

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SIMULÁTOR DOPRAVNEJ FIRMY
Bakalárska práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SIMULÁTOR DOPRAVNEJ FIRMY

Bakalárska práca

Študijný program: Aplikovaná informatika
Študijný odbor: 2511 Aplikovaná informatika
Školiace pracovisko: Katedra didaktiky matematiky, fyziky a informatiky
Školiteľ: PaedDr. Daniela Bezáková, PhD.

Čestné prehlásenie

Čestne prehlasujem, že som bakalársku prácu vypracoval samostatne pod vedením školiteľa a s použitím uvedenej literatúry.

V Bratislave, X.Y. 2019

Meno autora

Pod'akovanie

Ďakujem ...

Abstrakt

slovensky

Abstract

anglicky

Obsah

| | |
|---|-----------|
| ÚVOD | 8 |
| 1 VÝCHODISKÁ | 9 |
| 1.1 PREHĽAD TEÓRIE | 9 |
| 1.1.1 <i>Simulácia</i> | 9 |
| 1.1.2 <i>Doprava</i> | 9 |
| 1.1.3 <i>Dopravná firma</i> | 10 |
| 1.1.4 <i>MVC architektúra</i> | 10 |
| 1.1.5 <i>Active record</i> | 10 |
| 1.2 EXISTUJÚCE SOFTVÉROVÉ RIEŠENIA | 11 |
| 1.2.1 <i>Euro Truck Simulator 2</i> | 11 |
| 1.2.2 <i>TransOcean: The Shipping Company</i> | 13 |
| 1.3 TECHNOLOGIE | 15 |
| 1.3.1 <i>SASS</i> | 15 |
| 1.3.2 <i>Bootstrap</i> | 15 |
| 1.3.3 <i>Vue</i> | 15 |
| 1.3.4 <i>Bootstrap Vue</i> | 15 |
| 1.3.5 <i>Laravel</i> | 16 |
| 2 NÁVRH | 19 |
| POUŽITÁ LITERATÚRA | 20 |

Úvod

1 Východiská

V tejto kapitole vysvetlím teoretické pojmy súvisiace s mojou prácou, predstavím existujúce softvérové riešenia, ich výhody, prípadne nevýhody a popíšem technológie, ktoré využijem pri vývoji svojej aplikácie.

1.1 Prehľad teórie

1.1.1 Simulácia

Počítačová simulácia je softvérová imitácia nejakého reálneho systému alebo procesu vytvorená za účelom skúmania tohto systému, jeho vlastností a fungovania. Základom simulácie je vytvorenie modelu, ktorý reprezentuje kľúčové vlastnosti a správanie simulovaného systému. [1]

Simulačné hry sú také simulácie, do ktorých sa zapájajú viacerí „hráči“ (používatelia), pričom niektorých z nich môže zastupovať počítač. Hráči buď navzájom spolupracujú za účelom vyriešenia nejakého problému, alebo naopak súťažia. [2]

Moja hra bude len čiastočnou simuláciou dopravnej firmy. Vlastnosti, ktoré budú zodpovedať realite sú reálne značky kamiónov, skutočné názvy miest a štátov. Simulácia nebude prebiehať v reálnom čase, nakoľko cesta medzi dvoma mestami v realite trvá niekoľko hodín, čo by hráča nudilo. Hráči budú vystupovať v úlohe šéfa firmy, resp. jej zamestnancov. Rolu vodičov firmy prevezme počítač.

1.1.2 Doprava

Úlohou dopravy je zabezpečiť fyzické premiestnenie osôb a vecí v čase a priestore. Samotné premiestnenie z určitého miesta na iné miesto v určitom čase je produkt dopravy, čo môžeme nazvať tiež preprava. Preprava nemá hmotnú formu, nedá sa skladovať a je hneď spotrebovaná. Dopravu je nutne riadiť kvôli optimálnej del'be práce, optimálnej kvalite prepravy a minimalizácii nákladov na dopravu. Preprava musí byť dohodnutá so zákazníkom, ktorému musí byť umožnené získať informácie o preprave svojej zásielky.

1.1.3 Dopravná firma

Dopravná firma je spoločnosť, ktorá je spojená so všetkými službami, ktoré sa vzťahujú k preprave zásielok. Cieľom firmy je komplexné uspokojenie špecifických potrieb zákazníka.

Vo svojej bakalárskej práci budem pod pojmom dopravná firma (ďalej len *moja dopravná firma*, resp. *firma*) rozumieť takú firmu, ktorá sa zaoberá len prepravou tovaru, a to pomocou kamiónov, nič nevyrába ani neskladuje. Moja dopravná firma rieši zákazky na prepravu tovaru z jednej klientskej firmy do druhej. V každej zákazke sa nachádza iba jeden druh tovaru v prívese, ktorý vlastní jedna z klientských firiem. Moja dopravná firma vlastní garáže v rôznych mestách, pričom v každom meste môže byť len jedna garáž. Taktiež vlastní kamióny, pomocou ktorých prepravuje zásielky. Zisk firmy tvoria výlučne peniaze zarobené za dokončené zásielky, na ničom inom firma neprofituje. Náklady firmy tvoria platy všetkých zamestnancov, náklady na údržbu a chod kamiónov, kúpa nových garáží a kamiónov. Firmu tvoria zamestnanci troch typov. Prvým je šéf alebo riaditeľ, ktorý danú firmu vlastní, má všetky právomoci ako manažér. Šéf môže najímať ďalších manažérov. Manažéri vo firme prijímajú objednávky a pridelujú im vodičov a kamióny. Tiež môžu nakupovať garáže, kamióny a najímať vodičov. Posledným typom zamestnancov sú vodiči, ktorí prepravujú zásielky v kamiónoch.

1.1.4 MVC architektúra

Model-View-Controller je jeden z najpoužívanejších architektonických návrhových vzorov. Rozdeľuje aplikáciu na tri prepojené časti. Model reprezentuje štruktúru dát a informácií, s ktorými aplikácia pracuje. View zobrazuje dáta, ktoré sú v modeli, v podobe vhodnej pre používateľa. Controller riadi a reaguje na vstupy od používateľa a premieňa ich na príkazy, s ktorými vie pracovať model alebo view. [3]

1.1.5 Active record

Active record je architektonický návrhový vzor na prácu s dátami v relačných databázach.

Pre každý riadok v databáze je vytvorený objekt, ktorý obsahuje dáta z daného riadku a zodpovedá za základné operácie ako upravenie, vymazanie a čítanie. [4]

1.2 Existujúce softvérové riešenia

V tejto časti popisujem dve počítačové hry, ktorých základom je preprava tovaru. Obsahujú tiež istú formu riadenia vlastnej firmy, jej majetku a taktiež ziskov a nákladov. Je na hráčovi, akú stratégiu zvolí, aby sa stal úspešným.

Oba uvedené softvérové produkty sú platené a je potrebné ich nainštalovať na počítač, ktorý spĺňa určité minimálne požiadavky, čo môže byť pre potenciálnych záujemcov odradzujúce.

1.2.1 Euro Truck Simulator 2

Euro Truck Simulator 2 je počítačová hra, v ktorej je hráč v roli vodiča kamiónu. Jeho úlohou je prevážať tovar v návesoch z jednej firmy do inej. Na začiatku hráč vlastní iba malú garáž pre budúce kamióny. Prijíma zákazky, v ktorých je ponúknutý aj samotný kamión, nakoľko zatiaľ žiaden nevlastní. Hráč na svojich cestách stretáva iné autá a kamióny, ktoré sú riadené priamo hrou. Na cestách platia pravidlá ako v normálnej cestnej premávke, ktorými sa riadia tieto vozidlá. Samozrejmosťou sú aj udalosti počas cesty, napríklad nehody, ktoré môžu poškodiť náklad ako aj samotný kamión. Kamión je možné opraviť, no náklad sa opraviť nedá, a teda odmena bude menšia.



Obrázok 1– pohľad z kabíny kamiónu počas jazdy

Postupne si hráč môže rozširovať svoju firmu o ďalšie kamióny, ktoré sú realistické modely siedmych značiek. Každý kamión je možné upravovať, od vzhľadu kabíny, podvozku a kolies, cez výkon motora až po malé detaily ako sú svetlá alebo zrkadlá, až kým nie je kamión podľa predstáv hráča. Za každú zásielku hráč dostane určité skúsenosti, za ktoré dostáva body na získanie lepších zručností. Lepšie zručnosti zaručia nové typy zákaziek, pretože spoločnosti budú hráčovi viac veriť, čím dostane aj vyššiu odmenu. Zakúpením prvého kamiónu hra samozrejme nekončí, práve naopak, začína. Najímaním nových vodičov a kupovaním ďalších garáží sa hráčovi rozrastá jeho vlastná dopravná firma. Najatí vodiči pracujú samostatne, taktiež zákazky majú pridelené automaticky, takže hráč sa môže sústrediť na svoje jazdy.



Obrázok 2 – pohľad na jednu z garáží s jej profitom, vodičmi a kamiónmi

Táto hra ma najviac inšpirovala pri tvorbe mojej bakalárskej práce. V mojej aplikácii sa však nebudem zameriavať na úlohu hráča ako vodiča, ale skôr na úlohu hráča ako vlastníka dopravnej firmy, ktorý bude rozhodovať o zákazkách, garážach, kamiónoch a vodičoch s cieľom čo najlepšie riadiť svoju firmu.

1.2.2 TransOcean: The Shipping Company

TransOcean: The Shipping Company je počítačová hra, v ktorej hlavnou úlohou je riadenie svojich nákladných lodí, ktoré sa plavia s nákladom po celom svete. Obsahuje viac ako 50 autentických, rôzne veľkých prístavov, ako aj realistické lode a trasy, po ktorých sa plavia. V reálnom čase je možné sledovať plavbu lodí, pričom čas je možné zrýchliť, aby sa hráč nenudil, keďže počas plavby medzi prístavmi je loď riadená automaticky hrou. Hráč si vyberá zákazku, v ktorej je určité množstvo nákladu, ktorý treba prepraviť medzi dvomi prístavmi. Následne vyberie jednu alebo viac lodí, ktoré sú vhodné na prepravu takéhoto nákladu. Počas plavby vznikajú rôzne udalosti, s ktorými sa musí hráč vysporiadať ako tajfúny, pirátske nájazdy alebo colné kontroly. Pri príchode do prístavu má hráč možnosť minihry, pri ktorej sa stane kapitánom danej lode a musí ju čo najlepšie zaparkovať v prístave alebo môže zaplatiť za zaparkovanie. Loď v prístave môže vylepšovať, čo môže

trvá určitý čas a v prístavoch je státie spoplatnené. Treba udržať rovnováhu medzi výdavkami a ziskami a nájsť najlepšiu stratégiu na rozširovanie svojej firmy.

Možnosť voľby zákaziek pre všetky lode (v mojom prípade kamióňy) je jedna z vlastností, ktoré by som chcel mať aj v bakalárskej práci.



Obrázok 3 – pohľad na výber zákazky z jedného prístavu

1.3 Technológie

V tejto časti popisujem technológie, ktoré využívam vo svojej bakalárskej práci.

1.3.1 SASS

SASS je skriptovací jazyk, ktorý je interpretovaný alebo kompilovaný do kaskádových štýlov (CSS). SASS rozširuje CSS o niekoľko mechanizmov z tradičných programovacích jazykov, ktoré nie sú dostupné v samotnom CSS, ako napríklad premenné, cykly, pravidlá pre vnorenia alebo funkcie. [5]

1.3.2 Bootstrap

Bootstrap [6] je open source framework na vývoj webových stránok. Poskytuje pripravené základné CSS triedy či Javascriptové funkcie, ktoré umožňujú rýchlo vytvoriť responzívnu stránku. Ponúka pripravený responzívny grid systém, elementy ako navigácia, formuláre či posuvné galérie s obrázkami.

1.3.3 Vue

Vue je open source Javascriptový framework na vytváranie používateľského rozhrania ako aj single page aplikácií (aplikácií, ktoré tvorí jediná stránka, ktorá dynamicky prepisuje svoj obsah). Vue používa HTML syntax, ktorá umožňuje prepojenie vykreslených elementov s údajmi vo Vue inštancii. Vue je vybavený systémom reaktivity, ktorý používa objekty Javascriptu a optimálne vykresľovanie. Každý Vue element sa nazýva komponent a ten sleduje svoju reaktívnu závislosť na aktuálnych dátach v inštancii počas svojho vykresľovania, čím systém presne vie, kedy a ktoré komponenty treba znovu prekresliť. [7]

1.3.4 Bootstrap Vue

Bootstrap Vue [8] je, podobne ako Bootstrap, open source framework na vývoj webových stránok. Rozdiel je však v tom, že Bootstrap využíva Javascriptovú knižnicu jQuery, zatiaľ čo Bootstrap Vue už podľa názvu používa Javascriptový framework Vue. Vue a jQuery nie sú navzájom kompatibilné.

1.3.5 Laravel

Laravel je open source PHP framework. Je navrhnutý na vývoj webových aplikácií založených na MVC modeli.

Laravel core – jadro celého frameworku Laravel kernel ja voľne dostupný na githube. Laravel kernel používa IoC návrhový vzor, čo znamená, že každá časť sa dá prepísať prípadne upraviť. Vývojári Laravelu zapracovali veľa riešení a praktík z iných fungujúcich frameworkov.

Laravel Routing – okrem štandardných metód get a post podporuje aj metódy put, patch, delete a option, ktoré vieme zavolať na danú cestu s rôznym účelom. Ľahko sa zoskupujú cesty, ktoré majú nejakú časť spoločnú či automaticky transformujú parametre na model. Routes sú tak tiež chránené pomocou CSRF tokenu, ktorý sa posiela spolu s požiadavkou. Routes tiež umožňujú obmedziť počet požiadaviek od jedného používateľa za určitý čas na určitý počet. [9]

Laravel Eloquent – jednoduchá implementácie ActiveRecord na prácu s databázou. Každý model má vlastnú tabuľku s ktorou pracuje. Model umožňuje vytvárať nové záznamy, upravovať aktuálne či vyberať podľa určitých kritérií. Databázové spájanie tabuliek pomocou metódy join je nahradené reláciami, ktoré môžu byť rôznych typov, či už ide o obyčajné relácie ako one-to-many alebo one-to-one, ponúka Eloquent mnohé ďalšie ako napr. many-to-many, kde si automaticky vytvorí pomocnú tabuľku. [9]

Laravel Artisan – je rozhranie príkazového riadku v Laraveli. Poskytuje užitočné príkazy, ktoré sa používajú pri vývoji. Príkazom ``php artisan list`` sa v príkazovom riadku zobrazia všetky dostupné príkazy. Okrem týchto pripravených príkazov, Laravel ponúka možnosť vytvoriť si tak tiež vlastné príkazy, ktoré potrebujeme. Príkazom ``php artisan make:command SendEmails`` bude vytvorená nová trieda, v ktorej bude metóda, ktorá sa zavolá pri zadaní príkazu do príkazového riadku a tiež dve premenné, ktoré vytvorený príkaz popisujú, tie sa zobrazia pri zobrazení všetkých príkazov.[9]

Laravel Migration – je nástroj, ktorý slúži na synchronizáciu databázy medzi vývojármi. Spustením jednoduchého príkazu ``php artisan migrate`` sa vytvoria nové stĺpce, prípadne sa upravujú aktuálne. [9]

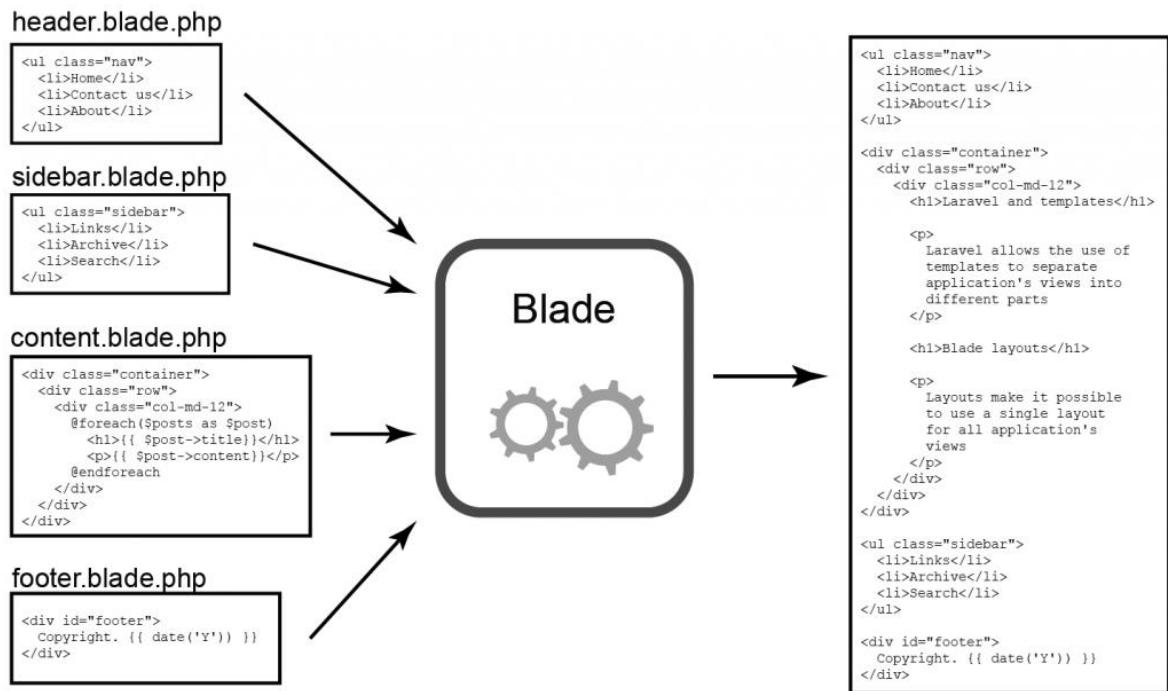
Laravel Seeding – je metóda na vygenerovanie veľkého počtu testovacích dát podľa zvolených pravidiel, ktoré môžu vývojári využívať počas vývoja. Jednoduchým príkazom ``php artisan make:seeder UserTableSeeder`` bude vytvorená nová trieda s jedinou metódou `run`, ktorá sa zavolá na príkaz ``php artisan db:seed``. Na vytváranie veľkého počtu dát sa používa model factory, ktorý sa zavolá metóde `run`. Príkazom ``factory(App\User::class, 50)->create()`` sa vygeneruje 50 rôznych záznamov. Na to, aby boli záznamy rôzne, sa využíva php knižnica Faker [10]. [9]

Laravel Events – jednoduchá implementácia pozorovateľa, ktorá umožňuje sledovať rôzne udalosti (events), ktoré sa udejú v aplikácií a následne na ne cez poslucháča (listener) zareagovať. [9]

Laravel Broadcasting – sa používa aktualizovanie používateľského rozhrania v reálnom čase. Keď sa na serveri dáta upravia, Laravel cez WebSocket pripojenie, pošle túto informáciu všetkým zariadeniam, ktoré sú práve pripojené aby to mohli okamžite spracovať a aktualizovať údaje na stránke. [9]

Laravel Queues – využívajú sa na plynulý chod aplikácie, keď vykonanie náročných alebo zdĺhavých úloh sa presúva do úzadia alebo na neskorší čas (napríklad na noc), aby mohol používateľ využívať aplikáciu. [9]

Laravel blade templating – Blade je jednoduchý a výkonný nástroj na vytváranie výsledných šablón. Využíva sa na generovanie html elementov. Príkazy začínajú znakom `@` a je tam možné používať cykly, podmienky a iné na to, aby výsledná stránka zobrazovala dáta, ktoré potrebujeme, v takej podobe ako potrebujeme.



Obrázok 4 – príklad ako blade vytvorí html

2 Návrh

Použitá literatúra

- [1] Talumis. *What is simulation*. [online] [cit. 15.2. 2019]. Dostupné na internete: <https://www.talumis.com/what-is-simulation/>
- [2] *Simulačné hry*. [online]. [cit. 15.2. 2019]. Dostupné na internete: <http://pdf.truni.sk/e-ucebnice/iktv/data/bb3260f3-f4d6-4311-bf09-ea49b652b427.html?ownapi=1>
- [3] *MVC: Model, View, Controller*. [online]. [cit. 15.2. 2019]. Dostupné na internete: <https://www.codecademy.com/articles/mvc>
- [4] FOWLER, Martin. 2003. *Patterns of Enterprise Application Architecture*. Addison-Wesley, 2003. ISBN 978-0-321-12742-6
- [5] *Sass Basics*. [online] [cit. 15.2. 2019]. Dostupné na internete: <https://sass-lang.com/guide>
- [6] *Bootstrap*. [online]. [cit. 15.2. 2019]. Dostupné na internete: <https://getbootstrap.com/>
- [7] *Introduction – What is Vue.js?* [online]. [cit. 15.2. 2019]. Dostupné na internete: <https://vuejs.org/v2/guide/>
- [8] *Bootstrap + Vue*. [online]. [cit. 15.2. 2019]. Dostupné na internete: <https://bootstrap-vue.js.org/>
- [9] *Laravel documentation*. [online]. [cit. 15.2. 2019]. Dostupné na internete: <https://laravel.com/docs/5.6>