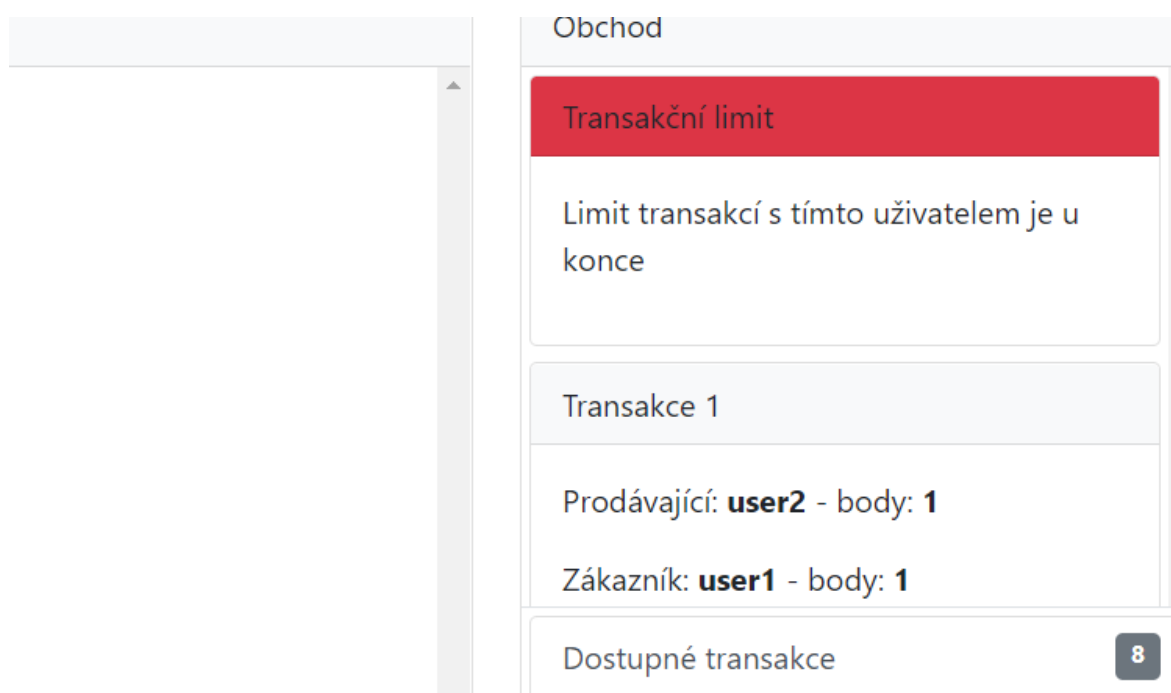
Obrázek 37 – Hra mezi hráči

To může snížit úroveň důvěry v prvního hráče a jeho dočasné vítězství může hrát proti němu. Současně se můžete pokusit manipulovat s ostatními hráči a dát jim nepravdivé informace o tom, jaké akce jeden z hráčů produkuje.



Obrázek 38 – Obchod

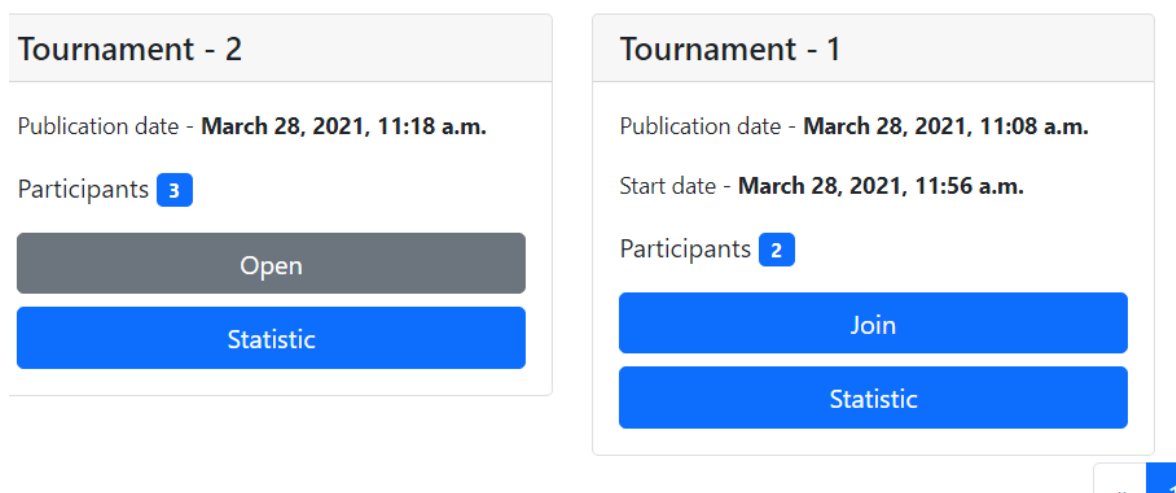
Cílem této hry tedy je ukázat, že v případě poctivých tržních vztahů vyhrává každý z hráčů. A také to, že v případě komunikace mezi hráči, je možné odhalit neopatrný hráč, tak a manipulovat s ostatními hráči, aby se skryli, jak neopatrný hráč.



Obrázek 39 – Transakce

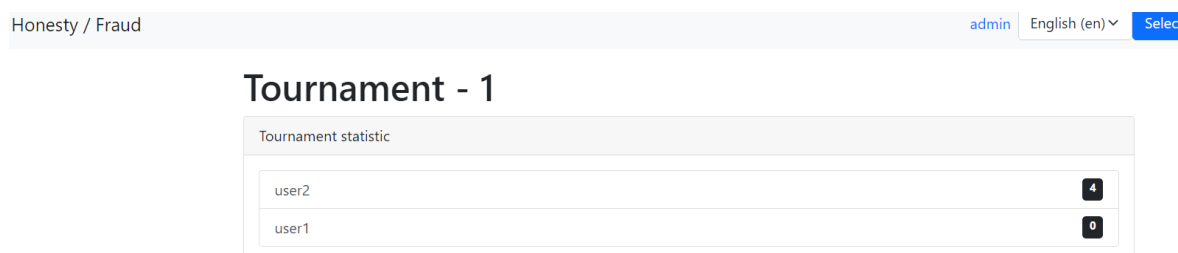
Druhým cílem hry je proto možnost ukázat, jak lidský faktor a komunikace umožňují vyhrávat nebo prohrávat na trhu.

## Tahová hra



Obrázek 40 – Turnaj

K turnaji se připojí hráči a každý hráč se dožije nalezení jiného hráče, který s ním bude chtít obchodovat. To znamená provést prodej nebo nákup zboží. Samotná operace nákupu nebo prodeje není tak důležitá jako akce, která je skutečná nebo podvodná, když hráč neposílá zboží nebo nezaplatí jinému hráči. To určuje výsledek transakce a počet bodů, které každý hráč obdrží.



Obrázek 41 – Body

		Prodejce (Pr)	
		Skutečný prodej	Podvodný prodej
Kupující (Ku)	Skutečný nákup	Pr – 1b, Ku – 1b	Pr – 2b, Ku – 0b
	Podvodný nákup	Pr – 0b, Ku – 2b	Pr – 0b, Ku – 0b

Tabulka - Rozdělení bodů po dokončení transakce

Poté, co hráč našel jiného hráče, který je ochoten provést transakci s ním, tento hráč odešle žádost o prodej druhému hráči zboží. Tato hra neuvádí typ položky, protože není důležitá. Druhý hráč musí přijmout nabídku transakce, nebo odmítnout. Poté, co hráč přijal předběžnou transakci, každý z hráčů si vybere jednu ze dvou Dejvic. Prodávající rozhoduje o skutečném prodeji nebo podvodném prodeji, když druhý hráč nedostane zboží. Zatímco kupující rozhoduje o skutečném nákupu nebo podvodném nákupu, když prodávající nedostane peníze. Jakmile oba hráči zvolili akci, transakce je dokončena a hráči obdrží informaci o tom, kolik bodů získali za transakci. Podle počtu získaných bodů určí, zda jejich druhý hráč podvedl, nebo ne.

Po dokončení transakce může hráč posoudit, zda jeho druhý hráč podváděl nebo hrál upřímně, a tyto informace sdělit jinému hráči nebo je skrýt, aby získal výhodnější podmínky při hraní s ostatními hráči.



Obrázek 42 – Transakce a body

V tomto případě je v turnaji omezení jak na celkový počet transakcí pro jednoho hráče, tak na celkový počet transakcí mezi dvěma hráči najednou. To omezení bude nutit hráče hrát pokud ne se všemi, tak alespoň s velkým počtem hráčů v turnaji, aby se získat maximální množství informací o chování hráčů v turnaji, stejně jako предотвратить тайна догода mezi dvojicí hráčů pro více jedním z hráčů, maximální počet bodů.

Dostupné transakce	8
Body	0
Prodej	1
Nákupy	2
Skutečné prodeje	1
Skutečné nákupy	1
Falešné prodeje	0
Falešné nákupy	0

Obrázek 43 – Co vidí hráči

Turnaj je dokončen poté, co všichni hráči provedli požadovaný počet transakcí. Po skončení turnaje budou hráči přesměrováni na stránku s konečným výsledkem turnaje. Pro více informací se lektor může ukázat, v administrativní panel výsledky транакций těch, nebo jiných hráčů, aby se statický portrét hráče, a také se může ptát hráčů o tom, jak se tyto nebo jiné rozhodnutí o nákupu nebo prodeji. A také o tom, zda jejich rozhodnutí ovlivnila informace, kterou dostali od jiných hráčů.



## 6 Závěr

Tato hra vám umožní ukázat tržní vztahy v podmínkách přítomnosti lidského faktoru a potenciální beztretnost hráče na trhu, za předpokladu, že regulace hráče akce pouze mírou důvěry k němu další hráči.

Při regulaci důvěry hráč v turnaji nemusí vůbec získat žádné herní body, pokud ostatní hráči ztratí veškerou důvěru v něj. To bude způsobeno tím, že ostatní hráči mohou začít používat pouze podvodné akce ve vztahu k tomuto hráči. V důsledku toho, bez ohledu na to, co hráč zvolí, bude vždy dostávat 0 bodů. Hráč tak bude potrestán samotným trhem.

Ve stejném případě, pokud se všichni hráči budou držet Fair Play, trh bude v rovnováze a výsledek turnaje by měl být stejný počet bodů u všech hráčů.

Pokud hráči uplatní podvodné akce v rozumných mezích a také si zvolí nějaké spolehlivé partnery, může se objevit tendence k většímu růstu skóre u části hráčů. V tomto případě je úspěch hráče může být обусловлен původní důvěrou k němu mají ostatní hráči, když se tento hráč rozhodl pouze poctivé akční a na konci hry prudký nárůst bodů v případě, že strategie byla změněna. Nesprávná změna strategie však může vést k zastavení růstu skóre hráčů.

Skutečná úspěšnost hráčů bude hodnocena na konci turnaje na základě akcí těchto hráčů, stejně jako popis strategie, kterou tyto hráči sami vybrali.

Cílem hry je také zjistit vliv na činnost hráčů přítomnost informací o tom, jak se ten či onen hráč chová. Tyto informace mohou hráči získávat od ostatních hráčů, ale samozřejmě, že tato информация může být buď nepřesné, nebo nepravdivé, že hráči by měli také vzít v úvahu při provádění transakcí s ostatními hráči.

## 7 Seznam použitých zdrojů

1. Jennifer Niederst Robbins Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics. - 4th Edition O'Reilly Media 2012.
2. Mark Lutz: Learning Python. - 4th Edition O'Reilly & Associates, 2009
3. Ustin Seitz Black Hat Python: Python Programming for Hackers and Pentesters. - 1st Edition No Starch Press, 2014.
4. Příručka k HTML a CSS. [broe.hyperlink.cz](http://broe.hyperlink.cz) [online]. [cit. 2010-01-06].
5. Bootstrap tutorial [online]. W3school [cit. 2014-11-11]. Dostupné online.
6. Bootstrap [online]. Bootstrap [cit. 2013-05-30]. Dostupné online. HAIS, Karel; HODEK, Břetislav. Velký anglicko-český slovník. I., A–M. V nakl. Leda 1. vyd., v nakl. Academia 3., přeprac. vyd. Voznice: Leda; Praha : Academia, 1997. 1504 s. ISBN 80-85927-35-7, ISBN 80-200-0673-7.
7. Bootstrap in A List Apart #342 [online]. Mark Otto, 2012-01-17 [cit. 2012-03-05].
8. Standard ECMA-262 [online]. Ecma International, 2017-07-03. Dostupné online. (anglicky).
9. JELÍNEK, Lukáš. Asm.js zrychluje. [www.linuxexpres.cz](http://www.linuxexpres.cz) [online]. [linuxexpres.cz](http://linuxexpres.cz), 2013-12-23 [cit. 2018-05-18]. Dostupné online.
10. URBAN, Petr. Epic Games ve spolupráci s Mozillou v prohlížeči zprovoznil Unreal Engine 4. [Cnews.cz](http://cnews.cz) [online]. 2014-03-13 [cit. 2018-05-18]. Dostupné online.
11. Version Information (JavaScript) [online]. Msdn.microsoft.com [cit. 2013-05-26]. Dostupné online.
12. CHAFFER, Jonathon; SWEDBERG, Karl. Learning JQuery: Better Interaction Design and Web Development with Simple JavaScript Techniques. [s.l.]: Packt Publishing, 2007. ISBN 978-1847192509. (anglicky)
13. BIBLEAULT, Bear; KATZ, Yehuda. JQuery in Action. [s.l.]: Manning Publications Co., 2008. ISBN 978-1933988351. (anglicky)
14. HEILMANN, Christian. Beginning JavaScript with DOM scripting and Ajax. [s.l.]: Apress, 2006. Dostupné online. ISBN 978-1590596807. (anglicky)
15. DARIE, Cristian; CHERECHES-TOSA, Filip; BUCICIA, Mihai. Ajax And Php: Building Responsive Web Applications. [s.l.]: Packt Publishing, 2005. ISBN 978-1904811824. (anglicky)
16. HEILMANN, Christian; FRANCIS, Mark Norman. Web development solutions. [s.l.]: Apress, 2007. Dostupné online. ISBN 978-1590598061. (anglicky)
17. ZIMOLA, Bedřich. Operační výzkum [online]. Zlín: Vysoké učení technické v Brně. Fakulta managementu a ekonomiky ve Zlíně, 2000 [cit. 2017-09-03]. Kapitola Operační výzkum jako nástroj řízení, s. 19. Dostupné online. ISBN 80-214-1664-5.
18. Zadie Smith, Swing Time (New York: Penguin Press, 2016), 315–16.
19. Grazer, Brian, and Charles Fishman. 2015. A Curious Mind: The Secret to a Bigger Life. New York: Simon & Schuster.
20. Kurland, Philip B., and Ralph Lerner, eds. 1987. The Founders' Constitution. Chicago: University of Chicago Press. <http://press-pubs.uchicago.edu/founders/>.