



ZADÁNÍ BAKALÁ SKÉ PRÁCE

Název:	Webová aplikace pro návrh kolejíšť modelové železnice
Student:	Martina Egrtová
Vedoucí:	doc. RNDr. Ing. Marcel Ji ina, Ph.D.
Studijní program:	Informatika
Studijní obor:	Softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce letního semestru 2017/18

Pokyny pro vypracování

Cílem práce je navrhnout a implementovat webovou aplikaci, která umožní navrhovat moduly kolejíšť pro modulární modelovou železnici.

- 1) Seznamte se s koncepcí modulární modelové železnice a na základ diskuze s vedoucím práce identifikujte konkrétními požadavky na návrh jednoho modulu modelové železnice.
- 2) Prozkoumejte jiná existující ešení (aplikace) pro návrh modelové železnice a inspirujte se jimi.
- 3) Navrhn te backend a frontend webové aplikace. Po diskuzi s vedoucím práce zvolte vhodné nástroje pro jejich implementaci.
- 4) Navržené ešení implementujte a ov te jeho funk nost.
- 5) Prove te uživatelské testování, vyhodno te ho a zp tnou vazbu zapracujte do vytvá ené aplikace.
- 6) Aplikaci zdokumentujte v etn celkové a kontextové nápov dy a manuálu k používání aplikace.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

prof. Ing. Pavel Tvrdí k, CSc.
d kan

V Praze dne 9. února 2017

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



Bakalářská práce

Webová aplikace pro návrh kolejiště modelové železnice

Martina Egrtová

Vedoucí práce: doc. RNDr. Ing. Marcel Jiřina, Ph.D.

16. května 2017

Poděkování

Ráda bych poděkovala panu doc. RNDr. Ing. Marcelovi Jiřinovi, Ph.D. za vedení této bakalářské práce, jeho nadšení k tomuto tématu, trpělivost, cenné rady a čas, který mi věnoval.

Prohlášení

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

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

V Praze dne 16. května 2017

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2017 Martina Egrtová. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Egrtová, Martina. *Webová aplikace pro návrh kolejiště modelové železnice*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2017. Dostupný také z WWW: (<https://railway-tracks.herokuapp.com>).

Abstrakt

Předmětem této práce je implementace webové aplikace, která by umožňovala interaktivní grafický návrh kolejiště modelové železnice. Hlavním cílem je usnadnit práci při návrhu částí modelu železničních drah a poskytnutí přehledu pro uživatele. Webová aplikace je určena pro uživatele se zálibou v modelových železnicích. V práci jsou popsány technologie použité k tvorbě aplikace, dostupná řešení, analýza a návrh aplikace, její implementace a testování.

Klíčová slova modulová železnice, návrh, webová aplikace, Node.js, Express.js, Fabric.js

Abstract

The subject of this thesis is the implementation of a web application that allows an interactive graphic design of the model railroads. The main purpose is to simplify work on designing model railway tracks and to provide to user an overview. The web application is intended for users interested in model railways. The thesis describes the technologies used to create the application, available solutions, analysis and design of the application, its implementation and testing.

Keywords modular railway, design, web application, Node.js, Express.js, Fabric.js

Obsah

Úvod	1
1 Cíl práce	3
2 Vymezení pojmů	5
2.1 Modelová železnice	5
2.2 Modulové kolejiště	5
2.3 Modul	6
3 Dostupná řešení	7
3.1 AnyRail	7
3.2 WinTrack	7
3.3 SCARM	8
3.4 TRAX	8
4 Analýza a návrh	9
4.1 Specifikace požadavků	9
4.2 Diagram případu užití	11
4.3 Architektura webové aplikace	12
4.4 Aplikační rozhraní	15
4.5 Datový model	16
4.6 Uživatelské rozhraní	17
5 Technologie	19
5.1 Programovací jazyk a webová platforma	19
5.2 Objektově orientovaná grafická knihovna	20
5.3 Databáze	21
5.4 Ostatní	21
6 Implementace	23

6.1	Konfigurace webového serveru	23
6.2	Třídy	26
6.3	Objektově relační mapování	27
6.4	Klientská část	28
6.5	Tvorba kolejí	34
6.6	Obrazovky a jejich části	36
7	Testování	39
7.1	Scénáře	39
7.2	Vyhodnocení testování	39
8	Uživatelská příručka	41
8.1	Seznámení se s webovou aplikací	41
	Závěr	47
	Literatura	49
A	Seznam použitých zkratk	51
B	Obsah příloženého CD	53

Seznam obrázků

2.1	Příklad modulového kolejiště	6
2.2	Příklad modulu	6
4.1	Uživatelské rozhraní	10
4.2	Diagram případu užití	12
4.3	Struktura webové aplikace	13
4.4	Adresářová struktura	14
4.5	Datový model	17
6.1	Ovládací prvky	31
6.2	Objekt pro napojení kolejí	35
8.1	Hlavní obrazovka	45
8.2	Okno nastavení	46
8.3	Seznam použitých kolejí	46

Seznam tabulek

4.1	Aplikační rozhraní	16
8.1	Klávesové zkratky aplikace	42
8.2	Menu aplikace	43

Úvod

Za modelovou železnicí stojí plno práce při její výrobě a skládání. Mojí prací je usnadnit tu část, kde se modelová železnice navrhuje. Jednoduchá a volně dostupná webová aplikace tomu má pomoci. Je určena pro uživatele se zálibou v modelových železnicích.

Má práce spočívá v tom vytvořit webovou aplikaci, která by umožňovala interaktivní grafický návrh modulové modelové železnice. Modulové kolejiště je typ modelové železnice, které se skládá z částí tratí tzv. modulů. Zajišťuje lepší modularitu a snadnější výrobu, přenositelnost a větší variabilitu při skládání. Modularita je zaručena tím, že koleje jsou z modulu vyvedeny kolmo a tím se dají snadno propojit s dalším modulem. Aplikace obsahuje návrhové pole, které představuje modul. Do tohoto návrhového pole je možné přidávat různé typy kolejí z nabídky. Aplikace poskytuje takovou funkcionalitu, aby se uživateli dobře pracovalo a podporovalo zásady modularity. Návrhové pole je možné ukládat v podobě různých formátů, na server i lokálně.

V části literární rešerše se zabývám výběrem vhodné knihovny pro interakci při návrhu modulu a popisem technologií, které pro svou práci použiji. Dílčí cílem práce je i návrh formátu pro ukládání dat. Dále představím aplikace, které se snaží řešit podobný problém.

V praktické části provádím návrh webové aplikace, její implementaci a testování. Částí této práce je i nakreslení kolejí ve vektorovém editoru a jelikož je funkčnost aplikace závislá na podobě kolejí, je v praktické části popsána také. Pro uživatele je připraven i manuál k používání aplikace.

Cíl práce

Cílem této bakalářské práce je vytvořit jednoduchou webovou aplikaci, která bude umožňovat interaktivní návrh kolejiště pro modelové železnice. Webová aplikace bude určena pro uživatele se zálibou v modelových železnicích. Bude určena přímo pro návrh modulových kolejišť. Aplikace by měla dodat lepší představu o tom, co by si chtěl uživatel vymodelovat a kolik kolejiště bude potřebovat. Aplikace bude obsahovat základní typy kolejí, ze kterých se snadno pomocí interaktivní manipulace budou moci sestavit části kolejišť. Aplikace bude schopna ukládat a načítat návrhy ve zvoleném formátu a exportovat je do obrázků. Mezi další cíle práce patří vytvoření uživatelské příručky webové aplikace a provedení uživatelského testování použitelnosti.

Cílem rešeršní části práce je seznámení se s dostupnými knihovnami, které by poskytovaly potřebné funkčnosti pro interakci při návrhu kolejiště a výběr nejvhodnější z nich. Dále také seznámení se s technologiemi a dostupnými řešeními.

Cílem praktické části práce je návrh a implementace webové aplikace na základě požadavků a provedení testování.

Vymezení pojmů

V této kapitole vysvětlím pojmy související s tématem práce z pohledu modelování železnic, aby si čtenář mohl udělat lepší představu o co se jedná.

2.1 Modelová železnice

Modelová železnice je znázorněním zmenšeného modelu dopravní železnice. Modelování těchto železnic je v této době poměrně rozšířená záliba a vznikají tak i spolky železničních modelářů. Modely mohou být domácí výroby, ale existuje i řada firem, která se touto činností zabývá.

Modelové železnice mohou vycházet z reálných prototypů nebo historických map a obrázků. Modeláři při výrobě mohou vycházet i z vlastní fantazie. Modelování spočívá v utváření scenerie a rozmístění železničních prvků do nich. Součástí mohou být budovy, příroda, lidé nebo další dopravní prvky. U některých modelů je cílem, aby železnice opravdu fungovala jako skutečná a opravdu po ní mohly jezdit vlaky.

2.2 Modulové kolejiště

Jedním z typů modelového kolejiště je kolejiště modulové. Modulové kolejiště se skládá z dílů tzv. modulů. Tímto způsobem skládání má výhody oproti klasickým kolejištím. Přináší větší variabilitu sestav. Moduly se mohou mezi sebou spojovat snadno a libovolně a vytvářet tak pokaždé jiné dráhy. Výroba je také méně náročnější, protože se nemusí stavět celá dráha. Modeláři pak mohou na sestavení železnicové dráhy sdílet své moduly a vytvořit tak rozsáhlé modely. Tyto železnicové dráhy jsou také lépe skladovatelné, dají se rozložit a nezabírají tak celou místnost. Na obrázku 2.1 je vidět, jak takové modulové kolejiště vypadá a že je složeno z menších částí. [1]

2. VYMEZENÍ POJMŮ



Obrázek 2.1: První modul TT kolejiště KŽM Praha 3 byl postaven 6/2001 [2]

2.3 Modul

Modul znázorňuje určitý úsek tratě. Je to část železniční dráhy. Okraje modulů, na které jsou vyvedeny koleje mají přesnou návaznost kolejnic na sebe, takže lze moduly snadno spojovat. Na obrázku 2.2 je možné vidět ukázkou reálného modulu.



Obrázek 2.2: Modulové kolejiště TT (Německo, 2001) [3]

Dostupná řešení

V této kapitole představím pár programů, které slouží pro návrh modelové železnice. Jedná se o programy, které jsou velice rozsáhlé, obsahují prvky modelové železnice od různých výrobců, kterých je opravdu spousta. Nicméně nejsou navrženy pro potřeby modulových železnic, i když by bylo možno je v těchto programech nakreslit. Slouží spíše pro návrh celé železniční dráhy spolu s okolím. Většina programů jsou desktopové verze, a tak je lze použít pouze na některých operačních systémech, obvykle na systému Windows. Ty jsou také většinou placené.

3.1 AnyRail

AnyRail je program pro vytváření železničních modelů. Je to desktopový program pouze pro operační systém Windows. Obsahuje mnoho předdefinovaných knihoven a databází pro různé vlakové soupravy, koleje a komponenty. Mimo návrhu železnic umožňuje vytvářet nákupní seznam a zobrazit návrhové pole v 3D. Má dostupnou omezenou verzi programu pro vyzkoušení zdarma, pro plné užití je potřeba koupit licenční klíč za 44 €. [4]

3.2 WinTrack

WinTrack obsahuje širokou škálu traťových systémů pro různá měřítka. Dále obsahuje další sady komponent jako signály, tunely, mosty, budovy, auta a další. Vše může být převedeno do 3D a ve 3D si lze také vytvořit vlastní objekty. Jako AnyRail pomáhá vytvořit nákupní list použitých kolejí. Má dostupnou omezenou verzi zdarma, která neumožňuje ukládat nebo tisknout plán. Plná verze programu stojí 99,5 €. [5]

3.3 SCARM

SCARM je volně dostupný program pro navrhování železničních tratí. Je to desktopový program pro operační systém Windows. Podporuje mnoho druhů kolejí a objektů jako mosty, tunely a další. Výhodou je možnost si kolejovou trať nakreslit pomocí křivek. Návrhy lze převést do 3D pohledu. Umožňuje vytvoření vlastních 2D objektů, které se pomocí nástroje dají zobrazit i jako 3D. [6]

3.4 TRAX

TRAX je webový nástroj pro plánování tras modelové železnice. Pro jeho použití je nutná registrace. Je navržen jako pro webové prohlížeče na počítačích tak i pro mobilní zařízení. Poskytuje prvky modelové železnice od různých výrobců, které lze vkládat do velké souřadnicové mapy. Nástroj je zcela zdarma, ale jelikož byl spuštěn v roce 2016 a je stále ve vývoji, mohou se projevovat chyby. [7]

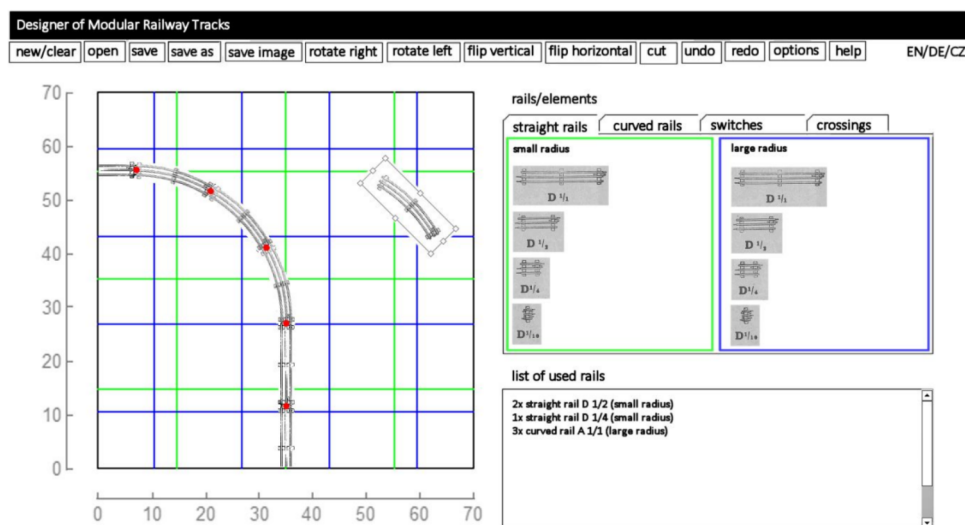
Analýza a návrh

V této kapitole popíšete analýzu a návrh řešení aplikace. Naleznete zde specifikaci požadavků na funkce aplikace. Dále je v této kapitole popsána struktura aplikace použitých technologií a ukázka vytvoření webového serveru s Node.js Express. Další částí je specifikace aplikačního rozhraní na webovém serveru, datový model a uživatelské rozhraní.

4.1 Specifikace požadavků

Cílem aplikace je umožnit interaktivně vytvářet návrhy kolejišť modulové modelové železnice. Jeden modul je čtverec o straně 70 cm, do kterého mohou být umístovány koleje libovolně s výjimkou hranic modulu, kde kolej musí být vyvedena kolmo k hranici modulu a vždy v ose nějaké zelené nebo modré čáry, které jsou pevně definované v mřížce (zde nazývané dvoj mřížka). Jednotlivé moduly lze následně sestavovat k sobě a vytvářet rozlehlejší kolejiště. Koleje jsou dvojího druhu. Základním rozlišením je, že tzv. menší varianta používá více zakřivené oblouky/výhybky (8 oblouků tvoří kruh o průměru 70 cm, křížení je pod 45°, rovné koleje jsou kratší). Větší varianta používá méně zakřivené oblouky/výhybky (12 oblouků tvoří kruh o průměru 119 cm, křížení je pod 30°, rovné koleje jsou delší). Obě varianty se dají libovolně kombinovat. Koleje tak mohou začínat na hraně modulu na zelené čáře a končit na jiné hraně modulu na modré čáře apod.

4. ANALÝZA A NÁVRH



Obrázek 4.1: Uživatelské rozhraní aplikace

Výchozím stavem je prázdná dvoj mřížka, do které lze umisťovat systémem drag&drop různé druhy kolejí z palety napravo. Paleta obsahuje několik záložek, ve kterých jsou jednotlivé druhy kolejí sdruženy do skupin. Vybrané elementy je možné během přenášení otáčet a překlápět dříve, než jsou umístěny do dvoj mřížky. I poté je ale možné kdykoliv libovolnou již umístěnou kolej vybrat a manipulovat s ní stejně, jako by byla vybrána z palety. Pokud se některý z konců umisťované koleje nachází blízko jiného konce již umístěné koleje, kolej by měla „zacvaknout“ do oné umístěné koleje při přidržení tlačítka Shift. Podobně by se měla kolej přichytit k mřížkám automaticky, s výjimkou přidržení tlačítka Ctrl, která danou funkcionalitu vyřadí. Prioritou je nejprve přichycení ke koleji a poté teprve k mřížce.

Každá umístěná kolej je vylistována v textové formě do okna pod paletou. Tento seznam je možné vybrat myší a zkopírovat do schránky.

Pomocí tlačítek na klávesnici se umožní následující funkcionality:

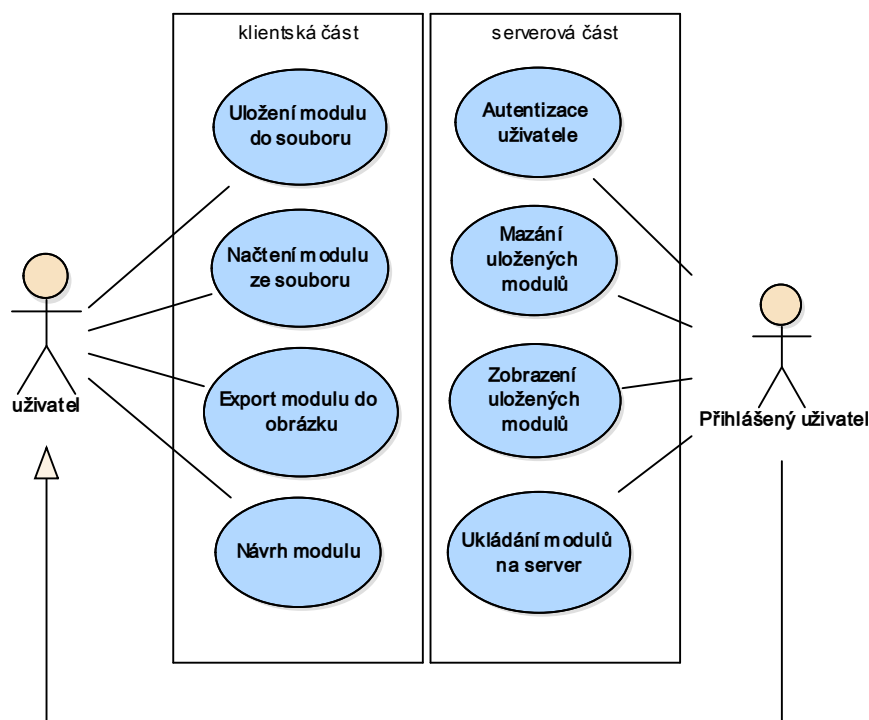
- s/a – vertikálně/horizontálně překloupit umisťovanou kolej.
- šipka vpravo/vlevo – otáčení umisťované koleje vpravo a vlevo.
- Delete – Smazání vybrané koleje.
- Shift – Zacvaknutí kolejí
- Ctrl – vyřazení funkcionality přichytávání

Funkce jednotlivých tlačítek:

- New/clear – vymaže vše v dvoj mřížce. Předtím vyskočí okno, zda chce uživatel modul uložit, pokud došlo k nějaké neuložené změně v dvoj mřížce.
- Open/Save/Save as – umožní uložit modul ve formátu XML (formát by měl být navržen tak, aby byl dobře intuitivně pochopitelný). Otevře se dialogové okno pro otevření/uložení modulu. Ideálně by se měl uživatel nejprve do aplikace zaregistrovat/přihlásit, aby mohl ukládat vytvořené moduly do své složky na serveru.
- Save image – umožní uložit modul (dvoj mřížku bez rozměrových os) jako bitmapu ve formátu PNG (případně JPEG).
- Rotate right/left – otáčení celé dvoj mřížky doprava doleva o 90 stupňů.
- Flip vertical/horizontal – překlopení celé dvoj mřížky vertikálně/horizontálně.
- Cut – ořízne všechny koleje, které přesahují dvoj mřížku (předpokládá se, že rovné koleje budou sloužit jako vyrovnávací koleje a hlediska délky, takže se často stane, že umístovaná rovná kolej bude muset být oříznuta tak, aby se vešla do modulu).
- Undo/redo – odebrání a vracení kolejí
- Options – zatím možnost nastavit rozteče mřížek v samostatném oknu. Rozteče zelené mřížky (malá varianta) jsou 20,5 cm, modré mřížky (velká varianta) jsou 16,33 cm. Mřížky jsou centrovány vzhledem ke středu modulu.
- Help – otevře se další webová záložka se strukturovaným návodem, jak aplikaci používat.

4.2 Diagram případu užití

Diagram případu užití zobrazuje chování systému z hlediska uživatele a jeho rolí. Hlavním účelem je zachycení účastníků systému a vztahů mezi poskytovanými službami. V diagramu je systém je rozdělen do dvou částí klientské a serverové, protože je větší část aplikace psána na front-endu. Diagram popisuje jednotlivé činnosti, které uživatel smí vykonat.

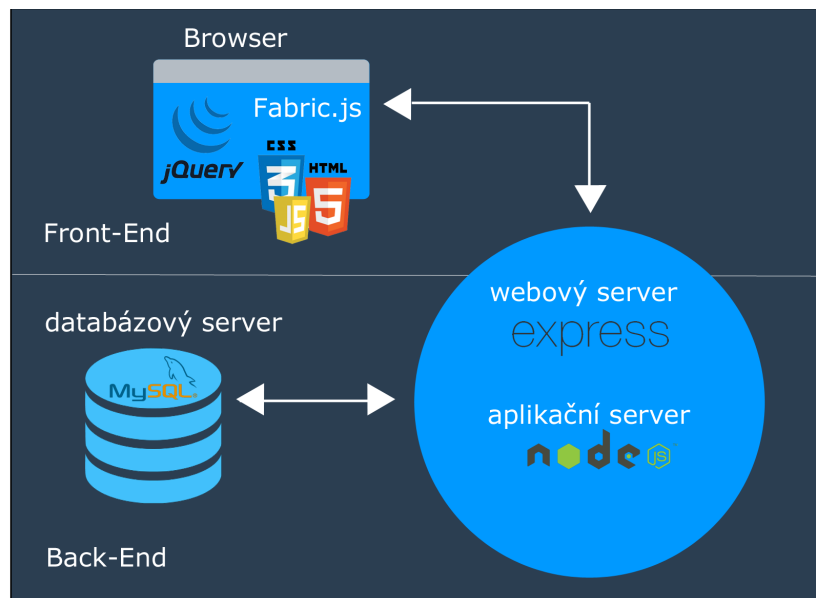


Obrázek 4.2: Diagram případu užití

4.3 Architektura webové aplikace

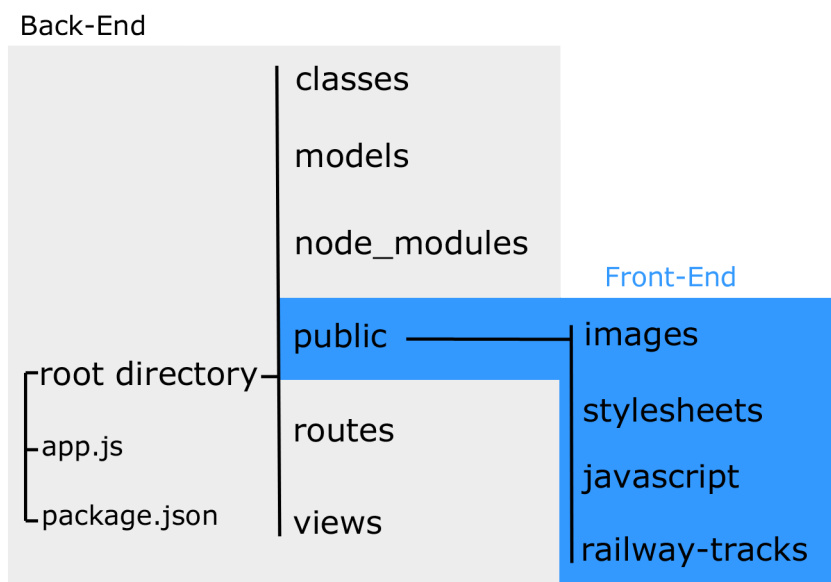
Architektura webové aplikace využívá MVC architektury, která rozděluje datový model, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent označovaných model, pohled (angl. view), řadič (angl. controller). Model reprezentuje informace, se kterými aplikace pracuje a definuje metody pro interakci s datovým úložištěm. Pohled řídí vše, co se koncovým uživatelům zobrazí na obrazovce. Řadič reaguje na události a změny v modelu nebo pohledu.

Aplikace běží na webovém serveru Node.js Express, protože je vhodný na jednoduché aplikace a dá se snadno zprovoznit. Pro zobrazení prezenční vrstvy je použit šablonovací systém Jade. Server přistupuje k databázi MySQL. Na klientské části je využito technologií HTML5, CSS3, Fabric.js a jQuery.



Obrázek 4.3: Struktura webové aplikace

4.3.1 Adresářová struktura



Obrázek 4.4: Adresářová struktura aplikace na serveru

Adresářová struktura je dána defaultní strukturou při instalaci Node.js Express. Přidané vlastní podadresáře jsou pouze `classes` a `models`. Na obrázku 4.4 je modře označena zóna, která se využívá zejména pro front-end aplikace. Ostatní jsou využívány čistě serverově.

Popis jednotlivých adresářů:

- **root directory** – obsahuje celý projekt
 - **app.js** – Jádro aplikace. Hlavní spouštěcí soubor, který spojuje části aplikace. Importuje důležité objekty z nainstalovaných balíčků potřebné pro základní fungování webové aplikace. Nastavují se zde cesty k adresářům a objektům, které chce využívat.
 - **package.json** – JSON dokument s informacemi o projektu. Vyžadovanými položkami jsou jméno aplikace a její verze. Nejdůležitější volitelnou položkou jsou závislosti, kde jsou definovány používané balíčky a jejich verze. Tyto závislosti se zapisují automaticky už při instalaci balíčků, pokud napíšeme příkaz s přepínačem `--save`.

- **classes** – Obsahuje třídy.
- **models** – Obsahuje objekt pro přístup k datům. Umožňuje manipulovat s obsahem databáze.
- **node_modules** – Obsahuje nainstalované balíčky. Tyto balíčky se importují do aplikace pomocí příkazu **require**.
- **public** – Adresář pro statické dokumenty.
 - **images** – Úložiště pro obrázky.
 - **stylesheets** – Úložiště pro kaskádové styly pro vzhled aplikace.
 - **javascript** – Obsahuje různé stažené javascriptové komponenty, které jsou používány v klientské části aplikace.
 - **railway-tracks** – Obsahuje hlavní funkcionalitu této práce, vektorové obrázky kolejí a XML soubor se seznamem těchto kolejí.
- **routes** – Stanovuje jak aplikace bude reagovat na klientovi požadavky. Definuje tzv. routy, které jsou popsány pomocí HTTP metody, URI a spouští předepsaný kód pokud se požadavek shoduje s danou routou.
- **views** – Obsahuje HTML šablony

4.4 Aplikační rozhraní

Aplikační rozhraní slouží pro komunikaci mezi serverem a klientem. Definuje cesty (koncové body) ke službám nebo datovým informacím serveru pomocí identifikátoru zdroje v síti (URI), které určují i metodu HTTP požadavku. Pokud přijde požadavek na cestu, která v aplikačním rozhraní není, je vrácena chybová stránka. Pokud cesta existuje, požadavek se zpracuje a vrací odpověď. V aplikaci jsou povolena HTTP cookies, což jsou malá data, která nejprve server odešle klientovi a která má klient uložené na zařízení. Při dalším požadavku se tato data odesílají zpět serveru.

V tabulce níže jsou popsány cesty, které v aplikaci jsou a k čemu slouží.

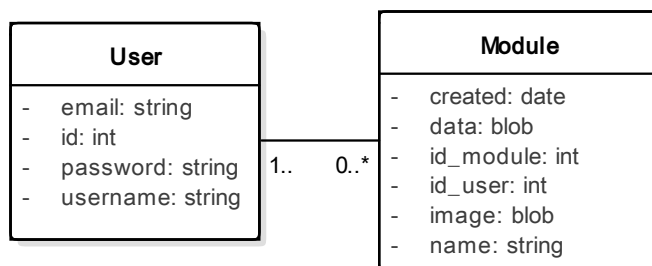
Tabulka 4.1: Aplikační rozhraní

Metoda	Požadavek	Popis
GET	/	Načte hlavní stránku
POST	/getTrails	Načte sadu kolejí
POST	/getParsed	Rozparsuje poslané XML
POST	/getSessionModule	Získá z cookies naposledy automaticky uložený modul
POST	/saveOptions	Uloží nastavení, které jsou na výběr při exportu do obrázku do cookies.
POST	/getOptions	Získá nastavení exportu modulu do obrázku z cookies.
GET	/help	Zobrazí stránku s návodem
GET	/auth/register	Zobrazí stránku s registrací
POST	/auth/register	Registruje uživatele
GET	/auth/login	Zobrazí stránku s přihlášením
POST	/auth/login	Přihlásí uživatele
GET	/auth/logout	Odhlásí uživatele
GET	/modules	Zobrazí stránku s moduly uloženými na serveru, pokud je uživatel přihlášen
POST	/modules	Získá moduly přihlášeného uživatele z databáze
POST	/modules/getModule	Získá konkrétní modul uživatele na základě jeho id
POST	/modules/saveModule	Vytvoří a uloží nový modul
POST	/modules/saveSessionModule	Uloží do cookies právě rozpracovaný modul
POST	/modules/deleteModule	Odstraní konkrétní modul uživatele na základě jeho id

4.5 Datový model

Pro požadavek ukládání modulů na server je potřeba uživatele identifikovat, a proto je zavedena jednoduchá autentizace. Pro autentizaci slouží v databázi tabulka uživatelů se základními informacemi jako přihlašovací jméno, email a heslo. Identifikátor je automaticky generován při přidávání uživatele a je to primární klíč. Tabulka modulů obsahuje id modulu, id uživatele, datum vytvoření, jméno modulu, obrázek modulu a jeho serializaci v podobě XML formátu. Identifikátor modulu je také automaticky generován spolu s datem vytvoření při ukládání modulu. Primárním klíčem je zde id modulu. Pro to, aby si mohl uživatel uložit více modulů slouží v tabulce modulů položka id

uživatele, která slouží jako cizí klíč a která identifikuje uživatele kterému modul patří. Vztah mezi tabulkami je 1:N, tedy jeden uživatel může mít uloženo více modulů, ale jeden modul patří pouze jednomu uživateli.



Obrázek 4.5: Datový model

4.6 Uživatelské rozhraní

Uživatelské rozhraní a jeho podoba je znázorněna ve specifikaci požadavků. Na obrázku 4.1 se vykytuje podoba hlavní stránky, ze které se bude vycházet. Na stránce by mělo být návrhové pole představující modul do kterého se budou přidávat pomocí funkce drag&drop koleje ze seznamu kolejí. Nalevo od modulu bude svislá osa, udávající výšku modulu. Pod modulem bude vodorovná osa, udávající šířku modulu. Koleje budou rozříděné do záložek podle typu kolejí. Na levé straně záložky se budou nacházet koleje menších variant, a na pravé koleje větších variant. Pod modulem a jeho osami se bude nacházet tabulka s použitými koleji v modulu. Nad modulem bude popisek se jménem modulu a ikony tužky představující editace popisku. Těsně pod menu se bude nacházet stavový panel kam se budou vypisovat informace o kolejích.

Hlavička stránky bude obsahovat název aplikace a jednoduchý jednořádkový formulář pro rychlé přihlášení. Ve formuláři pro přihlášení budou dvě okénka pro textový vstup. První pro uživatelské jméno, druhé pro heslo. Napravo ve formuláři bude tlačítko pro přihlášení. Po úspěšném přihlášení se místo formuláře bude zobrazovat uživatelské jméno a odkaz k odhlášení uživatele z aplikace. Hlavička zároveň bude obsahovat menu, které slouží jak pro navigaci, tak pro práci s modulem.

Pokud uživatel nebude přihlášen menu bude obsahovat následující položky v pořadí: 1. new/clear, 2. open, 3. save s rozbalovacím menu s 3.1 locally(xml), 3.2 as image, 4. rotate right, 5. rotate left, 6. flip vertical, 7. flip horizontal, 8.undo, 9.redo, 10.options, 11. help, 12. register.

Pokud uživatel bude přihlášen menu bude obsahovat následující položky v pořadí: 1. new/clear, 2. open, 3. save s rozbalovacím menu s 3.1 on server, 3.2 locally(xml), 3.3 as image, 4. rotate right, 5. rotate left, 6. flip vertical, 7. flip horizontal, 8.undo, 9.redo, 10.options, 11. help, 12. modules.

Při přidržení kurzoru nad položkami menu bude zobrazen tooltip. Při kliknutí na položku open bude vyvoláno systémové okno pro nahrání souboru. Při kliknutí na položky locally(xml) a as image v save bude vyvoláno systémové okno pro uložení souboru.

Položka options zobrazí modální okno, které se bude moct nechat přesouvat po stránce. Bude obsahovat následující tři formuláře: Nastavení exportu obrázku – ve formuláři bude textové pole, které po kliknutí zobrazí okénko pro výběr barvy. Dále dvě zaškrtačací políčka a rozbalovací menu spolu s tlačítkem. Nastavení vzdáleností mřížek – ve formuláři budou dvě obyčejná textová pole spolu s tlačítkem. Nastavení intervalu pro automatické ukládání modulu – ve formuláři bude jedno obyčejné textové pole spolu s tlačítkem. Při přidržení kurzoru nad položkami formuláře bude zobrazen tooltip.

Položka help přesměruje stránku na návod k aplikaci. Návod se bude skládat z obrázků (screenshotů) aplikace, popisů a tabulek pro klávesové zkratky a popis položek menu.

Položka register přesměruje stránku na registrační formulář. Formulář bude obsahovat textové pole pro uživatelské jméno, pro email a pro heslo a tlačítko.

Položka modules přesměruje stránku na seznam modulů uložených na serveru. Moduly zde budou zobrazeny pomocí obrázku, jména a data vytvoření a budou zarovnané do mřížky. U modulu v pravém horním rohu nad obrázkem budou tři ikony pro operaci s modulem. První ikona bude zobrazovat křížek, který poukazuje na smazání modulu. Při kliknutí na něj se vyvolá dialogové okno s tím zda chce uživatel opravdu smazat modul. Druhá ikona bude zobrazovat šipku stažení, pro stažení souboru, které vyvolá systémové okno pro uložení souboru. Třetí bude ikona obrázku, představující možnost stažení modulu jako obrázku a také vyvolá systémové okno pro uložení souboru. Při přidržení kurzoru nad ikonami se zobrazí tooltip.

Aplikace bude při pravém horním menu zobrazovat informace o stavu akcí, které uživatel provádí, v animované bublině. Ta po chvíli zmizí.

Technologie

Tato kapitola popisuje technologie, které byly zvoleny pro implementaci aplikace na základě zkušeností a splnění požadavků, které byly kladeny. Jedná se o výběr webové platformy, programovacího jazyka, databáze, a hlavně knihovny pro interaktivní objektový model.

5.1 Programovací jazyk a webová platforma

Při volbě programovacího jazyka a webové frameworky jsem se omezila na již mě známé věci, a to Symfony pro vývoj webových aplikací v jazyce PHP nebo Node.js Express pro vývoj v jazyce Javascript. Protože tato práce je zaměřena spíše na klientskou část aplikace, kde bych využívala jazyk Javascript, rozhodla jsem se pro použití Node.js Express. Node.js je také více vhodný pro vytváření jednoduchých webových aplikací. Lze s ním snadno a rychle vytvořit webový server a díky své implementaci je i rychlý.

5.1.1 Node.js

Node.js je platforma pro vytváření síťových aplikací, tedy i webových aplikací. Zahrnuje v sobě webový server, takže není třeba dalšího speciálního nástroje jako při vývoji v PHP, kde se většinou využívá Apache. Node.js je javascriptové běhové prostředí postavené na Chrome V8 javascriptovém interpretu. Node.js pracuje pouze v jednom vlákne, narozdíl od jiných webových serverů, kde se každý požadavek zpracovává v jiném vlákne. Zpracovávání na více vláknech bývá často zbytečné a neefektivní kvůli inicializacím a načítáním potřebných věcí. Při použití jednoho vlákna je vše inicializováno a načteno pouze jednou. Asynchronní zpracování je další optimalizační technikou, který Node.js používá. Asynchronní znamená, že se nečeká na zpracování příkazů a pokračuje se dál v kódu, a neblokuje další operace. Součástí při instalaci této platformy je správce balíčku NPM (Node Package Manager), kterým lze do aplikace jednoduše nainstalovat různé rozšiřující balíčky. [8][9]

5.1.2 Express

Express je webový aplikační framework pro Node.js. Poskytuje základní funkce pro webové ale i mobilní aplikace. Lze snadno nainstalovat pomocí NPM, protože je to vlastně rozšiřující balíček Node.js. Usnadňuje práci se zpracováním požadavků serverem a s odesláním odpovědi zpět klientovi. V aplikaci se pro tyto účely používá tzv. routování. Definují se identifikátory zdroje v síti (URI) a HTTP metody, kterými lze k serveru přistupovat. Ty pak mohou přistupovat k požadavkům a odpovědím pomocí tzv. middleware funkcí. Tyto funkce jsou asynchronní a odpovídají na požadavky zasláním dat nebo vygenerováním stránky pomocí šablonovacího systému.[10]

5.2 Objektově orientovaná grafická knihovna

Grafická knihovna slouží k vytvoření grafiky na webu pomocí SVG nebo elementu canvas z HTML5. Tyto knihovny pomáhají při zobrazování grafických prvků a jejich animací. Některé z nich podporují interaktivitu a javascriptové události grafických prvků a tak lze prvky ovládat. Při výběru takové knihovny byla stanovena kritéria, která by knihovna měla umět, a to právě manipulaci s objekty jako rotování, převrácení a pohyb. Dalším kritériem byla podpora událostí jako kliknutí na objekt, posunutí objektu a další. Dále aby šlo snadno exportovat objekty do obrázku a aby šli vykreslit obrázky formátu SVG. Posledním kritériem byla možnost grafiku serializovat, deserializovat, tedy uložit stavy grafických prvků tak, aby se případně mohly znovu načíst.

Výběr z knihoven byl celkem velký a rozhodování probíhalo mezi Fabric.js, Raphael, Paper.js a Easel.js. Nakonec byl vybrán Fabric.js, který splňoval všechna kritéria a zdál se pro tuto práci nejvhodnější.

5.2.1 Fabric.js

Fabric.js poskytuje interaktivní objektový model postavený na elementu canvas v HTML5. Má zabudovaný parser z canvasu do SVG a naopak. Umí vytvářet jednoduché geometrické útvary nebo složitější útvary pomocí cest. Podporuje ale i vykreslení obrázku nebo textu. Objektům vytvořené touto knihovnou lze nastavovat různé filtry, barevné přechody nebo stíny. Dále jsou k dispozici animace a rozsáhlý systém událostí. Objekty je možné slučovat do skupin a pracovat tak s celou skupinou jako s objektem. Dokáže serializovat canvas do formátu JSON nebo SVG a pomocí těchto formátů je znovu deserializovat. Export do obrázku různých formátů je také možný. [11]

Tato knihovna je opravdu velmi dobrá. Obsahuje mnohé další funkcionality. Manipulace s objekty je defaultně povolena a ovládací prvky jsou zde přívětivé a intuitivní. Objekty mají k dispozici mnoho vlastností a lze přidávat i vlastní. Reakce na události je také jednoduché změnit podle svého.

5.3 Databáze

5.3.1 MySQL

MySQL je softwarový nástroj vlastněný společností Oracle, který zprostředkovává práci s relační databází. Je to nejpoužívanější open source databáze pro webové aplikace. Komunikace s databází probíhá pomocí jazyka SQL (Structured Query Language) a lze jej používat na různých operačních systémech. Každá MySQL databáze je tvořena z tabulek, které mají sloupce a řádky. O řádcích mluvíme jako o záznamech a o sloupcích jako o attributech. Atributy určují datový typ záznamů ve společném sloupci. Modifikovat, získávat nebo mazat lze pomocí dotazů právě v jazyce SQL. Pro vzdálenou správu databáze přes webové rozhraní je používán nástroj phpMyAdmin.

5.4 Ostatní

Mezi ostatní jsou zařazeny méně důležité technologie. Jedná se o pomocné technologie a formáty souborů.

5.4.1 jQuery

jQuery je knihovna, která usnadňuje práci s javascriptem a pomáhá v interakci mezi javascriptem a HTML. Umí manipulovat, animovat a zachytávat události objektů HTML mnohem snadněji díky svému API. Je multiplatformní a podporuje HTML5 a CSS3. Nad touto knihovnou je postaveno jQuery UI, které je také často využíváno. Obsahuje předdefinované prvky pro webové aplikace jako dialogy nebo kalendáře.

5.4.2 Jade

Jade je šablonovací systém, který odděluje prezentační vrstvu od té programové. Slouží také ke generování HTML kódu na základě skriptu. Při generování stránky můžeme definovat nejen šablonu jaká se má použít, ale i data která má vypsát nebo se kterými má pracovat. Kdyby bylo šabloně předáno pole stejných objektů, potom by se dali snadno vypsát pomocí cyklu. Jade také zkracuje zápis HTML elementů. Nepoužívá běžné tagy ale pouze názvy elementů které musí být správně odsazeny a zanořeny v kódu a ukončený novým řádkem. Je vhodný spíše pro menší, jednoduché aplikace.

5.4.3 XML

XML je rozšiřitelný značkovací jazyk. Definuje soubor podmínek pro zakódování dokumentu ve formátu, který jde přečíst lidmi i počítačem. Používá se pro serializaci dat, a je především určen pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Klade důraz na informační obsah, není zaměřený na

prezentaci. Formát se skládá z elementů a jejich atributů, které popisují konkrétní data nebo dokumenty. Ve své práci používám XML k serializaci modulu a také k definici všech typů kolejí, které se pak z tohoto souboru načítají.

5.4.4 SVG

SVG je značkovací jazyk a formát souboru, který popisuje vektorovou grafiku pomocí XML. Prohlížeče jsou schopné tento jazyk vykreslit jako grafické objekty. Zatímco v rastrové grafice je obrázek popsán pomocí jednotlivých pixelů, vektorová grafika popisuje obrázek jako složení z předdefinovaných útvarů jako body, přímky, křivky, mnohoúhelníky a další. Výhodou tohoto formátu je velikost souboru, snadná přenositelnost a čitelnost jak pro počítač, tak člověka. SVG soubory se dají snadno kreslit v některém z vektorových editorů. Jedním z úkolů v této práci bylo nakreslení kolejí a ty byli nakresleny právě ve vektorovém editoru Inkscape. Inkscape jako svůj nativní formát používá právě SVG.

5.4.5 Canvas

Canvas je jedním z elementů v HTML5, je to plátno, na které se dá kreslit. Pomocí HTML5 pouze definujeme tento element spolu s velikostí plátna a id. Pomocí id pak můžeme přistupovat k plátnu a javascriptem vykreslovat objekty. Pro to, aby se dali objekty vykreslovat na pozice má canvas souřadnicový systém, který má počátek v levém horním rohu. Na plátno lze snadno vykreslit základní tvary jako obdélníky, kruhy, čáry anebo obrázky.

Implementace

V této kapitole je popsána implementace webové aplikace od vytvoření a konfigurace serveru, přes klasifikaci tříd a jejich objektově relačního mapování, až po řešení problému práce a tvorby kolejí.

6.1 Konfigurace webového serveru

Následující postup při instalaci a vytvoření projektu platí pro systém Windows. Nejprve je potřeba stáhnout instalační balíček Node.js z oficiálních stránek nodejs.org. Instalace je snadná, stačí spustit stažený instalační soubor a postupovat podle klasických instrukcí instalátoru. Je potřeba souhlasit s licenčním ujednáním a proklikat se pomocí tlačítka další na konec s ponechaným základním nastavením pro instalaci. Základní nastavení pro instalaci nainstaluje i NPM, balíčkovací systém, který je součástí Node.js.

Po úspěšné instalaci už je možno vybrat pracovní adresář a vytvořit tam projekt následujícím způsobem.

Zavolání příkazu `npm init` v pracovním adresáři, který vygeneruje soubor `package.json` s informacemi o projektu. Další fází je nainstalování balíčku s webovým frameworkem Express příkazem `npm install express --save`. Tento příkaz do adresáře vygeneruje složku `node_modules` s balíčkem Express. Pro vytvoření základního skeletonu aplikace poslouží následující příkaz `express --view=pug jmeno_aplikace`. To vytvoří v hlavním adresáři vedle `node_modules` novou složku s jménem aplikace a v ní už je vygenerovaná struktura, která je popsána výše v kapitole adresářová struktura aplikace. Z této složky je následně nutné zavolat `npm install`, aby se doinstalovali případné závislosti. Při generování struktury aplikace se vytváří aplikace nová, která si zkopíruje soubor `package.json`. Nová aplikace se nachází v hlavním adresáři ve složce se jménem aplikace, které se zadávalo při generování skeletonu pomocí Express. [12]

Dalším způsobem, jak jednoduše vytvořit projekt je stáhnout si IDE Webstrom, které přímo nabízí založení Node.js Express aplikace s vybraným ša-

blonovacím systémem. Do Webstrom IDE je v nastavení pouze třeba zadat cestu k Node.js spustitelnému souboru.

Před samotným řešením problému a přidávání funkcionalit je nutné nastavení a spuštění webového serveru. Konfigurace webového serveru se nachází v kořenovém adresáři ve spouštěcím souboru `app.js`, jehož obsah je popsán pod tímto textem.

Obsah `app.js`:

Na horních řádcích se nachází importované objekty a moduly přidané pomocí funkce `require()`. Mezi defaultně importované moduly patří:

express – importuje framework Express

path – jádro Node pro práci s cestami.

Hodí se například pro nastavení cesty k HTML šablonám. pomocí `path.join()` a jeho argumentů sestaví cestu.

```
app.set('views', path.join(__dirname, 'views'));
```

morgan – Slouží k přehledu požadavků a odpovědí na serveru. Hodí se pro spíše pro vývoj aplikace, kdy je třeba vědět činnosti serveru.

cookieParser – modul z `expressu`, která pomáhá s prací s HTTP cookies. Přidává požadavku objekt `cookies`, ke kterému lze snadno přistupovat a nastavovat ho.

bodyParser – modul z `expressu`, která přidává do požadavku objekt `body`. Ten obsahuje parametry z HTTP požadavků, například data z formulářů odeslaných metodou POST.

další přidání moduly:

express-session – modul z `expressu`, který pomáhá s nastavováním `session` a přidává požadavku objekt `session`. Hodí se například pro autentizaci uživatelů.

mysql – modul pro připojování a dotazování k MySQL databázi.

index, auth, modules, help – objekty, které obsluhují co se bude klientovi posílat a zobrazovat a na jaký požadavek. Pro přehlednost je lepší je mít rozdělené podle souvislosti do více souborů jako zde.

Následně se vytvoří aplikace pomocí příkazu `express()` a nakonfigurují se importované moduly a globální proměnné, které se mají používat, pomocí příkazů `app.use()` a `app.set()`.

Výpis 6.1: Ukázka importu balíčků a proměnných a vytvoření aplikace

```
var express = require('express');

var index = require('./routes/index');
var auth = require('./routes/auth');
var modules = require('./routes/modules');
var help = require('./routes/help');

var app = express();
```

Výpis 6.2: Nastavení šablonovacího systému a cesta k šablonám

```
app.set('views', path.join(__dirname, 'views'));
app.set('view engine', 'jade');
```

Výpis 6.3: Nastavení logování požadavků na server do konzole

```
logger('dev');
```

Loguje požadavky do konzole. Parametr funkce určuje jaké informace se budou vypisovat. Zde parametr `dev` vypisuje informace o HTTP metodě, URL, statusu, době a délce odpovědi.

Výpis 6.4: Nastavení pro parsování JSON formátu aplikací

```
bodyParser.json()
```

Výpis 6.5: Nastavení aplikace pro čtení dat z URL a jejich transformaci do objektu `body` v požadavku

```
bodyParser.urlencoded({ extended: false })
```

Výpis 6.6: Přidání cookies do požadavků a odpovědí

```
app.use(cookieParser());
```

Výpis 6.7: Nastavení cesty k veřejnému adresáři

```
express.static(path.join(__dirname, 'public'))
```

Výpis 6.8: Nastavení routování

```
app.use('/', index);
app.use('/auth', auth);
app.use('/modules', modules);
app.use('/help', help);
```

Routování je technika zpracování požadavků. V příkladu nad tímto textem je prvním parametrem funkce cesta, která se přidává za doménovou adresu a druhým parametrem je objekt který požadavky umí zpracovat. Tyto objekty jsou importovány v ukázce kódu 6.1.

Další nastavení, které nebudou popsány obsluhují zachytávání neexistujících url cest a chyb v aplikaci.

6.2 Třídy

Třídy jsou jakousi šablonou pro vytváření objektů (instancí tříd). Popisují vlastnosti reálného objektu pomocí atributů a chování pomocí metod. Javascript neobsahuje klasickou definici třídy i když je označován jako objektově orientovaný. V javascriptu není žádné třídní programování ale prototypové. Třidu lze zkonstruovat pomocí vlastnosti funkce prototype, které vytvoří prázdný objekt. Pokud je objekt vytvořen pomocí konstruktoru s new, pak dědí všechny vlastnosti a metody od prototypového objektu. V ukázce kódu níže 6.9 je příklad vytvoření třídy v javascriptu.[13]

Ve své práci mám definovány dvě třídy, a to třídu s názvem User a třídu názvem Module.

Vlastnosti třídy User představující uživatele:

- id – identifikační číslo uživatele
- username – uživatelské jméno
- email – emailová adresa

Vlastnosti třídy Module představující modul:

- id – identifikační číslo modulu
- userId – identifikační číslo uživatele, který modul vytvořil
- image – bytová data obrázku modulu
- data – serializace modulu v podobě XML
- created – datum vytvoření modulu

Výpis 6.9: Ukázka vytvoření třídy v javascriptu pomocí vlastnosti prototype.

```
function User(username) {
    this.username = username;
}
User.prototype = {
    constructor : User,
    username : '',
    setUsername : function (username) {
        this.username = username;
    }
};
module.exports = User;
```

6.3 Objektově relační mapování

Objektově relační mapování je programovací technika, která zjišťuje konverzi dat mezi databází a objektem. V databázi máme objekty reálného světa uloženy ve formě řádků v tabulkách. V Objektově orientovaných jazycích jsou tyto objekty instancí tříd. Reprezentace je odlišná, a proto je používána tato technologie ke zjednodušení práce. Jedná se o vytvoření objektů pro přístup k datům. Jsou to objekty, které mohou manipulovat a dotazovat se databáze pomocí metod, které vrací výsledek v podobě objektů. Vývojář pak použije předdefinované metody namísto dotazů do databáze. Pokud v databázi nastane změna, stačí přepsat pouze tyto metody. Kdyby se využívalo přímo dotazování do databáze, musel by se měnit kód na více místech.

Ve své práci mi stačí dva takové objekty, protože má databáze obsahuje dvě tabulky. Jsou to objekty userDAO a moduleDAO v serverovém adresáři s názvem model.

Definované metody objektu userDAO:

- **addUser** s parametry user a password
Slouží pro přidání uživatele, tedy pro registraci uživatele s daným heslem.
- **removeUser** s parametrem user
Maže uživatele na základě jeho identifikátoru
- **isUserExist** s parametrem username
Kontroluje zda v databázi je nebo není uživatel s daným username.
- **authenticate** s parametry username a password
Slouží pro autentifikaci uživatele. Vrací uživatele pokud existuje uživatel s daným username a password.

Definované metody objektu `moduleDAO`:

- **getModulesFromUser** s parametrem `user`
Vrátí všechny uložené moduly daného uživatele.
- **addModuleToUser** s parametry `user` a `module`
Uloží nový modul pro daného uživatele.
- **getModule** s parametrem `module`
Vrátí konkrétní modul na základě jeho id.
- **removeModule** s parametrem `module`
Vymaže konkrétní modul na základě jeho id.

6.4 Klientská část

V této kapitole jsou popsány funkcionality, které jsou řešené v klientské části aplikace pomocí javascriptu a knihovny `fabric.js`. Jedná se o práci při modelování železniční tratě v modulu jako manipulace s jednotlivými kolejmi, spojování kolejí a jejich zachytávání k vývodním linkám.

Aplikace obsahuje dva canvasy jeden slouží pro návrh modulu canvas a druhý jako výběrové menu s dostupnými kolejemi `canvasMenu`.

6.4.1 načtení kolejí

Načtení kolejí probíhá v `canvasMenu`. Podle vybrané záložky typu kolejí se vykreslí dané typy do tohoto canvasu. Pro načítání slouží XML soubor `trackCatalog` viz. výpis 6.10 v adresáři `railway-tracks` v `public` složce. V tomto souboru jsou všechny koleje rozdělené po typech. Pro načtení tohoto souboru se dotazují serveru pomocí POST požadavku na URI `getRails`, který mi vrátí obsah souboru v JSON formátu jako objekt, se kterým se na klientské části lépe pracuje. Tento objekt procházím a vyfiltruji dané typy kolejí, které nesou informace o této koleji a název obrázku příslušející koleji.

Popis funkce `addTrail`:

```
function addTrail(canvas, railInfo, callback)
```

parametry této funkce jsou `canvas` do kterého se má vykreslovat, informace o koleji a `callback` předávající objekt `fabricjs`, který představuje kolej. `Callback` pak slouží například pro nastavení pozice koleje v modulu nebo jinou změnu které nemají všechny koleje stejné. Následuje volání `fabric` metody `fabric.loadSVGFromURL` s parametry cesta k SVG obrázku a `callback` funkce vracející vytvořené objekty a možnosti objektů. Tato metoda používá SVG parser, prochází jednotlivé elementy a vytváří jim korespondující `fabricjs` objekty a vrací je vyvoláním `callback` funkce.

```
fabric.loadSVGFromURL(image, function (objects,
    options));
```

Tyto objekty následně procházím a rozdělují je na elementy které slouží ke spojování kolejí a na elementy patřící k obrázku koleje. Z elementů patřících k obrázku koleje vytvořím skupinu SVG elementů pomocí fabric metody fabric.util.groupSVGElements s parametry pole objektů, které se mají seskupit, a možnosti objektů. Z ostatních elementů sloužících pro spojování a vytvořené skupiny SVG elementů odstraním transformační matici pomocí metody objektu `_removeTransformMatrix`.

```
var obj = fabric.util.groupSVGElements(obr, options);
obj._removeTransformMatrix();
```

V SVG se různé transformace objektu jako otáčení, škálování totiž popisují pomocí transformační matice a tato metoda z této matice vypočítá konkrétní hodnoty různých transformací jako úhel otočení. Následně je spojím do obyčejné skupiny pomocí fabric.Group metody s parametry pole objektů a nastavím této skupině vlastní atributy `contacts` a `info`. Atribut `contacts` obsahuje pole elementy, které slouží pro spoj kolejí a atribut `info`, což je objekt udržující informace ze souboru o koleji.

```
var group = new fabric.Group(objects);
group.contacts = contacts;
group.info = {
    type: trailinfo.type,
    image: trailinfo.image,
    width: trailinfo.width,
    height: trailinfo.height,
    name: trailinfo.name,
    size: trailinfo.size
};
```

Výpis 6.10: Ukázka struktury souboru trackCatalog

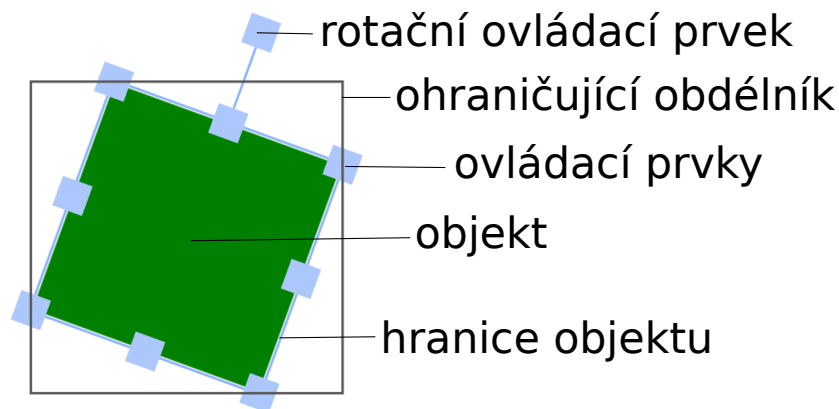
```
<trackCatalog>
  <straight>
    <small>
      <track typeID=\"1\">
        <name>small straight 1/1</name>
        <image>small-straight-1.svg</image>
        <width units=\"cm\">27,4</width>
        <height units=\"cm\">5</height>
      </track>
    </small>
    <large>
      <track typeID=\"4\">
        <name>large straight 1/1</name>
        <image>large-straight-1.svg</image>
        <width units=\"cm\">31,15</width>
        <height units=\"cm\">5</height>
      </track>
    </large>
  </straight>
</trackCatalog>
```

Jednotlivé koleje obsahují informace o názvu koleje, název příslušejícího vektorového obrázku a velikost.

6.4.2 manipulace s kolejemi

Frabic.js má implementovány základní operace manipulace s objekty, některé jsou defaultně nastaveny a některé nejsou. Každý objekt má tzv. bounding rectangle neboli ohraničující čtverec, který udává hranice objektu tak jako pozice objektu zleva, zprava, shora a zdola. Dále má ohraničení objekt pomocí čtverce, na kterém se nachází ovládací prvky. Ve své práci mám povoleno pouze objekt rotovat, a to pomocí rotačního ovládacího prvku. Vybraný prvek se kterým chceme manipulovat je označen ohraničujícím čtvercem. Lze vybrat i skupinu objektů výběrem nebo přidržením klávesy shift a následným po klikáním na dané objekty.

Manipulovat s objekty lze pomocí ovládacích prvků viz. obrázek 6.1) nebo pomocí kláves.



Obrázek 6.1: Ovládací prvky

- otáčení

Výchozí otáčení v knihovně povoluje všechny úhly otáčení. Bylo třeba předělat otáčení pouze na určité úhly, aby se uživateli dobře navrhovalo. Jelikož koleje mají být vyvedeny kolmo a některé koleje mají spojovací prvek pod určitým úhlem, musí být kolej schopna se o tyto úhly otočit. Pro rovné koleje jsou to úhly 0, 90, 180, 360. Pro malou variantu oblouků jsou to úhly 45, 22.5, 11.25 a jejich násobky pro velkou variantu zase 30, 15, 7.5 a jejich násobky kvůli zkracování kolejí na poloviční velikost. Ostatní složitější koleje jsou sestaveny z těchto dvou typů a nejsou pootočený o jiný úhel, než je již definovaný. Pokud by se přidávala nová kolej, která by měla spojovací prvek pod jiným úhlem než jsou definované úhly, musel by se tento úhel a jeho násobky do definičního pole přidat.

Pro otáčení slouží rotační ovládací prvek anebo levá šipka pro otáčení směrem vlevo a pravá šipka pro otáčení směrem vpravo. Pro řešení rotace při použití ovládacího prvku využívám posluchač událostí, kde otáčení probíhá za pomoci pozice myši. Otáčení pomocí šipek je provedeno jiným způsobem, protože pozici myši zde využít nelze.

Řešení s ovládacím prvkem:

Otáčení se dá nastavit pomocí inicializace posluchače události pomocí metody `on` a předání dané události `'object:rotating'`. Pomocí návratové funkce je vrácen objekt `options`, který obsahuje objekt se kterým se manipuluje a událost myši `mousemove` generující se při pohybu myši. Objekt má v obsluze události již nastavený úhel podle pozice myši, který je využíván pro nalezení nejbližší hodnoty z pole vzestupně seřazených definovaných úhlů s názvem `angles`. Při nalezení nejbližšího úhlu už jen stačí nastavit objektu atribut `angle`, pro změnu úhlu.

```
canvas.on('object:rotating', function (options) {
    var obj = options.target;
    var angle = obj.angle;
    for (var i = 0; i < angles.length; i++) {
        if (angle <= angles[i]) {
            obj.angle = angles[i];
            break;
        }
    }
});
```

Řešení s ovládáním šipek:

Aby mohl uživatel otáčet s objektem musí nejprve objekt označit a zvolit šipku. Detekci smáčknutí kláves zajišťuje posluchač událostí onkeydown na stránce. To, jaká klávesa byla smáčknuta na klávesnici lze zjistit pomocí atributu keyCode z předaného objektu z návratové funkce. Označený objekt v kanvasu se dá získat pomocí metody `getActiveObject()`. Z tohoto objektu je zjištěn jeho nynější úhel, a následně pozice tohoto úhlu v definovaném poli přístupných úhlů. Poté je nastaven buďto předešlý úhel z definovaného pole úhlů, pokud je prováděno otáčení vlevo nebo následující úhel při otáčení vpravo.

```
document.onkeydown = function(e){
    if (e.keyCode == '39') {
        var activeObject = canvas.getActiveObject();
        var index = angles.indexOf(activeObject.angle);
        activeObject.setAngle(angles[(index + 1)]);
    }
}
```

- **převrácení**

Pro převrácení objektů v knihovně `fabric.js` není ovládací prvek. Převrácení objektů je nastaveno pouze na klávesy. Převrácení je řešeno jednoduše, protože objekty obsahují atributy `flipX` pro vertikální převrácení a `flipY` pro horizontální převrácení. Nastavují se jím hodnoty typu `BOOL`, kde `true` označuje nepřevrácený objekt a `false` převrácený. Stačí tedy jen změnit atributy objektu `flipX` nebo `flipY` na opačnou hodnotu při smáčknutí klávesy.

- **spojování kolejí**

Spojování kolejí je prováděno při posunu objektů a smáčknuté klávese `shift`. Pro posuny objektů má `fabric` posluchač událostí `object:moving`, kde se přepočítává pozice podle polohy myši. Pro spojování je zde využito spojovacích prvků, které byli nakresleny v SVG obrázku a přidány jako objekty do atributu `contacts`. Při posunu objektem se kontrolují pozice a úhel otočení spojovacích prvků s pozicemi a úhly spojovacích

prvků ostatních objektů v modulu. Pokud jsou od sebe blízko a mají stejný úhel, posune se vybraný objekt tak aby byli oba spojovací prvky na sobě. Pozice u objektů jsou přepočítávány na transformační matici, aby bylo snadné rozeznat, jestli jsou na stejné pozici se stejným úhlem. Protože počátek objektu je nastaven na levý horní roh, pozice objektu jsou pak určeny pouze tímto bodem a při otočení objektu už tato pozice není nejlevějším nebo nejhornějším bodem objektu. K přepočtu na transformační matici slouží metoda `calcTransformMatrix` a převod této matice do čitelné podoby pak pomocí `fabric.util.qrDecompose`. Obě tyto metody jsou součástí knihovny `fabric.js`.

```
var matrix = obj1.calcTransformMatrix();
var iobj1 = fabric.util.qrDecompose(matrix);
```

- **zachytávání kolejí**

Zachytávání lze povolit na celý modul pomocí zapnutí `CapsLock` klávesy nebo pouze na okraje modulu při vypnutí této klávesy. Řešení jsou stejná, pouze při zachytávání na okrajích je kontrolována pozice objektu, zda už není v blízkosti okraje modulu. V tomto problému je také použit převod na transformační matici jako u spojování kolejí. Zachytávání je řešeno také pomocí spojovacích prvků, protože vývod koleje z modulu musí být kolmý, to znamená že spojovací prvek musí být rovnoběžný s okrajem modulu. Pokud nějaký spojovací prvek je rovnoběžný s některým okrajem modulu, zjistí se pozice nejbližší vývodní čáry od tohoto prvku a posune se celá kolej tak aby vývodní čára byla v centru spojovacího prvku. Jeli rovnoběžná s levým nebo pravým okrajem modulu, řeší se pouze pozice shora. Jeli rovnoběžná s horním a dolním okrajem modulu, řeší se pouze pozice zleva. Při zachytávání k okrajům je řešení obdobné, kolej se posune tak aby rovnoběžný spojovací prvek byl vnější půlkou ven z modulu a vnitřní půlkou uvnitř modulu.

6.4.3 Manipulace s modulem

Pro manipulaci s celým modulem je vytvářena skupina se všemi objekty v modulu. S touto skupinou se pak provede příslušné otočení nebo převrácení. To ale nestačí, protože se tato skupina otáčí a převrací vzhledem ke skupině a ne modulu. Skupina má jiné rozměry a výsledek nebude ten očekávaný. Stačí však ale nastavit správnou pozici skupiny v modulu. Při otáčení vlevo nastavíme pozici zleva na předešlou hodnotu pozice shora a pozici shora na předešlou pozici zprava. Podobným způsobem se provádí nastavení pozice při otáčení vpravo. Při převrácení vertikálně se nastaví pozice zleva na předešlou pozici zprava. Podobným způsobem při převrácení horizontálně. Poté co je skupina umístěna na správné místo ji stačí rozdělit zpátky na objekty. Objekty ve skupině lze získat pomocí metody `getObjects`, následně jsou obnoveny stavy

objektů pomocí metody `destroy`. Skupina se smaže pomocí metody `remove`, a všechny objekty ze skupiny už stačí přidat pomocí metody `add` do modulu.

```
var objects = group.getObjects();
group.destroy();
canvas.remove(group);
objects.forEach(function (obj) {
    canvas.add(obj);
})
```

6.4.4 Serializace modulu

Serializace je proces uložení stavu složitějšího objektu nebo instance do podoby, která se dá uložit na datové úložiště nebo odeslat přes síť. V aplikaci je modul ukládán do XML formátu.

Ve formátu XML je kořenový element `module` s následujícími atributy. Uživatelské jméno autora, pokud je uživatel přihlášený, jméno modulu a použité rozestupy mřížek. Kořenový element obsahuje elementy `track`, které znázorňují danou kolej. Atributy koleje slouží k definování správné pozice v modulu a vnořené elementy informace o koleji.

Výpis 6.11: Ukázka uloženího modulu s jednou kolejí v podobě XML

```
<module author="mar" name="test" spacing1="163" spacing2="205">
  <track flipX="false" flipY="false" image="small-straight-1.svg" left=
    "85.21413081282503" rotate="0" top="143.49153322834644">
    <type>straight</type>
    <image>small-straight-1.svg</image>
    <width units="cm">27,4</width>
    <height units="cm">5</height>
    <name>small straight 1/1</name>
    <size>small</size>
  </track>
</module>
```

6.5 Tvorba kolejí

Částí práce byla i tvorba různých typů kolejí. Pro kreslení a ukládání sem zvolila formát SVG. Tento formát `fabric` umí vykreslit jako několik objektů a lze pak lépe řešit problém napojování kolejí nebo zachytávání kolejí k vývodním čarám modulu.

Kreslení probíhalo ve vektorovém grafickém editoru `Inkscape`, který je bezplatný a volně dostupný ke stažení. `Inkscape` mimo grafického editoru obsahuje editor XML. Tyto dva editory jsou spolu propojené, a pokud proběhne změna v jednom editoru, projeví se i ve druhém.

Poměry velikostí kolejí byli spočítány na základě informace ve specifikaci požadavků, kde bylo řečeno že koleje jsou dvojího typu tzv. menší a větší varianty. Menší varianta používá více zakřivené oblouky (výhybky), kde osm

oblouků tvoří kruh o průměru 70 cm a křížení je pod 45° . Větší varianta používá méně zakřivené oblouky, kde dvanáct oblouků tvoří kruh o průměru 119 cm a křížení je pod 30° . Délka velkých rovných kolejí je potom vypočítána jako obvod kruhu dělený počtem oblouků a tyto délky jsou pak děleny po půlkách pro rovné koleje poloviční a čtvrtinové délky. Složitější koleje jako křížovatky nebo výhybky jsou už poskládány z jednodušších kolejí jako oblouky a rovné koleje. Tloušťku kolejnice jsem určila sama na 0,45 cm spolu s velikostí rozteče koleje na 4,1 cm. Celková šířka koleje je tedy 5 cm.

V Inkscape volím pro kreslení jednotky px a měřítko 10 px je rovno 1cm.

Tvorba rovných kolejí: Menší varianta

1. Vytvoření obdélníku 50x274px (výška podle délky koleje $\pi * 70/8$)
2. Vytvoření obdélníku 41x274px (výška podle délky koleje $\pi * 70/8$)
3. Zarovnat oba obdélníky, aby měli stejný střed pomocí zarovnání objektů na střed svisle i vodorovně
4. Označit oba obdélníky a vytvořit křivku pomocí rozdílů těchto obdelníků
5. Vytvoření dvou obdélníků 50x10px a zarovnat je na konce kolejí, kde se dají napojit další koleje, tak aby byl půlkou z koleje ven a půlkou sahal do koleje viz. obrázek 6.2. Tyto dva obdélníky po správném zarovnání je třeba nastavit na průhledné.



Obrázek 6.2: Objekt pro napojení kolejí

6. Pražce mohou být libovolné velikosti, protože nejsou využívány k funkcionalitě v aplikaci. Rozměr jsem nastavila na cca 12,5x63,5px a zarovnala je na horizontální střed koleje.

Větší varianta je tvořena stejně pouze s rozdílnou výškou podle délky koleje $\pi * 119/12$ Kratší typy rovných kolejí se liší také pouze délkou koleje.

Tvorba oblouků:

1. Vytvoření 4 kruhů o velikostech 750px, 741px, 659px, 650px a zarovnat je všechny na střed. v programu lze nastavit start a konec v úhlech pro kruh, který vytvoří výseč. Nastavit start na defaultní 0 a konec na 45.
2. Označit nejmenší a největší kruh a vytvořit křivku pomocí rozdílů objektů. to samé udělat i s druhým největším a druhým nejmenším kruhem.
3. Ze zbylých objektů vytvořit rozdíl a kolejnice jsou hotovy

4. Vytvoření dvou obdélníků 50x10px a zarovnat je na konce kolejí a nastavit pozadí na průhledné.
5. Pražce se nakreslí stejným způsobem jako při tvorbě rovných kolejí, akorát se musí ještě rotovat o určitý úhel

Větší varianta se bude dělat obdobně akorát s jinými velikostmi kruhů, a to 1240px, 1231px, 1149px a 1140px. Úseč bude menší s koncem na 30 stupních. Kratší typy obloukových kolejí se dělají obdobně až na velikost výseče, kde úhel je poloviční nebo čtvrtěční. Tedy pro malou verzi 22,5° a 11,25°. Pro velkou 15° a 7,5°. Ostatní složitější typy už lze poskládat z předchozích kolejí s menšími úpravami pražců a ponecháním vždy jednoho obdélníku u napojovací části koleje.

6.6 Obrazovky a jejich části

Stručné shrnutí a ukázka hlavních částí aplikace. Ukázku hlavních částí lze vidět na obrázcích 8.1, 8.3, 8.2

1. Modul
Návrhové pole představující modul s měřítkem o rozměru 70x70cm. Lze do něj přidávat koleje přetažením z výběrového menu a manipulovat s nimi. Manipulace probíhá pomocí myši, klávesových zkratk nebo některých položek hlavního menu.
2. Výběrové menu
Výběrové menu obsahuje seznam dostupných typů kolejí. Jsou rozříděné podle typu pomocí záložek a podle velikosti podle umístění v tomto menu. Nalevo se nachází koleje s malým rádiusem a napravo s rádiusem velkým. Kliknutím na konkrétní kolej se vytvoří duplikát, který je zachycen ke kurzoru a je nutné ho přetáhnout do modulu a dalším kliknutím ho položit. Další možností, jak umístit kolej do modulu je klasickým přetažením. S kolejí ve výběrovém menu je možno manipulovat pouze pomocí klávesových zkratk.
3. Hlavní menu
V hlavním menu se nachází položky, které manipulují celým modulem, udržují jeho historii, ukládají ho nebo načítají jiný. Dalšími položkami jsou možnosti nastavení aplikace a odkázání na registrační nebo pomocnou stránku.
4. Stavový panel
Stavový panel vypisuje informaci o právě vybraném objektu v modulu. Objektem je myšlena kolej nebo skupina kolejí. Vypisuje postupně informace o názvu koleje, pozice zleva, pozice zdola, úhel otočení, horizontální převrácení a vertikální převrácení.

5. Přihlašovací formulář

Rychlý přihlašovací formulář slouží k autentizaci uživatele a nabídnutí dalších funkcionalit aplikace.

6. Okno nastavení

V tomto okně je možnost nastavení, jak bude vypadat exportovaný obrázek a jaký bude mít formát. Dále se dá v nastavení měnit rozestup mřížek a interval pro automatické ukládání modulu.

7. Seznam použitých kolejí

Obsahuje výpis kolejí, které jsou použity v návrhu modulu.

Testování

Uživatelské testování je jednou z metod, jak zjistit hlavní nedostatky aplikace, popř. chyby, a jak se uživatelům s aplikací pracuje. Spočívá v pozorování chování a práce uživatele s aplikací. Uživatelům se zadají úkoly tzv. scénáře, které se mají pokusit splnit. Pozorování se zaměřuje na to, jestli uživatel vůbec úkol splnil, jak dlouho mu to trvalo, zda musel použít nápovědu pokud je dostupná nebo na to intuitivně přišel sám. Po pozorování jsou shrnuty výsledky a zváženy možnosti úpravy aplikace.

Uživatelům byla předložena aplikace s několika úkoly k splnění. Předtím než se pustili do práce byla jim vysvětlena účelnost aplikace a poukázána možnost se podívat na nápovědu v aplikaci.

7.1 Scénáře

- Scénář 1: Navrhněte jakoukoliv část železniční tratě, využijte spojení kolejí.
- Scénář 2: Uložte svůj návrh jako obrázek ve formátu SVG.
- Scénář 3: Přihlaste se a uložte svůj návrh do aplikace.
- Scénář 4: Přejděte na stránku se svými uloženými návrhy a návrh smažte.

7.2 Vyhodnocení testování

- Poznátky k 1. scénáři:

Uživatelé si nejprve hráli s návrhem modulu. Manipulovali s objekty pomocí myši, jak to aplikace umožňovala. Poté prostudovali rychle nápovědu, zejména tabulku zkratk. Koleje se poté snažili spojovat.

7. TESTOVÁNÍ

Z tohoto chování uživatelů byla přidána na hlavní stránku tabulka se zkratkami a zkráceným popisem co dělají, aby uživatelé hned neztráceli čas čtením a hledáním v nápovědě.

Dalším návrhem pro zlepšení přímo od uživatele byla možnost si modul přiblížovat/oddalovat. Tento návrh by stál za uvážení realizovat, ale z časového důvodu byl vynechán, protože by se jednalo o předělání větší části aplikace. Přiblížení a oddálení lze realizovat případně pomocí prohlížeče.

- Poznátky k 2. scénáři:

Uživatelé nejprve zkoušeli uložení jako obrázku, který má defaultní formát PNG. Poté se snažili najít způsob, jak tento formát změnit. Chvíli jim trvalo, než našli možnosti nastavení aplikace, ale tam jim už vše bylo jasné.

- Poznátky k 3. scénáři:

Uživatelé se registrovali, přihlásili a snadno našli způsob pro uložení návrhu do aplikace.

- Poznátky k 4. scénáři:

Uživatelé hledali, kde by mohli uložené moduly najít. Někteří našli hned, a některým to trvalo déle.

Návrhem od uživatele pro zlepšení přehlednosti bylo přejmenovat položku v menu s názvem modules na něco více vystihujícího. Tato položka byla nakonec změněna na saved modules.

Uživatelská příručka

Návrhář modulových železničních tratí je určen, jak už název napovídá pro návrh částí modelových železničních tratí tzv. modulů. Pro moduly je typické že koleje jsou vyvedeny ven z modulu kolmo pro lepší modularitu při výsledném sestavování reálných modulů. K dispozici pro návrh jsou základní typy kolejí v dvou různých variantách, malé a velké. Malá varianta s více zakřivenými oblouky, kde osm oblouků v kruhu má průměr 70 cm. Velká varianta s méně zakřivenými oblouky, kde dvanáct oblouků v kruhu má průměr 119 cm. Návrh je usnadněn pomocí zachytávání kolejí a snadnou manipulací s kolejemi. Aplikace obsahuje také tabulku použitých kolejí v modulu. Moduly lze ukládat lokálně jako obrázky nebo soubory, které je možno nahrát zpět do aplikace. Při přihlášení je možnost ukládat moduly i na server.

8.1 Seznámení se s webovou aplikací

Pro seznámení s aplikací jsou zde vypsány obrazovky, na které můžete při procházení aplikace narazit. Pozornost je věnována hlavní stránce, protože na této stránce se navrhuje.

Hlavní obrazovka

Na hlavní obrazovce viz. obrázek 8.1 se nachází modul spolu se seznamem dostupných kolejí a se seznamem použitých kolejí v modulu viz. obrázek 8.3. V horním menu je nabídka operací s modulem a odkazy na další obrazovky.

Do modulu lze přidávat koleje přetažením ze seznamu kolejí. Seznam kolejí je rozdělen do záložek podle typu koleje. Na levé straně se nachází koleje menší varianty a na pravé straně koleje větší varianty. V seznamu kolejí je možné s kolejí otáčet a převracet předtím, než je přetažena do modulu. V modulu lze pak spojovat koleje přidržetím klávesy shift. Spojení se provede pokud je v blízkosti kolej, která jde napojit. Dále je defaultně na okrajích modulu kolej přichytávána k vývodním čarám, aby vedla kolmo z modulu. Stejně přichytávání jde nastavit pro celou plochu modulu zapnutím CapsLocku, nebo zrušit

Tabulka 8.1: Klávesové zkratky aplikace

Klávesová zkratka	Použití
Shift	Přidržení této klávesy při pohybu s kolejí se kolej spojí s možnými dalšími kolejemi
CapsLock	Pokud je zaplý, zachycuje koleje k vývodní mřížce po celé ploše modulu, jinak pouze v blízkosti okrajů
Ctrl	Přidržení této klávesy lze pohybovat kolejí bez zachycování
a	Převrátí kolej vertikálně
s	Převrátí kolej horizontálně
šipka vpravo	Otočí kolej směrem doprava
šipka vlevo	Otočí kolej směrem doleva
Delete	Vymaže označenou kolej

přichytávání úplně přidržením klávesy Ctrl. Otáčení kolejí je proveditelné pomocí šipek vpravo a vlevo nebo myši táhnutím za rotační kontrolní prvek. Převrátit kolej lze pomocí kláves a, s. Smazání koleje probíhá při označení koleje nebo skupiny kolejí a smáčknutím klávesy Delete nebo stačí přetáhnout kolej mimo modul. Shrnutí klávesových zkratk můžete nalézt v tabulce 8.1.

V hlavním menu se nachází položky, se kterými lze manipulovat s celým modulem. Seznam položek a jejich použití najdete v tabulce 8.2.

Nastavení

V možnostech nastavení viz. obrázek 8.2 je možné definovat vzhled exportovaného obrázku. Jakou má mít barvu pozadí nebo zda nemá mít pozadí, jestli mají být vidět vývodní čáry a v jakém formátu chce uživatel mít obrázek uložen. Pro formát k obrázku je k dispozici výběr z SVG, PNG a JPEG. Dále je v možnostech nastavení pro vývodní čáry, kde uživatel může zadat jiné rozestupy než jsou. Změněném rozestupu po uložení se ihned projeví v modulu. Aplikace automaticky ukládá modul do paměti prohlížeče každých 30 vteřin, takže při znovu načtení stránky nebo znovu otevření prohlížeče se zobrazí naposledy uložený modul. Interval automatického ukládání lze zmenšit nebo zvětšit v nastaveních.

Registrace

Na stránce registrace je formulář, kam uživatel musí doplnit uživatelské jméno, email a heslo pod kterým chce být registrován. Pokud se registrace povede je uživatel přesměrován na stránku přihlášení. Jestliže uživatel pod takovým uživatelským jménem existuje nebo se registrace nezdaří je vypsána hláška a uživatel musí vyplnit formulář znovu. Když je uživatel přihlášen rozšíří se mu menu o položku s názvem save on server, která uloží modul na server, a o položku s názvem modules, kde může uložené moduly najít.

Modules

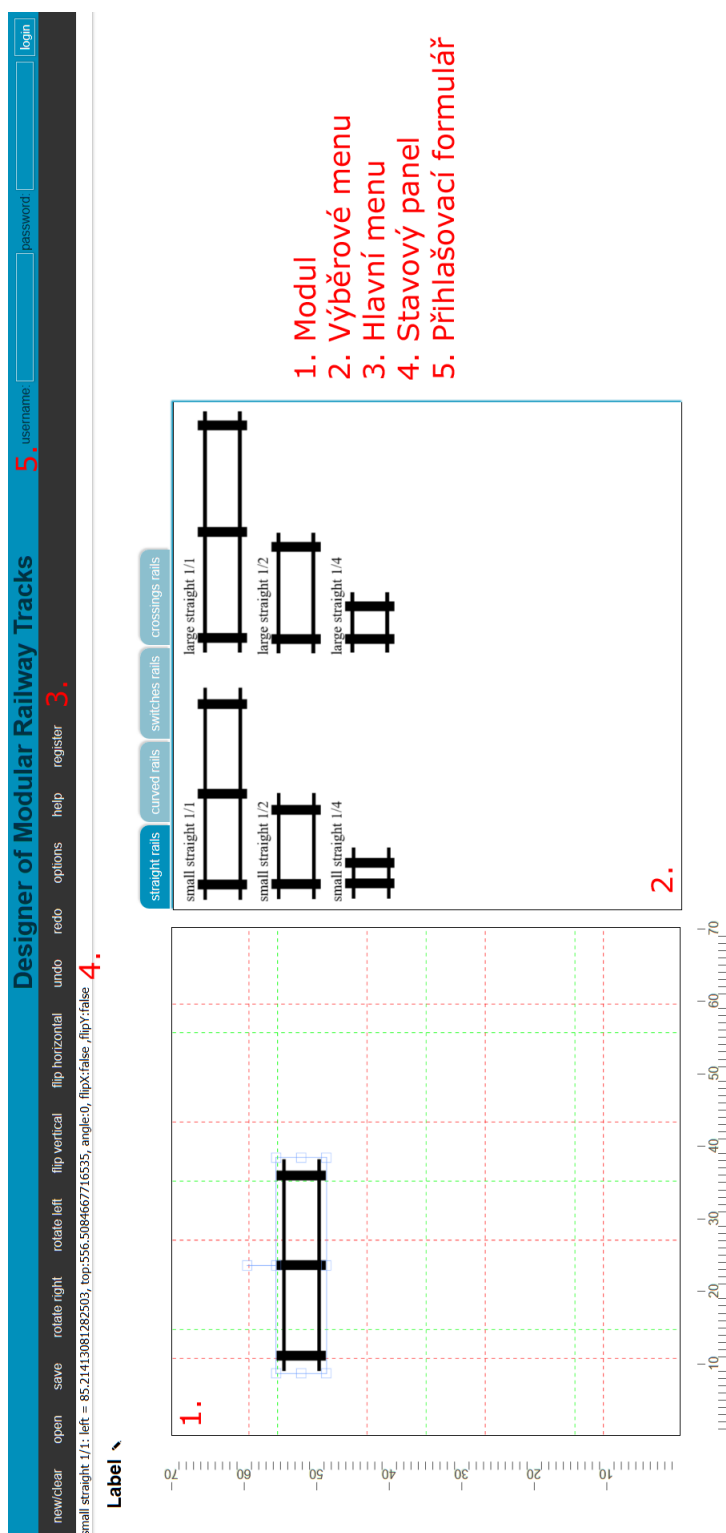
Tabulka 8.2: Menu aplikace

Menu	Použití
new/clear (nový/vymazat)	Vymaže všechny koleje z modulu
open (otevřít)	Vyvolá dialogové okno pro nahrání modulu ze souboru
save (uložit)	Nabídne možnosti uložení
save locally (uložit lokálně)	Vyvolá dialogové okno pro uložení modulu jako souboru lokálně
save on server (uložit na server)	Uloží modul na server, pokud je uživatel přihlášen
save as image (uložit jako obrázek)	Vyvolá dialogové okno pro uložení modulu jako obrázek
rotate left (otočení vlevo)	Otočí celý modul vlevo
rotate right (otočení vpravo)	Otočí celý modul vpravo
flip vertical (převrácení vertikálně)	Převrátí celý modul vertikálně
flip horizontal (převrácení horizontálně)	Převrátí celý modul horizontálně
undo (zpět)	Vrátí změny na modulu zpátky
redo (znovu)	Opakuje vrácenou změnu
options (nastavení)	Zobrazí okno pro nastavení aplikace
help (pomoc)	Otevře stránku se seznámením s aplikací
register (registrace)	Otevře stránku s registračním formulářem

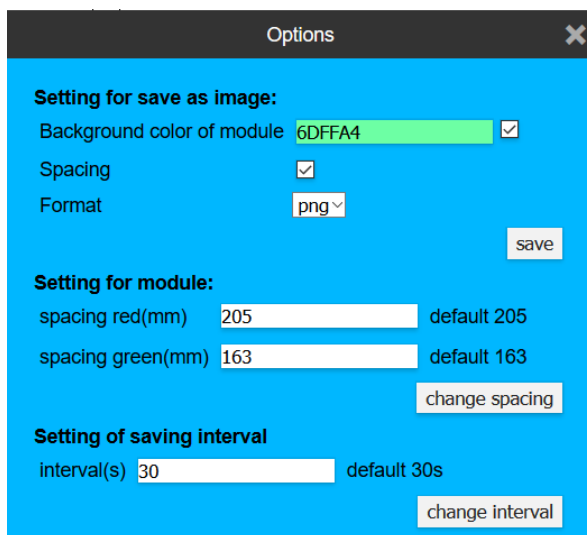
Na stránce s uloženými moduly jsou vidět miniatury modulu jako obrázku spolu s názvem a datem uložení. Modul může uživatel odstranit pomocí křížku v pravém horním rohu. Modul také může stáhnout buď jako obrázek nebo jako soubor. Při kliknutí na miniaturu modulu se stránka přesměruje na hlavní a načte do modulu vybraný modul.

Help

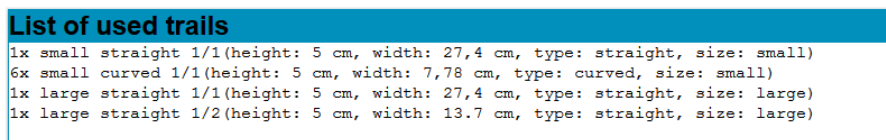
Na stránce nápovědy jsou k nalezení informace o webové aplikaci, s popisem hlavních prvků na stránce, klávesových zkratech a zacházení s kolejiemi v modulu.



Obrázek 8.1: Hlavní obrazovka



Obrázek 8.2: Okno nastavení



Obrázek 8.3: Seznam použitých kolejí

Závěr

Byla vytvořena webová aplikace, která umožňuje interaktivní návrh kolejiště pro modelové železnice. Aplikace obsahuje základní typy kolejí, které se dají rozšířit. S koleji lze snadno manipulovat a sestavovat části kolejišť tak, že splňují kritéria ze specifikace požadavků. Aplikace je schopna ukládat a načítat návrhy ve formátu XML. Návrhy je možné exportovat do různých typů obrázků a upravovat jejich výsledný vzhled. Dále byla vytvořena uživatelská příručka pro tuto webovou aplikaci a bylo provedeno uživatelské testování použitelnosti. Práce splnila všechny části zadání.

Literatura

- [1] KŽM Praha 3: *Definice modulového kolejiště [online]*. [cit. 2017-05-15]. Dostupné z: <http://www.kzm-praha.cz/krouzek/moduly.htm>
- [2] KŽM Praha 3: *Modulové kolejiště TT (Německo, 2001) [online]*. [cit. 2017-05-15]. Dostupné z: http://www.kzm-praha.cz/krouzek/img/VB_modul_01e.JPG
- [3] KŽM Praha 3: *První modul TT kolejiště KŽM Praha 3 byl postaven 6/2001 [online]*. [cit. 2017-05-15]. Dostupné z: <http://www.kzm-praha.cz/krouzek/img/demomodul1.jpg>
- [4] Copyright © 2017 DRail Modelspoor Software: *About AnyRail [online]*. [cit. 2017-05-15]. Dostupné z: https://www.anyrail.com/about_us_en.html
- [5] Ing. Büro Schneider: *About WinTrack [online]*. [cit. 2017-05-15]. Dostupné z: <http://www.wintrack.de/>
- [6] Peev, M.: *About SCARM [online]*. [cit. 2017-05-15]. Dostupné z: <http://www.scarm.info/index.php>
- [7] van Eldijk, F.: *About TRAX [online]*. [cit. 2017-05-15]. Dostupné z: <https://www.traxeditor.com/en/about/index.html>
- [8] Node.js Foundation: *About Node.js [online]*. [cit. 2017-05-15]. Dostupné z: <https://nodejs.org/en/>
- [9] Devel.cz Lab s.r.o.: *JavaScript na serveru: Architektura a první Hello World [online]*. [cit. 2017-05-15]. Dostupné z: <https://www.zdrojak.cz/clanky/javascript-na-serveru-architektura-a-prvni-hello-world/>
- [10] Node.js Foundation: *About Express [online]*. [cit. 2017-05-15]. Dostupné z: <http://expressjs.com/>

LITERATURA

- [11] Fabric.js: *Introduction to Fabric.js [online]*. [cit. 2017-05-15]. Dostupné z: http://fabricjs.com/fabric-intro-part-1#why_fabric
- [12] Node.js: *Installing Node.js Express [online]*. [cit. 2017-05-15]. Dostupné z: <http://expressjs.com/en/starter/installing.html>
- [13] Devel.cz Lab s.r.o.: *Třídy, dědičnost a OOP v Javascriptu [online]*. [cit. 2017-05-15]. Dostupné z: <https://www.zdrojak.cz/clanky/oop-v-javascriptu-i/>

Seznam použitých zkratk

- GUI** Graphical user interface
- XML** Extensible markup language
- JSON** JavaScript object notation
- SVG** JavaScript object notation
- HTML** HyperText markup language
- SQL** Structured query language
- PNG** Portable network graphics
- JPEG** Joint photographic experts group
- NPM** Node Package Manager

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
├─ impl.....	zdrojové kódy implementace
├─ database.sql.....	export databáze
├─ inkscape.svg.....	projekt Inkscape s nakreslenými koleji
├─ thesis.....	zdrojová forma práce ve formátu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$
text.....	text práce
├─ thesis.pdf.....	text práce ve formátu PDF