

Vysoká škola logistiky o.p.s.

**Vizualizace provozu železnice pomocí
auto ID**

(Bakalářská práce)



**Vysoká škola
logistiky**
o.p.s.

Zadání bakalářské práce

student	Tomáš Fejfar, DiS.
studijní program obor	Logistika Informační management

Vedoucí Katedry bakalářského studia Vám ve smyslu čl. 22 Studijního a zkušebního řádu Vysoké školy logistiky o.p.s. pro studium v bakalářském studijním programu určuje tuto bakalářskou práci:

Název tématu: **Vizualizace provozu železnice pomocí auto ID**

Cíl práce:

Práce předkládá řešení tvorby aplikace pro vizualizaci a identifikaci objektů na modelové železnici. Aplikace bude zobrazovat požadované informace ve webovém rozhraní prohlížeče.

Zásady pro vypracování:

Využijte teoretických východisek oboru logistika. Čerpejte z literatury doporučené vedoucím práce a při zpracování práce postupujte v souladu s pokyny VŠLG a doporučeními vedoucího práce. Části práce využívající neveřejné informace uveďte v samostatné příloze.

Bakalářskou práci zpracujte v těchto bodech:

- Úvod
- 1. Možnosti automatické identifikace
- 2. Použité technologie
- 3. Analýza požadavků na aplikaci
- 4. Tvorba aplikace
- 5. Vyhodnocení a využití aplikace
- Závěr

Rozsah práce: 35 – 40 stran textu

Seznam odborné literatury:

GROS, Ivan a kol. Velká kniha logistiky. Vyd. 1 VŠCHT Praha, 2016. 512 s. ISBN 978-80-7080-952-5

KOFLER, Michael a Bernd ÖGGL. PHP 5 a MySQL 5: průvodce webového programátora. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.

ŠVARCOVÁ, Ivana a Tomáš RAIN. Informační management. Praha: Alfa Nakladatelství, 2011. Informatika (Alfa Nakladatelství). ISBN 978-80-87197-40-0.

Vedoucí bakalářské práce:

Ing. Libor Kavka, Ph.D.

Datum zadání bakalářské práce:

31. 10. 2017

Datum odevzdání bakalářské práce:

5. 5. 2018

Přerov 31. 10. 2017



Ing. et Ing. Iveta Dočkalíková, Ph.D.
vedoucí katedry



doc. Ing. Ivan Hlavoň, CSc.
rektor

Čestné prohlášení

Prohlašuji, že předložená bakalářská práce je původní a že jsem ji vypracoval samostatně. Prohlašuji, že citace použitých pramenů je úplná a že jsem v práci neporušil autorská práva ve smyslu zákona č. 121/2000 Sb., o autorském právu, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů.

Prohlašuji, že jsem byl také seznámen s tím, že se na mou bakalářskou práci plně vztahuje zákon č. 121/2000 Sb., o právu autorském, právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, zejména § 60 – školní dílo. Beru na vědomí, že Vysoká škola logistiky o.p.s. nezasahuje do mých autorských práv užitím mé bakalářské práce pro pedagogické, vědecké a prezentační účely školy. Užiji-li svou bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti Vysokou školu logistiky o.p.s.

Prohlašuji, že jsem byl poučen o tom, že bakalářská práce je veřejná ve smyslu zákona č. 111/1998 Sb., o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších předpisů, zejména § 47b. Taktéž dávám souhlas Vysoké škole logistiky o.p.s. ke zpřístupnění mnou zpracované bakalářské práce v její tištěné i elektronické verzi. Tímto prohlášením souhlasím s případným použitím této práce Vysokou školou logistiky o.p.s. pro pedagogické, vědecké a prezentační účely.

V Přerově, dne 18. 08. 2018

.....

podpis

Poděkování

Děkuji tímto panu Ing. Liboru Kavkovi, Ph.D za rady a nápady při zpracování mé bakalářské práce.

Anotace

Díky modelové železnici, umístěné na Vysoké školy logistiky v Přerově, přichází myšlenka na vytvoření vizuálního zpracování informací zapsaných do databáze pomocí RFID antén umístěných na modelu železnice. Pomocí modelové železnice je možné zkusit fungování RFID komunikace v praxi.

V první části bakalářská práce informuje o možnostech automatické identifikace. V dalších částech práce je kompletní návrh a vytvoření aplikace vizualizace provozu na železnici pomocí webového prohlížeče.

Klíčová slova

Identifikace, železnice, vizualizace, aplikace, model, web, databáze

Annotation

Thanks to the railway model located at VŠLG Přerov, comes the idea to create a visual processing of Information entered into the database using RFID antennas, which are placed on the railway model. Using railway model is possible to test the functioning of RFID communication in practice.

In the first part the bachelor thesis informs about possibilities of automatic identification. In the other parts of the thesis is a complete design and creation of a traffic visualization application on the railway using a web browser.

Keywords

Identification, railway, visualization, application, model, web site, database

Obsah

Seznam ilustrací a tabulek	9
Seznam zkratk a značek	10
Terminologie.....	11
Úvod.....	12
1 Možnosti automatické identifikace	13
1.1 Technologie čárového kódu	13
1.1.1 Code 2/5	14
1.1.2 UPC.....	15
1.1.3 Code 39	16
1.1.4 EAN	17
1.1.5 Code 11	18
1.1.6 Code 128	18
1.1.7 PDF417	19
1.1.8 Data Matrix	20
1.1.9 QR code	20
1.2 Technologie RFID.....	21
1.2.1 Typy RFID transponderů	22
1.2.2 Kategorie RFID čteček	23
1.3 Technologie NFC.....	23
1.4 Technologie OCR	24
2 Použité technologie	25
2.1 (X)HTML.....	25
2.1.1 HTML 5	25
2.1.2 HTML 4.01	26
2.1.3 XHTML	26
2.2 CSS	27

2.2.1	CSS 1	28
2.2.2	CSS 2	28
2.2.3	CSS 3	29
2.3	APACHE.....	29
2.3.1	PHP	29
2.3.2	MySQL	29
2.3.3	Javascript.....	30
2.4	PSPad	30
2.5	Model železniční dopravy	30
3	Analýza požadavků na aplikaci.....	31
3.1	Analýza proměnných	31
4	Tvorba aplikace.....	33
4.1	Návrh aplikace	33
4.2	Struktura databáze.....	33
4.3	Grafický návrh	34
4.4	Přímá tvorba aplikace.....	36
4.4.1	Příprava HTML.....	36
4.4.2	Příprava CSS	37
4.4.3	Příprava PHP.....	38
4.4.4	Uživatelský výstup.....	42
5	Vyhodnocení a využití aplikace	45
	Závěr	46
	Soupis použitých zdrojů.....	47
	Tištěné zdroje:	47
	Internetové zdroje:	47
	Ostatní zdroje:	48

Seznam ilustrací a tabulek

Obrázek 1.1 Znázornění Code 2/5	14
Obrázek 1.2 Znázornění UPC	16
Obrázek 1.3 Znázornění Code 39	16
Obrázek 1.4 Znázornění EAN-13	17
Obrázek 1.5 Znázornění EAN-8	17
Obrázek 1.6 Znázornění Code 11	18
Obrázek 1.7 Znázornění Code 128	19
Obrázek 1.8 Vysvětlení PDF417	19
Obrázek 1.9 Znázornění Data Matrix	20
Obrázek 1.10 Znázornění QR code.....	21
Obrázek 2.1 Ukázka a popis HTML kódu	25
Obrázek 2.2 Příklad nových tagů v HTML 5	26
Obrázek 2.3 Složení CSS pravidla.....	27
Obrázek 3.1 Požadavky aplikace	31
Obrázek 3.2 Seznam proměnných	32
Obrázek 4.1 Struktura tabulky events. Označeny potřebné sloupce z tabulky.....	34
Obrázek 4.2 Grafické rozvržení webové stránky.....	34
Obrázek 4.3 Připravené obrázky vlaků a vagonů	35
Obrázek 4.4 Kompletní kód HTML	36
Obrázek 4.5 První část CSS stylu	37
Obrázek 4.6 Druhá část CSS stylu.....	38
Obrázek 4.7 První část PHP kódu.....	40
Obrázek 4.8 Druhá část PHP kódu	41
Obrázek 4.9 Nové načtení webové stránky	42
Obrázek 4.10 Zadání položky datum	43
Obrázek 4.11 Zadání položky datum a čas	43
Tabulka 1.1 Kódovací tabulka Code 2/5	15

Seznam zkratek a značek

ASCII	anglická zkratka pro American Standard Code for Information Interchange, kódová tabulka definující znaky anglické abecedy a další používané znaky v informatice
CSS	Cascading Style Sheets, kaskádové styly
EAN	European Article Number
HTML	HyperText Markup Language
ID	identifikační číslo
ISO	International Organization for Standardization, mezinárodní organizace pro normalizaci
ISO/IEC 18092	definice komunikačních módů pro NFC
MySQL	My Structured Query Language
NFC	Near Field Communication
OCR	Optical Character Recognition
PDF417	Portable Data File 417
PHP	Hypertext Preprocessor, dříve Personal Home Page
PSPad	PSPad Editor
QR code	Quick Response kód
RFID	Radio Frequency Identification
UPC	Universal Product Code
URL	Uniform Resource Locator
XHTML	eXtensible HyperText Markup Language

Terminologie

Atribut	vlastnost objektu
Deklarace	označení jména a datového typu
Element	jednodušší část prvku
Hlavička HTML	určuje základní údaje o HTML souboru
Linux	otevřený operační systém
Multimédia	jedná se o video, fotografie, nosiče CD a DVD
Proměnná	označení pro identifikaci prvku
Příkaz	prvek vyjadřující nějakou činnost, která má být provedena
RFID transponder/tag	čip obsahující informace, které je možné získat pomocí RFID čtečky
Selektor	umožňuje označení prvku na webové stránce
Server	počítač poskytující služby pro uživatele
Skript	prostý text spustitelný jako soubor převeden v daném skriptovacím jazyce
Software	počítačový program
Vizualizace	zobrazování skutečnosti
Windows	nejrozšířenější operační systém od společnosti Microsoft

Úvod

Na modelové železnici umístěné v prostorách Vysoké školy logistiky v Přerově je možné za dozoru vyučujících manipulovat s lokomotivami a vagony. Je možné spustit model a tím se aktivuje systém, zapisující všechny údaje do připravené databáze.

V této práci je na základě připravené databáze, do které je zapisováno, jako cíl určena vizualizace a identifikace objektů na železnici. Aplikace bude zobrazovat požadované informace ve webovém prohlížeči.

V teoretické části jsou popsány možnosti automatické identifikace od čárových kódů až po NFC technologii.

V praktické části je popsána kompletní tvorba požadované aplikace.

1 Možnosti automatické identifikace

Automatická identifikace je technologie založená na radiofrekvenčním, optickém, magnetickém, případně dalším principu. Technologie se používá pro identifikaci objektů, získávání informací a ukládání přímo do počítačového systému.

V logistice se automatická identifikace používá pro jednodušší vyhledávání předmětů a míst pro jejich uložení. V maloobchodních prodejnách se používá při příjmu a výdeji zboží optický systém. Kontroluje se stav zásob. U modelové železnice se kontroluje složení vlaků, ze které je použita databáze pro tuto bakalářskou práci.

1.1 Technologie čárového kódu

Čárový kód je nejvíce používaným prostředkem automatické identifikace. Používá se pro sběr dat o určitém objektu. Pro člověka je nemožné přečíst čárový kód, stejně jako u všech technologií pro automatickou identifikaci. Pro čtení se používají speciální čtečky či skenery. Při manipulaci s čárovými kódy je důležitá opatrnost, protože jsou náchylné na poškození, a následně může být problém s jejich čtením.

Čárový kód je tvořen černými pruhy a bílými pruhy. Tento systém má svojí logiku, pomocí které jsou schopny čtečky či skenery rozeznat, o co se jedná v daném čárovém kódu. Černý i bílý pruh má rozdílnou šířku a pomocí těchto kombinací se dokáže rozlišit, o jaký znak se jedná. Každý kód se skládá z prvního znaku start a posledního znaku stop. Mezi těmito znaky jsou zbylé informace. Černé pruhy se používají z důvodu pohlcování světla a bílé pruhy se používají z důvodu odražení světla. Odraz a pohlcování světla je důležité, protože čtečky a skenery vyzařují určité světlo, a právě pomocí odrazu se rozpozná, o jaké informace se v daném čárovém kódu jedná.

Čárové kódy dělíme do několika skupin podle toho, jak se kódují¹:

- Code 2/5
- UPC
- Code 39
- EAN
- Code 11

¹ Zdroj: <https://cs.wikipedia.org>

- Code 128
- PDF417
- Data Matrix
- QR Code

1.1.1 Code 2/5²

Kód 2/5 je nejstarší ze všech čárových kódů. Byl vyvinut v roce 1968. Složení kódu 2/5 je pouze z hodnot start, stop a číselných znaků od 0 do 9.

Obrázek 1.1 Znárodnění Code 2/5



Zdroj: https://en.wikipedia.org/wiki/Two-out-of-five_code

Na obrázku 1.1 je možné si všimnout složení každého čísla. Každé číslo je složeno z 5 černých pruhů, kdy vždy jsou 3 úzké a 2 široké.

V roce 1972 byl vyvinut nový kód 2/5 nazvaný prokládaný 2/5. Bylo použito kódování i pomocí mezer a mohl se tam využívat větší prostor čárového kódu.

² Zdroj: <https://cs.wikipedia.org>

Tabulka 1.1 Kódovací tabulka Code 2/5

Znak	C1	C2	C3	C4	C5
0	0	0	1	1	0
1	1	0	0	0	1
2	0	1	0	0	1
3	1	1	0	0	0
4	0	0	1	0	1
5	1	0	1	0	0
6	0	1	1	0	0
7	0	0	0	1	1
8	1	0	0	1	0
9	0	1	0	1	0
Start	1	1	0		
Stop	1	0	1		

Zdroj: vlastní zpracování podle <https://en.wikipedia.org/wiki/Barcode> v českém jazyce

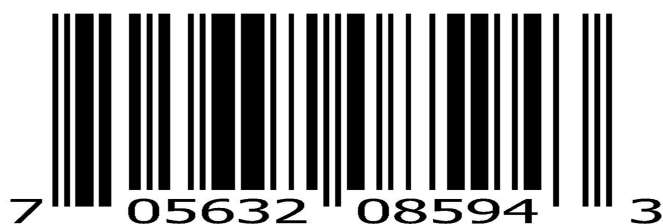
1.1.2 UPC³

UPC kód byl vyvinut v roce 1973 a byl úspěšně zaveden v supermarketech. Tento kód má dostupných 5 verzí: UPC⁴ A, B, C, D a E. Tento kód se skládá z 12místného čísla, kdy první číslice určuje znak systému číslování, 5 následujících čísel je pro identifikaci výrobce, dalších 5 je číslo výrobku a poslední číslice je kontrolní znak. V aktuální době se UPC používají výhradně v USA nebo v Kanadě.

³ Zdroj: <https://cs.wikipedia.org>

⁴ UPC – Universal Product Code, používán s kódy EAN a další.

Obrázek 1.2 Znárodnění UPC



Zdroj: <https://barcodes.cz/carove-kody-upc/?lang=cs>

1.1.3 Code 39⁵

Kód 39 byl vyvinut v roce 1974. Jako první začal tento čárový kód používat číslice, písmena a speciální znaky (-, ., \$, /, +, % a mezera). Start a stop není u tohoto kódu řešen číslicí, ale pomocí znaku hvězdičky (*). U kódu 2/5 bylo uvedeno, že každý znak je rozpoznán 3 úzkými a 2 širokými pruhy. U kódu 39 je to změněno, kdy každý znak je kódován 5 pruhy a 4 mezerami. Šířka pruhu a šířka mezery určuje, o jaký znak se bude jednat. Je 39 různých kombinací pruhů a mezer, které vyjadřují znak. Pro kód 39 existuje rozšíření tzv. modulo 43, které obsahuje kontrolu a je schopno opravit případné chyby.

Obrázek 1.3 Znárodnění Code 39



Zdroj: https://en.wikipedia.org/wiki/Code_39

⁵ Zdroj: <http://www.gaben.cz/cz>

1.1.4 EAN⁶⁷

EAN je nejrozšířenější čárový kód. Byl vyvinut v Evropě v letech 1976. EAN má 2 možnosti používání. Jsou jimi EAN-13 a EAN-8. Dle názvu se dá usoudit, že EAN-13 obsahuje celkem 13 znaků a EAN-8 obsahuje 8 znaků. Kratší verze EAN jsou používány pro méně rozměrné výrobky.

První 3 pozice kódu obsahují vždy číslo země, kde je výrobce zaregistrovaný. Následující 4 až 6 číslic určují přímo výrobce. Zbývající číslice, kromě úplně poslední číslice, určují číslo výrobku. Poslední číslice je používána pro kontrolu správnosti kódu. Každý znak EAN kódu určují vždy 2 černé pruhy a 2 bílé mezery.

Obrázek 1.4 Znárodnění EAN-13



Zdroj: <http://www.gaben.cz/cz/faq/carove-kody-teorie#ean-13-a-ean-8>

Obrázek 1.5 Znárodnění EAN-8



Zdroj: <http://www.gaben.cz/cz/faq/carove-kody-teorie#ean-13-a-ean-8>

⁶ Zdroj: <https://www.kodys.cz>

⁷ EAN – European Article Number, používán se systémy Code 11 a další.

1.1.5 Code 11⁸

Kód 11 se nepoužívá jako ostatní čárové kódy v obchodní sféře, ale používá se převážně pro označování telekomunikačních zařízení od roku 1978. Skládá se stejně jako všechny ostatní čárové kódy z číslic od 0 do 9, obsahuje ještě znak pomlčky (-) a samozřejmě znak Start a Stop. Každý znak obsahuje 3 černé a 2 bílé pruhy a kombinace těchto prvků jsou 2 široké a 3 úzké pruhy nebo 1 široký a 4 úzké pruhy. Jiné možnosti nejsou.

Pro kontrolu jsou zde dvě možnosti s tím, že jedna možnost se používá při kódu kratším jak 10 znaků, a druhá používá při delším kódu. Obě tyto možnosti používá kontrolní součet dělení číslem 11, tzv. modulo 11 a následným zbytkem je kontrolní číslice.

Obrázek 1.6 Znárodnění Code 11



Zdroj: https://en.wikipedia.org/wiki/Code_11

1.1.6 Code 128⁹

Kód 128 byl vyvinut v roce 1981 a používá se hlavně v logistice. Dle názvu se dá poznat, že může obsahovat až 128 zakódovaných znaků. Každý znak obsahuje 3 černé pruhy a 3 mezery. Kód může obsahovat písmena, číslice a speciální znaky, vše bráno z ASCII¹⁰ tabulky.

Jsou 3 varianty kódu:

- 128A, který používá znaky ASCII tabulky od pozice 0 do pozice 95.
- 128B, který používá znaky ASCII tabulky od pozice 32 do pozice 127
- 128C, který používá znaky ASCII tabulky od pozice 0 do pozice 99

⁸ Zdroj: <http://www.barcodeisland.com>

⁹ Zdroj: <https://en.wikipedia.org>

¹⁰ ASCII – American Standard Code for Information Interchange – kódová tabulka používaná pro definici anglických znaků a dalších znaků používaných v informatice

Obrázek 1.7 Znárodnění Code 128

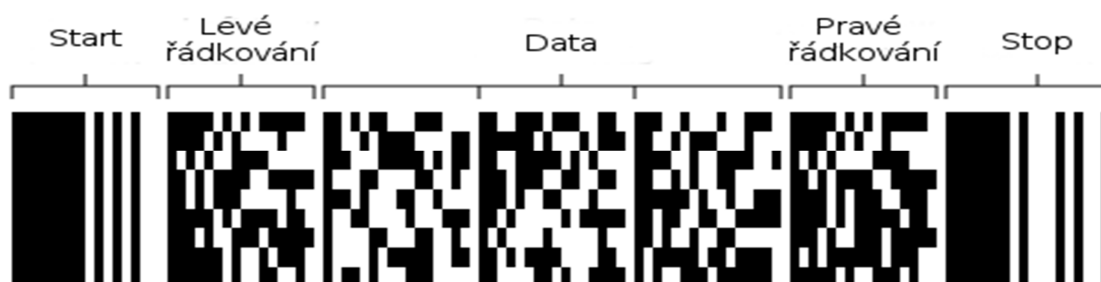


Zdroj: https://en.wikipedia.org/wiki/Code_128

1.1.7 PDF417¹¹

Kód PDF417 byl vyvinut v roce 1991. Je to první z dvourozměrných čárových kódů¹². Z tohoto vyplývá, že není jako u jednorozměrných čárových kódů použit jen jeden řádek, ale můžeme používat od 3 až do 90 řádků. Každý znak obsahuje 4 černé pruhy a 4 bílé mezery. Pomocí tohoto kódu nemusíme odkazovat jen na databázi jako u předešlých jednorozměrných kódů, ale tento kód může obsahovat potřebné informace i sám o sobě. Vzhled dvourozměrných kódů je pro člověka k nepochopení. Vše je lépe vysvětleno na obrázku 1.8 a popsáno v následujících řádcích.

Obrázek 1.8 Vysvětlení PDF417



Zdroj: vlastní zpracování podle <https://en.wikipedia.org/wiki/PDF417>

- Start – identifikuje počátek čtení řádku
- Levé řádkování – obsahuje informace o čísle řádku a případné opravě
- Data – hlavní oblast obsahující zakódovaná data
- Pravé řádkování – obsahuje další informace o řádku

¹¹ Zdroj: <https://en.wikipedia.org>

¹² PDF417, DataMatrix, QR code

- Stop – identifikuje ukončení čtení řádku

1.1.8 Data Matrix¹³

Data Matrix byl vyvinut v roce 2005. Pro snímání štítku Data Matrix se už nepoužívá klasický skener, ale je použita digitální kamera, pomocí které je zvýšena vzdálenost přečtení kódu až na 15m. Díky možnosti čtení pomocí kamery není rozhodující úhel čtení kódu. Můžeme kód natáčet jakýmkoliv směrem a vždy bude správně přečten. Velikost štítku je 8x8 až 144x144 bloků, kdy každý blok je prezentován černým nebo bílým čtvercem. Na obrázku 1.9 je možné si všimnout 2 stejných stran s černými bloky a 2 stejných stran se střídajícími bloky. Výhoda tohoto kódování je, že není důležité, zda je štítek vytištěn na tiskárně nebo je vyryt pomocí laseru do kovové desky. Toto kódování se používá pro označování elektronických součástek.

Obrázek 1.9 Znárodnění Data Matrix



Zdroj: https://en.wikipedia.org/wiki/Data_Matrix

1.1.9 QR code¹⁴

QR kód byl vyvinut v roce 1994. Jak název napovídá, jedná se o velmi rychlý kód. První použití bylo v automobilovém průmyslu, ale později se zjistilo, že se dá používat i pro marketing či další oblasti. Do QR kódu můžeme zakódovat text dlouhý až 4300 znaků. V dnešní době se do QR kódu zaznamenávají různé doplňující informace v časopisech, na vizitkách a webových stránkách. Je možnost zakódovat do QR kódu

¹³ Zdroj: <http://www.gaben.cz>

¹⁴ QR code – Quick Response code, patří mezi dvourozměrné čárové kódy

jen URL¹⁵ adresu. V případě vizitky je možnost přidání informací do mobilního telefonu.

QR kód je vždy čtvercový od velikosti 21x21 bloků. První verze QR kódu měla právě velikost 21x21 bloků a každá další verze se zvětšovala o 4 bloky. Aktuální verze nese číslo 40 a tak její velikost je 177x177 bloků.

Obrázek 1.10 Znárodnění QR code



Zdroj: https://en.wikipedia.org/wiki/QR_code

1.2 Technologie RFID¹⁶

RFID je nástupcem čárových kódů. Používá se pro komunikaci na krátkou vzdálenost. Oficiálně byla technologie RFID uvedena v roce 1983 pro obchodní účely, vývoj začal již v polovině 20. století. RFID používá pro komunikaci transponder a čtečku. RFID transpondery je možné použít pro čtení nebo kombinaci čtení a zápisu. RFID transponder je důležitým nosičem kódu. Obsahuje uložená zakódovaná data a je umístěn na objektu, který je identifikován.

RFID čtečka, jak už název napovídá, slouží ke čtení dat z RFID transponderu. Zároveň se pomocí čtečky komunikuje se systémem.

Výhody RFID¹⁷:

- Rychlá manipulace

¹⁵ URL – Uniform Resource Locator – nejpoužívanější jako řetězec znaků odkazující na webovou stránku.

¹⁶ RFID – Radio-frequency identification používá rádiovou frekvenci pro identifikaci

¹⁷ Zdroj: <https://www.eprin.cz/rfid-technologie.html>

- Možnost snímat více objektů v jeden moment
- Možnost řízení zásob
- Možnost chránit zboží před odcizením
- Možnost číst transpondery bez přímé viditelnosti

1.2.1 Typy RFID transponderů¹⁸

RFID čipy jsou rozděleny podle způsobu přenosu a napájení na pasivní, aktivní a polopasivní.

1.2.1.1 Pasivní čipy

Jak název napovídá, pasivní čipy nemají vlastní zdroj napájení. Všechny informace mají uložené na integrovaném obvodu a obsahují pouze stálé informace zapsané do paměti čipu. Pasivní čipy mají nevýhodu ve vzdálenosti čtení. Jelikož nemají napájení, tak je možné číst čipy jen v řádu desítek centimetrů až jednotek metrů. V obou případech záleží na velikosti a síle čtečky a její frekvenci. Pasivní čipy je možné nalézt na oblečení v obchodech.

1.2.1.2 Aktivní čipy

Aktivní čipy obsahují napájecí zdroj, který zajišťuje provoz vysílače. Snižuje to nároky na signál vysílaný čtečkou, jelikož transponder dokáže vysílat informace. RFID čtečka může zároveň prepisovat údaje uložené na čipu. Vzdálenost čtení se prodlužuje až na 100 m.

1.2.1.3 Polopasivní čipy

Polopasivní čipy obsahují také napájecí zdroj, ale ten není stále využíván jako u aktivních čipů. Důvod je šetření energií v době nenačítání RFID čtečkou. Když je použita čtečka a je vyslán rádiový signál, tak se transponder spustí a odešle informace.

¹⁸ Zdroj: GROS, Ivan. *Velká kniha logistiky*. Praha: Vysoká škola chemicko-technologická v Praze, 2016. ISBN 978-80-7080-952-5.

1.2.2 Kategorie RFID čteček¹⁹

RFID čtečky dělíme do dvou hlavních kategorií podle jejich vlastností a podmínkami použití.

1.2.2.1 Stacionární RFID čtečky

Používají se pro komplexnější zařízení. Jsou používány ve výrobních a logistických aplikacích. Je možnost připojit více vnějších antén, díky kterým je dosaženo lepšího pokrytí prostoru. Stacionární RFID čtečky mohou mít vysokou odolnost v náročných podmínkách.

1.2.2.2 Mobilní RFID čtečky

Jedná se o přenosná zařízení, která ukládají snímaná data do vnitřní paměti. Takováto zařízení jsou většinou vybavena i čtečkami čárových kódů. Nejčastěji se používají v logistice či průmyslové výrobě.

1.3 Technologie NFC²⁰

NFC je nástupcem technologie RFID. První použití pro spotřebitele bylo v roce 2007. Jedná se o bezdrátovou technologii na krátkou vzdálenost do 4 cm. Je tedy důležité přiblížit přístroje. NFC je obsaženo hlavně v mobilních telefonech, kde je možné přenášet data mezi přístroji nebo možnost platit pomocí mobilního telefonu u platebního terminálu u obchodníka.

Pro technologii NFC jsou důležité standardy ISO²¹. Nejdůležitější je standard ISO/IEC 18092²², který definovala společnost NFC Forum²³. Jedná se o definování režimů pro bezdrátové komunikační rozhraní, jako jsou kódování, přenosové rychlosti, modulační schémata a transportní protokoly.

¹⁹ Zdroj: Hájek, Tomáš. Systémy automatické identifikace. Praha, 2010. Bakalářská práce. Vysoká škola regionálního rozvoje a Bankovní institut – AMBIS, a.s.. Vedoucí práce Ing. Vladimír Beneš, Ph.D..

²⁰ NFC – Near field communication, bezdrátová komunikace mezi zařízeními

²¹ ISO – International Organization for Standardization, jedná se o mezinárodní organizaci pro normalizaci.

²² Zdroj: <https://www.iso.org/standard/56692.html>

²³ NFC Forum – Nezisková společnost založená společnostmi Sony, Philips a Nokia

Bezpečnost u NFC je bohužel problém. Samo NFC neposkytuje žádnou odolnost vůči útokům. NFC musí být chráněno přímo v aplikacích pomocí protokolů SSL pro vytvoření zabezpečení.

1.4 Technologie OCR²⁴

OCR se používá pro zdigitalizování tištěných textů pomocí skeneru. V dnešní době je používána technologie OCR v klasických skenerech, které jsou používány pro digitalizaci různých dokumentů v papírové formě. OCR je schopné převádět dokonce i ručně psané texty, obrázky a další. OCR je schopné správně oskenovat i české znaky. Neúspěšnost převodu je nižší než 5%, je ale důležité, aby byl text psaný správně a znaky nebyly navzájem překryté. V tomto případě se skenovací doba prodlužuje nebo je to oskenováno špatně.

²⁴ OCR – Optical Character Recognition, jedná se o optické rozpoznávání znaků

2 Použité technologie

2.1 (X)HTML²⁵

HTML²⁶ je značkovací jazyk, jehož aktuální verze nese číslo 5. Jazyk HTML navrhl Tim Berners-Lee v roce 1990. HTML se tedy začalo pomalu vyvíjet a byly vydány postupně různé verze, které toho uměly více. Na konci roku 1999 byla vydána verze 4.01 a byla to poslední verze až do roku 2014, kdy byla vydána verze 5. Aktuální verze HTML nese číslo 5.3. Byla vydána 9. srpna 2018.

Obrázek 2.1 Ukázka a popis HTML kódu

```
<!DOCTYPE html>           <!-- podle zápisu doctype se pozná, o jakou verzi HTML se jedná -->
<html>                    <!-- Zde je začátek html dokumentu -->
<head>                    <!-- v sekci head je titulek stránky a obsahuje i
                           různé metatagy a připojené css a javascript soubory -->
<title>Titulek stránky</title> <!-- titulek stránky -->
</head>
<body>                    <!-- začátek těla HTML dokumentu -->
<div id="header">         <!-- sekce, která má id header, což se používá jako
                           jedinečný identifikátor -->
<div class="uvnitř">     <!-- sekce, která má class uvnitř, což se používá jako třída a může
                           být použita vícekrát v dokumentu => může se opakovat -->
<p> Hezký text</p>      <!-- odstavec textu -->
</div>
</div>
</body>
</html>
```

Zdroj: vlastní zpracování

2.1.1 HTML 5

Verze HTML 5 a její následníci 5.1, 5.2 a 5.3 přináší do webových prohlížečů řadu novinek, které mají zjednodušit práci kodérům a programátorům, kteří se zabývají právě tvorbou webových stránek. HTML 5 přináší přímou podporu multimédií v prohlížeči a podporu offline aplikací, které nepotřebují k provozu připojení k internetu.

Pro HTML 5 byly vytvořeny speciální tagy²⁷, které definují strukturu stránky.

²⁵ Zdroj: <https://cs.wikipedia.org>

²⁶ HTML – Hypertext Markup Language

²⁷ Tag – základní značky v oblasti HTML

Obrázek 2.2 Příklad nových tagů v HTML 5

```
<section></section> <!-- určuje části stránek -->
<footer></footer> <!-- určuje patičku stránky, kde by
měl být určen autor stránek -->
<nav></nav> <!-- určují navigační sekci stránek -->
<video></video> <!-- vloží na stránku video -->
```

Zdroj: vlastní zpracování

U HTML 5 je důležité dát si pozor na fungování ve všech dostupných webových prohlížečích. Většina nových prohlížečů už tuto verzi podporují, ale zastaralé verze prohlížečů mohou mít problémy se zobrazením webových stránek. Je možné tyto problémy obejít. Jsou tu 2 možnosti:

- Napsat HTML kód pro dvě různé verze HTML – zdlouhavé a komplikované
- Použít Javascript²⁸ knihovnu na webové adrese www.modernizr.com, kde jsou pomocí právě Javascriptu vytvořeny různé elementy pro webové stránky, aby vše fungovalo i u starších verzí prohlížečů.

Používanou verzi HTML je možné poznat podle zdrojového kódu, který začíná textem `<!DOCTYPE HTML>` znázorňující obrázek 2.1.

2.1.2 HTML 4.01

HTML 4.01 je starší verze HTML, která je stále ještě nejrozšířenější verzí. Tato verze nepodporuje nové tagy z HTML 5, takže multimédia se musí vkládat jiným stylem než použitím tagu `<video>`, ale je potřeba použít tag `<iframe>`²⁹ z jiné webové stránky nebo použít Javascript.

2.1.3 XHTML³⁰

XHTML³¹ je značkovací jazyk, který měl nahradit starší HTML 4.01. V XHTML musejí všechny tagy, i ty nepárové, být ukončeny:

²⁸ Javascript – programovací jazyk používaný pro webové stránky

²⁹ `<iframe>` - tag umožňující vložit do stránky rám o určité velikost a zobrazit v něm požadovanou webovou stránku.

³⁰ XHTML – eXtensible Hypertext Markup Language

³¹ Zdroj: <https://cs.wikipedia.org>

- U obrázků nestačí jen tento zápis ``, ale musí být na konci tagu uvedeno lomítko ``, takto to musí být u všech nepárových tagů.
- U párových tagů to zůstává stejné, kde to začíná takto `<table>` a končí vždy `</table>`

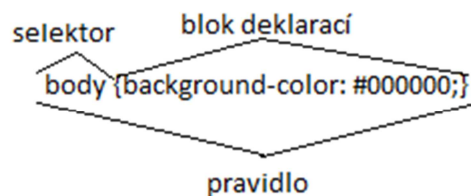
XHTML je založeno na XML³² dokumentu, kde je tedy nutné, aby všechny tagy byly ukončeny, ať už párové nebo nepárové. Jsou i další důležitá pravidla, která se musí dodržovat, aby XHTML web byl validní³³:

- `` musí obsahovat atribut `alt`³⁴
- `<script>` a `<style>` musí obsahovat atribut `type`³⁵

2.2 CSS³⁶

Kaskádové styly³⁷, jak se tomuto česky říká, se používají pro popis způsobu zobrazení elementů na stránkách napsaných v jazyce HTML nebo XHTML. Kaskádové styly byly vyvinuty v roce 1997. V dnešní době jsou nejdůležitější částí webových stránek, bez nichž by webové stránky vypadaly jako pouhé části textu případně obrázků.

Obrázek 2.3 Složení CSS pravidla



Zdroj: vlastní zpracování

Jako selektory se udávají různé části z HTML dokumentů. Selektory jsou:

- Body – určuje celou část těla HTML dokumentu
- Div – určuje určitý blok
- P – určuje odstavce v textu

³² XML – eXtensible Markup Language

³³ Validita – webová stránka neobsahuje žádnou chybu v zápisu zdrojového kódu

³⁴ Alt – obsahuje text, který se zobrazí, když se obrázek nenačte

³⁵ Type – určuje typ kódu, zda je to javascript nebo jiný druh scriptu

³⁶ CSS – Cascading Style Sheets

³⁷ Zdroj: <https://cs.wikipedia.org>

- #id – určuje id prvku
- .class – určuje třídu prvku

Zápis selektorů je možný nalézt na obrázku 2.1. Selektory je možné mezi sebou kombinovat:

- Body p – nastaví pro všechny elementy p deklarace požadované v pravidlech
- Body > p – nastaví pro všechny dceřiné elementy p deklarace požadované v pravidlech. Výjimku by tvořily elementy, které jsou v HTML dokumentu vloženy do jiného bloku.

Kaskádové styly je možné vkládat do stránek pomocí 3 různých možností:

- Vložením přímo do HTML dokumentu, kdy je zapsáno CSS do elementu <style>
- Vložením externího souboru pomocí elementu <link> do sekce <head> v HTML dokumentu

Přímým zápisem stylu k určitému elementu, kdy je přidán atribut <style="blok deklarací">, toto funguje jen pro ten určitý element.

Při práci s kaskádovými styly se musí dávat pozor, protože všechny webové prohlížeče obsahují už přednastavené parametry u většiny elementů. Je potřeba dbát na kontrolu ve více prohlížečích a nejlépe všechny vlastnosti řádně nastavit na požadované hodnoty. Nestane se tak, že budou problémy se zobrazením v různých prohlížečích.

2.2.1 CSS 1

CSS 1 je první verzí kaskádových stylů. Od první verze je už možné prvky vybírat podle selektorů a nastavit jim potřebné vlastnosti. Je možné nastavovat potřebnou barvu textu (color), potřebnou velikost textu (font-size), v případě tučnosti textu (font-weight) a mnoho dalšího.

2.2.2 CSS 2

CSS 2 jsou druhou verzí kaskádových stylů. Tato verze přináší další rozšíření:

- Max-width, min-width – maximální a minimální šířka
- Max-height, min-height- maximální a minimální výška

- Position – nastavení relativní či absolutní pozice v dokumentu
- Top, bottom, right, left – nastavení vzdálenosti od okrajů, při použití absolutní pozice

2.2.3 CSS 3

CSS 3 je v tuto chvíli poslední verzí kaskádových stylů. Touto verzí je přidáno další množství parametrů a je propojena s HTML 5. CSS 3 hlavně přidává:

- Animace prvků – není nutné používat JQuery knihovnu³⁸
- Vlastnosti drag'n'drop – možnost označit prvek a přetáhnout
- Opacity – možnost nastavení průhlednosti prvků
- Možnost nahrání stylu písma z externího zdroje

2.3 APACHE

Apache je softwarový balíček pro webový server používaný pro většinu internetových stránek. Je nainstalovaný na serveru. Pro ovládání je důležité znát vlastnosti a možnosti nastavení.

2.3.1 PHP³⁹

PHP je skriptovací jazyk. Je určen hlavně pro tvorbu dynamických webových aplikací v HTML či XHTML. Skript je prováděn na straně serveru, takže k uživateli se dostane pouze výsledek. PHP je možno provozovat na různých platformách jako Linux či Windows. PHP je nejrozšířenějším programovacím jazykem. PHP bylo vytvořeno v roce 1995 a aktuální verze nese označení 7.2.

2.3.2 MySQL⁴⁰

MySQL je databázový systém, kde probíhá komunikací pomocí jazyka SQL. MySQL je možné používat na různých platformách a nejvíc se využívá právě s PHP.

³⁸ JQuery knihovna – Javascript knihovna obsahují možnosti rozšíření a použití pro webové stránky

³⁹ PHP – Hypertext Preprocessor, dříve Personal Home Page.

⁴⁰ MySQL – My Structured Query Language

2.3.3 Javascript⁴¹

Javascript byl vytvořen v roce 1995 autorem Brendanem Eichem. Javascript je objektově orientovaný jazyk, který se používá převážně pro webové stránky. Zapisuje se přímo do HTML kódu. Při použití Javascript je důležité si dát pozor na aktivaci u uživatele. Může se stát, že uživatel bude mít vypnutý Javascript, tak by mohl být problém s použitím Javascript kódu. Javascript se používá po načtení webových stránek na straně uživatele. Pomocí toho se nezatěžuje server, na kterém jsou webové stránky umístěny. Pomocí Javascript kódu je možné vytvářet funkce, které mají oživit webové stránky nebo kontrolovat vyplnění různých formulářů.

2.4 PSPad⁴²

PSPad je freewarový textový editor a editor zdrojových kódů. PSPad byl vytvořen v roce 2001 českým programátorem Janem Fialou. PSPad je jedním z nejrozšířenějších editorů pro webové stránky. Pomocí této aplikace je možné vytvářet i upravovat PHP, Javascript, HTML, XML a další soubory pro webové stránky.

2.5 Model železniční dopravy

Jelikož se tato bakalářská práce zabývá vizualizací a identifikací objektů na modelové železnici ve webovém prohlížeči, tak zde není popsáno fungování a složení modelu železniční dopravy umístěného ve školní budově Vysoké školy logistiky o.p.s. Důvod tohoto rozhodnutí je, že se pracuje pouze s databází, do které jsou ukládány všechny potřebné informace.

Vše o tomto modelu železnice je možné nalézt v Bakalářské práci s názvem Aplikace standardů GS1 v modelu železniční dopravy⁴³ od autorky Petry Setlíkové z roku 2017.

⁴¹ Zdroj: <https://cs.wikipedia.org>

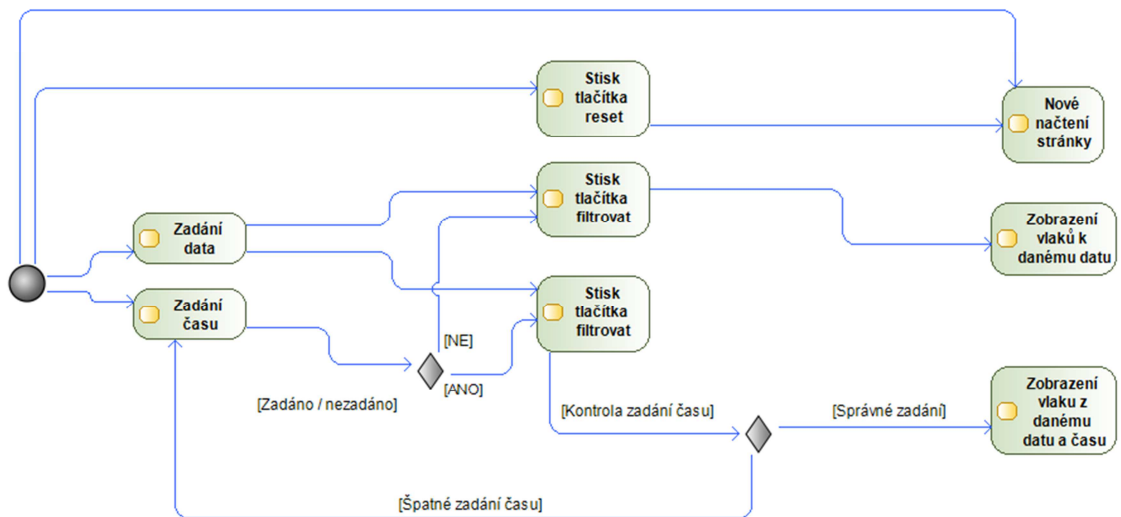
⁴² PSPad – celý název PSPad Editor

⁴³ Zdroj: Setlíková DiS., Petra. Aplikace standardů GS1 v modelu železniční dopravy. Přerov, 2017. Bakalářská práce. Vysoká škola logistiky o.p.s.. Vedoucí práce doc. Dr. Ing Oldřich Kodym.

3 Analýza požadavků na aplikaci

Důležité je připravit možnosti požadavků, které má aplikace splňovat. Hlavním kritériem je, co má aplikace zobrazovat, jaké informace jsou potřeba a jak bude aplikace ovládána. Dalším faktorem je určení možností uživatele, jestli bude schopen měnit údaje v databázi nebo jestli nebude mít přístup do této sekce. Pro analýzu je potřeba připravit diagram, který bude znázorňovat možnosti použití. Je potřeba udělat kompletní přehled použitelných proměnných, které budou potřeba k fungování aplikace.

Obrázek 3.1 Požadavky aplikace



Zdroj: vlastní zpracování

3.1 Analýza proměnných

Následující obrázek 3.2 obsahuje kompletní soupis potřebných proměnných použitých v aplikaci a jejich opodstatnění.

Obrázek 3.2 Seznam proměnných

```
$con = "";  
$doplnenidatum = "";  
$doplnenicas = "";  
$sqlvysl = "";  
$sqlvzhled = "";  
$vysledek = "";  
$zaznam = "";  
$vysledek2 = "";  
$zaznvzhled = "";  
$rfid = "";  
$rfidnahrada = "";  
$popisvagon = "";  
$zaklcas = "";  
$urcitycasplus = "";  
$urcitycasminus = "";  
$sqlvyslvagony = "";  
$vysledekvagony = "";  
$kontrolacasu = "";
```

Zdroj: vlastní zpracování

Pomocí proměnných je možné analyzovat složitost sepsání kódu. V případě použití velkého počtu proměnných může nastat problém s kontrolou zadaného kódu. Je důležité proměnné rozlišovat a vždy správně uvádět. V případě použití více podobných proměnných může nastat chyba velice snadno.

4 Tvorba aplikace

V této kapitole bude popsána kompletní tvorba webové aplikace Vizualizace provozu železnice pomoci auto ID⁴⁴.

4.1 Návrh aplikace

Důležité pro návrh této aplikace je zjištění struktury databáze železničního modelu a použité kódování. Dalším krokem je příprava informací, které jsou potřeba zobrazovat. Bylo navrženo, že bude možné vyplňovat přesné datum a čas projetí vlaku nebo jen zadání požadovaného dne. V případě vyplnění obou položek bude zobrazen správný vlak, který v danou chvíli projel RFID čtečkou. V případě vyplnění požadovaného dne bude vypsán kompletní seznam vlaků, které projely RFID čtečkami. Při novém načtení webové stránky bude vždy načteno posledních několik seřazených vlaků s vagony.

Důležitá je grafická příprava lokomotiv a vagonů, které mají být zobrazeny ve webovém prohlížeči. Celková grafická úprava webové stránky je posledním krokem pro návrh aplikace.

4.2 Struktura databáze

Databáze modelové železnice obsahuje 14 různých tabulek, které obsahují informace a RFID čtečkách či anténách, informace o všech vagonech, nebo nejdůležitější tabulka, která obsahuje všechny zapsané průjezdy přes RFID čtečky.

Nejdůležitější tabulka nese název events, která obsahuje všechny průjezdy přes RFID antény. Zapisuje automatické ID, přesný čas průjezdu, data z RFID štítku a číslo RFID antény, která zaznamenala průjezd. Zbylé položky z tabulky nejsou důležité pro správné znázornění vlaku.

⁴⁴ ID – identifikační číslo

Obrázek 4.1 Struktura tabulky events. Označeny potřebné sloupce z tabulky

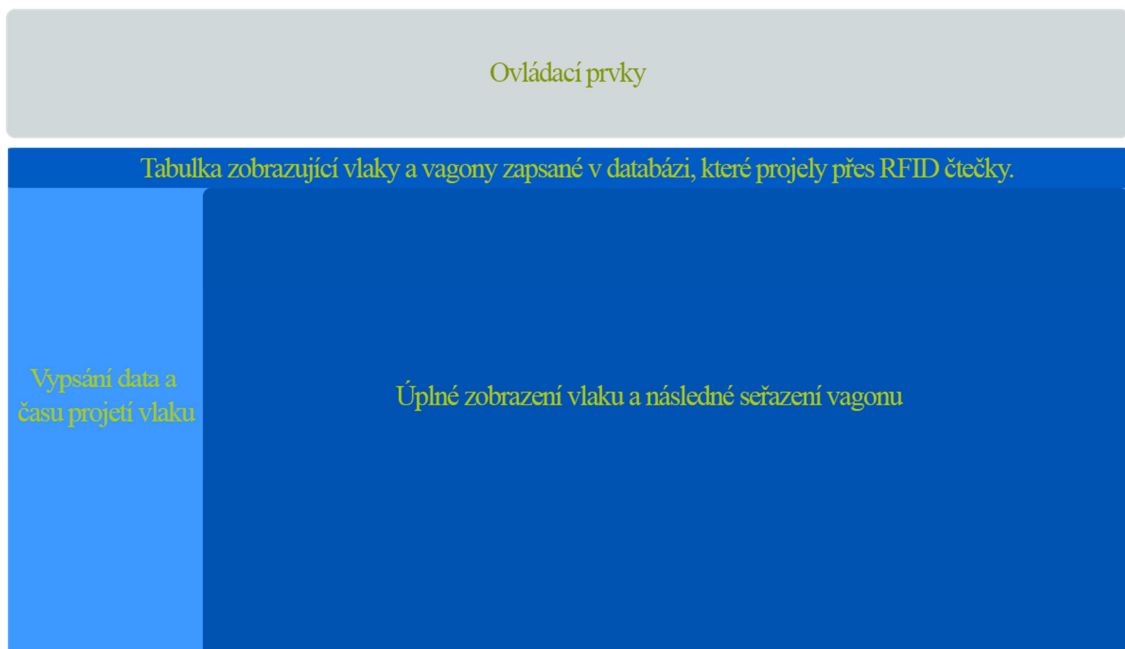
#	Název	Typ	Porovnávání	Vlastnosti	Nulový	Výchozí	Komentáře	Další	Operace
<input type="checkbox"/>	1	id	int(11)		Ne	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	2	timestamp	timestamp		Ano	CURRENT_TIMESTAMP			Změnit Odstranit Více
<input type="checkbox"/>	3	timestamp_processed	timestamp		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	4	type	int(11)		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	5	data	varchar(64) utf8_general_ci		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	6	board	int(11)		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	7	dio_nr	int(11)		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	8	operator	int(11)		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	9	extint	int(11)		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	10	extstr	varchar(40) utf8_general_ci		Ano	Žádná			Změnit Odstranit Více
<input type="checkbox"/>	11	status	int(11)		Ano	Žádná			Změnit Odstranit Více

Zdroj: vlastní zpracování

4.3 Grafický návrh

Pro grafický návrh je důležité, aby to bylo přehledné a jednoduché na ovládání. Časté změny grafické rozvržení nejsou žádány u uživatelů.

Obrázek 4.2 Grafické rozvržení webové stránky



Zdroj: vlastní zpracování

V sekci ovládací prvky bude obsažena možnost vyplnění přesného data, hodiny a následně filtrování potřebných informací.

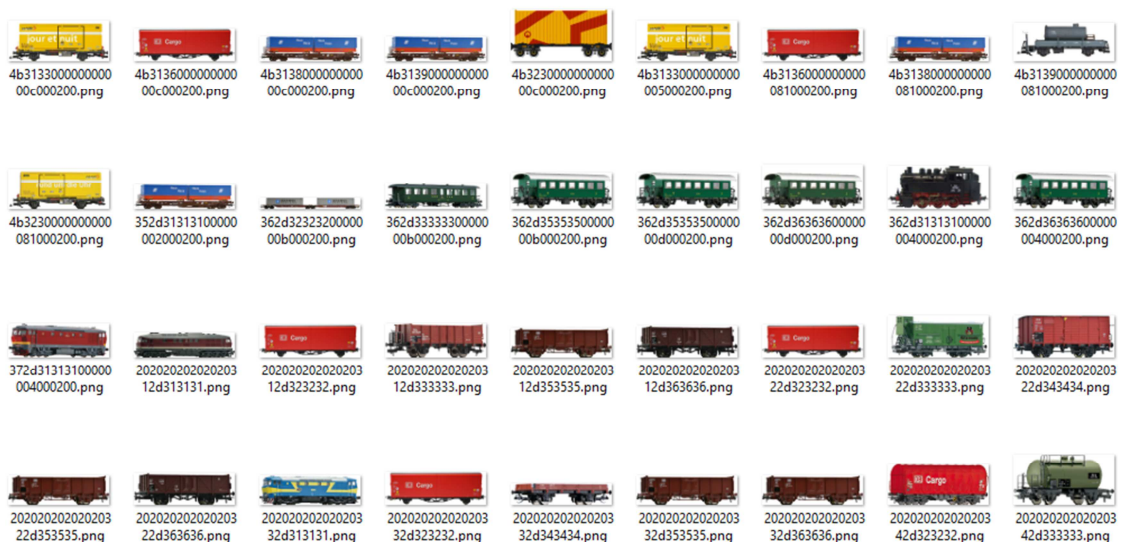
Vyplnění přesného data bude vytvořeno pomocí možnosti vybrat datum z kalendáře. Ulehčí to uživateli vyplňování a bude docíleno správného formátu data.

Vyplnění přesné hodiny a minuty bude záležet na uživateli. Bude možnost vyplnit hodiny, hodiny a minuty nebo nevyplnit žádný čas.

Vše toto se odešle přes tlačítko filtrovat a vypíše to požadované informace v tabulce. V ovládacích prvcích bude ještě možnost resetovat filtr a následně to opět vypíše poslední projeté vlaky.

Tabulka zobrazující vlaky a vagony bude obsahovat datum a čas projetí vlaku a následné kompletní seřazení vagonů. Seřazení nemusí být vždy přesné, záleží na tom, jestli byly vagony zapsány do databáze.

Obrázek 4.3 Připravené obrázky vlaků a vagonů



Zdroj: vlastní zpracování

Jak je vidět na obrázku 4.3, tak všechny názvy souborů obsahují kód, který je stejný s položkou z databáze v tabulce events případně v tabulce wagons, ze které je brán název lokomotivy nebo vagonu. Pomocí této informace se bude porovnávat a zobrazovat správný obrázek pro lokomotivu nebo vagon.

4.4 Přímá tvorba aplikace

Pro tvorbu aplikace budou použité všechny potřebné technologie popsané v kapitole 2. Při tvorbě je důležité mít alespoň začátečnické znalosti HTML, CSS a PHP. Všechny potřebné soubory jsou umístěny na příloženém CD.

4.4.1 Příprava HTML

HTML bude obsahovat všechny položky, které se mají vyplňovat případně zobrazovat. Jediné co nebude HTML obsahovat, je tabulka, která bude nastavena v kódu PHP.

Obrázek 4.4 Kompletní kód HTML

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
  <head>
    <meta http-equiv="content-type" content="text/html; charset=utf-8">
    <meta name="generator" content="PSPad editor, www.pspad.com">
    <title></title>
    <link REL=STYLESHEET type="text/css" href="styles/styles.css">           <!--připojení CSS souboru-->
    <script type="text/javascript" src="js/scripts.js"></script>           <!--připojení JavaScript souboru-->
  </head>
  <body>
    <p>
    <h1>Vítejte na stránce „Vizualizace provozu železnice pomoci auto ID“</h1>
    </p>
    <form method="post" action="<?php echo $_SERVER["PHP_SELF"]?>">           <!--formulář, který má nastavenu akci pro server.
    Postupně obsahuje všechny potřebné položky pro vyplnění-->
      <label for="dat">Zadejte datum:</label>
      <input type="date" id="dat" name="datum" value="" />           <!--položka připravená pro vyplnění data-->
      <label for="casik">Zadejte čas: </label>
      <input id="casik" type="text" name="hodina" value="">
      <input type="Submit" name="odesli" value="FILTROVAT">           <!--položka připravená pro vyplnění času-->
      <input type="reset" name="reset" value="RESET" onclick="reseticko()"> <!--tlačítko pro RESET celé stránky-->
    </form>
    <p>
      Datum je povinné! Datum zadejte přesně!
    </p>
    <p>
      Čas můžete zadat přesně (hh:mm), není to nutné, pokud neznáte přesný čas, stačí například jen hodina.
    </p>
    <p>
      *případné odchylky od skutečnosti jsou možné (pokud nebyl vlak či vagon zapsán do DB, tak není možné ho správně zobrazit)
    </p>
    <?php ... ?>
  </body>
</html>
```

Zdroj: vlastní zpracování

Jak je vidět na obrázku 4.4, je připojen soubor styles.css, který dodává požadovaný vzhled webové stránky. Následuje připojení souboru scripts.js, který obsahuje Javascript. Všechny tyto položky jsou umístěny v hlavičce HTML. Následuje položka body, tedy tělo HTML dokumentu. V těle se budou zobrazovat všechny položky, jako tabulky, texty, tlačítka a další. Nejdůležitější položkou je zde formulář, který začíná a končí tagy <form></form>. Tento formulář obsahuje 2 kolonky pro vyplnění a 2 tlačítka, jedno pro odeslání formuláře a druhé pro resetování stránky. Poslední řádky

obsahují informativní text pro uživatele. V sekci `<?php ... ?>` je obsažen kompletní kód, který bude popsán kapitole 4.4.3.

4.4.2 Příprava CSS

CSS bude obsahovat všechny položky, u kterých se nastavuje požadovaný vzhled.

Obrázek 4.5 První část CSS stylu

```
body {
    text-align: center;
    background-color: rgb(224,224,224);
    padding: 0;
    margin: 0;
}
input[type="text"], input[type="date"] {
    font-size: 17px;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
    background-color: rgb(255,255,255);
    margin: 0 5px;
}
input[type="submit"], input[type="reset"]{
    font-size: 17px;
    background-color: rgb(255,255,0);
    border: 1px solid #000000;
    -webkit-border-radius: 5px;
    -moz-border-radius: 5px;
    border-radius: 5px;
    padding: 2px 25px;
    margin: 0 0 0 15px;
    cursor: pointer;
}
input[type="reset"] {
    background-color: rgb(255,0,0);
    color: #fff;
}
input[type="submit"]:hover, input[type="reset"]:hover{
    text-decoration: underline;
}
table {
    min-width: 100%;
    width: intrinsic;
    width: -moz-max-content;
    width: -webkit-max-content;
    border: 1px solid #000000;
    border-collapse: collapse;
}
table tr {
    background-color: rgba(176,240,255,.5);
    display: block;
    overflow-x: auto;
}
table tr:nth-child(odd) {
    background-color: rgba(208,208,208,.4);
}

```

*/*vzhled pro tělo dokumentu*/
/*vycentrování textu na stránce*/
/*nastavení pozadí celé stránky*/
/*nastavení vnitřního rámu*/
/*nastavení venkovního rámu*/*

*/*položky pro vyplnění, type určuje o jaký input se jedná*/
/*velikost písma nastavena na 17px*/*

*/*nastavení zaoblení okraje*/*

*/*tlačítko pro odeslání nebo reset stránky*/*

*/*nastavení šířky okraje*/*

*/*položka bude mít po najetí jiný kurzor*/*

*/*reakce položky na najetí myši*/
/*podtržení textu*/*

*/*nastavení stylu pro celou tabulku*/*

*/*nastavení stylu pro jednotlivé řádky tabulky*/*

*/*nastavení stylu pro sudé řádky*/*

Zdroj: vlastní zpracování

Na obrázku 4.5 je možné si všimnout tagů, které odkazují na položky v HTML. Všechny položky nastavují požadovaný vzhled, aby stránky nebyly pouhým textem. Je zde nastavena barva pozadí, velikosti textu, případně barvy textu. Všechny důležité položky mají vytvořený komentář, který popisuje, k čemu daná položka slouží.

Obrázek 4.6 Druhá část CSS stylu

```
table tr th {                                     /*nastavení stylu pro buňky nadpisu sloupců*/
  background-color: rgb(0,0,255);
  color: #fff;
  padding: 10px 5px;
  font-size: 20px;
  width: 90%;
}
table tr th:first-child {
  width: 200px;
}
table tr th:second-child {
  width: intrinsic;
  width: -moz-max-content;
  width: -webkit-max-content
}
table tr td {                                     /*nastavení stylu pro buňky tabulky*/
  border: 0px;
  display: inline-block;
}
table tr td:first-child {                         /*nastavení stylu pro první buňky řádku v tabulce*/
  width: 200px;
  text-align: center;
}
```

Zdroj: vlastní zpracování

Na obrázku 4.6 je zapsán zbytek položek, kterými se nastavuje vzhled. V tomto případě se jedná o řádky tabulky případně buňky řádků.

4.4.3 Příprava PHP

Jak bylo popsáno v kapitole 2, tak PHP slouží pro skriptování⁴⁵ na straně serveru.

PHP kód obsahuje různé podmínky, cykly a jiné složité příkazy, které jsou potřebné udělat na straně serveru. Na obrázku 4.7 je na prvních řádcích vidět připojení k databázi, kontrola nastavení správného kódování textu a následná kontrola připojení k databázi, kdy při špatném připojení to vypíše chybu pomocí příkazu echo. V následujících řádcích je příprava tabulky a následná kontrola vyplnění dvou položek pro datum a čas. Pokud jsou tyto položky vyplněny, tak proběhne kompletní skript, který obsahují obrázky 4.7 a 4.8. Řádky s proměnnými \$doplnenidatum a \$doplnenicas kontrolují vyplnění položek, aby se nestalo, že by byla napadena databáze. Příkaz `mysqli_real_escape_string` vymaže speciální znaky z vloženého textu. Proměnná `$sqlvysl` obsahuje SQL⁴⁶ příkaz, který vybírá všechny položky z tabulky events, a vybírá je pro přesné datum a čas, který byl vyplněn zároveň. Tento výpis je seřazen od nejnovějšího. Do proměnné `$vysledek` se ukládá pole⁴⁷, které bude obsahovat všechny

⁴⁵ Skriptování – provádění příkazů

⁴⁶ SQL – Structured Query Language

⁴⁷ Pole – datový typ k ukládání více hodnot stejného typu

položky dle výběru z tabulky. Následuje cyklus⁴⁸ while, pomocí kterého budou vypsány všechny položky z pole \$vysledek. Na vypisování položek z pole je vždy nutné používat cykly. Nejlépe cyklus while, který bude pokračovat do té doby, co bude pole obsahovat uložené položky. Jakmile bude vypsána poslední položka, tak bude cyklus ukončen. Dále bude skript pokračovat dalším řádkem. Do proměnné \$vysledek2 jsou ukládány položky z tabulky wagons, protože je nutné získat sloupec s názvem lokomotivy nebo vagonu. Následuje podmínka if, která hledá v tabulce events položky data a porovnává jejich obsah. Pokud se rovnají celému textu, který je zadán za dvojitým rovnítkem, tak jsou vypsány následující hodnoty do HTML tabulky. Řádkem echo date je vypsána položka timestamp, která obsahuje přesné datum a čas projetí lokomotivy přes RFID⁴⁹ anténu. Následuje cyklus for, který funguje na jiném principu než cyklus while. Cyklus for je používá vždy, když je znám počet opakování daného cyklu. V tomto případě je znám, jelikož tabulka wagons obsahuje 67 různých položek. Následuje postupné vypisování uložených položek z pole \$vysledek2, který obsahuje úplný výběr položek z tabulky wagons. Proměnná \$rfid má za úkol připravit textový řetězec, který bude používán v dalších částech kódu. Do proměnné \$rfidnahrada je uložen textový řetězec, který bude nahrazovat případný chybějící obrázek. Příkaz file_exists kontroluje v hlavní složce serveru, jestli je daný obrázek nahrán na serveru. Pokud není nahrán, tak se vypíše právě náhradní proměnná \$rfidnahrada.

Na obrázku 4.8 je znázorněna poslední část PHP kódu. Do proměnné \$zaklcas je ukládán aktuální čas nalezené lokomotivy. Do proměnných \$urcitycasplus a \$urcitycasminus je uložena právě proměnná \$zaklcas a přičítá se nebo odečítá 5 sekund. Tímto se zajistí určitý interval času, pro který se bude vypisovat následný výběr z tabulky events. Do proměnné \$sqlvyslwagony je uložen příkaz vypisující z tabulky events všechny položky s datem a časem z proměnných \$urcitycasplus a \$urcitycasminus a následně stejného čidla, kterým projela lokomotiva. Následuje opět cyklus while, který postupně prochází pole \$sqlvyslwagony a vypisuje všechny položky, které nemají stejné RFID jako lokomotiva. Potom následuje kontrola, zda obsahuje tento vagon obrázek, který je dále použit jako výstup v tabulce. Zároveň je přidán k obrázku titulek s názvem vagonu či lokomotivy, který se objeví po najetí kurzoru na daný obrázek.

⁴⁸ Cyklus – opakující se část kódu dokud není splněna podmínka ukončení

⁴⁹ RFID – Radio Frequency Identification

Obrázek 4.7 První část PHP kódu

```

<?php
$con = @mysqli_connect('localhost', 'vsig', 'vsigprerov', 'vsig_prerov');
$con->set_charset("utf8");
if (!$con) {
    echo "Je tu problém: " . mysqli_connect_error();
    exit();
}
echo "<table>";
echo "<tr><th>Datum a čas</th><th colspan='0'>Seřazení vlaku</th></tr>";
if (!(empty($_POST['datum'])) AND !(empty($_POST['hodina']))) {
    $doplnenidatum = mysqli_real_escape_string($con,$_POST['datum']);
    $doplnenicas = mysqli_real_escape_string($con,$_POST['hodina']);
    $sqlvysl = "select * from events WHERE timestamp LIKE '' .$doplnenidatum." " .$doplnenicas."% ORDER BY id DESC ";
    $sqlvzhled = "SELECT * FROM wagons";
    $vysledek=@mysqli_query($con, $sqlvysl);
    while ($saznam=mysqli_fetch_assoc($vysledek)):
        $vysledek2 = @mysqli_query($con, $sqlvzhled);
        if (($saznam["data"] == "362d3131310000004000200") OR ($saznam["data"] == "372d3131310000004000200")) {
            echo "<tr>";
            echo "<td>";
            echo date("Y-m-d H:i:s", strtotime($saznam['timestamp']));
            echo "</td>";
            for($j = 1; $j <= 67; $j++) {
                $saznvzhled=mysqli_fetch_assoc($vysledek2);
                if ($saznam["data"] == $saznvzhled["rfid"]) {
                    $rfid = "/bp/images/" . $saznam["data"] . ".png";
                    $rfidnahrada = "/bp/images/no-image.png";
                    $popisvagon = $saznvzhled["description"];
                    if (file_exists($_SERVER['DOCUMENT_ROOT'],$rfid)) {
                        echo "<td><img width='200' height='80' src='".$rfid."' title='".$popisvagon."' ></td>";
                        break;
                    }
                }
            }
            echo "<td><img width='200' height='80' src='".$rfidnahrada."' title='".$popisvagon."' ></td>";
            break;
        }
    }
}

```

Zdroj: vlastní zpracování

Obrázek 4.8 Druhá část PHP kódu

```
//uložení času pro kontrolu a následné procházení DB v určitých intervalech
//k času, který byl uložen při nalezení lokomotivy, bylo přičteno 5 sekund
//k času, který byl uložen při nalezení lokomotivy, bylo odečteno 5 sekund
$sysledek2 = @mysql_query($con, $sqlsysvagon);
while ($sysledek2 = @mysql_fetch_assoc($sysledek2)) {
    $sysledek2 = @mysql_query($con, $sqlvzhled);
    for($h = 1; $h <= 67; $h++) {
        $szazvzhlid=mysql_fetch_assoc($sysledek2);
        if ($szaznam["data"] == $szazvzhlid["rfid"]) {
            $rfid = "/bp/images/" . $szaznam["data"] . ".png";
            $rfidnahrada = "/bp/images/no-image.png";
            $popisvagon = $szazvzhlid["description"];
            if (file_exists($_SERVER["DOCUMENT_ROOT"] . $rfid)) {
                if ($szaznam["data"] != "362d3131310000004000200" AND ($szaznam["data"] != "372d3131310000004000200")) {
                    echo "<td><img width=200 height=80 src='" . $rfid . "' title='" . $popisvagon . "' ></td>";
                    break;
                }
            }
            else {
                if (($szaznam["data"] != "362d3131310000004000200" AND ($szaznam["data"] != "372d3131310000004000200")) {
                    echo "<td><img width=200 height=80 src='" . $rfidnahrada . "' title='" . $popisvagon . "' ></td>";
                    break;
                }
            }
        }
    }
}
endwhile;
echo "</tr>";
endwhile;
$kontrolacasu = $_POST['hodina'];
if (!preg_match('/^([0-2]?[0-5])?$/',$kontrolacasu)) {
    echo "<td colspan=5' style=text-align: center; font-size: 25px; ><b>chyba zápisu času</b></td>";
}
else {
    if (mysql_num_rows($sysledek) == 0){
        echo "<td colspan=5' style= 'text-align: center; font-size: 25px; ><b>žádný záznam nenalezen</b></td>";
    }
}
```

//přidání kontrola vyplnění času, jestli obsahuje jen číslice nebo i jiné znaky kromě :

Zdroj: vlastní zpracování

Po posledním cyklu while je ještě připravena kontrola, která má za úkol kontrolu správnosti zadaného času. Tato kontrola může být v této části kódu, protože se jedná o podmínku, a ta musí být umístěna ve správném cyklu. Příkaz preg_match kontroluje obsah čísel a jejich počet v daném textu. Vykřičník určuje negativitu na tento příkaz. V případě splnění se vypíší následující řádky.

Poslední podmínka s příkazem mysqli_num_rows kontroluje, zda se počet hodnot v poli \$vysledek nerovná nule. Pokud je tomu tak, tak se vypíší následující řádky.

4.4.4 Uživatelský výstup

V této kapitole budou uvedeny obrázky, jak uživatel uvidí výstupy při práci s aplikací.

V případě nového načtení webové stránky se uživatel setká s výpisem posledních uložených vlaků viz. obrázek 4.9.

Obrázek 4.9 Nové načtení webové stránky



Zdroj: vlastní zpracování

Pokud uživatel vyplní položku datum, budou vypsány všechny vlaky pro ten určitý den viz. obrázek 4.10.

Obrázek 4.10 Zadání položky datum

Vítejte na stránce „Vizualizace provozu železnice pomocí auto ID“

Zadejte datum: Zadejte čas:

Datum je povinné! Datum zadejte přesně!
Čas můžete zadat přesně (hh:mm), není to nutné, pokud neznáte přesný čas, stačí například jen hodina.
*případné odchylky od skutečnosti jsou možné (pokud nebyl vlak či vagon zapsán do DB, tak není možné ho správně zobrazit)

Datum a čas	Seřazení vlaku
2017-03-02 12:39:57	
2017-03-02 12:39:47	
2017-03-02 12:39:36	
2017-03-02 12:39:12	
2017-03-02 12:39:01	
2017-03-02 12:38:58	
2017-03-02 12:38:43	
2017-03-02 12:38:28	

Zdroj: vlastní zpracování

V případě vyplnění obou položek datum a čas budou vypsány všechny vlaky, které danou hodinu nebo hodinu a minutu byly zaznamenány. Čas je možné zadat jako určitou hodinu nebo hodinu a minutu. V případě sekund toto není podporováno, proto aplikace vypíše chybu, že je čas špatně zadán.

Obrázek 4.11 Zadání položky datum a čas

Vítejte na stránce „Vizualizace provozu železnice pomocí auto ID“

Zadejte datum: Zadejte čas:

Datum je povinné! Datum zadejte přesně!
Čas můžete zadat přesně (hh:mm), není to nutné, pokud neznáte přesný čas, stačí například jen hodina.
*případné odchylky od skutečnosti jsou možné (pokud nebyl vlak či vagon zapsán do DB, tak není možné ho správně zobrazit)

Datum a čas	Seřazení vlaku
2017-03-02 12:38:58	
2017-03-02 12:38:43	
2017-03-02 12:38:28	
2017-03-02 12:38:17	
2017-03-02 12:38:07	

Zdroj: vlastní zpracování

V případě vyplnění pouhého času se vypíše chyba, že datum je nutná položka. Všechny potřebné informace jsou zapsány přímo na webové stránky pod možností filtrování. V případě nedodržení popsaných informací bude vždy uživatel upozorněn a neprovede se zadaný příkaz, dokud neopraví vstupní hodnoty.

5 Vyhodnocení a využití aplikace

Tato aplikace má sloužit výhradně pro uživatelskou sféru, které požaduje zobrazení informací ve srozumitelné formě a nejlépe v grafickém provedení. Pro uživatele je to nejlepší možnost pochopení výsledných výstupů. V případě textové formy výstupu to zdržuje uživatele při práci s textem. V této formě je uživatel spokojen s webovou prezentací, jelikož vidí přesné informace přímo na obrazovce zařízení. Nemusí se zdržovat čtením nebo složitým hledáním potřebných dat. V případě nutnosti či stížností uživatele je vždy možné zasáhnout do kompletního kódu a upravit vše, jak bude požadováno.

Jak je psáno výše, využití této aplikace je hlavně pro uživatele z důvodu zjednodušení. Zobrazení vlaků používají již webové stránky www.idos.cz při hledání určitých vlakových spojů. Uživatel si tak může zkontrolovat seřazení vagonů a být správně připraven na nástupišti.

V případě této aplikace je seřazení potřebné k logistickým důvodům, aby byla zajištěna případná rychlejší manipulace s vagony. Informace o umístění dokáže pomoci i v případě nehody. Každá pozice vagonu je uložena a v případě řešení nehody se dají zjišťovat důvody problému.

Závěr

Během práce byla úspěšně vytvořena aplikace pro vizualizaci provozu na železnici.

Při grafickém znázornění železniční dopravy dochází k chybě v případě zobrazení všech vagonů za určitou lokomotivou. Je to uskutečněno rychlostí a počtem zápisů do databáze v jednu chvíli. Může se stát, že nebude daný RFID tag ze štítku na vagonu řádně zapsán, a tak není možné daný vagon zobrazit v aplikaci. Pokud budou správně přečteny a uloženy všechny RFID tagy, tak budou vlaky obsahovat všechny vagony. Bohužel není možné vyřešit správné seřazení vagonů z důvodu neukládání času v milisekundách. V případě použití časového horizontu milisekund by bylo možné seřadit dle zápisu všechny vagony tak, jak jsou seřazeny.

Touto bakalářskou prací doporučuji rozšířit ukládání aktuálního času až na milisekundy.

V případě problému s ukládáním RFID tagů je důležité zrychlit vysílání RFID antén. Mohlo by to zrychlit ukládání a předejít tak chybějícím údajům.

Soupis použitých zdrojů

Tištěné zdroje:

GROS, Ivan. *Velká kniha logistiky*. Praha: Vysoká škola chemicko-technologická v Praze, 2016. ISBN 978-80-7080-952-5.

KOFLER, Michael a Bernd ÖGGL. *PHP 5 a MySQL 5: průvodce webového programátora*. Brno: Computer Press, 2007. ISBN 978-80-251-1813-9.

OUDOVÁ, Alena. *Logistika: základy logistiky*. Aktualizované 2. vydání. Prostějov: Computer Media, 2016. ISBN 978-80-7402-238-8.

ŠVARCOVÁ, Ivana a Tomáš RAIN. *Informační management*. Praha: Alfa Nakladatelství, 2011. Informatika (Alfa Nakladatelství). ISBN 978-80-87197-40-0.

Internetové zdroje:

Wikipedia. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2018-08-14]. Dostupné z: https://cs.wikipedia.org/wiki/Hlavn%C3%AD_strana

Kodys. *Kodys* [online]. Praha: online, 2018 [cit. 2018-08-14]. Dostupné z: <https://www.kodys.cz/>

LOGISTIKA. *Studentske.cz* [online]. online, 2018 [cit. 2018-08-14]. Dostupné z: <http://logistika-cz.studentske.cz/2011/02/logistika-obsah.html>

RFID řešení. *Eprin* [online]. Brno: online, 2018 [cit. 2018-08-14]. Dostupné z: <https://www.eprin.cz/rfid-technologie.html>

Barcode Island. *Barcode Island* [online]. online, 2006 [cit. 2018-08-14]. Dostupné z: <http://www.barcodeisland.com>

Ostatní zdroje:

Hájek, Tomáš. Systémy automatické identifikace. Praha, 2010. Bakalářská práce. Vysoká škola regionálního rozvoje a Bankovní institut – AMBIS, a.s.. Vedoucí práce Ing. Vladimír Beneš, Ph.D..

JANOUSHKOVÁ, Ivana. Identifikace výrobků s využitím čárových kódů. Zlín, 2011. Bakalářská práce. Univerzita Tomáše Bati ve Zlíně, Fakulta logistiky a krizového řízení. Vedoucí práce doc. Ing. Zdeněk Čujan, CSc..

Setlíková DiS., Petra. Aplikace standardů GS1 v modelu železniční dopravy. Přerov, 2017. Bakalářská práce. Vysoká škola logistiky o.p.s.. Vedoucí práce doc. Dr. Ing Oldřich Kodym.

Autor (vypracoval)	Tomáš Fejfar, DiS.
Název BP	Vizualizace provozu železnice pomocí auto ID
Studijní obor	Informační management
Rok obhajoby BP	2018
Počet stran	35
Počet příloh	0
Vedoucí BP	Ing. Libor Kavka, Ph.D.
Oponent BP	
Anotace	<p>Díky modelové železnici, umístěné na Vysoké školy logistiky v Přerově, přichází myšlenka na vytvoření vizuálního zpracování informací zapsaných do databáze pomocí RFID antén umístěných na modelu železnice. Pomocí modelové železnice je možné zkoušet fungování RFID komunikace v praxi.</p> <p>V první části bakalářská práce informuje o možnostech automatické identifikace. V dalších částech práce je kompletní návrh a vytvoření aplikace vizualizace provozu na železnici pomocí webového prohlížeče.</p>
Klíčová slova	Identifikace, železnice, vizualizace, aplikace, model, web, databáze
Místo uložení	ITC (knihovna) Vysoké školy logistiky v Přerově
Signatura	