VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INTELIGENTNÍCH SYSTÉMŮ
DEPARTMENT OF INTELLIGENT SYSTEMS

INTERAKTIVNÍ VIZUALIZACE VÝROBNÍCH ROZVRHŮ S MOŽNOSTMI EDITACE

INTERACTIVE VISUALISATION OF MANUFACTURE SCHEDULES WITH EDITING CAPABILITIES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — **DAMIÁN GLADIŠ**
AUTHOR

VEDOUCÍ PRÁCE — **Ing. MARTIN HRUBÝ, Ph.D.**
SUPERVISOR

BRNO 2021

Vysoké učení technické v Brně
Fakulta informačních technologií

Ústav inteligentních systémů (UITS) Akademický rok 2020/2021

Zadání bakalářské práce

Student: **Gladiš Damián**
Program: Informační technologie
Název: **Interaktivní vizualizace výrobních rozvrhů s možnostmi editace**
Interactive Visualisation of Manufacture Schedules with Editing Capabilities
Kategorie: Uživatelská rozhraní

Zadání:

1. Prostudujte programování webových aplikací. Prostudujte základní schémata výrobních rozvrhů.
2. Navrhněte webovou aplikaci, která zobrazí uživatelsky přívětivě výrobní rozvrh ze zadaného souboru ve formátu SQLITE3. Navrhněte ovládací prvky pro navigaci rozvrhem (zobrazení detailu, posuv v čase, změna měřítka časové osy, selekce zdrojů, apod.).
3. Aplikaci implementujte. Aplikace musí umožnit interaktivně modifikovat jednotlivé záznamy rozvrhu a tyto ukládat do vstupní DB.
4. Testujte aplikaci s použitím několika různých rozvrhů.

Literatura:

- Dle pokynů vedoucího.

Pro udělení zápočtu za první semestr je požadováno:

- První dva body.

Podrobné závazné pokyny pro vypracování práce viz https://www.fit.vut.cz/study/theses/

Vedoucí práce: **Hrubý Martin, Ing., Ph.D.**
Vedoucí ústavu: Hanáček Petr, doc. Dr. Ing.
Datum zadání: 1. listopadu 2020
Datum odevzdání: 12. května 2021
Datum schválení: 11. listopadu 2020

Abstrakt

Bakalárska práca sa zaoberá vizualizovaním výrobných rozvrhov pomocou webovej aplikácie, kde zobrazený výrobný rozvrh je možné interaktívne editovať. Podľa Ganttovho diagramu bola navrhnutá schéma na zobrazenie výrobného rozvrhu a jeho editačné vlastnosti. Na vývoji tejto webovej aplikácie boli použité technológie ako PHP, HTML, MySQL, XAMPP, CSS, Javascript a SQLite3. Riešenie bolo otestované na niekoľkých rôznych výrobných rozvrhoch.

Abstract

Bachelor thesis deals with visualizing production schedules using a web application where the displayed production schedule can be interactively edited. According to Gantt's diagram, a scheme was designed to display the production schedule and its editing properties. Technologies such as PHP, HTML, MySQL, XAMPP, CSS, JavaScript and SQLITE3 were used to develop this web application. The solution was tested on several different production schedules.

Kľúčové slová

webová aplikácia, výrobný rozvrh, HTML, DayPilot, PHP, SQLite, výrobné operácie, interaktívna vizualizácia, Ganttov diagram, editačné vlastnosti webovej aplikácie

Keywords

web application, production schedule, HTML, DayPilot, PHP, SQLite, production operations, interactive visualization, Gantt chart, web application editing properties

Citácia

GLADIŠ, Damián. *Interaktivní vizualizace výrobních rozvrhů s možnostmi editace*. Brno, 2021. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Hrubý, Ph.D.

Interaktivní vizualizace výrobních rozvrhů s možnostmi editace

Prehlásenie

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne pod vedením pána Ing. Martina Hrubýho, PhD. Uviedol som všetky literárne pramene, publikácie a ďalšie zdroje, z ktorých som čerpal.

.....
Damián Gladiš
10. mája 2021

Poďakovanie

Týmto by som sa chcel poďakovať môjmu vedúcemu práce Ing. Martinovi Hrubýmu, PhD. za vedenie v mojej bakalárskej práci, cenné rady a všetkú pomoc.

Obsah

Kapitola 1

Úvod

Digitálne technológie sa stále vyvíjajú a vďaka tomuto vývoju podniky a firmy využívajú najmodernejšie nástroje pri optimalizácií vo výrobnom programe. Hlavným cieľom pre firmy je optimalizovať plánovanie a riadenie výroby. Jedným z týchto nástrojov sú aj MES systémy, ktoré sa zaoberajú týmito cieľmi a napomáhajú pracovníkom vo výrobe príjmať dôležité rozhodnutia alebo odhaliť prípadné problémy, čo vedie k zefektívneniu výroby.

Hlavným cieľom mojej bakalárskej práce je vizualizácia výrobných rozvrhov pomocou webovej aplikácie, ktorá bude podobná ako nástroje pre optimalizáciu výroby, a ktorá bude interaktívna a pre zobrazený výrobný rozvrh bude ponúkať možnosti editácie. Vo webovej aplikácii si bude možné zvoliť obdobie, po ktoré sa majú zobraziť výrobné operácie vo výrobnom rozvrhu a typ zobrazenia tohto výrobného rozvrhu. Webová aplikácia bude obsahovať aj ovládacie prvky pre navigáciu vo výrobnom rozvrhu.

V kapitole 2 sa venujem problematike výroby, výrobného procesu, z čoho sa skladá výrobný proces, akú má štruktúru, problematike výrobného rozvrhu, ako vyzerá ER diagram k zadanej databáze a popis databázy, čo je to vizualizácia a tvorba vizualizácie pomocou PHP, MySQL, Xampp, SQLite, DayPilot a podobne. Ďalej sa venujem architektúre systému webovej aplikácie, ktorá bola pri vývoji použitá, z čoho sa daná architektúra skladá a ako s ňou užívateľ interaguje.

V kapitole 3 sa venujem návrhu webovej aplikácie, čo všetko má daná webová aplikácia spĺňať, ako bude vyzerať užívateľské rozhranie webovej aplikácie, aké sú vstupné parametre pre vizualizáciu výrobného rozvrhu, aké ovládacie prvky pre navigáciu vo výrobnom rozvrhu má webová aplikácie obsahovať a aké editačné vlastnosti ponúka webová aplikácia.

V kapitole 4 je popísaná implementácia systému webovej aplikácie, vstupných parametrov pre vizualizáciu výrobného rozvrhu, spracovania vstupných dát vo formulári, zmeny mierky časovej osi, detailu, hľadania a interaktívnej modifikácie výrobnej operácie. Taktiež je v tejto kapitole popísaná inštalácia balíčkov, nasadenie a spustenia webovej aplikácie.

Nakoniec, v kapitole 5 je zhrnuté testovanie webovej aplikácie. Táto časť obsahuje zhodnotenie funkčnosti celej webovej aplikácie a zodpovedá na otázku, či webová aplikácia spĺňa požiadavky, ktoré boli určené v zadaní. Zahŕňa v sebe testovanie zobrazenia operácií vo výrobnom rozvrhu a testovanie s použitím niekoľkých rôznych výrobných rozvrhov podľa rôznych vstupných údajov.

Kapitola 2

Teoretický rozbor témy

V tejto kapitole je podrobne predstavená problematika výroby, výrobného procesu a výrobného rozvrhu, čo znamená pojem vizualizácia a aká to je interaktívna vizualizácia, ďalej je to tvorba vizualizácie pomocou technológií HTML, CSS, XAMPP, PHP, JavaScript, MySQL, programovacích a skriptovacích jazykov, ktoré navzájom spolupracujú a modulu DayPilot, ktorý je súčasťou jazyka JavaScript. Patrí sem aj použitá architektúra systému webovej aplikácie, ER diagram a popis k zadanej databáze.

2.1 Čo je to výroba, výrobný proces a výrobný rozvrh?

2.1.1 Výroba

Výroba označuje súvislú činnosť, ktorú firma vykonáva za účelom, aby poskytla službu alebo výrobok a na základe toho bola patrične ohodnotená od svojich zákazníkov. Je možné ju definovať ako transformáciu výrobných faktorov do podoby ekonomických služieb a statkov, ktoré následne prejdú spotrebou. Služby sú označované ako nehmotné statky a sú to úkony, po ktorých existuje dopyt. V ekonómii sú statky označované ako fyzické komodity, resp. veci vyrábané pre spotrebu alebo výmenu, ktoré výrazne zvyšujú úroveň ekonomického blahobytu [6].

2.1.2 Výrobný proces

Výrobný proces vedúci k vypracovaniu finálneho výrobku sa nazýva technologický postup. Je zostavený postupnosťou operácií, ktoré sa ďalej členia na úseky, úkony a pohyby. Obvykle sa na zostavenie technologického postupu podieľajú špecialisti, technológovia a normovači výkonu. Pri plánovaní a riadení je veľmi dôležité, aby bolo presne stanovené, na ktorom pracovisku alebo oddelení sa bude daná operácia prevádzať. Tiež je veľmi dôležité, aby bolo dobre stanovený odhad času prevádzania danej operácii na konkrétnom pracovisku alebo oddelení. Technologické postupy neslúžia pre plánovanie a riadenie výroby len ako základný zdroj informácií, ale aj ako fyzický nosič týchto informácií.

V technologických postupoch sa uvádzajú potrebné údaje k spracovaniu danej operácie, ako čas potrebný pre spracovanie operácie. Následne sú to údaje, ktoré komentujú priebeh menších operácií a výsledky medzioperácií. Vo vzťahu k danej operácii, sa k nej uvádzajú aj dodatočné informácie, napríklad informácie o potrebnom špeciálnom náradí, upozornenie na použitie zvláštnych výrobných postupov, špeciálny postup na kontrolu kvality výrobku a podobne. Na obrázku 2.1 je znázornený príklad pracovného postupu, konkrétne sa jedná

o pracovný postup pre výrobu 300ks súčiastky nazvanej *teleso ložiska*. Na tomto obrázku je znázornených prvých 6 operácií vo fáze zhotovenia dielu. Technologický postup je znazornený vo chvíli, keď je diel predávaný z technologického oddelenia na oddelenie plánovania výroby. *ZAK* a *C.V.* sú údaje, ktoré identifikujú výrobnú zakázku a súčasti. Oddelenia sú označené číslami platmými v danom výrobnom podniku. *TK* definuje kusový čas v minútach, ktorý označuje čas potrebný na zhotovenie dielu na danom oddelení. *Třída* vyjadruje náročnosť danej operácie z kvalifikačného hľadiska a zároveň určuje výšku mzdovej sadzby za jednu minútu práce. *TP* vyjadruje čas potrebný na prípravu a zakončenie operácie v minútach na danom oddelení. Operácie, ktoré neobsahujú časové údaje o spracovaní danej operácie, sú *režijné* a pracovníci sú platení podľa hodinových sadzieb [6].

TEPO – TZ132

C.V.: 2-96577 ZAK: 123 567 894 B.C.: 13009 Ks: 300

Název: těleso ložiska
Materiál: odlitek 2710760 Jakost: 422650.5
Vystavil: J. Chalupa Změnový ref.: J. Vostrý Dne: 13. 3. 97

Operace 1	Prac: 98620	Třída:	TK:	TP:
Stav dodání: 422650, žíhaný normalizačně a popuštěný, apretovaný, se základním nátěrem, přejímka dle ČSN 421261.00. Zapsat číslo tavby a odlitku do postupu!				
Operace 2	Prac: 94120	Třída:	TK:	TP:
Proměřit, prorýsovat, zkontrolovat rozměry				
Operace 3	Prac: 52880	Třída: 32	TK: 30	TP:
Odstranit nálitky v otvoru				
Operace 4	Prac: 52880	Třída: 52	TK: 250	TP: 50
Rovnat dle rýsování, přerovnat pomocné plochy, hrubovat základnu a boční plochu M180 s přídavkem 1 mm na plochu, boční plochy M600, MD11, horní plochy včetně zámku a nálitky M5 pro šrouby hotově s ohledem na přídavek na další obrábění. Nářadí: fréza 15 st. EFU 106				
Operace 5	Prac: 94240	Třída: 27	TK: 21	TP: 8
Odhrotit				
Operace 6	Prac: 46323	Třída: 41	TK: 50	TP: 20
Vrtat otvory 4 x D = 30, 4 x M24, srazit hrany				

Obr. 2.1: Pracovný postup vo výrobe [6]

2.1.3 Výrobný rozvrh

Výrobný rozvrh zobrazuje dané operácie alebo kooperácie v danom časovom úseku, na y – ovej osi sa nachádzajú identifikačné čísla alebo názvy strojov, na x – ovej osi sa nachádzajú operácie alebo kooperácie v danom čase na danom stroji. Pri operáciach sú uvedené aj informácie, ako napr. presný čas začiatku a konca danej operácie, požadovaný personál alebo nástroje, ktoré sú potrebné pre výrobu danej súčiastky, typ operácie, typ nástroja ale aj prioritu výrobnej operácie. Rôzne typy výrobných rozvrhov obsahujú aj produkciu materiálu a jeho konzumáciu. Obsahujú aj oblasť, v akom časovom úseku sa pre operáciu žiada o stroj, personál a nástroje.

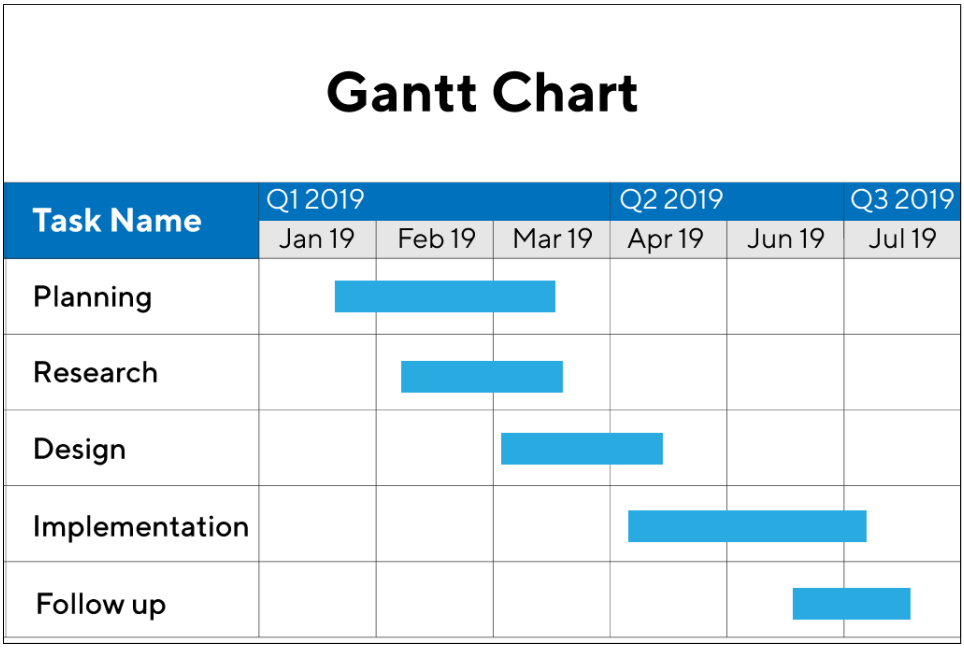
Schéma výrobného rozvrhu by taktiež mala obsahovať správne posúvanie v časovom merítku, prípadne zmenu tohto merítka. Dôležitou časťou je aj editácia udalostí v tomto rozvrhu, ak ich chceme presúvať medzi strojmi alebo len na časovej osi v rovnakom stroji.

Väčšinou nepotrebujeme zobraziť úplne všetky operácie, zaujíma nás len konkrétna operácia alebo konkrétne obdobie, v ktorom sa výrobné operácie vykonávajú. Výrobný rozvrh by mal zaisťovať aj správnu selekciu zdrojov pre zobrazený výrobný rozvrh.

Podľa zamerania vo výrobe alebo obecne, výrobný rozvrh môže zobrazovať výrobné operácie ako celkové časové obálky alebo môže slúžiť na zobrazenie čiastkových výrobných operácií, takže výrobný rozvrh by mal ponúkať aj možnosti zobrazení, ktoré závisia na tom, čo si chce práve používateľ zobraziť.

2.2 Vizualizácia výrobného rozvrhu

Povedali sme si, čo všetko by mal obsahovať a zobrazovať výrobný rozvrh. Príkladom 2D zobrazenia výrobného rozvrhu je aj **Ganttov diagram**. Zobrazuje nám, kedy má byť ktorá operácia vykonaná. Umožňuje nám sledovať, či daná operácia je vykonávaná načas, s meškaním alebo je vykonávaná s časovým predstihom. Ganttov diagram je veľmi prehľadný nástroj, v ktorom sa vieme jednoducho pohybovať v čase. Vieme usúdiť, či náhodou nie je daný stroj preťažený a prípadne prerozdeliť operácie na danom stroji podľa priority. Optimalizáciu operácií na danom stroji je potrebné vykonať aj pri nedostatku personálu alebo nástrojov k vykonaniu danej operácie. Na obrázku 2.2 vidíme ukážku Ganttovho diagramu.



Obr. 2.2: Ganttov diagram [11].

Dizajnovo je to typ stĺpcového grafu, ktorý znázorňuje projekty a rozvrhy pomocou farebných pruhov rôznej dĺžky. Graf okrem začiatočných a koncových dátumov odráža aj dôležité úlohy, udalosti a miľníky. Ganttove diagramy môžu dokonca odrážať viacnásobné časové rámce a spôsob, akým sa jedna udalosť týka tej druhej. Výsledkom je vizuálna reprezentácia časového obdobia, ktorá nám ľahko nadväzuje konečný cieľ, ale aj cestu potrebnú na to, aby sa tam dostala. Z tohto dôvodu sa Ganttove diagramy aplikujú na masívne projekty. Ak sa digitálny obchodník pokúša spravovať obsah a diskutuje o rozvrhu s interným tímom alebo

s externými klientmi, Ganttove diagramy môžu vizuálne vysvetliť pôvodný plán a ukazovať, kde sa projekt nachádza v rámci časovej osi. Užívatelia dokážu rýchlo pochopiť kroky, ktoré boli prijaté na dosiahnutie cieľov a aj to, ako sa jednotlivé úlohy vzájomne spoliehajú na ich dokončenie. Čas nie je vôbec statický.

Digitálny obchodníci môžu pomocou Ganttovho diagramu na komunikáciu plánov a časových rámcov poskytovať presné reprezentácie svojich kampaní. Môžu ukázať vzájomnú závislosť úloh v jasnom a stručnom dizajne, ktorá nepotrebuje žiadne extra tlmočenie. Jediný Ganttov diagram je možné exportovať a zdieľať v príspevkoch na sociálnych sieťach. Ganttové diagramy sú tiež ideálne aj pre obsahové a redakčné kalendáre. Odpovedať na otázky kto vytvára, čo vytvára a kedy je to potrebné zverejniť je možné pomocou jednoduchého kontrolného záznamu. Ale ak je obsah závislý od niekoľkých tímov alebo jednotlivcov na dokončení, Ganttov diagram dokáže ilustrovať, kto a v akom poradí musí dokončiť úlohu, aby niekto iný mohol začať ďalšiu úlohu v poradí. Ganttov diagram vysvetľuje presne to, čo sa deje a súčasný stav projektu, vrátane toho, čo je už splnené a toho, čo sa ešte len splní [9].

2.3 Technológie pre vývoj interaktívnej vizualizácie

2.3.1 PHP

PHP je open source programovací skriptovací jazyk, používaný širokou verejnosťou webového odvetvia, ktorý je aj veľmi univerzálny. Je určený najmä pre vývoj webových aplikácií. To, čím sa líši tento jazyk od ostatných jazykov, napr. od jazyku Javascript, na strane klienta, je to, že PHP kód je spustený na strane servera, vygeneruje sa HTML kód, ktorý je neskôr odoslaný naspäť klientskej strane. Klient vidí len výsledný HTML kód, nevie ako pôvodný PHP kód vyzeral. Tento jazyk je veľmi prívetivý aj pre začiatočníkov [1].

Jazyk PHP získal aj negatívne hodnotenia, hlavne kvôli pomalej rýchlosti spracovania príkazov a aj kvôli menšej bezpečnosti oproti iným jazykom. Dnes to však už neplatí. Tvorcovia PHP vynaložili veľa síl na vyčistenie jazyka a odstránenie alebo aspoň obmedzenie tých funkcií, ktoré umožňovali zneužitie alebo priamy útok na aplikáciu. Čo sa týka komunikácie so serverom MySQL, v jazyku PHP sú na výber dve možnosti, MySQLi alebo PDO. Existuje aj tretie rozšírenie, nazvané len MySQL, ktoré dnes je už považované za zastaralé a tak kód, ktorý budete sami programovať, by mal využívať rozšírenie PDO alebo MySQLi [2].

2.3.2 JavaScript

JavaScript je programovací počítačový jazyk, najviac používaný zo súčastí webových stránok. Implementáciou týchto súčastí sa webstránka stáva dynamickou webstránkou. Tento interpretovaný programovací jazyk disponuje objektovo-orientovanými možnosťami. Je otvorený a multiplatformový jazyk navrhnutý na vytváranie sieťovo orientovaných aplikácií. Dopĺňa a je integrovaný jazykmi ako je Java alebo HTML [10].

2.3.3 DayPilot JavaScript Scheduler

JavaScript Scheduler je vizuálna komponenta jazyka HTML, ktorá zobrazuje časový harmonogram pre viaceré zdroje. Na horizontálnej osi (x) sa zobrazuje čas a na vertikálnej osi (y) sú ako riadky zobrazené zdroje. Správanie tejto komponenty je možné prispôsobiť vďaka JavaScript API referenčnej príručke pre programovanie aplikácií. V balíku API

nájdeme aj pár tém, vďaka ktorým vieme prispôsobiť aj vzhľad danej aplikácie. Existuje aj online konfigurátor UI Builder, v ktorom si sami nakonfigurujeme základné vlastnosti komponenty [4].

2.3.4 MySQL

MySQL je multiplatformový open-source systém, vyvinutý a založený na SQL dotazoch. Pôvodne ho vyvinula švédska spoločnosť MYSQL AB. Vychádza z jazyka SQL, ktorý pozostáva z príkazov pre vytváranie a údržbu tabuliek, pre vkladanie, riadenie a načítanie dát a aj z príkazov pre spravovanie databáz.

Komunikácia môže prebiehať napríklad programovo z prostredia aplikácie, ktorú sme si sami napísali alebo môže prebiehať interaktívne pomocou vyhradeného klientského programu. Možnosť načítania dát nám dáva jeden zo základných pilierov jazyka SQL, príkaz *SELECT*. O udržovanie dát v aktuálnom stave sa starajú príkazy *UPDATE* a *DELETE*, ktoré spoliehajú na klauzulu *WHERE* definujúcu tie záznamy, s ktorými má byť prevedená daná operácia. Pre pridávanie záznamov do tabuľky slúži príkaz *INSERT*. Samotné programovanie databázy nám uľahčujú vstavané funkcie, uložené procedúry, triggery, udalosti a aj užívateľsky definované funkcie [2].

2.3.5 SQLite

SQLite je medziprocesná knižnica transakčného databázového nástroja SQL, ktorá je samostatne implementovaná, bežiaca bez servera a s takmer nulovou žiadanou konfiguráciou. Kód pre SQLite je verejne publikovaný a je teda dostupný na akýkoľvek účel, komerčný alebo súkromný. SQLITE je najrozšírenejšia databáza na svete s viacerými aplikáciami, než sme schopní spočítať.

Na rozdiel od ostatných SQL databáz, SQLite nemá samostatné serverové spracovávanie. SQLite priamo číta a zapisuje do bežných súborov. Kompletne celá databáza SQL s viacerými tabuľkami, triggermi, indexmi, a pohľadmi je obsiahnutá v jednom disku alebo súbore. Formát databázového súboru je multiplatformový, teda môžete kopírovať databázu medzi 32 – bitovými a 64 – bitovými systémami alebo medzi *big – endian* a *little – endian* architektúrami. Tieto rozšírenia robia SQLite obľúbenou voľbou pre formát aplikačného súboru.

Je ťažké predpovedať budúcnosť, ale zámerom vývojarov je podporovať SQlite aspoň do roku 2050, programovo aj dizajnovo [12].

2.3.6 HTML

HTML je skratka pre HyperText Markup Language. HTML dokumenty sú jadrom všetkých obsahov, ktoré sú zavesené na WWW(World Wide Web) a pozostávajú z 2 základných častí: obsah informácií a zložku pokynov, ktorá usmerní počítač, ako správne zobraziť jeho obsah. Nie je to typický programovací jazyk, ale skôr sada pokynov pre zobrazenie daného obsahu. Preklad tohto obsahu zabezpečuje aplikácia *webový prehliadač*. Bez ohľadu na operačný systém, na ktorom sa nachádza používaný webový prehliadač, by mal HTML obsah vyzerať rovnako. HTML súbor vyžaduje minimálne 4 základné elementy:

<html> ... </html>

<head> ... </head>

<title> ... </title>

<body> ... </body>

Tieto elementy označujú základné časti dokumentu HTML: samotný dokument, sekciu hlavičky, sekciu titulku a samotné telo dokumentu. Každý element tvorí pár „tagov“, začiatočný a koncový tag. Tagy sú definované uhlovými zátvorkami a koncový tag obsahuje pre názvom elementu aj lomítko. Niektoré HTML elementy majú len jeden tag. Aby mohol byť podporený skriptovací jazyk, akým je aj JavaScript, tak k základným štyrom elementom sa pridáva aj element:

<script> ... </script>

HTML dokument skoro vždy obsahuje kód jazyka Javascript. Výsledný HTML kód v spolupráci s kódom jazyka JavaScript bude vyzerať nasledovne:

```
<html>
    <head>
        <title> ... </title>
        ...
        <!-- Voliteľné prvky skriptu podľa potreby -->
        <script> ... </script>
    </head>
    <body>
    ...
    </body>
</html>
```

<html> tag uzatvára všetky ostatné elementy a definuje rozsah HTML dokumentu. *<script>* tagy sa objavujú najmä v sekcii hlavičky, ale môžu sa objaviť aj na inom mieste v dokumente. Odsadenie vnorovaných tagov je voliteľné, ale sprehľadňuje danú štruktúru pre vytváranie, čítanie a úpravu HTML dokumentu. Takýto spôsob sa používa vo všetkých jazykoch v praxi.

Keďže HTML a JavaScript sú veľmi úzko späté navzájom, pre používanie jazyka JavaScript musíme ovládať základy jazyka HTML. Pre každého, kto používa alebo sa ešte len učí používať HTML a JavaScript, tak rôzne webové prehliadače interpretujú daný kód rôznymi spôsobmi. Zároveň, každý webový prehliadač podporuje rôzne propiertárne funkcie jazykov HTML a JavaScript. Našťastie, je možné pracovať s určitou štandardizovanou skupinou týchto jazykov.

HTML dokumenty sa zvyčajne používajú ako spôsob distribuovania informácií cez web vzdialeným prístupom, ale sú rovnako cenné, ako keď sa používajú na lokálnom počítači, otvorené webovým prehliadačom. Takže pomocou HTML v spojení s JavaScriptom a neskôr aj s PHP, dokážeme vytvoriť plnohodnotné prostredie pre správu problémov, ktoré je možné buď nasadiť globálne na webový server alebo lokálne na danom počítači.

Dobré programovacie spôsoby zahŕňujú oddelenie časti, kde sa spravujú výpočty od vstupno-výstupného rozhrania. HTML v spojení s JavaScriptom poskytuje prívetivú implementáciu tejto programovacej taktiky. HTML sa stará o vstupno – výstupné rozhranie a JavaScript/PHP zabezpečuje spracovanie výpočtov. Výhodou HTML je, že poskytuje bohaté množstvo možností rozhrania, ktoré prekonali výraznou mierou jazyky založené na textových jazykoch, akým je aj jazyk C [3].

2.3.7 CSS

CSS je skratka pre Cascading Style Sheets (kaskádové štýly) a je to doplňujúci jazyk k jazyku HTML. S jazykom CSS prispôsobujeme vzhľad našich webových stránok. Ukážeme si ako takýto kód vyzerá:

```
header{
    background-color: #111;
    color: white;
    font-size: 16pt;
}
```

V tomto kóde je definované *pravidlo štýlu*. Podstatné je všimnúť si, že v každom z tých 3 riadkov sa nachádza dvojbodka aj bodkočiarka. Všetko vo vnútri zložitých zátvoriek sa nazýva *blok deklarácie*.

Selektor je časť, ktorá sa nachádza pred ľavou zloženou zátvorkou a ktorá nám určuje, v akej časti webovej prezentácie budeme upravovať jej vzhľad. Cieľom nášho pravidla je úprava štýlu elementu *header*. Každý riadok v bloku deklarácií definuje v tomto prípade iba jednu deklaráciu. Deklarácia sa skladá z *vlastnosti* a *hodnoty*. Vlastnosť je časť deklarácie pred dvojbodkou a hodnota časť za dvojbodkou. Každá deklarácia je ukončená bodkočiarkou.

Toto bol veľmi triviálny príklad kódu v jazyku CSS. Iné kódy môžu byť oveľa zložitejšie, ale princíp zapísaného kódu je veľmi ľahké odhaliť samotným experimentovaním. Nie je potrebné sa obávať, keď narazíte na niečo, čo nebude dávať zmysel alebo určitý význam [8].

2.3.8 XAMPP

XAMPP je jeden z multiplatformových webových serverov, ktorý pomáha najmä vývojárom vytvárať a testovať ich programy na lokálnych webových serveroch. Bol vyvinutý spoločnosťou Apache Friends a jeho natívny zdrojový kód môže byť upravený alebo prepracovaný verejnosťou. Pozostáva z Apache HTTP servera, systému správy relačných databáz MySQL MariaDB a interpréta pre rôzne programovacie jazyky, ako je PHP alebo Perl.

XAMPP je skratka, kde X znamená multiplatformový, A predstavuje Apache, M ako MySQL, prvé P predstavuje PHP a druhé predstavuje Perl. Xampp pomáha miestnemu používateľovi alebo serveru otestovať svoju webovú stránku a klientom prostredníctvom počítačov a notebookov pred umiestnením na hlavný webový server. Je to platforma, ktorá poskytuje vhodné prostredie na vytváranie, testovanie a overovanie práce projektov založených na Apache, Perl, MySQL databáze a PHP prostredníctvom systému lokálneho používateľa. Medzi týmto technológiami je Perl, čiže programovací jazyk používaný na vývoj webového vývoja, PHP je backend programovací skriptovací jazyk a MariaDB je najobľúbenejšie používaná databáza vyvinutá spoločnosťou MySQL.

Tu sú vysvetlené tieto, ale aj ďalšie časti z kolekcie tohto softvéru:

- **Cross-Platform:** Rôzne lokálne systémy majú nainštalované rôzne operačné systémy. Táto multiplatformová komponenta bola zahrnutá v zmysle zvýšiť užitočnosť apublicitu pre tento balík distribúcií Apache.
- **Apache:** Multiplatformový webový HTTP server. Používa sa pre poskytovanie webového obsahu na celom svete. Aplikácia serveru je povolená pre inštaláciu zdarma a používaná pre komunitu vývojárov pod záštitou nadácie Apache Software. Vzdialený server Apache poskytuje používateľovi požadované obrázky, súbory a iné dokumenty.

- **MariaDB:** Najprv, systém správy databáz MySQL (DBMS) bol priamou súčasťou XAMPPu, ale teraz bol nahradený s MariaDB. Je to jedno z najpoužívanejších relačných DBMS, vyvinutých spoločnosťou MySQL. Ponúka online služby vyhľadávania, usporiadania, vymazania, manipulácie a ukladania dát.
- **PHP:** Je to backend programovací skriptovací jazyk používaný primárne na vývoj v oblasti webu. Umožňuje používateľom vytvárať dynamické webové stránky a aplikácie. Bol implementovaný pomocou jazyka C. Jeho inštalácia môže byť prevedená na hocijakej platforme a podporuje rôzne systémy správy databáz. PHP predstavuje **hypertextový procesor**. Hovorí sa, že je odvodený z nástrojov osobnej domovskej stránky, čo vysvetľuje jeho jednoduchosť a funkčnosť.
- **Perl:** Je to kombinácia dvoch dynamických jazykov na vysokej úrovni, a to Perl 5 a Perl 6. Perl je možné aplikovať pri hľadaní riešení problémov zameraných na administráciu systému, vývoju webu a vytvárania sietí. Používatelia pomocou jazyka Perl dokážu naprogramovať dynamické webové aplikácie. Je veľmi robustný a flexibilný.
- **phpMyAdmin:** Je to nástroj používaný na riešenie transakcií MariaDB. Jeho hlavnou úlohou je správa DBMS.
- **OpenSSL:** Ide o verejne otvorenú implementáciu protokolom Secure Socket vrstvy a protokolom transportnej vrstvy.
- **XAMPP Control Panel:** Je to ovládací panel, ktorý poskytuje kontrolu nad ostatnými komponentami balíku XAMPP.
- **Webalizer:** Je to softvérové riešenie analýzy webu používané pre užívateľské správy a poskytuje detailnejšie podrobnosti o používaní.
- **Mercury:** Je to poštový dopravný systém, ktorý pomáha riadiť emaily naprieč celým webom.
- **Tomcat:** Je to servlet[1] založený na Jave, aby poskytoval funkcionality jazyku Java.
- **Filezilla:** Je to server protokolu FTP, ktorý podporuje a uľahčuje operácie založené na prenose súborov.

Predtým, ako prejdete cez XAMPP tutoriál, potrebujete mať základné vedomosti o jazykoch vývoja vo webovom prostredí ako sú napr. HTML a PHP [5].

2.4 Architektúra systému webovej aplikácie

Webová aplikácia sa skladá z logickej časti, kde sú spracovávané operácie a z viditeľnej časti, ktorú vidí používateľ. Tieto dve časti je potrebné prepojiť, aby navzájom spolu komunikovali. K tomuto slúži architektúra MVC.

[1] program, ktorá je napísaný v jazyku Java a ktorý sa používa ako nástroj pre tvorbu webových aplikácií

2.4.1 Architektúra MVC

MVC (Model-View-Controller) je architektúra, ktorá je veľmi populárna a ktorá oddeľuje dátový model a riadiace používateľské rozhranie od logiky softvéru. Využíva sa hlavne pri vývoji webových aplikácií. Je tak súčasťou série, či už PHP frameworkov alebo JavaScript frameworkov.

2.4.2 Štruktúra MVC

Je nutné brať MVC architektúru ako pohľad na to, ako by sme mali pristupovať k vývoju webových aplikácií. V praxi sa môžeme stretnúť s rôznymi vysvetleniami jej implementácie alebo princípu, keďže každý jej framework je z pohľadu funkčnosti trochu odlišný od jej iného frameworku. Ale základom je stále jej stavba:

- model
- view (pohľad)
- controller (radič)

Model definuje aplikačnú logiku webovej aplikácie. Model má na starosti všetky výpočty alebo vyhodnocovanie pravdepodobnosti a uchováva v sebe databázu alebo len samotné dáta. Napríklad na stránke e-shopu spravuje prihlásenie užívateľa k jeho účtu a tovar, ktorý užívateľ vložil do košíku. Taktiež jednou z jeho úloh môže byť definícia pravidla pre validáciu dát z formulára alebo môže definovať pravidlá, ktoré musia byť splnené pred odoslaním objednávky.

Nemá vedomosti o tom, na čom pracujú ďalšie dve zložky, slúži len k spracovaniu jednotlivých procesov. Dáta nie sú nikam odosielané, view (pohľad) si ich sám zoberie a spracuje. Z tohto dôvodu do modelu nepatria elementy, ktoré sú závislé na view (pohľade) alebo na controller (radiči).

View (pohľad) nám ukazuje dáta, ktoré sú spracované užívateľským rozhraním webovej aplikácie a modelom, napríklad rôzne tlačidlá a podobne. Určuje nám, akým spôsobom majú byť dáta zobrazované vo webovom prehliadači. Požiadavky, ktoré boli spracované modelom, si od neho sám vyzdvihne.

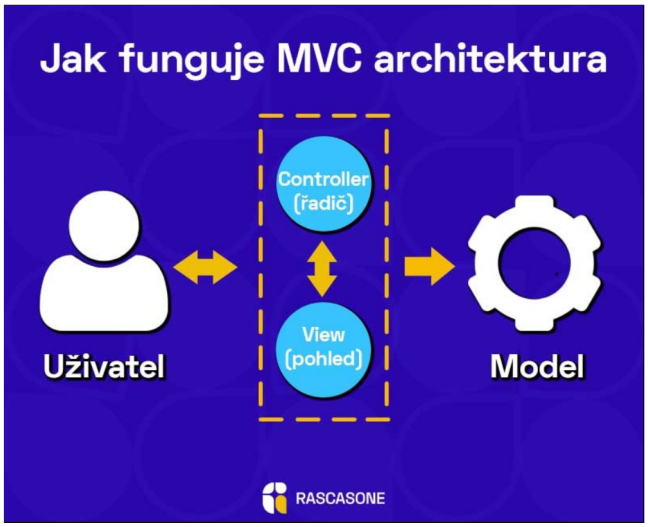
Controller (radič) je pre celú architektúru jej riadiacou jednotkou. Vstupné užívateľské dáta príjma a priraďuje ich k danej aplikačnej logike, kde ich následne view informuje o vykonávaní požiadavky, aby si ich mohol model vyzdvihnúť [7].

2.5 ER Diagram

Entity Relationship diagram je entitno-relačný diagram, ktorý sa využíva pri modelovaní dát pri práci s databázami, je do určitej miery abstraktný a ľahko pochopiteľný [13]. V tejto kapitole je popísaný ER diagram k zadanej databáze a popis jednotlivých entít tejto databázy.

2.5.1 ER diagram a popis k zadanej databáze

Databáza tejto webovej aplikácie bola zadaná vedúcim práce a sa skladá celkovo zo siedmich tabuliek. ER diagram tejto databázy je znázornený na obrázku 2.4 . V podsekciách si vysvetlíme význam daných entít, ich atribútov a ich vzťahy medzi nimi.



Obr. 2.3: Princíp MVC architektúry [7]

Entita Resource

Táto entita nám poskytuje informácie o použitých zdrojoch v tejto databáze.

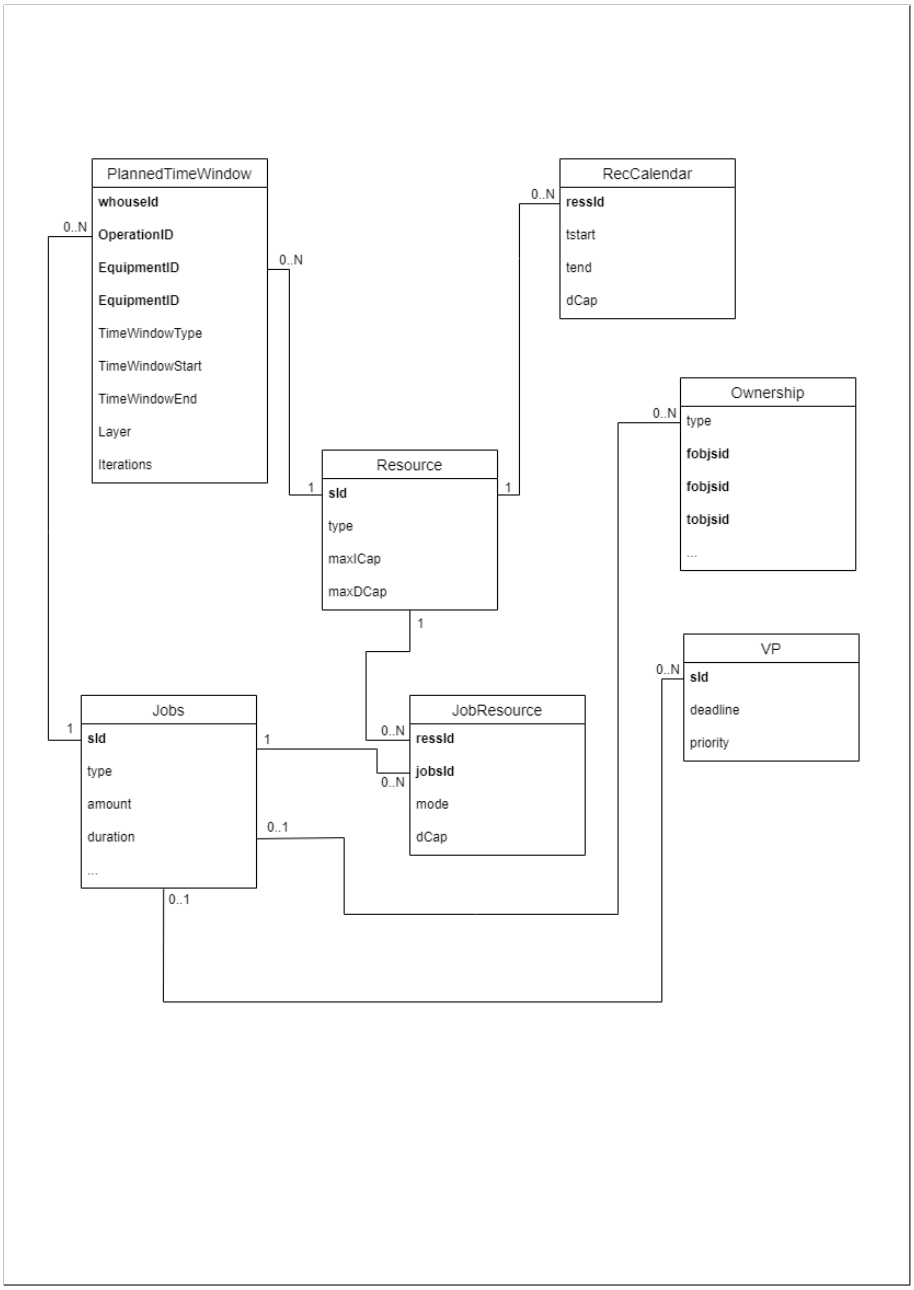
Primárnym kľúčom tejto entity je *sId*. Táto entity uchováva atribúty okrem ID zdroja aj atribút *type*, ktorý predstavuje typ zdroja. Hodnota 0 tohto atribútu predstavuje stroj, hodnota 2 predstavuje personál a hodnota 3 predstavuje nástroj.

Ďalej sú to atribúty *maxICap* a *maxDCap*, kde *maxICap* predstavuje maximálnu kapacitu pre stroj a *maxDCap* predstavuje maximálnu kapacitu pre personál.

Entita ResCalendar

Táto entita nám poskytuje informácie o dostupnosti zdrojov formou kalendárov.

Primárnym kľúčom tejto entity je *ressId*. Entita ďalej obsahuje aj atribút *tstart*, ktorý označuje dátum začiatku bloku postupnosti, atribút *tend*, ktorý označuje dátum konca bloku postupnosti a atribút *dCap*, ktorý nám poskytuje informáciu o kapacite dostupnosti, ktorá je relevantná pre personál.



Obr. 2.4: ER diagram pre zadanú databázu

Entita Jobs

Táto entita nám poskytuje informácie o výrobných operáciach.

Primárnym kľúčom tejto entity je *sId*. Tento atribút definuje ID aktivity (fázovej operácie, plnej operácie). Plná operácia sa skladá z fázových operácií. Atribút *type* nám označuje typ aktivity. Tabuľka rozlišuje tieto typy aktivít:

- 0 - plná operácia
- 2 - fázová operácia inštalácia (INST)
- 3 - fázová operácia rozjazd (ROZJ)
- 4 - fázová operácia výroba (ITER)
- 5 - fázova operácia dokončenie (FIN)
- 6 - údržba
- 7 - kooperácia

Operácia je buď plná operácia (0) alebo kooperácia (7). Kooperácia je operácia vykonávaná mimo rámec podniku, rozvrhuje sa, ale jej fáza nás nezaujíma.

Plná operácia (0) sa skladá z postupnosti fázových operácií INST, ROZJ, ITER, FIN. Fázové operácie sú voliteľné až na fázovú operáciu ITER (výrobná fáza je typicky povinná).

Ďalším atribútom je atribút *amount*, ktorý označuje požadované množstvo do výroby. Atribút *duration* vyznačuje časový úsek aktivity (ak to je fázová operácia ITER, tak hodnota atribútu je v milisekundách, inak je v sekundách).

Entita JobResource

Daná entita označuje požiadavky operácií na zdroje.

Táto entita má dve cudzie a zároveň primárne kľúče, ktorými sú atribúty *ressId* a *jobsId*. *ressId* označuje ID zdroje (stroj, nástroj, personál) a *jobsId* označuje ID aktivity (plná operácia, fázová operácia).

Ďalšími atribútmi sú atribút *mode*, ktorý informuje o čísle módu stroja pre plnú operáciu, hodnot atribútu -1 je pre zdroje typu nástroj a personál. Posledným atribútom je atribút *dCap*, ktorý označuje požadovanú kapacitu pre personál.

JRR (Job-Resource Request) je vo vzťahu buď k plnej operácii (primárny zdroj) alebo k fázovej operácii (sekundárny zdroj). Každá plná operácia vyžaduje aspoň jeden stroj (musí byť primárny). Nástroje, personál a materiál sú voliteľné.

JRR plnej operácie je štruktúrovane:

- množina módov určených strojom (operácia môže byť vykonaná na stroji A alebo B alebo C . . .)
- v každom móde potom platia ostatné požiadavky (mode=-1) na nástroje a personál
- zdroje sú vybrané počas celej doby rozvrhu plnej operácie, resp. počas všetkých jej fáz

JRR fázovej operácie:

- vyžadovanie zdroja len na vykonanie fázovej operácie (typicky personál na INST /ROZJ/ITER)

Entita Ownership

Táto entita označuje vzťahy objektov medzi jednotlivými úlohami.

Táto entita má dve cudzie a zároveň primárne kľúče, ktorými sú atribúty *fobjsId* a *tobjsId*. *fobjsId* označuje ID objektu vytvárajúceho vzťah a *tobjsId* označuje ID objektu vo vzťahu.

Ďalším atribútom je atribút *type*, ktorý označuje typ vzťahu. Hodnota 0 tohto atribútu označuje vzťah vlastníctva, kde plná operácia vlastní fázovú operáciu. Hodnota 1 tohto atribútu označuje vzťah vlastníctva, kde výrobný príkaz VP vlastní plnú operáciu.

Entita VP

Daná entita obsahuje záznamy o výrobných príkazoch.

Primárnym kľúčom tejto entity je *sId*, ktorý označuje ID výrobného príkazu. Ďalšími atribútmi entity sú *deadline*, ktorý označuje termín dokončenia a *priority*, ktorý označuje užívateľom zadanú prioritu.

Celkovo sa štruktúra aktivít hierarchicky skladá z:

- úloha je zadaná ako množina VP
- VP sa skladá z plných operácií a kooperácií
- plná operácia sa skladá z fázových operácií

Entita PlannedTimeWindow

Daná entita obsahuje záznamy o rozvrhnutí daných výrobných operácií.

Táto entita má dve cudzie a zároveň primárne kľúče, ktorými sú atribúty *OperationID* a *EquipmentID*. *OperationID* popisuje ID vykonávanej operácie a *EquipmentID* popisuje ID zdroja alebo materiálu.

Ďalšími atribútmi entity sú *whouseId*, ktorý označuje ID skladu, *TimeWindowType*, ktorý označuje typ spracovávaného bloku, *TimeWindowStart*, ktorý označuje dátum začiatku bloku, *TimeWindowEnd*, ktorý označuje dátum konca bloku, *Layer*, ktorý označuje číslo pracoviska s rozlíšením strojov s násobnou kapacitou a *Iterations*, počet prevedených cyklov výroby pri fázovej operácii ITER.

Atribút *TimeWindowType* rozlišuje tieto typy blokov:

- 0 - rozvrh fázy ITER
- 1 - rozvrh fázy INST
- 2 - rozvrh fázy ROZJ
- 3 - časová obálka rozvrhu plnej operácie (celková doba plnej operácie)
- 5 - údržba
- 7 - prebieha produkcia materiálu
- 8 - rozvrh fázy FIN
- 9 - rozvrh kooperácie

Rozvrh plnej operácie sa skladá z čiastkových blokov zaberania zdrojov (stroj, nástroj, personál, materiál).

Kapitola 3

Návrh riešenia

V tejto kapitole sa venujem spracovaniu dosiahnutých informácií o preskúmaných technológiách a k samotnému návrhu webovej aplikácie, ktorá bude zobrazovať výrobný rozvrh v požadovanom formáte, ďalej k užívateľskému rozhraniu webovej aplikácie, vstupným parametrom pre vizualizáciu výrobného rozvrhu, ovládacím prvkom pre navigáciu vo výrobnom rozvrhu a editačným vlastnostiam webovej aplikácie.

3.1 Návrh webovej aplikácie

Na základe preštudovania programovania webových aplikácií som navrhol webovú aplikáciu, ktorá bude splňovať kritéria napísané v zadaní práce a zároveň, nebude kopírovať funkcionalitu už existujúcej aplikácie.

Ako už bolo spomenuté, výsledkom práce bude vizualizácia výrobných rozvrhov pomocou webovej aplikácie. Táto webová aplikácia, čo sa týka vstupno – výstupného rozhrania, bude užívateľsky prívetivá, ale nebude lipnúť na dizajne kvôli veľkému spracovaniu dát zo vstupnej databázy. Avšak, zovňajškom nebude úplne zaostávať za podobným webovými aplikáciami. Pre zmienený vzhľad budú použité kaskádové štýly CSS.

Aplikácia musí vedieť spracovať veľký objem dát, ktoré obsahujú informácie o rôznych operáciach, preto musí byť komunikácia medzi výpočtovým rozhraním a vstupno – výstupným rozhraním čo najjednoduchšia a najplynulejšia. Rovnako po používateľovej interakcii s webovou aplikáciou musí aplikácia rýchlo zareagovať na prípadne zmeny v požiadavkách na dané operácie a zdroje, a tieto zmeny zobraziť.

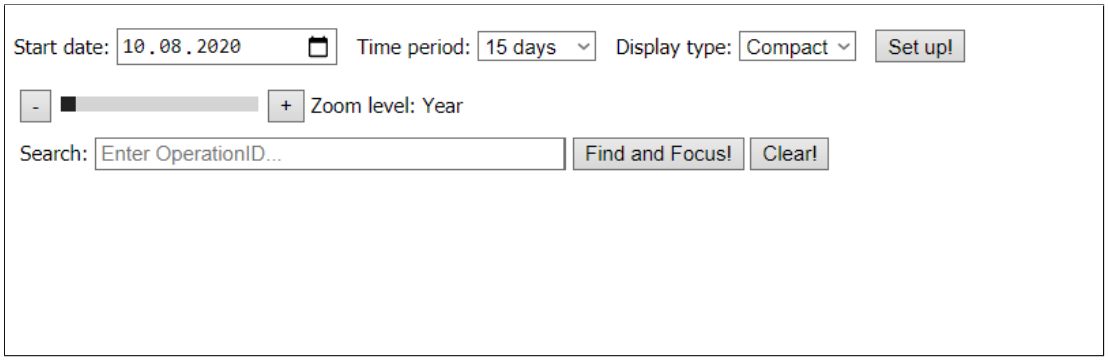
Vo požadovaných vstupných hodnotách je, aby webová aplikácia obsahovala ako vstupnú hodnotu dátum, od ktorého sa majú zobrazovať výrobné operácie, časový úsek, po ktorý sa majú operácie zobraziť, a typ zobrazenia operácií, či sa majú zobraziť len celkové časové obálky operácií alebo aj ich jednotlivé fázy a kooperácie. Po zobrazení výrobných operácií priradených k svojim strojom, ku ktorým sú priradené, používateľ systému bude schopný si zmeniť mierku zobrazeného výrobného rozvrhu. Mierka bude rozdelená do viacerých typov. Ročný typ, kde budú operácie zobrazené len po dňoch. Mesačný typ, ktorý už rozdeľuje v zobrazenom dni vykonávanie operácie dopoludnia alebo popoludní. Týždňový typ nám zobrazí podrobne jednotlivé hodiny v danom dni. Posledným typom je hodinový typ, kde sú operácie priblížené až ku štvrtinám zo zobrazenej hodiny. Aby používateľ sa nemusel zdĺhavo posúvať v zobrazenom výrobnom rozvrhu, tak pre zameranie sa na hľadanú operáciu bude webová aplikácia obsahovať aj vstupné políčko pre hľadaný názov operácie.

Medzi požiadavkami na webovú aplikáciu je aby bola interaktívne editovateľná. Editácia bude umožnená v type zobrazenia *Compact* kvôli prehľadnosti, a bude implementovaná tým spôsobom, že užívateľ presunie časovú obálku výrobnej operácie po časovej osi alebo medzi jednotlivými strojmi. Používateľ bude aj vyzvaný, aby potvrdil vykonanie zmien a prípadne doupravil čas, rádovo v hodinách alebo minútach. V prípade malej zmeny časového úseku používateľ dokáže zmeniť začiatok vykonávania aj kliknutím na danú časovú obálky výrobnej operácie. V prípade odmietnutia zmien sa spracovávaná časová obálka výrobnej operácie vráti do pôvodného stavu.

3.2 Užívateľské rozhranie webovej aplikácie

Vzhľadom k tomu, že táto webová aplikácia má načítavať a spracovávať veľké množstvo dát z konkrétnej databázy, je rozmiestnenie a zobrazenie jednotlivých elementov v HTML dokumente pomerne jednoduché. Dôraz som kládol na to, aby užívateľ vedel, čo má zadať ako vstup pre zobrazenie rozvrhu a ako so zobrazeným rozvrhom ďalej manipulovať.

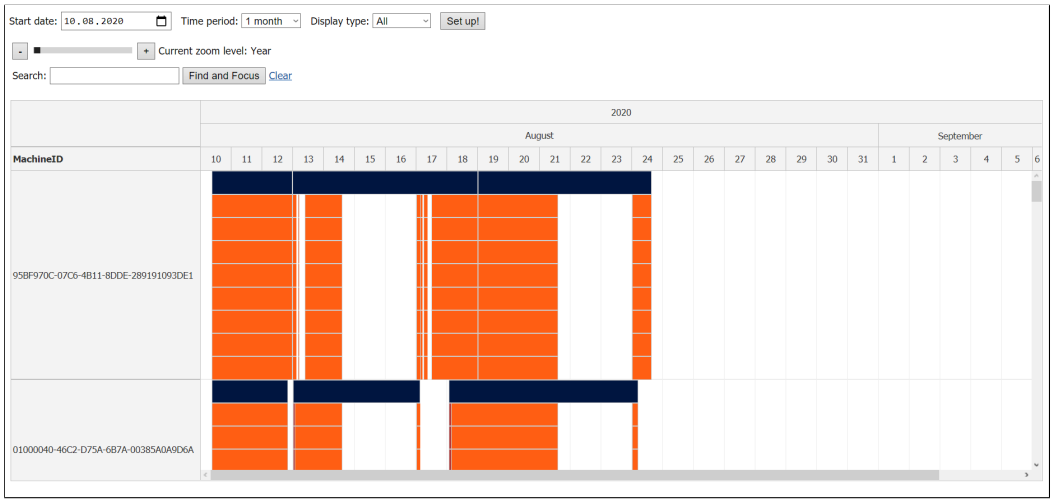
Na obrázku 3.1 vidíme úvodné zobrazenie webovej aplikácie. Po načítaní webovej aplikácii používateľ si zvolí deň, od ktorého chce zobraziť výrobné operácie, časový úsek, po ktorý sa majú zobrazovať výrobné operácie a typ zobrazenia výrobného rozvrhu. Typy rozvrhov sa delia na typ *All*, kde sa zobrazujú plné operácie so svojimi fázovými operáciami a typ *Compact*, kde sa zobrazia len plné operácie bez fázových operácií.



Obr. 3.1: Úvodné načítanie webovej aplikácie

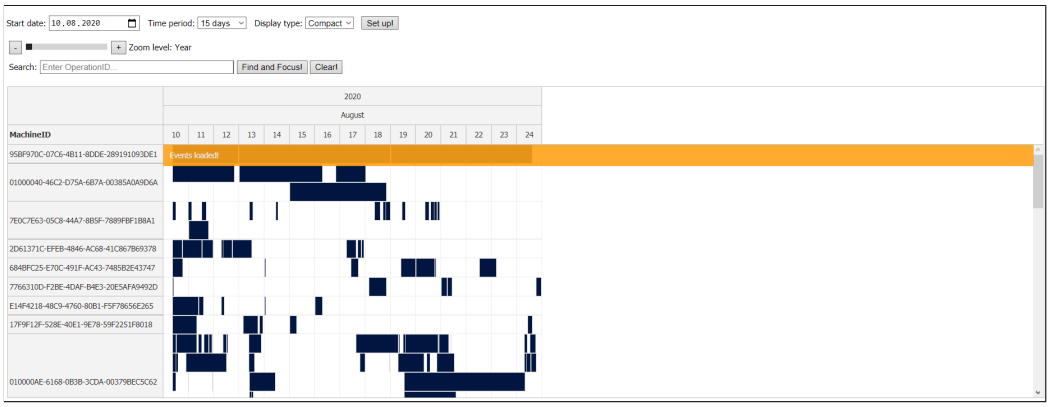
Zároveň, vidíme úroveň priblíženia výrobných operácií v danom časovom úseku. Webová aplikácia podporuje 4 úrovne priblíženia: *Year*, *Month*, *Week* a *Hour*. Úroveň *Year* poskytuje priblíženie na dni a *Hour* poskytuje priblíženie až na štvrť hodiny. Ďalším prvkom vo webovej aplikácii je vstup pre hľadanie výrobnej operácie v rozvrhu, ktorý automaticky zameria časovú os na danú operáciu a zvýrazni ju sivou farbou.

Na obrázku 3.2 vidíme zobrazenie typu *All*, kde sú tmavomodrou farbou zobrazené celkové časové obálky operácií a oranžovou, červenou a inými farbami sú zobrazené ich fázové operácie.



Obr. 3.2: Zobrazenie *All*

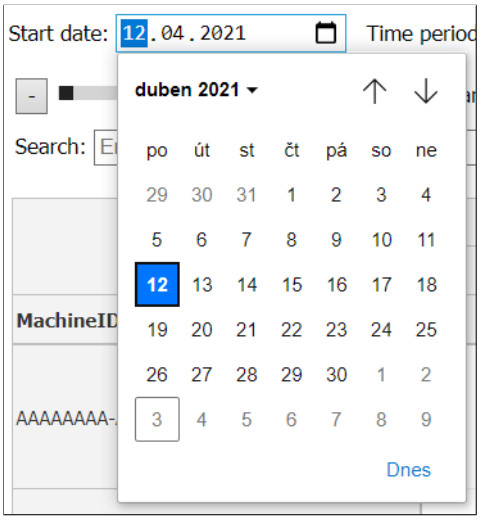
Na obrázku 3.3 vidíme zobrazenie typu *Compact*, kde sú tmavomodrou farbou zobrazené celkové časové obálky operácií a v tomto type zobrazenia sa už nezobrazujú ich fázové operácie. Celkový vzhľad je kompaktnejší. Vidíme aj nápis „Events loaded!“, ktorý sa zobrazí po úplnom načítaní výrobných strojov a operácií.



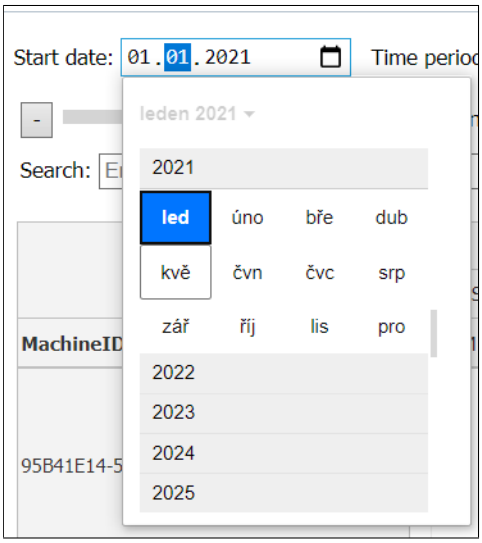
Obr. 3.3: Zobrazenie *Compact*

3.2.1 Vstupné parametre pre vizualizáciu výrobného rozvrhu

Predtým než sa zobrazí požadovaný výrobný rozvrh, je potrebné vyplniť vstupný formulár, kde sa nachádzajú nevyhnutné vstupné požiadavky na zobrazenie výrobného rozvrhu. Prvým je HTML element *<input>* pre dátum, od ktoré sa nám vizualizujú výrobné operácie. Je typu *date*, čo nám umožňuje interaktívne si zvoliť jednotlivé políčka tohto dátumu. Na obrázku 3.4 vidíme, ako si môžeme zvoliť deň v mesiaci hneď po tom, ako klikneme na ikonku kalendára v *<input>* HTML elemente. Na ďalšom obrázku 3.5 vidíme, ako si môžeme zvoliť mesiac a rok začiatočného dátumu, ktoré sa nám zobrazí po tom, čo klikneme na roletku mesiaca a roku v predchádzajúcom obrázku. Samozrejme, celý dátum si vieme vypísať aj ručne.

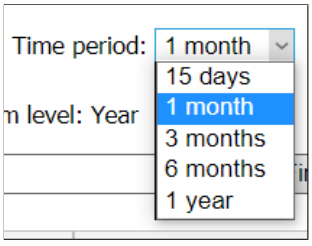


Obr. 3.4: Voľba dňa v mesiaci pri začiatočnom dátume zobrazenia



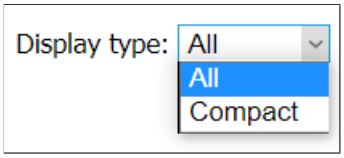
Obr. 3.5: Voľba mesiaca a roku pri začiatočnom dátume zobrazenia

Ďalšou požiadavkou je časový úsek, počas ktorého sa nám majú zobraziť výrobné operácie. Táto požiadavka je implementovaná HTML elementom *<select>* a ponúka nám na výber z 5 možností. Výber týchto možností vidíme na obrázku 3.6.



Obr. 3.6: Výber možností pri volení časového úseku zobrazenia

Poslednou požiadavkou je typ zobrazenia, ktorý je viac priblížený v kapitole 3. Taktiež je táto požiadavka implementovaná pomocou HTML elementu *<select>*, kde máme na výber z 2 možností. Výber týchto možností vidíme na obrázku 3.7



Obr. 3.7: Výber možností pri volení typu zobrazenia

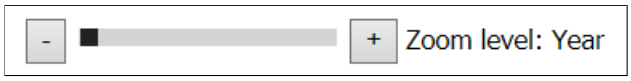
3.2.2 Ovládacie prvky pre navigáciu v rozvrhu

Po načítaní výrobných operácií už môžeme zobrazený výrobný rozvrh ovládať. Máme k dispozícii viacero ovládacích prvkov. Pre zobrazenie detailu výrobnej operácie stačí, ak používateľ prejde myšou na výrobnú operáciu, o ktorej chce zistiť bližšie informácie. Ako vyzerá takýto detail je znázornené na obrázku 3.8.

Obr. 3.8: Detail výrobnej operácie

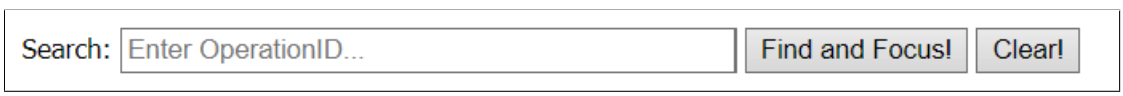
Ďalším ovládacím prvkom je posuv čase. To nám zabezpečuje horizontálny posuvník v zobrazenom výrobnom rozvrhu. Ak horizontálny posuvník nie je zobrazený, vidíme celé časové obdobie, v ktorom majú byť zobrazené výrobné operácie.

Zmenu merítka časovej osi nám zabezpečuje HTML element *<input>* typu *range*, v ktorom máme rozsah tohto elementu rozdelený do 4 pozícií. Pozíciu ukazovateľa nastavujeme pomocou tlačidiel „+“ a „-“. Na obrázku 3.9 vidíme ako merítko vyzerá a aká je jeho predvolená hodnota.



Obr. 3.9: Merítko časovej osi

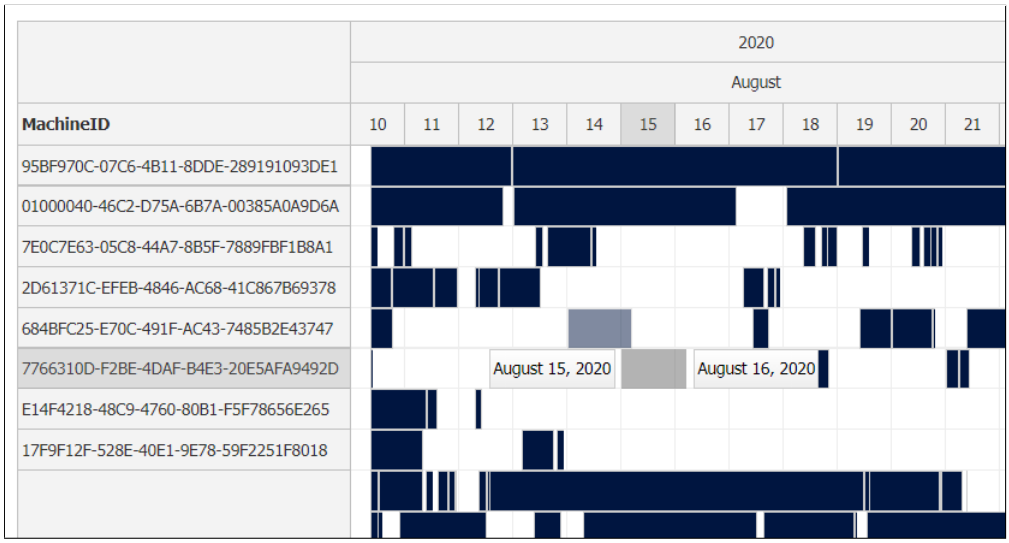
Selekcia zdrojov je už vykonávaná pri vstupných parametroch pre vizualizáciu výrobného rozvrhu, ale taktiež tu máme HTML element *input* typu *text*, ktorý slúži na vyhľadávanie ID operácie, ktorá je zadaná používateľom. Hľadanie sa spustí tlačidlo „Find and Focus!“, pohľad používateľa sa presunie na stred výrobnej operácie a výrobná operácia sa zvýrazní sivou farbou. Tlačidlo „Clear!“ vyprázdni hodnotu hľadaného ID výrobnej operácie. Táto selekcia je zobrazená na obrázku 3.10.



Obr. 3.10: Hľadanie výrobnej operácie

3.3 Editačné vlastnosti webovej aplikácie

Zobrazený výrobný rozvrh je možné aj interaktívne editovať. Editáciu je možné previesť dvoma spôsobmi a to v type zobrazenia *Compact*. Prvým spôsobom je myšou chytiť časovú obálku výrobnej operácie a presunúť ju na miesto, kde a kedy sa má operácia vykonávať. Názorná ukážka je na obrázku 3.11. Po tom, ako udalosť presunieme, sa nám zobrazí okno pre prípadnú zmenu času štartu časovej obálky výrobnej operácie. Ak zaklikneme „Cancel“ alebo klikneme mimo modálne okno, zmeny sa neuložia a detaily časovej obálky výrobnej operácie sa vrátia do pôvodného stavu. Ak zmeníme dátum a čas začiatku časovej obálky výrobnej operácie a zaklikneme „OK“, zmeny sa uložia aj do databázy. Návrh modálneho okna vidíme na obrázku 3.12. Druhý spôsob je pre situáciu, keď chceme len posunúť čas začiatku časovej obálky výrobnej operácie o pár minút, hodín. Nie je potrebné presúvať časovú obálku výrobnej operácie, stačí na ňu kliknúť a zobrazí sa rovnaké modálne okno, ako pri predchádzajúcom spôsobe. Taktiež aj spracovanie tejto editácie je rovnaká ako pri predchádzajúcej editácii.



Obr. 3.11: Presunutie časovej obálky výrobnej operácie

Obr. 3.12: Úprava dátumu a času začiatku časovej obálky výrobnej operácie

Kapitola 4

Implementácia

Táto kapitola sa venuje implementácii webovej aplikácie, pre ktorú boli špecifikované požiadavky a následne návrhy v predchádzajúcich kapitolách. Ako prvé je vysvetlené použitie triedy Scheduler modulu DayPilot, implemenetácia vstupných parametrov pre vizualizáciu výrobného rozvrhu, spracovanie dát vo formulári, zmeny mierky časovej osi, detailu, hľadania a interaktívnej modifikácie výrobnej operácie. V tejto kapitole je zahrnutá aj inštalácia balíčkov, nasadenie a spustenie webovej aplikácie. Ďalej je uvedená aj spolupráca jazyka PHP so súborovou databázou SQLite3, načítanie údajov z tejto databázy ako JSON polí a spracovanie týchto informácií na zdroje a udalosti pre triedu Scheduler v module DayPilot.

4.1 Implementácia systému webovej aplikácie

V tejto sekcii sú podrobne rozpísané časti webovej aplikácie, hlavná stránka i príslušné moduly a funkcie, ktoré boli využité.

4.1.1 Výrobný rozvrh ako objekt triedy Scheduler v module DayPilot

Výrobný rozvrh je implementovaný ako objekt triedy Scheduler (viz kapitola 2.3.3). Knižnice tejto triedy a časť zdrojových kódov som prevzal práve od DayPilot [4]. Funkcia `Scheduler()` vytvorí objekt a nastavia sa atribúty podľa požiadavkov. Príkladom atribútov je že výška bunky vo výrobnom rozvrhu, mierka časovej osi, zobrazenie víkendov, možnosť kliknúť na udalosť alebo presúvať udalosť. Takto vyzerá implementácia objektu DayPilot.Scheduler v jazyku Javascript:

```
//deklaracia objektu triedy DayPilot Scheduler
// a nastavenie prvotnych atributov
var dp = new DayPilot.Scheduler("dp", {
    //predvolene hlavicky casoveho meritka
    timeHeaders: [{"groupBy":"Month"},{"groupBy":"Day","format":"d"}],
    //predvoleny typ hlaviciek casového meritka
    scale: "Day",
    //zobrazenie vikendov
    businessWeekends: true,
    //sirka bunky
    eventHeight: 40,
    //zarovanie na presny sas
```

```
    allowEventOverlap: true,
    //povolena zmena casu
    timeRangeSelectedHandling: "Enabled",
    //nepovolena zmena dlzky casu operacie
    eventResizeHandling: "Disabled",
    //typ zobrazenia detailu operacie
    eventHoverHandling: "Bubble"
});
```

Pomocou premennej `dp` teraz môžeme pristupovať k všetkým atribútom tohto objektu a nastaviť ich podľa požiadavok používateľa.

4.1.2 Vstupné parametre pre vizualizáciu výrobného rozvrhu

Na hlavnej stránke webovej aplikácie sú umiestnené HTML elementy, typu `select` a `input`. V prvom riadku sú vstupné premenné, kde používateľ zadá vstupné hodnoty pre zobrazenie výrobného rozvrhu s daným rozsahom výrobných operácií. Najprv používateľ zadá dátum, od ktorého sa majú zobraziť výrobné operácie a kooperácie vo výrobnom rozvrhu. Následne si vyberie časový úsek, po ktorý sa majú dané operácie a kooperácie zobrazovať, a posledným vstupným parametrom je typ zobrazenia, v ktorom si používateľ zvolí, či chce vidieť len plné operácie alebo aj ich kooperácie. Takto vyzerá ukážka kódu implementácie tohto formulára v jazyku HTML:

```
<form method="POST" action="index.php">
      <label for="start_date">Start date:  </label>
    <input type="date" name="start_date" id="start_date"
    value="2020-08-10">
      <label for="time_period">Time period:  </label>
    <select name="time_period" id="time_period">
        <option value="15 day">15 days</option>
        <option value="1 month">1 month</option>
        <option value="3 month">3 months</option>
        <option value="6 month">6 months</option>
        <option value="1 year">1 year</option>
    </select>
      <label for="type">Display type:  </label>
    <select name="type" id="type">
        <option value="All">All</option>
        <option value="Compact">Compact</option>
    </select>
      <input type="submit" name="scheduler" value="Set up!">
</form>
```

Niektoré HTML elementy už majú prednastavené predvolené hodnoty podľa databázy zadanej vedúcim práce.

4.1.3 Spracovanie vstupných dát vo formulári

Po zvolení a vyplnení vstupných hodnôt vo formulári užívateľ klikne na tlačidlo „Set up!“ a odošle formulárové dáta na spracovanie. Pomocou jazyku PHP (viz kapitola 2.3.1) sa

dáta spracujú a skontroluje validita týchto dát. Tu si môžeme ukázať hlavné spracovanie dát formulára v jazyku PHP:

```
<?php
  //spracovanie udajov pomocou metody POST pri odoslani formulara
  if(isset($_POST['scheduler'])){

    $start_date = $_POST["start_date"];
    $time_period = $_POST["time_period"];
    $type = $_POST["type"];
    //uprava atributov vyrobneho rozvrhu
    if($type=="All"){
      echo "dp.eventMoveHandling = \"Disabled\";";
      echo "dp.eventClickHandling = \"Disabled\";";
    }
    else{
      echo "dp.eventMoveHandling = \"Update\";";
      echo "dp.eventClickHandling = \"Enabled\";";
    }

    //upravenie datumov na pozadovany format
    $date = strtotime(date("Y-m-d H:i:s",
        strtotime($start_date)) ." ". $time_period);
    $start_date = date("Y-m-d H:i:s",strtotime($start_date));
    $end_date = date("Y-m-d H:i:s",$date);
    $end_date = date("Y-m-d",$date);

    //nastavenie vstupnych parametrov na odoslane parametre
    //pre lepsiu vizualizaciu
    echo "document.getElementById('start_date').value =
    '".$_POST["start_date"]."';
    document.getElementById('time_period').value =
    '".$_POST["time_period"]."';
    document.getElementById('type').value =
    '".$_POST["type"]."';";
    echo "dp.startDate = \"$start_date\";";

    //prepocitanie atributu zobrazeneho casoveho useku vyrobneho rozvrhu
    if($time_period==="15 day")$period=15;
    if($time_period==="1 month")$period=30;
    if($time_period==="3 month")$period=90;
    if($time_period==="6 month")$period=183;
    if($time_period==="1 year")$period=365;

    //nastavenie atributu zobrazeneho casoveho useku vyrobneho rozvrhu
    echo "dp.days = $period;";

    //predanie parametrov JavaScriptovej funkcii na dalsie spracovanie
```

```
    echo "loadResources(\"$start_date\",\"$end_date\",\"$type\");";

    //inicializacia vyrobneho rozvrhu
    echo "dp.init();";

    //nastavenie meritka casovej osi
    echo "applyLevel($zoom);";
  }
?>
```

Ďalším krokom je volanie JavaScriptovej funkcie, ktorej sa predajú spracované dáta a pomocou asynchrónnej komunikácie odošle dáta PHP súboru, ktorý ich spracuje a následne posiela dotazy do zadanej databázy. O to sa v súbore `get_resources.php` stará funkcia `tasklist(items)`. Premenná `items` predstavuje zoznam ID zdrojov, na ktorých sa v danom časovom období vykonávajú operácie. Tento zoznam jej posiela funkcia `db_get_tasks( id, start, end, type)`, zo súboru `db.php`, ktorá spravuje aj otvorenie SQLITE (viz kapitola 2.3.5) súboru a hľadá tieto ID strojov podľa vykonávaných operácii v danej databáze. Parameter `type` určuje, či má funkcia hľadať a vrátiť ID strojov, ID fázových operácií a ich časové obálky výrobných operácií alebo len ID časových obálok výrobných operácií. V skratke, ako prvé vyhľadáva ID strojov výrobných operácií v danom časovom úseku a tie následné vracia a do modulu DayPilot (viz kapitola 2.3.3) ich ukladá do atribútu „resources“. Následne sa posiela dotaz na konkrétne výrobné plné/fázové operácie podľa typu zobrazenia, a tie vracia ako vstupné pole pre atribút „events.list“. Posledným krokom je samotné zobrazenie výrobného rozvrhu podľa zadaných parametrov.

4.1.4 Zmena mierky časovej osi

Po zobrazení výrobného rozvrhu je možné so zobrazenými operáciami ďalej pracovať. Konkrétne v tomto prípade to je mierka časovej osi. Máme na výber zo štyroch možností: „Year“,„Month“,„Week“ a „Hour“. Táto udalosť je obsluhovaná JavaScriptovou funkciou, kde pri zmene možnosti sa mení aj trvanie bunky a menia sa aj názvy hlavičiek. Funckia `setZoom()` je imeplementovaná takto:

```
    //funkcia pre zmenu meritka casovej osi
    function setZoom() {

        //obsluha tlacidla minus
        elements.plus.addEventListener("click", function(ev) {
          if (elements.range.value < dp.zoomLevels.length - 1) {
            elements.range.value++;
            applyLevel(elements.range.value);
          }
        });

        //obsluha tlacidla minus
        elements.minus.addEventListener("click", function(ev) {
          if (elements.range.value > 0) {
            elements.range.value--;
            applyLevel(elements.range.value);
```

```
        }
    });

    //predvoleny stav
    applyLevel(elements.range.value);
}
```

4.1.5 Hľadanie operácie v zobrazenom výrobnom rozvrhu

Ďalšou možnosťou pre pre prácu so zobrazenými operáciami je hľadanie konkrétnej operácie podľa jej ID. Užívateľ zadá konkrétne ID operácie a stlačí tlačidlo „Find and Focus!“. Obsluhujúca funkcia v JavaScripte spracuje zadaný parameter a začne prehľadávať ID operácie v zobrazenom liste operácií. Ak dané ID operácie nájde, tak pohľad na výrobný rozvrh sa zameria na stred časového rozmedzia výrobnej operácie. Hľadaná operácia zmení aj farbu pozadia, aby bola ľahšie rozpoznateľná medzi takým kvantom operácií.

4.1.6 Detail samotnej operácie

Funkcia `Bubble()` sa stará o spracovanie udalosti, kde sa majú zobraziť detaily o operácii, na ktorú užívateľ prejde myšou. Funkcia rozlišuje časovú obálku výrobnej operácie a fázovú operáciu. Po prejdení myšou na danú operáciu sa užívateľovi zobrazí ID operácie, dátum a čas začiatku operácie, dátum a čas konca operácie a ak sa jedná o fázovú operáciu, tak sa zobrazí aj príslušný typ fázovej operácie. Implementácie detailu samotnej operácie vyzerá takto:

```
//obsluha detailu vyrobnej operacie
dp.bubble = new DayPilot.Bubble({
    onLoad: function (args) {
      var ev = args.source;
      //synchronizacia s operaciou pre zobrazemie detailu
      args.async = true;

      //oneskorena simulacia nacitania informacii zo strany serveru
      setTimeout(function () {

        //celkova casová obalka vyrobnej operacie
        if(ev.data.type=="OPER"){
          args.html = "<div style='font-weight:bold'>"
          + ev.data.name + "</div><div>Start: "
          + ev.start().toString("yyyy-MM-dd HH:mm")
          + "</div><div>End: "
          + ev.end().toString("yyyy-MM-dd HH:mm") + "</div>";
        }
        //fazova operacia
        else{
           args.html = "<div style='font-weight:bold'>"
           + ev.data.name + "</div><div>Start: "
           + ev.start().toString("yyyy-MM-dd HH:mm")
           + "</div><div>End: "
```

```
            + ev.end().toString("yyyy-MM-dd HH:mm")
            + "</div><div>Type: " + ev.data.type + "</div>";
          }
          args.loaded();
          //casove rozmedzie medzi prejdenim mysou na operaciu
          //a zobrazenim detailu
        }, 500);
      }
});
```

4.1.7 Interaktívna editácia výrobnej operácie

Ako som spomínal v predchádzajúcej kapitole, interaktívnu editáciu je možné previesť v type zobrazenia *Compact*. Funkcia pre obsluhu editácie výrobnou operáciou premiestnením `onEventMoved` spracúva požiadavok na presunutie, buď v časovej osi alebo medzi zobrazenými strojmi, ktoré budú danú operácie spracovávať, a pošle požiadavok do databázy. Najprv sa zobrazí modálne okno, v ktorom je vyplnený údaj o zmene začiatku časové obdobia výrobnej operácie. Pri zrušení modálneho okna alebo kliknutí mimo modálne okno je vo funkcii uvedená podmienka, kde sa údaje o výrobnej operácie vrátia späť na pôvodné údaje pred zmenou a do databázy sa neposiela žiadny dotaz. Pri potvrdení modálneho okna, sa zavolá funkcia `update_db( id, oper_id, equip_id, start, end)`, ktorá preberie zmenené údaje výrobnej operácii a pomocou asynchrónnej komunikácii so súborom `event_update.php` sa odošle dotaz na všetky pôvodné informácie týkajúce sa celkovej časovej obálky výrobnej operácie a jej fázových operácií, prepočítajú sa časové údaje a po ich prepočítaní sa odošlú dotazy do databázy na uvedené zmeny pomocou MySQL príkazov *UPDATE*. Zmeny sa aktualizujú aj vo výrobnom rozvrhu pomocou funkcie `db.update()`. Veľmi podobnú úlohu má aj funkcia `onEventClick`, ktorá avšak obsluhuje editáciu výrobnej operácie kliknutím na požadovanú časovú obálku výrobnej operácie. Modálne okno a spracovanie údajov je rovnaké, ale nemení sa stroj, na ktorom je operácia vykonávaná. V jazyku JavaScript je funkcia `onEventMoved` implementovaná nasledovne:

```
//obsluha pre nastavenie modalneho okna pri editacii
//vyrobnej operacie presunutim vyrobnej operacie
dp.onEventMoved = function (args) {

    //data vo formulari v modalnom okne
    var form = [
      {name: "Start", id: "start", dateFormat: "d.M.yyyy H:mm:ss"},
    ];

    //nacitanie detailov upravovanej vyrobnej operacie
    var data = args.e.data;

    var options = {
      focus: "start"
    };

    //pomocny vypis
    //console.log(data.name,data.resource);
```

```
        //modalne okno
        DayPilot.Modal.form(form, data, options).then(function(modal) {
          dp.clearSelection();

          //postup pri zruseni zmien
          if (modal.canceled) {

            //najdenie danej operacie
            var e = dp.events.find(data.id);

            //zrusenie zmien pri zruseni modalneho okna
            e.data.start = restore_start;
            e.data.end = restore_end;
            e.data.resource = restore_resource;
            dp.events.update(e);
            return;
          }

          //uprava informacii do databazy
          update_db(data.id,data.name,data.resource,
          modal.result.start,data.end);
        }
    )};
```

4.2 Inštalácia balíčkov, nasadenie a spustenie webovej aplikácie

V tejto sekcii je popísaná inštalácia balíčku XAMPP s webserverom Apache spolu s nasadením a spustením webovej aplikácie.

4.2.1 Inštalácia balíčku XAMPP

Na oficiálnej stránke http://www.xampp.org/ si nájdeme a stiahneme inštalátor podľa operačného systému, kde bude nasadená webová aplikácia. Inštaláciu je lepšie vykonať ako administrátor. Ak sa vám zobrazí hláška o UAC (User Account Control), tak iba klikneme na tlačidlo „OK“.

Ďalšiu hlášku, ktorá sa môže zobraziť pri inštalácii, sa týka Microsoft Visual C ++ runtime, ktorá je pre XAMPP vyžadovaná. Túto hlášku potvrdíme a budeme presmerovaný na webstránku, kde túto komponentu je možné stiahnuť. Po jej naištalovaní pokračujeme ďalej v inštalácii balíčku XAMPP. Nasledujúce okná stačí už len potvrdiť.

V rámci zvolenia súčastí pre správne fungovanie webovej aplikácie je potrebné zaklknúť v časti „Server“ možnosť *Apache*, webserver, na ktorom aplikácia pobeží. V rámci časti „Program Languages“ možnosť *PHP*, jazyk, v ktorom je webová aplikácia napísaná. V nasledujúcom okne môžeme ponechať ako inštalačnú zložku „C:\xampp\“, pretože v zložke „Program Files\“ by mohol nastať problém s právami zápisu. Počas inštalácie budeme pravdepodobný vyzvaný na potvrdenie prístupu k sieti pre nové služby, túto akciu potvrdíme.

Po doinštalovaní balíčku XAMPP môžeme potvrdiť spustenie ovládacieho panelu aplikácie XAMPP.

4.2.2 Spustenie ovládacieho panelu XAMPP

Aplikáciu XAMPP Control Panel buď spustíme pomocou vyhľadávania nainštalovaných aplikácií alebo v rámci inštalácie. Ak je aplikácia spustená, pred tým ako spustíme webserver Apache, klikneme na tlačidlo „Config“ a vyberieme súbor „php.ini“ pre editáciu. Vo vybranom súbore odkomentujeme rozšírenia „extension=pdo_sqlite“ a „extension=sqlite3“, ktoré sú potrebné pre prácu s databázovým súborom SQLITE3. Zmeny uložíme a súbor zavrieme. V ovládacom paneli v prvom riadku „Apache“ zaklikneme tlačidlo „Start“. Ak by sa webserver nespustil, príčinou môže byť aj aplikácia Skype, ktorá zaberá požadované porty, na ktorých beží webserver. Riešením je vypnutie aplikácie Skype alebo jej prenastavenie na iný port než port 80 a spustenie webservera.

4.2.3 Nasadenie a spustenie webovej aplikácie

Z priloženého média rozbalíme súbor `xgladi00.zip` a presunieme priečinok `bachelor/` s celým jeho obsahom do priečinku `C:/xampp/htdocs/`. Otvoríme si webový prehliadač a zadáme URL: `http://localhost/bachelor/`. V okne sa nám zobrazí pripravená webová aplikácia.

Kapitola 5

Testovanie

Dôležitou časťou pri vývoji webových aplikácií je testovanie. Účelom testovania je to, aby webová aplikácia bola verifikovateľná a validná, a prípadne aby sme odhalili nedostatky a chyby v danej webovej aplikácii.

Verifikácia nám overuje, či je výsledný produkt vyhovuje správnosti vzhľadom k formulovaným požiadavkám. Verifikácia porovnáva návrh s výsledným produktom, v tomto prípade návrh webovej aplikácie s výslednou webovou aplikáciou. Verifikáciu vykonáva programátor v priebehu vývoja webovej aplikácie.

Validácia nám overuje, či je výsledný produkt vyhovuje správnosti vzhľadom k reálnym požiadavkám. Validácia porovnáva, či výsledný produkt splňuje požiadavky zadané zákazníkom. Na kontrolu validácie slúžia validačné testy, ktoré skontrolujú, či webová aplikácia naozaj odpovedá zákazníkovým požiadavkám.

Táto webová aplikácia bola v priebehu celého vývoja testovaná. Pri každom zakomponovaní novej súčasti bola táto súčasť otestovaná na to, či spĺňa svoju požadovanú funkciu. Postupné testovanie webovej aplikácie bolo vykonávané po malých častiach. Nakoniec bola otestovaná celá webová aplikácia ako celok.

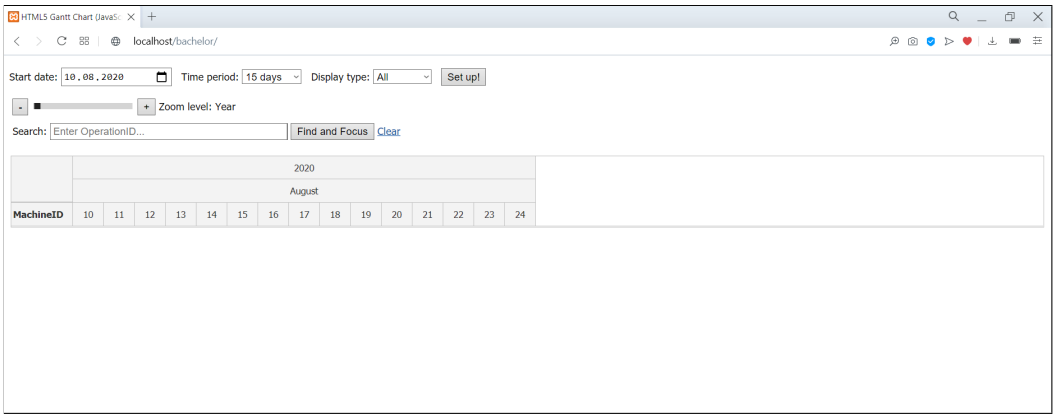
5.1 Zhodnotenie funkčnosti

Podľa predchádzajúcich odstavcov je v tejto sekcii ukázané, ako daná webová aplikácia funguje. Funkcia webovej aplikácie bude demonštrovaná pomocou základných činností, ktoré boli požadované. Viac o požiadavkách na základné činnosti webovej aplikácie sa dozviete v kapitole 2. Testovacími stránkami sú správne zobrazenie výrobných operácií, čo v sebe zahŕňa zmeny časového merítka, zobrazenie detailu výrobnej operácie, farebné odlíšenie výrobných operácií, hľadanie výrobnej operácie a zobrazenie výrobných operácií podľa požadovaných časových úsekov.

5.1.1 Testovanie načítania a zobrazenia operácií vo výrobnom rozvrhu

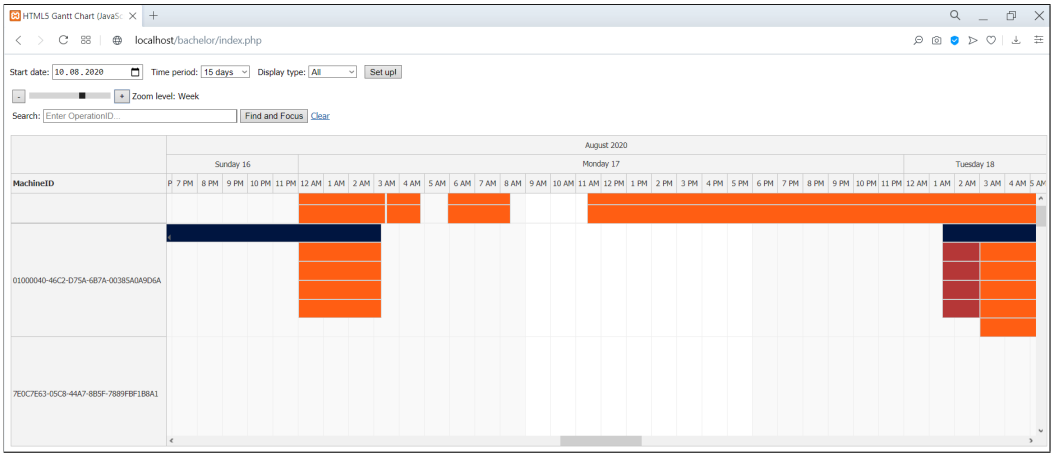
Užívateľovi sa po zadaní linku na webovú aplikáciu, kde je webová aplikácia uložená, zobrazí hlavná stránka s úvodnými nastaveniami. Táto stránka je na obrázku 5.1. Tu užívateľ vyplní požadované údaje. Medzi tieto údaje patrí dátum začiatku, ktorý určuje hranicu, od ktorého sa majú zobrazovať časové obálky výrobných operácií s ich fázovými operáciami. Ďalším údajom je časové obdobie, po ktoré sa výrobné operácie majú zobrazovať, a do tohto obdobia aj končia. Posledným údajom je typ zobrazenia, od ktorého závisí, či sa zobrazia časové obálky výrobných operácií aj s ich fázovými operáciami alebo bez nich.

Potom, čo užívateľ vyplní všetky tieto údaje, stlačí tlačidlo „Set up!“, začne sa proces spracovania požiadavok a zobrazenie výsledného výrobného rozvrhu. Teraz je už možné meniť úroveň priblíženia vo výrobnom rozvrhu a aj hľadať a zamerať sa na požadované ID časovej obálky výrobnej operácie alebo fázovej operácie. Nie je potrebné najprv vyplniť požadované údaje a až potom meniť úroveň merítka časovej osi výrobného rozvrhu.



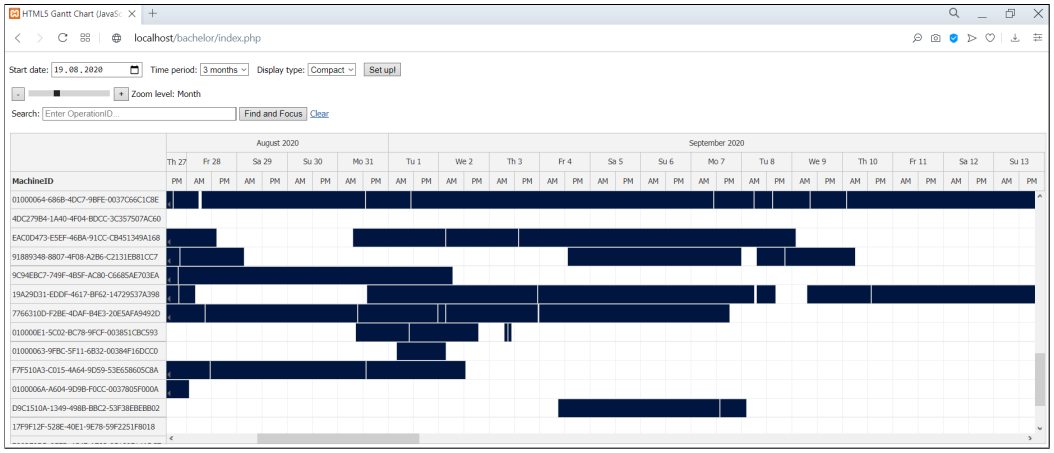
Obr. 5.1: Snímka obrazovky hlavnej stránky s úvodnými nastaveniami. Tu sa vyplňujú informácie ako dátum začiatku, od ktorého sa majú vybrané výrobné operácie zobrazovať, časové obdobie, po ktoré sa majú výrobné operácie zobrazovať a typ zobrazenia.

Ako vyzerá zobrazený rozvrh môžeme vidieť na obrázku 5.2. Na snímke je ako typ zobrazenia zvolený „All“, takže s časovými obálkami výrobných operácií sú zobrazené aj fázové operácie. Môžeme vidieť farebné rozlíšenie operácií, kde časové obálky výrobných operácií sú zobrazené tmavomodrou farbou a ich fázové operácie sú zobrazené inými farbami ako tmavomodrou.

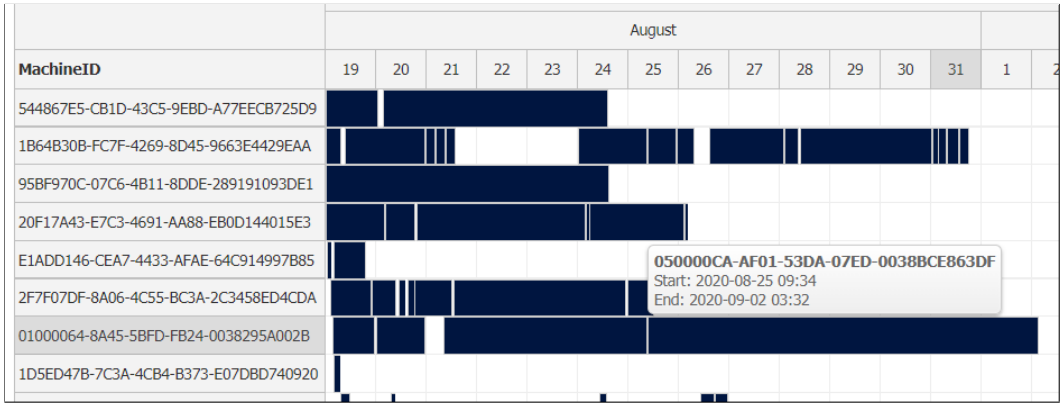


Obr. 5.2: Snímka obrazovky hlavnej stránky so zobrazeným rozvrhom, kde je ako typ zobrazenia zvolený „All“

Výrobný rozvrh s typom zobrazenia môžeme vidieť na obrázku 5.3. Na danom obrázku vidíme len plné operácie označené tmavomodrou farbou. Posledný obrázok 5.4 nám poukazuje na to, ako sa zobrazí detail, ak prejde používateľ myšou na časovú obálky výrobnej operácie a na to, aké informácie nám tento detail poskytuje.



Obr. 5.3: Snímka obrazovky hlavnej stránky so zobrazeným rozvrhom, kde je ako typ zobrazenia zvolený „Compact“



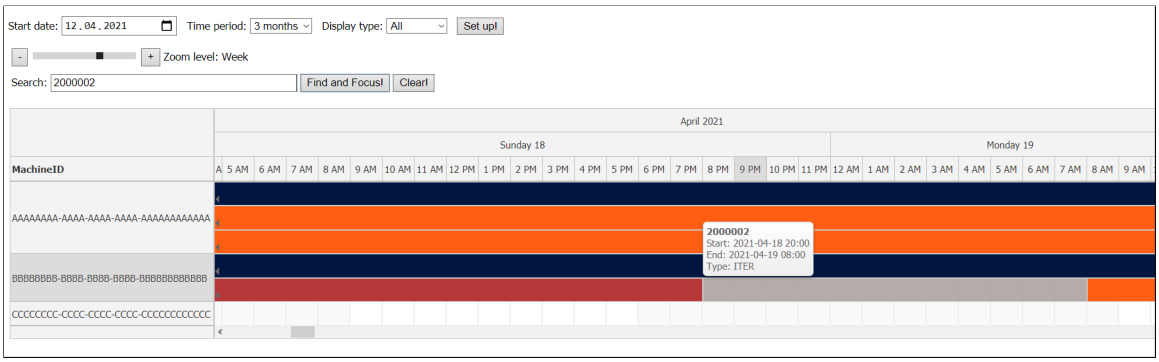
Obr. 5.4: Snímka obrazovky hlavnej stránky so zobrazeným rozvrhom a so zameraním na detail plnej výrobnej operácie

5.2 Testovanie aplikácie s použitím rôznych rozvrhov

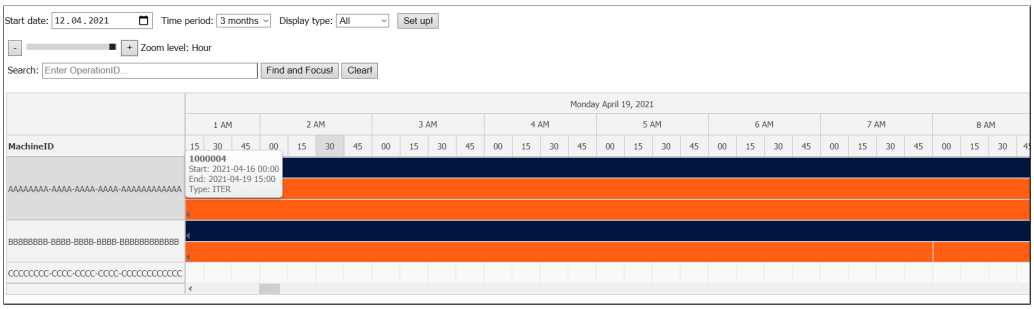
5.2.1 Testovanie na rozvrhu č.1

Na tomto vygenerovanom rozvrhu bolo otestované hľadanie ID výrobnej operácie, správne zobrazenie výrobného rozvrhu v hodinovom a v mesačnom časovom merítku a správnosť zobrazenia výrobného rozvrhu v časovom intervale jeden mesiac. Webová aplikácia splnila zadané požiadavky a zobrazila požadované výrobné operácie, na obrázku 5.5 je znázornené hľadanie ID výrobnej operácie, na obrázku 5.6 je znázornený výrobný rozvrh v hodinovom časovom merítku, na obrázku 5.7 je znázornený výrobný rozvrh v hodinovom časovom

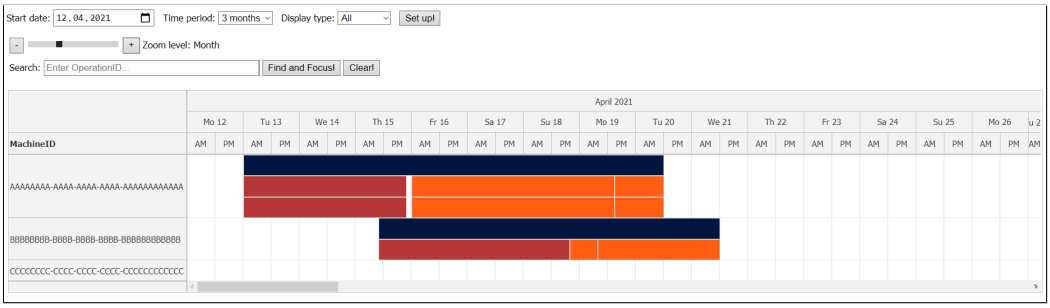
merítku a na poslednom obrázku je 5.8 sú znázornené výrobné operácie v časovom úseku 1 mesiac.



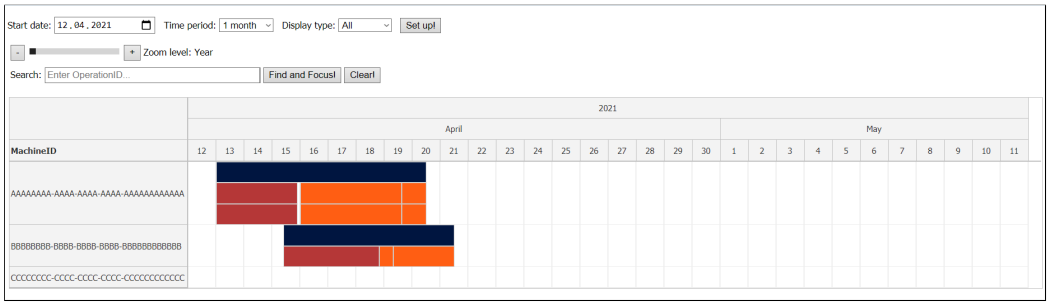
Obr. 5.5: Hľadanie ID operácie



Obr. 5.6: Zobrazenie v hodinovom časovom merítku



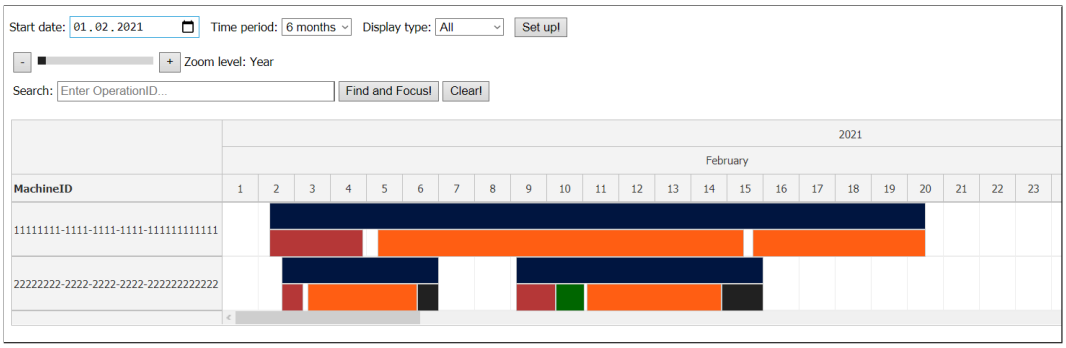
Obr. 5.7: Zobrazenie v mesačnom časovom merítku



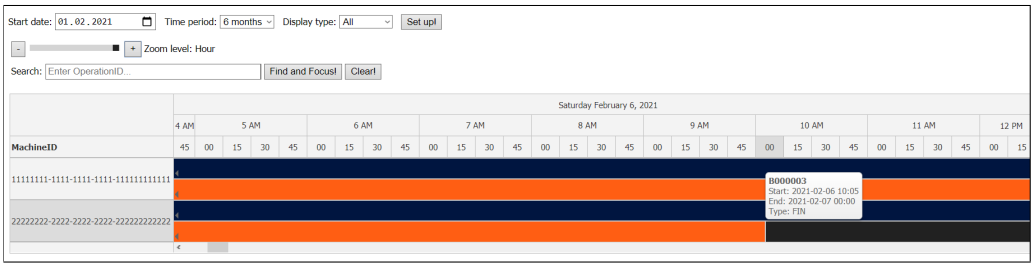
Obr. 5.8: Zobrazenie výrobných operácií v časovom úseku 1 mesiac

5.2.2 Testovanie na rozvrhu č.2

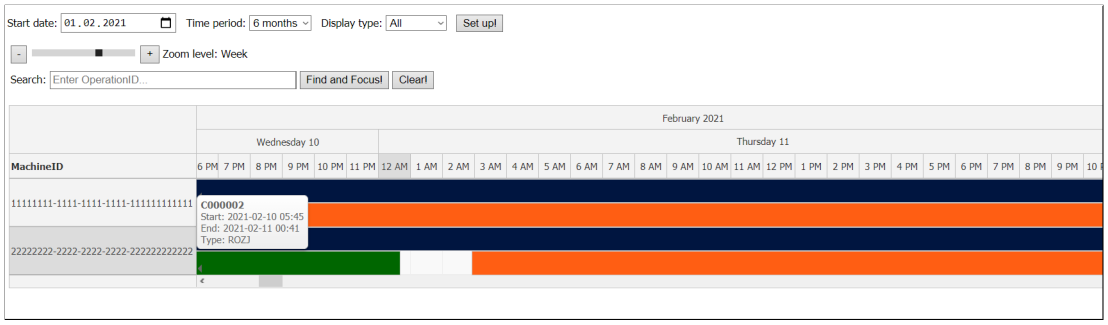
Na tomto vygenerovanom rozvrhu bolo otestované farebné odlíšenie výrobných operácií, časové zarovnanie výrobných operácií v zadanom časovom merítku a zobrazenie výrobných operácií v týždňovom časovom merítku. Webová aplikácia splnila zadané požiadavky a zobrazila požadované výrobné operácie, na obrázku 5.9 je znázornené farebné odlíšenie výrobných operácií, na obrázku 5.10 je časové zarovnanie výrobných operácií v zadanom časovom merítku a na obrázku 5.7 je znázornené zobrazenie výrobných operácií v týždňovom časovom merítku.



Obr. 5.9: Farebné odlíšenie výrobných operácií



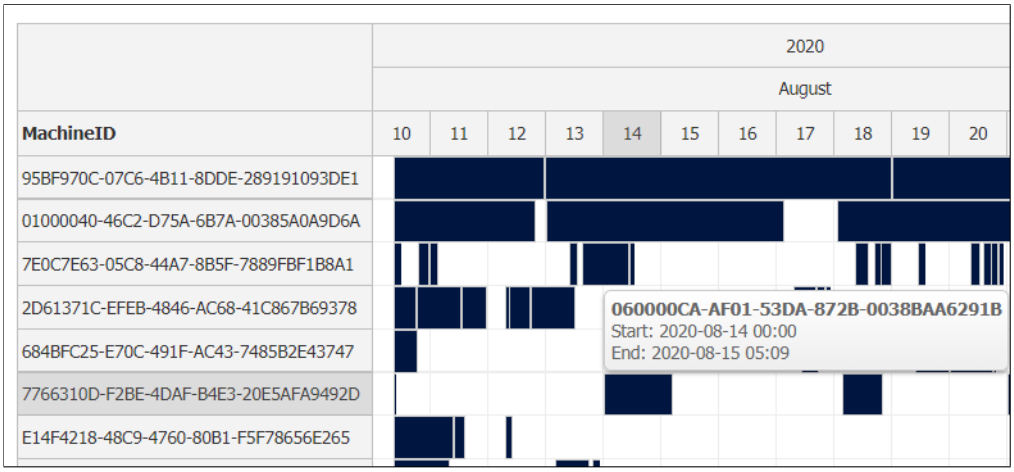
Obr. 5.10: Časové zarovnanie výrobných operácií v zadanom časovom merítku



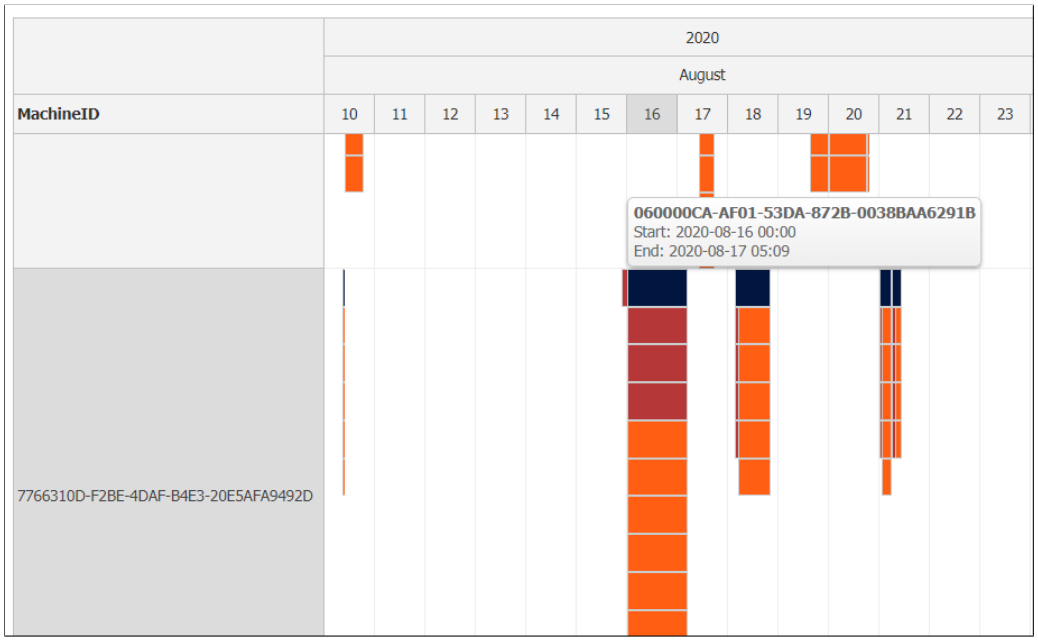
Obr. 5.11: Zobrazenie v týždňovom časovom merítku

5.2.3 Testovanie na rozvrhu č.3

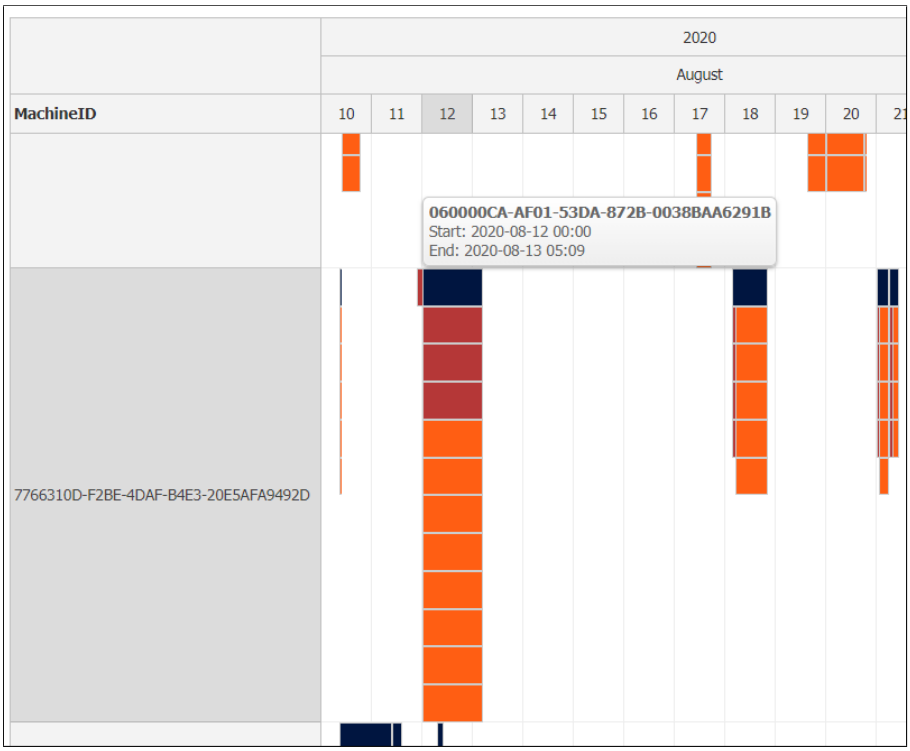
Na tomto rozvrhu, ktorý bol zadaný vedúcim práce, bolo otestované posúvanie výrobných operácie v rámci časovej osi smerom vpred a aj smerom vzad, či sa posúvajú všetky operácie týkajúce sa jednej časovej obálky, a aj posúvanie výrobných operácií medzi strojmi. Webová aplikácia splnila zadané požiadavky a zobrazila požadované výrobné operácie, na obrázku 5.12 vidíme pôvodnú pozíciu časovej obálky fázových operácií, na obrázku 5.13 je časová obálka operácie posunutá vpred v rámci časovej osi, na obrázku 5.14 je časová obálka operácie posunutá vzad v rámci časovej osi a na obrázku 5.15 je znázornená časová obálka posunutá na iný stroj v rámci výrobného rozvrhu.



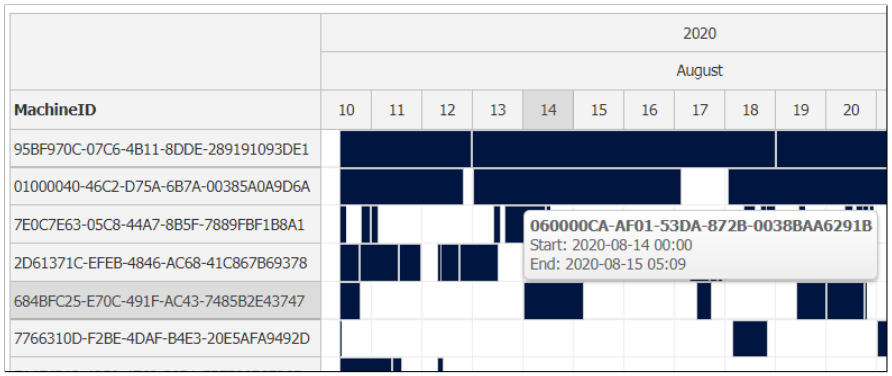
Obr. 5.12: Pôvodná pozícia časovej obálky operácie



Obr. 5.13: Posunutie celej časovej obálky výrobnej operácie vpred



Obr. 5.14: Posunutie celej časovej obálky výrobnej operácie vzad



Obr. 5.15: Posunutie celej časovej obálky výrobnej operácie na iný stroj

Kapitola 6

Záver

Ako sa spomína v úvode, cieľom tejto práce bolo vizualizovať výrobné rozvrhy pomocou webovej aplikácie, ktorá bude podobná ako nástroje pre optimalizáciu výroby, a ktorá bude interaktívna a pre zobrazený výrobný rozvrh bude ponúkať možnosť editácie. Vo webovej aplikácii si bude možné zvoliť obdobie, po ktoré sa majú zobraziť výrobné operácie vo výrobnom rozvrhu a typ zobrazenia tohto výrobného rozvrhu.

Najprv bolo potrebné analyzovať problematiku výrobných procesov a výrobných operácií, vizualizácie výrobných procesov, aby bolo jasné, aké úlohy má webová aplikácia spĺňať. Po spísaní teoretického rozboru nasledoval návrh riešenia, konkrétne návrh užívateľského rozhrania webovej aplikácie a celkového návrhu, kde boli zahrnuté aj priložené materiály od vedúceho práce Ing. Martina Hrubýho PhD., hlavne v rámci vytvárania ER diagramov, ktoré pomohli určiť vzťahy medzi jednotlivými entitami.

Ďalšou časťou bola implementácia webovej aplikácie, kde bola opísaná použitá architektúra systému a časti systému, z ktorých sa webová aplikácia skladá. V rámci tejto časti je opísané aj nasadenie a spustenie webovej aplikácie.

Nakoniec bola otestovaná webová aplikácia na požiadavky, ktoré sú spomenuté v zadaní práce a na koľko dané požiadavky implementovaná webová aplikácia splňuje. Avšak, už počas vývoja bola aplikácia testovaná po menších celkoch. To pomohlo odhaliť chyby a doladiť systém ešte pred konečným testovaním.

Po zhodnotení funkčnosti práce sa prišlo na pár vylepšení, ktoré by spravili aplikáciu viac prenositeľnou. Jedným z týchto vylepšení je aj lepšie grafické spracovanie pomocou softvérov, ktoré sú zamerané na responzívne vlastnosti webových aplikácií, a tak by bolo možné optimalizovať webovú aplikáciu či už pre mobilné telefóny alebo tablety.

Počas vývoja tejto webovej aplikácie som získal skúsenosti hlavne z asynchrónnej komunikácii medzi jazykmi PHP a JavaScript, a správe databáz z SQLite súborov. Najväčšiu skúsenosť som nadobudol z údržby a testovania webovej aplikácie.

Literatúra

[1] Achour, M., Betz, F. et al. *PHP: What is PHP?* [online]. 2020 [cit. 2021-03-22]. Dostupné z: https://www.php.net/manual/en/intro-whatis.php.

[2] Boronczyk, T. *MySQL Okamžitě*. 2. vyd. Brno: Academia, 2016. ISBN 978-80-251-4737-5.

[3] Brooks, D. R. *Guide to HTML, JavaScript and PHP: For Scientists and Engineers*. 1. vyd. London: Springer-Verlag, 2011. ISBN 978-0-85729-448-7.

[4] DayPilot. *JavaScript Scheduler* [online]. 2021 [cit. 2021-03-26]. Dostupné z: https://javascript.daypilot.org/scheduler/.

[5] Javatpoint. *XAMPP TUTORIAL* [online]. 2018 [cit. 2021-03-26]. Dostupné z: https://www.javatpoint.com/xampp.

[6] Keřkovský, M. a Valsa, O. *Moderní přístupy k řízení výroby*. 3. vyd. Praha: C. H. Beck, 2012. ISBN 978-80-7179-319-9.

[7] Koďousková, B. *Architektura MVC: definice, struktura, frameworky*. Rascasone s.r.o., Apr 2021. Dostupné z: https://www.rascasone.com/cs/blog/architektura-mvc-struktura-frameworky.

[8] Lazaris, L. a Baše, O. *CSS Okamžitě*. 1. vyd. Brno: Computer Press, 2014. ISBN 978-80-251-4176-2.

[9] Lile, S. *How to Use Gantt Charts for Better Data Visualization Spin Sucks* [[online]]. July 2018. [vid. 2021-04-30]. Dostupné z: https://spinsucks.com/marketing/gantt-charts-data-visualization/.

[10] Point, T. *JavaScript Tutorial* [online]. 2015 [cit. 2021-03-22]. Dostupné z: https://www.tutorialspoint.com/javascript/javascript_tutorial.pdf.

[11] ProductPlan. *Gantt Chart* [online]. 2021 [cit. 2021-04-08]. Dostupné z: https://www.productplan.com/glossary/gantt-chart/.

[12] SQLite. *SQLite* [online]. 2021 [cit. 2021-03-26]. Dostupné z: https://www.sqlite.org/about.html.

[13] Watt, A. a Eng, N. *Database Design, 2nd Edition* [online]. The BCcampus Open Textbook Project, 2014 [cit. 2021-04-10]. Dostupné z: https://open.umn.edu/opentextbooks/textbooks/354.

Príloha A

Obsah priloženého pamäťového média

Priložené CD obsahuje `xgladi00.zip` súbor, ktorý po rozbalení má nasledovnú štruktúru:

- `bachelor/` - obsahuje zdrojové súbory webovej aplikácie
- `pdf/` - obsahuje pdf súbor s technickou správou
- `pdf_latex/` - obsahuje zdrojové súbory technickej správy napísanej v jazyku LaTeX