



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV INFORMAČNÍCH SYSTÉMŮ

DEPARTMENT OF INFORMATION SYSTEMS

**WEBOVÁ VIZUALIZACE DAT ZE ZAŘÍZENÍ
URČENÝCH PRO KONTROLU KVALITY**

WEB VISUALIZATION OF QUALITY CONTROL DATA FROM INDUSTRIAL DEVICES

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

DMITRII KOZHEVNIKOV

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. PETR JOHN

BRNO 2023

Zadání bakalářské práce



147784

Ústav: Ústav informačních systémů (UIFS)
Student: **Kozhevnikov Dmitrii**
Program: Informační technologie
Specializace: Informační technologie
Název: **Webová vizualizace dat ze zařízení určených pro kontrolu kvality**
Kategorie: Webové aplikace
Akademický rok: 2022/23

Zadání:

1. Prostudujte možná použití chytrých zařízení v průmyslu, zaměřte se na zařízení pro kontrolu kvality a způsoby jejich komunikace.
2. Prostudujte principy tvorby přehledových obrazovek typu dashboard, způsoby jejich návrhu a existující typy vizualizací.
3. Prostudujte dostupné technologie pro tvorbu webových uživatelských rozhraní, vizualizaci dat a jejich testování (např. React, Nivo, D3.js, Cypress, apod.).
4. Proveďte analýzu požadavků na zobrazení dat získaných ze zařízení pro kontrolu kvality používaných firmou Sanezoo.
5. Navrhněte webové komponenty a sadu pohledů tvořených z nich, které dokáží zpracovat a vizualizovat data ze zařízení.
6. Navržené pohledy a komponenty implementujte.
7. Výsledné řešení otestujte a navrhněte další možná rozšíření.

Literatura:

- Few, S. Information Dashboard Design: Displaying Data for At-a-glance Monitoring. Analytics Press, 2013. ISBN 9781938377006.
- Eckerson, W. Performance Dashboards: Measuring, Monitoring, and Managing Your Business. Wiley, 2010. ISBN 9780470589830.
- *React: Getting Started* [online]. [cit. 2022-09-14]. Dostupné z: <https://reactjs.org/docs/getting-started.html>
- *Cypress: Why Cypress?* [online]. [cit. 2022-09-14]. Dostupné z: <https://docs.cypress.io/guides/overview/why-cypress#What-you-ll-learn>

Při obhajobě semestrální části projektu je požadováno:
Body 1 až 5.

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

Vedoucí práce: **John Petr, Ing.**
Vedoucí ústavu: Kolář Dušan, doc. Dr. Ing.
Datum zadání: 1.11.2022
Termín pro odevzdání: 31.7.2023
Datum schválení: 25.10.2022

Abstrakt

V současné době se každý specialista potýká s velkým množstvím dat, a jeho cílem je pracovat s nimi co nejefektivněji. Často ale chybí platformy, které jsou schopny poskytnout efektivní zobrazení informace pro každého uživatele a umožňují individuálního přizpůsobení. Tato bakalářská práce se zabývá ukládáním dat a zobrazováním za pomoci dashboardů. Cílem práce bylo navrhnout a vytvořit dashboard pro organizace zaměřené na kontrolu kvality, který umožní nejen efektivní zobrazení potřebných dat, ale také poskytne uživatelům možnost přizpůsobit zobrazené informace podle jejich požadavků a úkolů. Některé části tohoto řešení byly nasazeny ve firmě Sanezoo. Výsledkem provedené práce je aplikace pro vizualizaci dat, která uživatelům poskytuje data v tabulkách v pohodlné podobě a umožňuje také vytváření a úpravu grafů, které budou zobrazeny na speciálním dashboardu. Aplikace byla testována jak za pomoci automatických testů, tak reálnými uživateli. V budoucnu budou přidány nové možnosti na základě testovacího feedbacku a aplikace bude integrována do aplikace Sanezoo.

Abstract

Currently, each specialist is faced with a large amount of data, and his goal is to work with them as efficiently as possible, but there are not often platforms that are able to provide an effective display of information for each user and have the possibility of individual customization. This bachelor thesis deals with data storage and display using dashboards. The aim of the work was to design and create a dashboard for organizations focused on quality control, which will allow not only to effectively display the necessary data, but also provide the user with the opportunity to customize the displayed information depending on his requirements and tasks. Some parts of this solution were deployed by Sanezoo. The result of the work done is an application for data visualization, which provides the user with data in tables in a convenient form, and also allows you to create and edit charts that will be displayed on a special dashboard. The application has been tested both with the help of automatic tests and by real users. In the future, new options will be added according to the test feedback and the application will be integrated into the Sanezoo application.

Klíčová slova

dashboard, vizualizace dat, chytrá zařízení, automatické testování, cypress

Keywords

dashboard, data visualization, smart devices, automatic testing, cypress

Citace

KOZHEVNIKOV, Dmitrii. *Webová vizualizace dat ze zařízení určených pro kontrolu kvality*. Brno, 2023. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Petr John

Webová vizualizace dat ze zařízení určených pro kontrolu kvality

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Petra Johna. Uvedl jsem všechny literární prameny, publikace a další zdroje, ze kterých jsem čerpal.

.....
Dmitrii Kozhevnikov
19. července 2023

Poděkování

Chtěl bych poděkovat svému vedoucímu, který mi celou dobu pomáhal s realizací této práce. Chci také poděkovat všem akademickým pracovníkům VUT FIT a pracovníkům firmy Sanezoo, kteří mi poskytli potřebné znalosti k realizaci této práce. A samozřejmě své rodině, přítelkyni a kamarádům, kteří mě podporovali.

Obsah

1	Úvod	2
2	Chytrá zařízení	4
2.1	Ruční kontrola kvality	4
2.2	Automatizovaná kontrola kvality	4
2.3	Výhody automatizace	5
2.4	Analýza dat	6
3	Informační dashboardy	7
3.1	Typy dashboardů a jejich klasifikace	8
3.2	Požadavky na dashboardy	9
3.3	Způsoby vizualizace dat	13
4	Technologie vizualizace dat a testování	17
4.1	Vizualizace dat	19
4.2	Testování	20
5	Analýza	23
6	Návrh	25
6.1	Návrh komponent a dashboardu	25
6.2	Existující řešení	28
7	Implementace	30
7.1	Backend	30
7.2	Frontend	30
8	Testování	35
8.1	Automatické testování	35
8.2	Testování s uživateli	37
9	Závěr	38
	Literatura	39

Kapitola 1

Úvod

V současné době se každý specialista potýká s velkým množstvím dat, a jeho cílem je pracovat s nimi co nejefektivněji. K tomu se používají různé nástroje, které potřebují třídit informace a efektivně je zobrazovat podle požadovaného kontextu. Nejčastěji se k tomu používají speciální vizualizační nástroje, které se nazývají dashboardy. Dashboardy jsou navrženy tak, aby zobrazovaly data v grafické podobě. Popularita dashboardů neustále roste, protože poskytují firmám velké příležitosti na všech úrovních. Roste také jejich nabídka na trhu.

Bohužel však grafický design a nástroje používané v takových panelech nemusí vždy splňovat požadavky uživatele. Většina dashboardů se zaměřuje na vnější vzhled a zapomíná na svůj hlavní cíl - efektivní zobrazení informací, které umožní uživateli rychle a efektivně pracovat s nimi. Tyto panely by měly být dobře promyšleny a navrženy s ohledem na principy komunikace člověka s grafickým rozhraním. Dashboard se také snaží poskytnout tyto informace způsobem, který uživatel pochopí co nejrychleji, což je poskytnout uživateli pouze informace, které potřebuje k provedení práce.

V dnešní době nejsou časté platformy, které jsou schopny poskytnout efektivní zobrazení informace pro každého uživatele a poskytují možnost individuálního přizpůsobení. Tento úkol není tak jednoduchý, protože každá výroba vyžaduje jedinečný přístup k zobrazování charakteristických dat, a proto individuálně navržený dashboard. Hlavním úkolem je nejen zobrazování dat, ale také umožnění změny tohoto dashboardu tak, aby s ním mohli pracovat různí uživatelé, kteří s ním nejsou seznámeni, a také zobrazit data, která budou srozumitelná všem. Příkladem takové oblasti mohou být procesy, ve kterých využívá chytrá zařízení, například zařízení pro hodnocení kvality.

Tyto zařízení umožňují firmám zjednodušit a zrychlit výrobní procesy. Umožňují také kontrolovat každý výrobek tak, aby splňoval všechny požadavky. Tato zařízení každý den zpracovávají velké množství výrobků a pokud něco není v pořádku, musí o tom informovat uživatele. Takové množství informací by mělo být podáno srozumitelným způsobem pro uživatele a s těmito informacemi mohou pracovat různí pracovníci: provozovatelé zařízení, inženýři pro kontrolu kvality, manažeři a ostatní. K tomu je zapotřebí dashboard, který lze použít v různých firmách a který bude flexibilní v přizpůsobení.

Řešení pro přizpůsobení se uživateli a různým uživatelským schopnostem se zabývají různé firmy. Jednou z takových firem je Sanexoo. Tato firma se zabývá vyhledáváním vad u výrobků pomocí strojového učení a potřebuje ve svých nástrojích využít takový dashboard, který umožní efektivní práci s ním různým specialistům. Tento projekt vychází z údajů poskytnutých touto firmou.

Cílem této bakalářské práce je navrhnout a vytvořit dashboard, který umožní nejen efektivně zobrazit potřebná data, ale také poskytne uživateli možnost přizpůsobit zobrazené informace v závislosti na jeho požadavcích a úkolech. Výsledkem bude knihovna, kterou mohou společnosti integrovat do aplikací zabývajících se zpracováním dat a kontrolou kvality. K tomu bylo rozhodnuto použít automatické testování ke kontrole kvality vyvinuté knihovny.

Kapitola 2 obsahuje informace o chytrých zařízeních a také informace o kontrole kvality a její automatizaci. Popis toho, co je dashboard, jeho typy a požadavky na jeho vytváření, popis klíčových ukazatelů výkonu a způsoby vizualizace dat jsou obsaženy v kapitole 3. Kapitola 4 obsahuje popis technologií a knihoven pro vizualizaci dat a také popis způsobů automatického testování softwaru. V kapitole 5 je obsažena analýza aktuálního stavu ve firmě Sanezoo. Návrh řešení je obsažen v kapitole 6. Použité technologie, implementace a vytváření pracovních komponentů jsou popsány v kapitole 7. Proces psaní testovacích scénářů a automatických testů a také hodnocení práce s implementovanou aplikací je popsán v kapitole 8. Kapitola 9 obsahuje závěry o provedené práci.

Kapitola 2

Chytrá zařízení

Stále více výrobců zavádí ve svých podnicích chytrá zařízení. Jedná se o elektronická zařízení, která jsou schopna se spojit, sdílet a interagovat s uživatelem a dalšími chytrými zařízeními. Ve výrobě tato zařízení umožňují automatizovat kontrolu kvality pro preventivní detekci vad a úsporu odpovídajících nákladů [10]. Tato chytrá zařízení jsou speciální roboti a systém pro jejich kontrolu. Tato řešení také umožňují zlepšit celkovou kvalitu výrobků a zvýšit produktivitu tím, že se snižuje potřeba časově náročných manuálních kontrol, což pozitivně ovlivňuje konkurenceschopnost.

Podle firmy Sanexoo kontrola kvality znamená ověření, zda výrobek nebo proces odpovídá kvantitativním a kvalitativním vlastnostem, na kterých závisí kvalita výrobků stanovená technickými požadavky. Kontrolu kvality lze rozdělit na ruční a automatizovanou. Při implementaci automatizovaných systémů musí výrobce určit, pro jaké konkrétní účely a cíle plánuje zařízení používat a jaká kvalita a přesnost koncových výrobků budou vyžadovány.

2.1 Ruční kontrola kvality

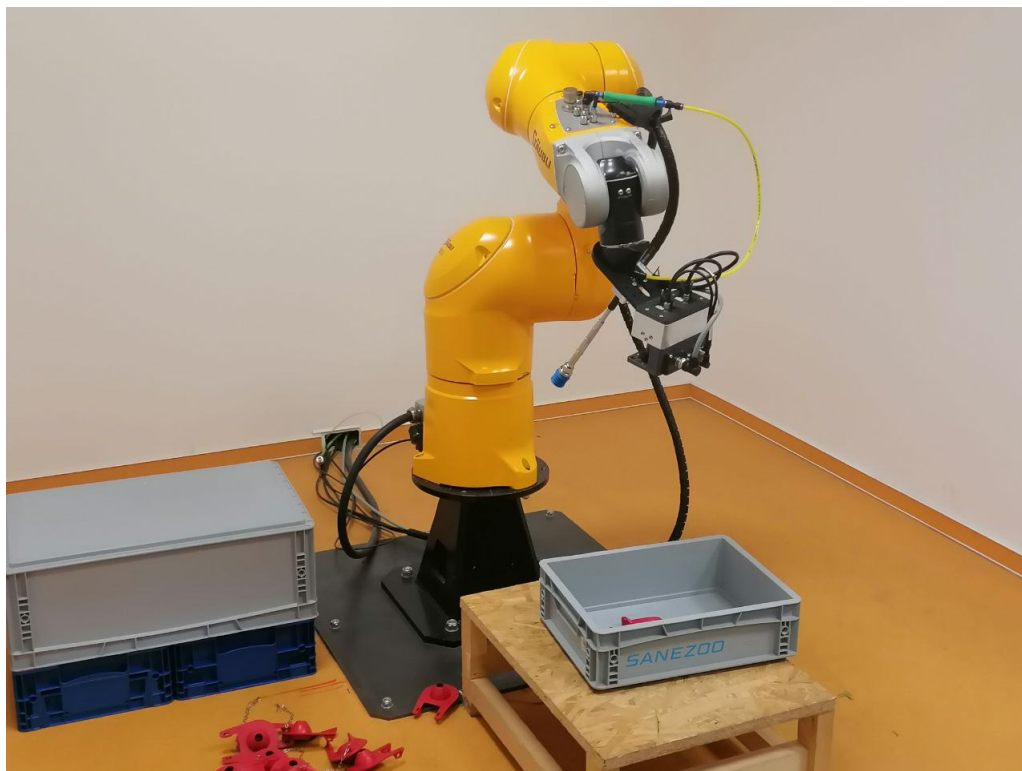
Při ruční kontrole technologové a operátoři měří, kontrolují a vyhodnocují výrobky ve všech fázích výroby nebo v určitých intervalech technologického cyklu [14]. Pokud zařízení a technologie pro kontrolu kvality chybí, znamená to, že proces je zcela manuální. V takovém případě se výrobce musí výhradně spoléhat na výsledky pozorování zaměstnanců.

Při kontrole tohoto typu hraje velkou roli lidský faktor a dovednosti odborníků, kteří provádějí samotná měření, což může negativně ovlivnit přesnost a spolehlivost měření. Ruční kontrola může také vážně zpomalit tempo výroby v závislosti na efektivitě procesu a dostupnosti laboratoře a měřicích strojů. Největší nevýhodou však představuje skutečnost, že pracovníci oddělení technické kontroly nemohou fyzicky zkontrolovat každý kus nebo výrobek v podmínkách velkoobjemové výroby.

2.2 Automatizovaná kontrola kvality

Kontrola kvality pomocí automatizace odstraňuje prakticky všechny nedostatky manuálního procesu a pomocí tradičních souřadnicových měřicích strojů. Při automatické kontrole jsou všechny práce prováděny speciálními roboty a lidský faktor je minimalizován. Často si podniky volí vlastní optický 3D skener, který je namontován na robotické rameno, nebo hotové řešení v podobě automatizovaného systému 3D skenování pro optimalizaci výrobních cyklů [3] a [12]. Tato řešení nevyžadují lidský zásah s výjimkou stahování a vykládání, což

umožňuje vysoký výkon, snižuje provozní dobu ve fázi kontroly kvality a zvyšuje kvalitu výrobků. Příklad robota pro kontrolu kvality je vidět na obrázku 2.1



Obrázek 2.1: Robot pro kontrolu kvality

Tyto systémy se používají k řešení mnoha problémů, včetně kontroly prvního výrobku na dopravníku a výrobní lince, testování dílů a nástrojů ve výrobním procesu, hodnocení dodaných dílů a překlad výsledků měření do digitálního formátu pro archivaci a sledování dat.

2.3 Výhody automatizace

Automatizované kontrolní systémy tak mají mnoho výhod, které pomáhají zvyšovat kvalitu dílů a konečných výrobků. Podle [2] mezi ně patří:

- Vzhledem k nedostatku zkušených provozovatelů měřicích strojů se výrobci stále častěji obracejí na automatizovaná řešení, která se snaží nejen eliminovat negativní faktory, ale také zlepšit kvalitu dílů, protože tyto systémy nejsou závislé na lidském faktoru. Kvalifikovaní zaměstnanci se navíc mohou zabývat důležitějšími úkoly, jako je analýza dat a proces kontroly.
- Automatizace zvyšuje spolehlivost kontroly tím, že zajišťuje přesné měření složitých dílů, včetně podmínek velkoobjemové výroby.
- Robotické systémy mohou sbírat více dat pro velké množství dílů s větší rychlostí. V kombinaci s pokročilými funkcemi pro sestavování zpráv to umožňuje výrobcům, aby se více rozhodovali na základě spolehlivých výsledků na více součástech.

Podniky mohou provádět automatizované kontroly kvality přímo na dopravníku pro optimalizaci výkonu. Lze měřit:

- více dílů se stejnou velikostí za hodinu,
- pouze nejdůležitější velikosti velkého množství dílů, zvýšení výkonu,
- stejný počet dílů, ale získat více informací o každém z nich pro další archivaci a sledování dat,
- více detailů a více parametrů v časovém intervalu.

To umožňuje výrobcům rychleji identifikovat vady a nesrovnalosti a přijmout nápravná opatření, aby se zabránilo prostojům a dodržení výrobního plánu.

Toto také umožňuje vyřešit problém nedostatku kvalifikovaných odborníků. Na vysoce konkurenčním trhu je často obtížné najít potřebné odborníky. Efektivním řešením jsou automatizované systémy. Zařízení nové generace založená na technologii 3D skenování, která je mimořádně srozumitelná a snadno použitelná, může řídit výrobní personál, který se nespécializuje na oblasti robotiky nebo metrologie. Samozřejmě, nelze provádět práce úplně bez zaměstnanců oddělení technické kontroly, kteří budou dělat plány na takové práce. Výrobní pracovníci však mohou používat řešení založená na 3D skeneru a pracovat na každodenním základě.

Je důležité, aby integrace softwaru do automatizovaného systému od technologů a inženýrů nevyžadovala žádné odborné znalosti v oblasti robotiky. Nejnovější generace softwaru pro sběr dat a metrologii automaticky počítá, modeluje a aplikuje cesty pohybu robotické ruky na základě integrovaného plánu [12]. Automatizované systémy kontroly kvality tak snižují náklady na vyhledávání, adaptaci a školení zaměstnanců.

2.4 Analýza dat

Jak jednotlivé přístroje určují, jestli se jedná o vadný výrobek nebo ne? Každá výroba se specializuje na svůj výrobek a všechny díly se mohou lišit podle typů mezi sebou. Přesně to dělají pracovníci, kteří provádí učení těch robotů. Tímto způsobem tyto značky zabrání chybám v budoucnosti a zjistí porušení. Když je robot naučený, může určit typy dílů, vady, jejich rozměry, tvary a další. Všechny tyto údaje spolu s čísly šarží výrobků, sériovými čísly a časem detekce vady jsou zaneseny do databáze a následně poskytnuty analytikům, kteří budou muset zjistit příčinu vady. A právě správné zobrazení těchto dat ovlivňuje, jak správně a rychle s tím lidé budou moci pracovat. Přestože je kontrola kvality automatizovaná, nelze vyloučit úplnou absenci lidí ve výrobě. S těmito systémy mohou pracovat také operátoři, kteří budou sledovat správnost výkonu práce a pomáhat při údržbě robotů. Operátoři budou také vidět tato data každý den během provozu a správné zobrazení dat jim umožní efektivně najít ztracené chyby.

Kapitola 3

Informační dashboardy

V dnešní době se pro efektivní zobrazení a analýzu dat používají různé tabulky a grafy [1]. Jednotlivě však nejsou tak efektivní, protože uživatel musí získat informace z různých zdrojů. Za tímto účelem jsou všechny tyto způsoby vizualizace dat kombinovány do monitorovacích panelů nebo takzvaných dashboardů. Lidé se každodenně setkávají s dashboardy, využívají se například v podnikání, v automobilech, letadlech, na burzách a podnicích. Vzhled a grafické provedení dashboardu se odvíjí od případu jeho použití [4]. Díky tomu mohou různé dashboardy obsahovat různé grafické doplňky: grafy, tabulky, histogramy nebo mapy, ale jsou navrženy tak, aby splňovaly jeden nejdůležitější úkol - poskytnout uživateli aktuální data ve vhodném formátu. Takže co je vlastně dashboard? Stephen Few, který se zabývá analýzou vývoje dashboardů, mu dal následující definici [7]:

„Dashboard je vizuální zobrazení nejdůležitějších informací potřebných k dosažení jednoho nebo více cílů; konsolidované a uspořádané na jedné obrazovce, takže informace lze sledovat na první pohled.“

Tohle je jen jedna z definic, existují i jiné, ale základní podstata zůstává nezměněna - dashboard je jedním z nejúčinnějších způsobů zobrazení informací.

Dashboardy umožňují zpracovat a poskytnout uživateli obrovské množství informací, ale tuto informaci je nutné představit v takové podobě, ve které uživatel může pracovat s ní co nejefektivněji. Z toho vyplývá, že dashboardy musí být správně vybrány, navrženy a vizualizovány. Při správném návrhu poskytuje dashboard uživatelům možnost na první pohled pochopit kontext dat a pomáhá jim rychle interpretovat. Podle Stephena Few by při navrhování dashboardu mělo být vše umístěno na jedné obrazovce a uspořádáno tak, aby uživatel nemusel přepínat mezi zdroji informací, protože člověk nemůže efektivně vnímat několika zdrojů informací a neztratit při tom kontext dat [8]. To vše připomíná panel auta, při pohledu na který může řidič rychle a snadno pochopit stav svého vozu.

Ale vzhledem k velkému množství a rozmanitosti dat je však obtížné splnit požadavky na jejich zpracování a zobrazení na monitorovacím panelu. Je třeba porozumět nejen prioritním údajům, ale také tomu, jak budou tato data využívána lidmi, kteří s nimi pracují. Tuto otázku zkoumá Colin Ware [17] a ve své knize se dotýká způsobů, jak člověk vnímá informace. Na základě jeho práce Stephen Few zdůraznil požadavky na dashboardy, které se zabývají nejen otázkou kontextu dat zobrazených uživatelům, ale také zvláštnostmi lidského vnímání grafiky:

- Jednoduchost a intuitivní srozumitelnost - klíčová data by měla být zobrazena tak, aby nezabírala mnoho místa na obrazovce. Na obrazovce je také důležitý výběr komponent, které budou zobrazovat data v nejpohodlnějším formátu. Vzhled komponenty také hraje roli ve vnímání informací člověkem, protože někteří uživatelé musí pracovat s tímto dashboardem každý den, takže by se měli cítit co nejpohodlněji.
- Agregace dat - uživatel potřebuje především data, která mu umožňují pochopit, co se v systému děje, a ne příčinu toho, co se stalo. Hlavním úkolem dashboardu je sledování stavu systému, nikoliv analýza. Analýza může být součástí dashboardu, ale není jeho klíčovým rysem.
- Adaptabilita - data musí být zobrazena ve formátu srozumitelném uživateli, k tomu musí být použita grafická zařízení, která jsou pro to nejvhodnější: grafy, histogramy, tabulky. Dashboard by měl být navíc navržen tak, aby zobrazená data byla srozumitelná každému uživateli, který s daným panelem neměl zkušenosti.

3.1 Typy dashboardů a jejich klasifikace

Dashboardy lze rozdělit podle různých kritérií, ale základní klasifikace - podle účelu panelů, jejich typu a interaktivity. Ve svých dílech Stephen Few [8] a Wayne Eckerson [5] identifikovali typy monitorovacích panelů, které můžete vidět dále.

Rozdělení podle typu

Hlavním kritériem každého dashboardu je ten, s jakým typem dashboard byl navržen a vytvořen.

Operační

Úkolem operačního dashboardu je řídit a kontrolovat základní procesy. Hlavní rozdíl spočívá v tom, že tento typ se používá k monitorování dynamických dat. Takový dashboard je nezbytný pro úkoly, které vyžadují okamžitou reakci uživatele na jakoukoli změnu dat. Tyto panely by měly velmi jednoduše zobrazovat kritické informace, protože uživatel musí pochopit kontext změn a reagovat na ně co nejrychleji. Data by také měla být aktualizována s vysokou frekvencí, aby poskytovala nejaktuálnější informace. Takové dashboardy mohou používat další grafické nástroje, které umožňují uživateli vidět kriticky důležité rozsahy hodnot.

Strategický

Tyto informační panely obvykle zobrazují informace, které mohou uživatelům pomoci posoudit dlouhodobý stav firmy. Zobrazená data umožňují uživatelům vytvářet předpovědi pro různé události, poskytují očekávané hodnoty a umožňují plánování. Tyto dashboardy používají různé metriky po dlouhou dobu, které umožňují předpovídat a identifikovat budoucí změny.

Analytický

Na základě názvu je hlavním cílem těchto panelů zobrazení dat vhodných pro analýzu. Tyto panely mohou poskytovat nástroje pro porovnávání dat podle různých kritérií. Takové

dashboards jsou flexibilnější při poskytování dat v určitém časovém období, na rozdíl od strategických. Právě agregace dat umožňuje zpracovat jejich obrovské množství a analytické nástroje je umožňují využít je k identifikaci vzniklých problémů. Tyto panely nepotřebují zobrazení dat v reálném čase, ale musí používat sofistikovanější nástroje, které umožní sledování potřebných změn.

Rozdělení podle účelu

Tato klasifikace umožňuje pochopit, proč byl tento dashboard vytvořen. V závislosti na jeho účelu bude mít svůj jedinečný vzhled, zobrazovat potřebná data a používat různá metrika a grafická zařízení. Tyto panely lze rozdělit pro práci s financemi, pro podnikání, pro výrobu nebo pro práci s lidmi.

Každý z těchto typů potřebuje zobrazit své unikátní data a jejich složitost. Například finanční dashboards používané na burzách by měly zobrazovat nejaktuálnější data, zatímco údaje o výrobě by to nemusely potřebovat. Tyto panely mohou být navrženy jak pro úzký okruh pracovníků, kteří s nimi budou pracovat, tak pro společnosti, ve kterých bude s tímto dashboardem pracovat velké množství pracovníků.

Rozdělení podle interaktivity

Jedná se o rozdělení panelů podle toho, jaké zobrazují data: statické nebo dynamické. Statické dashboards pracují s informacemi založenými na stavových snímcích v různých časových bodech v minulosti. Tímto způsobem mohou uživatelům poskytnout informace za určité období v minulosti nebo v rámci určité práce. Díky tomu jsou skvělé pro analýzu dat a umožňují vytvoření dashboardů pro analytiku. Dynamické dashboards naopak poskytují uživateli informace o systému v reálném čase. Tímto způsobem může uživatel okamžitě pochopit a reagovat na jakýkoli problém, který se v systému vyskytuje. Na tomto kritériu jsou postaveny klasifikace, které jsou uvedeny výše. Výběr zobrazených dat podle této klasifikace přímo závisí na úkolu, který je předán uživateli.

3.2 Požadavky na dashboards

Kvalita dashboardu závisí na zpracovaných datech, která jsou uživatelům poskytována, stejně jako na vizualizačních komponentách. Ale vzniká otázka: S čeho začít vývoj dashboardu? Před vytvořením dashboardu je třeba odpovědět na otázky:

- Jaké informace chceme zobrazit?
- Kdo je naším cílovým uživatelem a jaké jsou jeho úkoly?
- V jakém formátu by měly být informace zobrazeny?

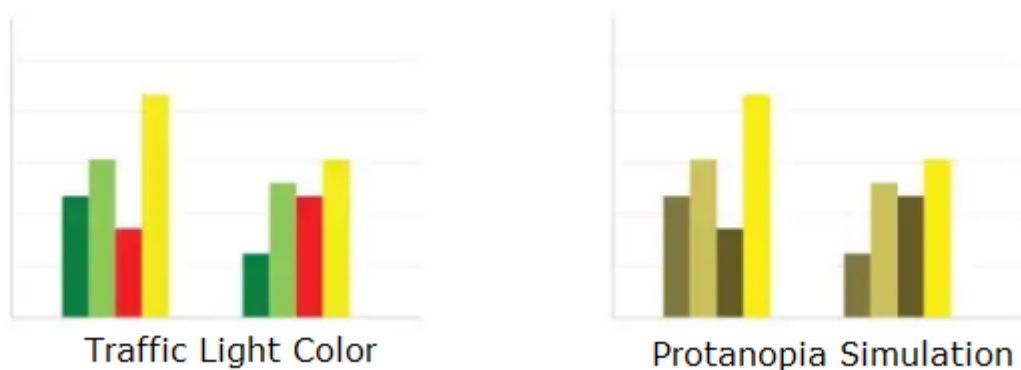
Důležitá je nejen vizualizace dat a jak tato data vypadají, ale také to, jak člověk tato data vnímá [6]. Vnímání dat lidským mozkem je zkoumáno různými autory [9] a [16].

Je důležité si uvědomit, že s dashboardem člověk komunikuje především prostřednictvím zraku. Jeho oko rozlišuje barvy, čáry, čísla a přenáší tyto informace do mozku [8]. Mozek zpracovává získané informace a přesně to, jak jsou data zobrazena, ovlivňuje na rychlost jejich vnímání. Nejprve, když se člověk podívá na dashboard, identifikuje nejdůležitější informace, pak si v krátkodobé paměti uloží základní údaje, které si může zapamatovat při pohledu na dashboard. A konečně v dlouhodobé paměti dochází k ukládání informací.

Stephen Few zdůrazňuje různá kritéria vnímání informací: barva, pozice, tvar, animace, seskupení. Právě podle těchto charakteristik lze rozlišit účinnost zobrazení prvků.

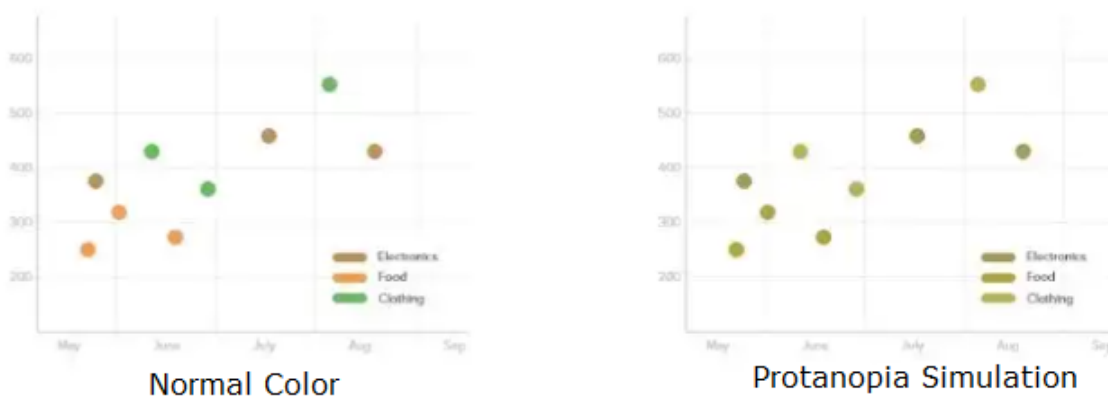
Barva

Prvkům, které se liší barvou od všech ostatních, je mnohem snazší věnovat pozornost. To může být použito k zobrazení kritických hodnot, například v červené barvě. Je také třeba pamatovat, že uživatel nemusí být schopen rozlišovat mezi barvami. Správným termínem pro tento problém je nedostatek barevného vidění způsobený nedostatkem jednoho ze tří kuželů v oku. Hlavním problémem, kterému tito lidé čelí, jsou červené a zelené barvy. Takže v podstatě kombinace semaforů, která je široce používána v informačních panelech, pro ně vypadá jako stejné barevné odstíny [11]. Příklad lze vidět na obrázku 3.1



Obrázek 3.1: vnímání barev semaforu u lidí se zrakovým postižením

Kromě toho osoba se silným kardiovaskulárním onemocněním nebude schopna rozlišit červenou, zelenou, oranžovou a hnědou barvu spíše zobrazeno na obrázku 3.2. Lze tedy s jistotou říci, že je třeba se vyhnout použití těchto 4 barev společně k vizualizaci různých významů.

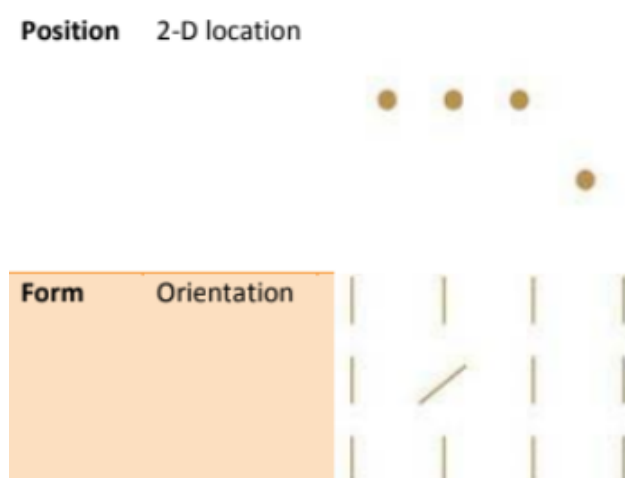


Obrázek 3.2: vnímání barev u lidí se kardiovaskulárním onemocněním

V tomto případě je třeba promyslet tento bod a použít kontrast k zobrazení různých dat, stejně jako způsoby, které zabrání vlivu takových parametrů [18]. Kromě toho uživatel může pracovat s dashboardem každý den po dobu několika hodin, takže není nutné používat jasné, agresivní barvy, kterými budou označena data. Navíc při použití těchto barev se zbývající data mohou jednoduše ztratit na pozadí jasných prvků, což vede k tomu, že uživatel může přeskočit něco důležitého.

Pozice

Umístění prvků je důležitou součástí při navrhování dashboardu. Pro uživatele je mnohem jednodušší věnovat pozornost samostatně stojící položce. Vliv pozice prvku je uveden na obrázku 3.3 To je třeba vzít v úvahu, protože data poskytovaná touto položkou mohou být kombinována s jinými daty na panelu.



Obrázek 3.3: vnímání pozice

Animace

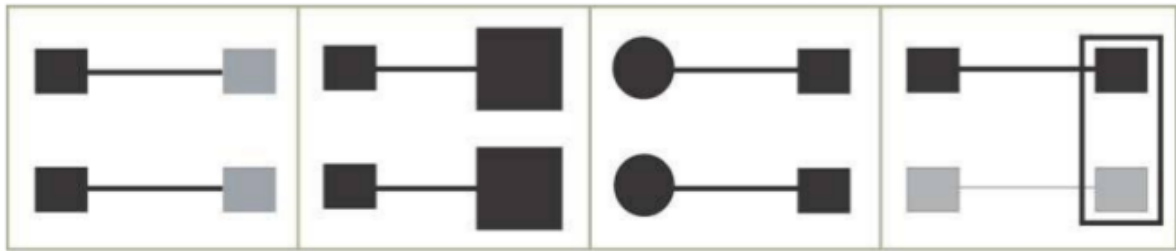
Jedním ze způsobů, jak přilákat pozornost uživatele, je použít animaci. Pohyblivý obraz přitahuje pozornost uživatele mnohem silněji než statický obraz. Tyto prvky by měly být použity velmi pečlivě a pouze ve situacích, kdy je to opravdu nutné, aby uživatel neztratil kontext zobrazovaných informací.

Seskupení

Mnoho prvků je mnohem jednodušší vnímat, když jsou v nějaké skupině. Tímto způsobem můžeme rozdělit data do skupin, čímž uživateli uvědomíme, že jsou propojeny a měly by být považovány společně.

Skupiny se mohou lišit tvarem, barvou, velikostí nebo izolací. Všechny tyto metody seskupení jsou uvedeny na obrázku 3.4 a nejúčinnější je izolace skupiny.

Tyto body jsou velmi důležité pro pochopení, v jaké formě musí být informace poskytnuty, aby člověk mohl co nejrychleji a správně porozumět, ale kromě toho je důležitá i struktura dashboardu.



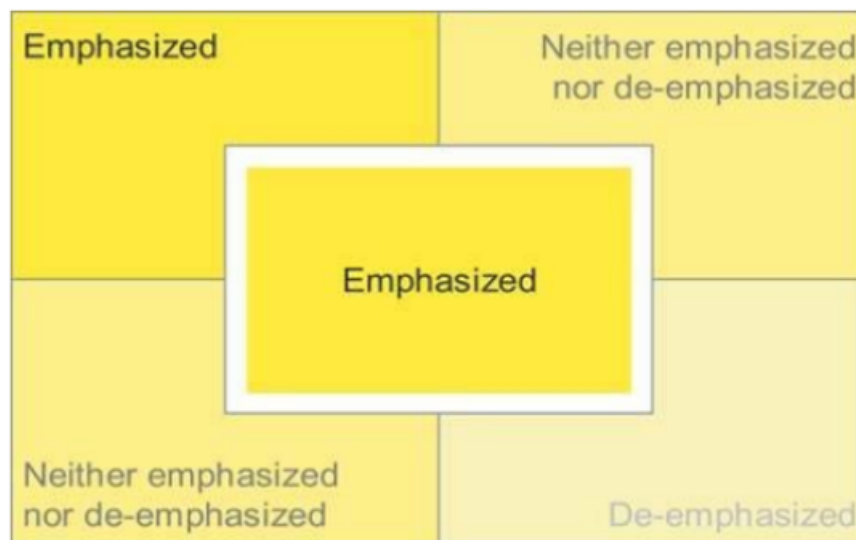
Obrázek 3.4: vliv seskupení na vnímání

Struktura

Při zodpovězení původně položených otázek je třeba pochopit, k čemu slouží každý prvek přidaný do dashboardu. Všechny informace by měly být umístěny na jedné obrazovce bez potřeby rolování. Lidé si mohou do své pracovní paměti uchovat pouze tři až čtyři prvky vizuálních informací, takže je třeba všechny důležité informace uložit na jednom místě v dashboardu. Pokud existuje potřeba vizualizovat další informace, stojí za to zvážit vytvoření samostatných dashboardů určených pro provedení úkolu uživatele.

Po napsání seznamu priorit a vytvoření modulů je možné začít přidávat prvky do dashboardu. Nejdůležitější prvek musí být umístěn v levém horním rohu a moduly musí být umístěny zleva doprava a shora dolů, jako je znázorněno na obrázku 3.5. Jedná se o standardní vzor pro lidi, kteří jsou zvyklí číst v západních jazycích: jejich oči obvykle nejprve upozorňují na levý horní roh [8]. Nejdůležitější informace lze rozdělit do dvou kategorií:

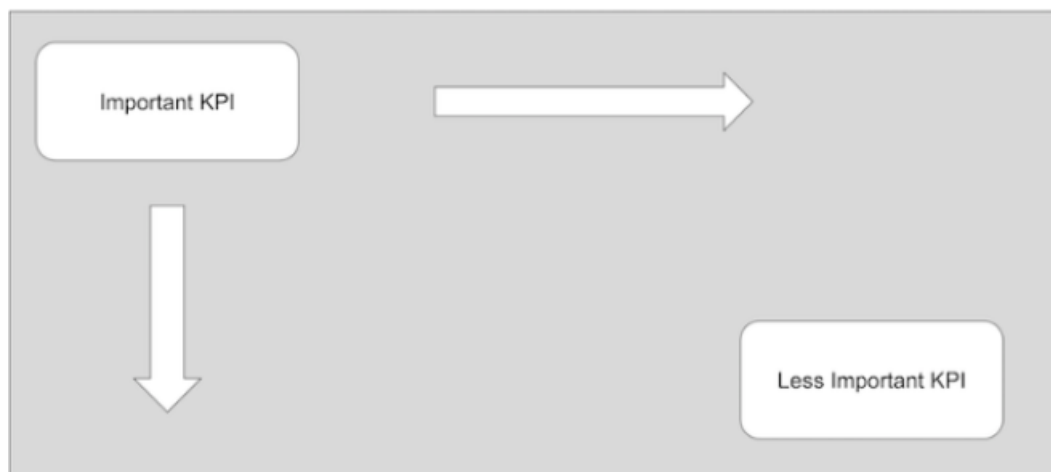
- Informace, které jsou vždy důležité
- Informace, které jsou důležité pouze v tuto chvíli



Obrázek 3.5: Vnímání prvků podle jejich polohy

Výběr dat

Kromě správného zobrazení dat je třeba pochopit, jak tyto údaje vybrat. V optimálním případě by měly být na dashboardu pouze klíčové ukazatele výkonnosti - KPI. Klíčové ukazatele výkonnosti jsou z angličtiny key performance indicators. KPI jsou nezbytné pro seznámení se situací, nikoliv pro studium dat. Klíčové ukazatele výkonnosti představují soubor opatření zaměřených na ty aspekty organizačního výkonu, které jsou nejvíce rozhodující pro současný a budoucí úspěch organizace [13]. Takový KPI by měl být správně umístěn při vizualizaci, příkladem může být obrázek 3.6.



Obrázek 3.6: Efektivní zobrazení KPI

Existuje však mnoho případů, kdy kromě standardních KPI jsou vyžadovány konkrétnější informace. V tomto případě je nutné zaměřit se na cíle uživatele. Možná stojí za to přidat osobní jednotlivé grafy nebo časové widgety pro sledování nových a zajímavých KPI, nebo jen data důležitá pro aktuální úkoly.

Pouze kombinací všech výše uvedených kritérií je možné vytvořit dashboard, který uživateli umožní trávit co nejméně času na zpracování informací, které mu jsou poskytovány.

3.3 Způsoby vizualizace dat

Po pochopení toho, jaká data by měla být uživateli zobrazena, stojí za to zvážit, v jakém formátu je zobrazit a jaké grafické zařízení k tomu použít. Existuje již mnoho způsobů a nástrojů pro zobrazení informací, které mohou usnadnit vytváření dashboardu. Všechny vybrané nástroje by měly být založeny na principech vizuálního vnímání a účinně plnit svou roli. Výběr grafů, tabulek, senzorů a dalších vizualizačních zařízení je poměrně složité řešení.

Stephen Few identifikoval několik základních pohledů pro efektivní zobrazení informací [8]. Mezi tyto způsoby patří tabulky, sloupcové grafy, spojnicové grafy, sparkline, skládané grafy, bodové grafy a bullet grafy. Existují další, ale popsané způsoby umožňují vizualizaci dat nejefektivnějším způsobem.

Je důležité si uvědomit, že každá možnost má své výhody a nevýhody, takže je třeba pečlivě zvážit, který z grafů je třeba použít a který nestojí za to. V této bakalářské práci se budu zabývat základními vizualizačními nástroji, které lze nalézt téměř v každém dashboardu a které mohou nejefektivněji přenášet informace.

Tabulky

Samozřejmě, že tabulky nejsou grafy, ale pokud uživatel potřebuje zobrazit jednotlivé hodnoty, není důvod používat grafy, protože to může znesnadnit vnímání uživatele. Například z tabulky zobrazené na obrázku 3.7 může uživatel získat informace v pohodlném stavu a okamžitě pochopit, jaká data zobrazuje tabulka.

Metric	Actual	Variance
Revenue	\$913,394	+\$136,806
Profit	\$193,865	-\$73,055
Avg Order Size	\$5,766	-\$297
On Time Delivery	104%	+4%
New Customers	247	-62
Cust Satisfaction	4.73 / 5	+0.23

Obrázek 3.7: Příklad tabulky

Tabulky jsou běžné a dobře srozumitelné vizualizace pro vyhledávání hodnot. Výše uvedená tabulka usnadňuje vyhledávání hodnot a souvisejících parametrů. Pokud tabulka obsahuje velké množství sloupců, je nutné vybrat základní údaje, které uživatel může potřebovat. Pokud uživatel potřebuje znát jednu hodnotu, například součet, průměr nebo okamžitou hodnotu, je ideální použít jednotlivá čísla nebo tabulky.

Lineární grafy a tepelné mapy

Lineární grafy jsou vhodné pro většinu zobrazených dat - jsou to body spojené přímými segmenty. Je také třeba poznamenat, že na grafu časové posloupnosti tato interpolace dat dává uživateli pocit spojení času. To lze vidět na obrázku 3.8.



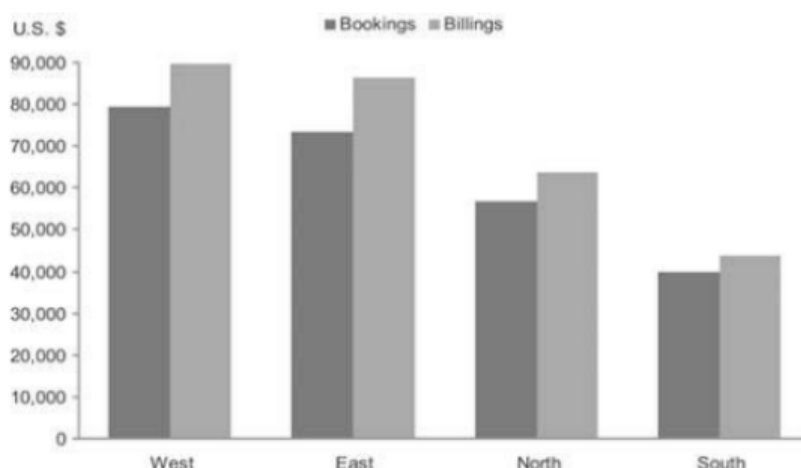
Obrázek 3.8: Příklad lineárního grafu

Lineární grafy jsou populární vizualizace, protože ztělesňují mnoho atributů na úrovni podprahové pozornosti v minimálním počtu pixelů. Linky mají sklon a úhel, polohu a barvu, což umožňuje bez váhání extrahovat spoustu informací. Je možné trochu komplikovat zobrazení a použít k zvětšení pozornosti tloušťku a jas čar. Lineární grafy interpolují data mezi dvěma body pomocí čáry. Ve většině případů je to žádoucí, protože tvar vytvářený liniemi pomáhá vnímání. Bohužel, interpolace může skrýt chybějící informace nebo dokonce ublížit: přinutit si myslet, že existují údaje, které ve skutečnosti nejsou.

Také při kreslení několika lineárních grafů může být skryt tvar grafu, mít barevné kolize, zvýšit zpoždění vnímání a učinit jej zbytečným. V tomto případě nám pomohou tepelné karty. Tepelné karty jsou praktické, když uživatel má lineární graf s příliš velkým počtem řádků. Tepelné karty používají změnu jasu barvy, takže do malého prostoru lze zabalit velmi velké množství dat. Tam, kde v lineárních grafech překrytí linek narušuje vnímání, tepelné karty v případě překrytí zvyšují barvu. Ale v tepelných kartách klesá přesnost, protože lidé nemohou snadno rozlišovat různé barevné jasy. Few ukazuje, že hranice rozlišitelnosti je 5 stupňů jasu [8]. To znamená, že tepelné karty jsou lépe aplikovány v situacích, kdy je přesnost méně důležitá než celková forma vizualizace.

Sloupcové grafy

Lineární grafy ukazují tvary, sloupcové grafy ukazují „přesné“ hodnoty. Tento typ grafu je vhodnější, když je třeba zobrazit hodnoty měření, které spadají do určité kategorie nebo porovnat několika metrik. Hodnoty v tomto grafu jsou reprezentovány sloupci umístěnými v souřadnicovém systému. V závislosti na umístění osy kategorií je graf vodorovný nebo svislý. Příklad sloupcového grafu je znázorněn na obrázku 3.9 a umožňuje uživateli přesně pochopit, jak probíhá číselné rozdělení podle regionů.

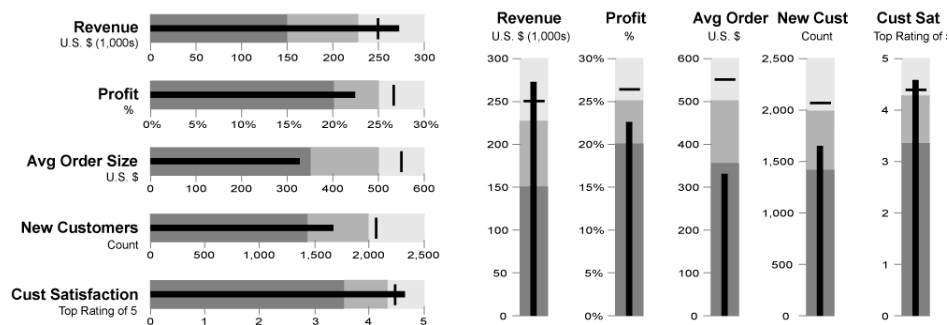


Obrázek 3.9: Příklad sloupcového grafu

Sloupcový graf pomáhá při porovnávání hodnot v každém intervalu díky jasnému spojení intervalů a snadnému porovnání délek.

Bullet grafy, značky, intervaly

Tyto typy vizualizace mohou být také použity, ale stojí za to je používat, když je to opravdu nutné. Například zobrazení počtu dotazů za sekundu jako měřítko je matoucí, protože tento ukazatel nemá žádné maximum. Pokud však je nutné zobrazit limit, například procent nebo frontu s maximální hloubkou, může být použití podobných vizualizací neuvěřitelně užitečné. Stupnice umožňují hlásit hodnotu limitu a také zjistit, jak blízko je hodnota k tomuto limitu. Lze také použít bullet grafy, jejichž příklad je uveden na obrázku 3.10



Obrázek 3.10: Příklad bullet grafu

Ostatní grafy

Jiné grafy se používají méně často a pouze tehdy, když je třeba zobrazit určité údaje. Existují také typy grafů, které nemusí být použity k zobrazení dat. Například koláčové nebo kruhové grafy vypadají vizuálně přívětivě, ale obvykle lidé nemohou odhadnout oblast sektoru grafu a efektivnějším způsobem bude použití sloupcového grafu. Samostatně stojí za zmínku sparkliny [15]. Jsou skvělé pro zobrazení jednotkových hodnot, ale bez známých prvků lineárních grafů často postrádají ten graf. Ostatní typy grafů se používají již pro specifické datové typy a jsou vybrány individuálně pro každý dashboard v závislosti na jeho účelu.

Kapitola 4

Technologie vizualizace dat a testování

Po zvolení způsobu využití a zobrazení dat je nutné zvolit technologii pro implementaci. V dnešní době je praktické vytvářet aplikace poskytující dashboardy jako webové aplikace. Webová aplikace je výkonný a moderní způsob implementace, který umožňuje uživateli používat systém z jakéhokoli zařízení, které má připojení k internetu, aniž by bylo nutné něco instalovat. Při implementaci aplikace, bude klient komunikovat s webovým serverem pomocí prohlížeče. Pokud zvolíme tento přístup, může koncový uživatel zobrazit námi vytvořené pohledy ve všech moderních webových prohlížečích. Hlavním požadavkem těchto prohlížečů je jejich podpora specifikace HTML5 v kombinaci s jazykem JavaScript. Aplikace musí být také rychlá a citlivá. Důležitým rozhodnutím je i to, kterou technologii použijeme pro práci s grafikou.

Standard HTML5¹ poskytuje dvě možnosti pro práci s webovou grafikou: Canvas a SVG. Tyto dvě technologie se od sebe dost liší. Je důležité vědět o jejich výhodách a nevýhodách, aby si vybral nejvhodnější technologii pro řešení úkolu. SVG² umožňuje vytvářet vektorovou grafiku, zatímco Canvas je navržen tak, aby vytvářel rastrovou grafiku. Jedním z nejdůležitějších parametrů při výběru technologie je výkon. Tento parametr závisí na počtu objektů a ploše povrchu. Obecně platí, že s rostoucím počtem objektů na obrazovce, klesá výkon SVG, protože objekty se postupně přidávají do modelu DOM³, zatímco výkon Canvas se téměř nemění. SVG je vhodné pro vytváření statických obrázků, vysoce kvalitních komplexních vektorových dokumentů a interaktivních grafů. Použití Canvas umožňuje rychle vykreslování obrázků na obrazovce, což je důvod, proč je tato technologie vhodná pro zobrazení složitých scén a animací v reálném čase. Při použití této technologie je však poměrně obtížné vytvořit obrázek nebo graf interaktivní. Existují frameworky, které to umožňují, ale není tak jednoduché udělat obrázek interaktivní, reponzivní a jednoduše upravitelný.

Je také důležité, že vytvořené komponenty mohou zobrazovat všechny prvky uživatelského rozhraní, nejen vizualizaci dat. Poté nutné vytvořit komponenty, které budou opakovaně použity v procesu vývoje. Tyto komponenty mohou být vytvořeny ručně, ale by bylo nepraktické a časově náročné. Nejčastěji se proto používají populární knihovny pro vývoj, které umožňují urychlit a zjednodušit tento proces. Tyto knihovny poskytují vývojářům

¹HTML5 je v informatice verze značkovacího jazyka HTML sloužícího pro tvorbu webových stránek

²SVG (z anglického Scalable Vector Graphics škálovatelná vektorová grafika) je značkovací jazyk a formát souboru, který popisuje dvojrozměrnou vektorovou grafiku pomocí XML.

³DOM (akronym anglického Document Object Model – objektový model dokumentu) je objektově orientovaná reprezentace XML nebo HTML dokumentu.

možnost nejen vytvářet komponenty, ale také je kombinovat pomocí pokročilých funkcí, které urychlují interakci komponent a vývoj uživatelského rozhraní.

Právě tyto knihovny umožňují řešit problémy na základě myšlenky deklarativního programování, nikoliv imperativního přístupu. Deklarativní přístup spočívá v popisu konečného výsledku (co chceme získat). Při imperativním přístupu jsou popsány konkrétní kroky k dosažení konečného výsledku (jak chceme něco získat). V současné době jsou nejoblíbenější knihovny React, Angular a Vue.

React

React je JavaScriptová knihovna pro vytváření uživatelských rozhraní, která byla vyvinuta společností Facebook. React se liší od základní aplikace napsané v HTML a JavaScriptu tím, že používá specifickou syntaxi - JSX. Jedná se o rozšíření jazyka JavaScript, který se používá pro popis Reactu struktury komponenty. Kromě toho se celá aplikace skládá z funkcí, které se nazývají komponenty. Komponenta je fragment rozhraní, který obsahuje rozložení a v případě potřeby související logiku. Takže při vývoji jednotlivých komponent může vývojář sestavit svou aplikaci z předem vytvořených bloků. Navíc je React velmi minimalistický a poskytuje vývojáři pouze základní nástroje, nicméně tam je obrovské množství rozšíření a komponent, které umožňují přidávání nových nástrojů pro řešení úkolů.

Angular

Angular je framework vyvinutý společností Google. Cílem tohoto frameworku je vytvořit pokročilé jednostránkové webové aplikace v programovacích jazycích JavaScript, TypeScript a Dart. Angular je složitější aplikační platforma než React. Stejně jako React, používá komponenty, které jsou velkými složkami aplikace nezávislé na sobě. Ale samotné komponenty rozdělují vzhled, strukturu a funkčnost. Také existují moduly, které jsou zodpovědné za správu komponent.

Angular pomáhá vázat komponenty aplikace k sobě, přenášet data, animovat rozhraní atd. To představuje větší příležitost pro vývojáře než React. S ním vývojář může vytvořit nejen webovou aplikaci, ale také napsat kód, který lze přizpůsobit jinému prostředí: mobilní nebo desktopové aplikace. U jednoduchých projektů proto může být jeho funkčnost nadbytečná.

Vue

Vue je další framework pro vytváření uživatelských rozhraní. Vue je vytvořen pro postupné zavádění, což ji odlišuje od Reactu a Angularu. Jeho jádro primárně řeší úkoly na úrovni zobrazení, což usnadňuje integraci s dalšími knihovnami a existujícími projekty. Vue je zcela vhodný pro vytváření komplexních jednostránkových aplikací, pokud bude použit ve spojení s moderními nástroji a dalšími knihovnami. Vue také staví aplikaci z komponent, které mají nastavené parametry a chování připojené pomocí kompilátoru. Také je možné měnit funkcionalitu a použité zařízení pomocí dalších knihoven a nástrojů. Vue je reaktivní framework, sleduje změny v modelu a překresluje pohled podle potřeby. Na rozdíl od knihovny React je Vue přísnější a více deklarativně zaměřen. Přísnost se projevuje v syntaxi, stejně jako v tom, jak jsou komponenty samy postaveny. Na rozdíl od knihovny React je Vue méně

flexibilitu a to hlavně kvůli tomu, že využití určitých knihoven je předurčeno, ale vše je také skvělé pro vývoj webových aplikací.

4.1 Vizualizace dat

Přístup k vytváření komponent, které umožňují vývojářům vytvářet grafy a další vizualizační nástroje, prošel řadou vývojářů. Díky nim vznikly knihovny s běžnými typy vizualizací. Poskytovaná funkcionalita těchto knihoven poskytuje flexibilitu a urychluje proces vytvoření vizualizace a tím urychluje vývoj samotné aplikace. Při výběru knihovny je důležité věnovat pozornost kompatibilitě s vybranou vývojovou technologií pro vytvoření uživatelského rozhraní. Je také třeba zvážit, jak pracovat s grafikou, stejně jako počet vizualizačních prvků, které je třeba zobrazit. Při výběru knihoven jsem se řídil množstvím vizualizací, které poskytují, tím, jak se provádí práce s webovou grafikou a kompatibilitou s knihovnou React.

D3.js

D3.js je sada nástrojů pro vizualizaci dat. Knihovna je složená z několika desítek malých modulů, z nichž každý řeší svůj vlastní úkol. Kromě modulů pro vytváření různých tvarů jsou uvnitř D3.js moduly pro práci s prvky na stránce, načítání dat, formátování a škálování dat, matematické funkce a další. Ale D3.js je složitá kvůli tomu, že je nízkoúrovňová. Více připomíná knihovnu na práci s daty, než na vizualizaci. Tato knihovna vyžaduje schopnost s ní pracovat. Takovým způsobem je D3.js komplexní nástroj s nadměrnou funkčností pro daný úkol. Navíc integrace D3.js do webové aplikace napsané v React způsobuje problém, že D3.js a React oba provádějí manipulaci se změnou DOM stromu, což může způsobit chyby v aplikaci. To znamená, že jejich sdílení není možné, ale třeba přidat ohraničení na manipulaci s DOM stromem pro D3.js, které omezí možnost implementace funkcionality.

Recharts

Recharts je knihovna, která je vytvořena pomocí D3.js. Recharts zjednodušuje vývoj, protože je jednotná, modulární a jednoduchá. Mřížka, řádkový ukazatel, prvky řetězcové prvky jsou všechny komponenty vhodné pro opakované použití. Díky tomu je mnohem jednodušší přizpůsobit grafy a dokonce použít vlastní přizpůsobené komponenty. Recharts nabízí SVG elementy, které jsou založeny na částech knihovny D3 a neumožňuje použití Canvas. Kromě toho tato knihovna neumožňuje vytvářet grafiku adaptivní pro mobilní zařízení.

Nivo

Nivo je také vytvořeno pomocí D3.js. Na rozdíl od Recharts má uživatelské rozhraní, s možností přizpůsobení, stejně jako jednoduchou interaktivní dokumentaci. Nivo umožňuje pracovat jak s Canvas, tak s SVG. Tato knihovna také umožňuje vytvářet grafy adaptivní pro různá zařízení. Přítomnost velkého množství grafů, stejně jako adaptivní vlastnosti této knihovny, je velmi flexibilní a pohodlné pro vývoj.

React-chartjs-2

Knihovna pro React, která je založena na Chart.js. Chart.js je jedna z nejpopulárnějších knihoven pro vizualizaci dat v Javascript. Na rozdíl od ostatních knihoven nemá vlastní dokumentaci, protože je vázanou knihovnou. Také tato knihovna nabízí pouze možnost práce s Canvas. Grafy mají flexibilitu a přizpůsobitelnost podle návrhu aplikace. Nicméně ne všechny typy grafů jsou adaptivní, což může komplikovat vývoj a řešení zadaného úkolu.

4.2 Testování

Také po vývoji je nutné zkontrolovat, zda produkt splňuje očekávání a požadavky, které jsou k němu předloženy. Určitě stojí za to provádět modulární testování ve vývojových fázích, ale v tomto případě jsem se zaměřil na end-to-end testování.

End-to-end nebo E2E je testovací proces, při kterém dochází k podrobné emulaci uživatelského prostředí. To znamená, že při tomto testování se simulují: kliknutí na tlačítka, vyplnění formulářů, přechody na stránky a odkazy. Cílem tohoto testování je zjistit, zda program pro konečného zákazníka funguje tak, jak původně počítal. E2E-proces je konečná fáze testování, po něm se žádné testování neprovádí.

Testování softwaru pomáhá nejen detekovat chyby, ale také ověřit splnění požadavků pro zákazníka. Také testování webových aplikací je jednou z nejdůležitějších součástí vývoje a vzhledem k vysokým požadavkům na kvalitu produktů se každým dnem zvyšuje potřeba stabilních nástrojů pro automatizaci testování. V dnešním světě se stále více využívá automatizované testování. Jedná se o metodu testování softwaru, která se provádí pomocí speciálních softwarových nástrojů, které jsou zase nezbytné pro provedení sady zkušebních příkladů. Automatizace umožňuje zvýšit efektivitu, stejně jako zvýšit dosah a rychlost testování. Na rozdíl od ručního testování, které provádí člověk, je kontrola všech scénářů mnohem rychlejší a data získaná při testování jsou stabilnější.

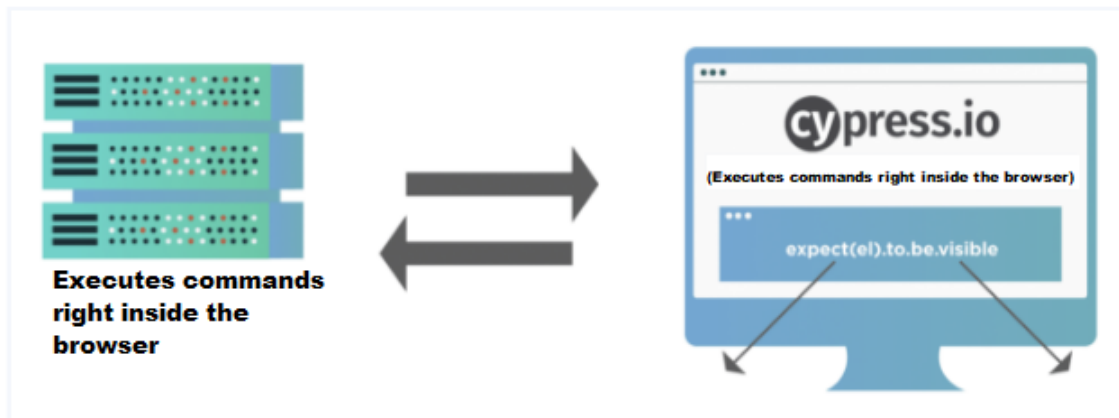
Samozřejmě, ne všechny případy stojí za to automatizovat. Základní věci, pro jejichž testování stojí za to používat automatizaci, jsou:

- Pravidelně se opakující testovací scénáře,
- Testovací skripty, které jsou složité a únavné provádět ručně,
- Testy, které vyžadují spoustu času.

Při automatizaci je nutné nejprve zvolit testovací nástroj, pomocí kterého se bude testování provádět. Poté je nutné určit rozsah automatizace, provést plánování testování, vývoj a návrh testů a poté provést testování a zpracovat získaná data. Výběr testovacího nástroje do značné míry závisí na technologii, na které je testovaná aplikace postavena.

Cypress

Jedná se o moderní testovací nástroj UI vytvořený pro moderní web. Řeší kritické problémy, se kterými vývojáři čelí při testování moderních aplikací, jako jsou problémy s synchronizací, nesoulad testů kvůli tomu, že položky nejsou viditelné nebo nedostupné.



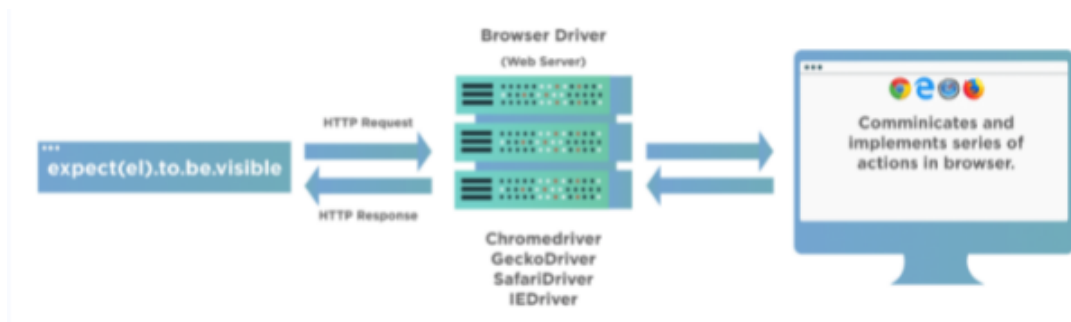
Obrázek 4.1: Architektura Cypress

Architektura Cypress vyobrazená na obrázku 4.1 poskytuje rychlé, konzistentní a spolehlivé provedení testů. Cypress automaticky čeká na spuštění příkazů před pokračováním, což zabraňuje problémům s asynchronností. Poskytuje také různé funkce, které umožňují pohodlnější testování. Cypress běží přímo v prohlížeči, který provede testovací kód. To umožňuje Cypress sledovat a měnit chování prohlížeče přímo během běhu testu, manipulovat s DOM stromem a měnit dotazy a odpovědi sítě za běhu. Z tohoto důvodu také vzniká nevýhoda, nemá podporu pro práci s externími funkcemi, například s vyskakovacími okny, kartami atd. Na druhou stranu to otevírá příležitost pro nový druh testování spolu s plnou kontrolou nad aplikací: frontendem a backendem. Cypress je poměrně nový nástroj, ve srovnání s ostatními testovacími prostředky se však aktivně rozvíjí a získává popularitu. Studoval jsem nejznámější automatizační technologie a vybral jsem si Cypress, také proto, že tuto technologii používá firma Saneezo.

Selenium

Selenium je projekt, v rámci kterého se vyvíjí řada softwarových produktů s otevřeným kódem (open source):

- Selenium WebDriver - je softwarová knihovna pro správu prohlížečů
- Selenium RC - je předchozí verze knihovny pro správu prohlížečů
- Selenium Server - je server, který umožňuje spravovat prohlížeč ze vzdáleného počítače, přes síť
- Selenium Grid - je cluster složený z několika Selenium serverů
- Selenium IDE - je plugin pro prohlížeč Firefox, který může zaznamenávat akce uživatele



Obrázek 4.2: Architektura Selenium

Kvůli tomu se architektura Selenium liší od Cypress. To je vidět na obrázku 4.2. Selenium se používá pro testování webových aplikací a správu webových stránek lokálně a online. Programy Selenium umožňují automatizovat aktivity prohlížeče. Tato technologie umožňuje psát skripty v téměř jakémkoli programovacím jazyce a je klíčovou součástí mnoha otevřených a proprietárních automatizačních nástrojů. Selenium umožňuje vzdáleně ovládat prohlížeč, takže uživatel může si vytvářet distribuované zdroje, které se skládají z mnoha strojů s různými operačními systémy a prohlížeči, a dokonce spustit prohlížeče v oblacích. Pro práci s Selenium je nutné mít pokročilé programovací dovednosti a jeho úzké zaměření umožňuje testovat pouze webové aplikace.

Kapitola 5

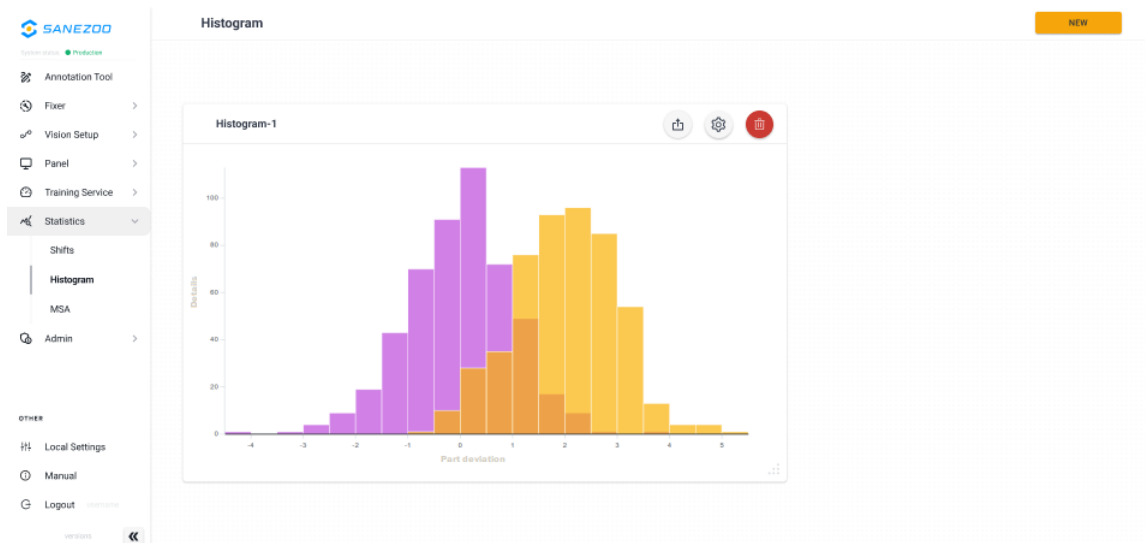
Analýza

Před výběrem technologie je nutné pochopit, kdo bude cílovým uživatelem tohoto softwaru. Je to klíčový krok, protože každý uživatel může mít na systém různé požadavky. To přímo ovlivní data a jejich způsoby zobrazení, které daný uživatel vyžaduje. Cílovým uživatelem může být například obchodní analytik, který je zodpovědný za zisk ve společnosti. V takovém případě budeme potřebovat analytický informační panel, který poskytne informace o počtu vyrobených produktů za určité období, jakož i počtu vadných dílů a jejich typů. Pokud uživatel bude provozovatelem výroby, bude pro něho nejprve důležité získat informace o aktuální sérii dílů, se kterou se práce provádí. Všechny tyto požadavky ovlivňují výběr technologií. K vyřešení tohoto problému jsem se zaměřil na dashboardy, které se používají při výrobě v průmyslu. Tuto otázku již zpracovala společnost Sanezoo.

Firma Sanezoo se zaměřuje na návrh a vývoj software v oblasti řízení robotů pomocí umělé inteligence a kontroly kvality pomocí strojového vidění. Firma již vyvinula projekt, který poskytuje možnost pracovat s robotem ve výrobě, specifikovat typy defektů a provádět nastavení konfigurace robotů. Cílem tohoto zmíněného projektu je poskytnout uživateli možnost integrovat roboty do konkrétní výroby, označit vady pro nastavení robota, získat a zpracovat data o výrobcích ve výrobě a vytvořit a obdržet zprávu o provedené práci. Cílem poskytování údajů o výrobě je umožnit uživatelům Sanezoo získat všechny informace, které se vyskytují ve výrobě, a také umožnit práci s těmito informacemi všem zaměstnancům firmy, kteří budou se systémem pracovat. Tímto způsobem budou pracovníci na kontrolu kvality a analytici potřebovat určité vizualizační nástroje, které lze přizpůsobit každému uživateli a poskytnout jim obecnou představu o tom, co se děje ve výrobě. Tyto vizualizační nástroje by měly být jasné a snadno použitelné, protože s nimi mohou pracovat různé typy zaměstnanců. Proto je nutné vytvořit několik komponent, které umožní zobrazit vybrané údaje o výrobě z různých hledisek.

Analýza poskytnutých dat také ukázala, že ne všechna data lze zobrazit uživatelsky přívětivou formou. Například data z platformy pro kontrolu kvality obsahují většinou proměnné logického typu s průběžnými výsledky. Nelze tedy přiřadit průběžný výsledek jako jednu z os grafu pro uživatele, protože uživatel by získal neúplné a neinformativní vizualizace. Za tímto účelem byla poskytnutá data analyzována, a pro každý typ grafů byly vybrány možnosti výběru dat pro vizualizaci, aby nezkušený uživatel nemusel rozumět velkému množství dat, která systém poskytuje.

V tuto chvíli společnost Sanezoo tuto část aplikace pro statistiky nevyvinula a má pouze předběžné rozložení toho, má pouze předběžný návrh vzhledu aplikace a poskytované funkcionality.



Obrázek 5.1: Návrh firmy Sanezoo

Obrázek 5.1 ukazuje předběžný pohled na to, jak bude tato součást vypadat podle firmy. Jedná se o histogram, který může operátor vytvořit. Osa X ukazuje odchylku od normy pro detaily a osa Y ukazuje počet dílů. Na obrázku je také vidět, že je zobrazen pouze graf a chybí data, podle kterých byl postaven. Neexistuje také žádné vysvětlení, co znamenají barvy na tomto grafu. Vpravo od grafu je volné bílé místo. Toto místo by bylo možné použít ke zvětšení velikosti grafiky nebo pro přidání legendy, která uživateli poskytne informace o tom, co je zobrazené na grafu. Kvůli těmto nedostatkům může mít i uživatel, který tento graf vytvořil, potíže s pochopením zobrazených dat, protože graf neobsahuje jakékoli vysvětlení. Kromě toho návrh nezobrazuje způsob výběru dat z tabulky.

Tato část aplikace je však navržena přímo pro provoz operátora a nevyřeší hlavní úkol - poskytnout příležitost pro práci různým uživatelům. Tato komponenta byla původně vyvinuta pro provoz provozovatele výroby. Tímto způsobem umožňuje vytvořit pouze jeden typ grafů a pracovat pouze s konkrétními daty. Kromě toho zcela chybí dashboard, se kterým by bylo možné pracovat a provádět analýzu.

Kvůli těmto nedostatkům byl jeden z hlavních požadavků vyvinout nový design této komponenty. Při navrhování šablony může být návrh volně změněn, přičemž hlavním cílem je poskytnout funkčnost pro řešení daného úkolu: umožnit různým pracovníkům pracovat s danou aplikací a také umožnit vytvářet různé typy grafů, které tito pracovníci budou moci používat. Je také nutné vyvinout dashboard, který lze nakonfigurovat pro každého uživatele a zobrazit potřebná data.

Kapitola 6

Návrh

Po provedení analýzy současné situace a všech požadavků, které je třeba splnit, bylo nutné navrhnout možné efektivní řešení. Nové řešení využívá existující komponenty tabulek, ale z důvodu, že se nepředpokládala podpora dashboardu a různých typů grafů bylo nutné vytvořit nové komponenty. Tyto komponenty budou splňovat nezbytné požadavky firmy a budou integrovány do existující aplikace. Za tímto účelem bylo nutné vyvinout návrh těchto komponent, protože v původním designu neexistovaly. Kromě komponent grafů byla vyvinuta také nová komponenta dashboardu, která umožňovala uživatelům systému konfigurovat zobrazení dat a grafů a také rozhodnout, jak dojde k ukládání a zpracování dat a interakci mezi komponentami.

6.1 Návrh komponent a dashboardu

Pro návrh komponenty byl použit grafický editor Figma. Pomocí tohoto editoru bylo vytvořeno rozhraní nových komponent, včetně dashboardu. První komponentou, která byla vyvinuta, jsou tabulky, které by uživateli poskytly potřebná data. Tyto tabulky obsahují data, která jsou přijímána ve výrobě z robotů. Bylo nutné zajistit jasné rozhraní a zobrazení těchto dat. V návrhu firmy Sanezoo již existoval návrh tabulek, který byl použit v jiných částech aplikace. Návrh tabulek nebyl v této části předložen z důvodu, že tato funkcionality byla pouze naplánovaná pro implementaci a nebyl vytvořen pro ni design. Proto jsem vytvořil nový návrh tabulky, který by zobrazoval data logického typu v pohodlné formě. Příklad tabulky je vidět na obrázku 6.1. Tato tabulka představuje vizualizovaná data o výrobcích získaných od firmy. Různé proměnné logického typu se zobrazují pomocí textu a speciální barvy, což uživateli umožňuje rychleji porozumět kontextu dat. Údaje o výrobku obsahují jeho ID, dobu skenování, vady a výsledek zpracování.

Sample	Serial Number	Date	ID	Readability	Repeatability	ID	Result	Defects
1	123242455555	23.10.22	2424	PASS	TRUE	1	NOK	overroll
3	365686385638	28.12.22	7464	FAIL	TRUE	2	NOK	overroll
2	365686385638	23.12.22	2424	FAIL	TRUE	3	OK	overroll
2	244222224456	24.12.22	2467	PASS	TRUE	4	OK	overroll
2	984824444442	28.12.22	8956	PASS	FALSE	5	OK	overroll
3	234468446755	26.12.22	6858	PASS	FALSE	6	OK	overroll
1	257884464646	28.12.22	6858	PASS	TRUE	7	OK	overroll
2	256895644667	21.12.22	6858	PASS	TRUE	8	NOK	overroll
1	754880563463	28.12.22	8044	PASS	TRUE	9	OK	overroll
2	465876636363	28.12.22	5785	PASS	TRUE	10	NOK	overroll

Obrázek 6.1: Navrhovaná tabulka

Také bylo nutné přidat návrh grafů, které by byly vytvořeny na základě dat z těchto tabulek. Bylo také nutné vyřešit problém, jakým způsobem by se při vytváření určovala pozice grafu na dashboardu. Při vývoji dashboardu bylo důležité udělat možnost nastavení zobrazených grafů a rozhodnout, jakým způsobem bude každý uživatel konfigurovat dashboard pro své použití. Existují různé způsoby, jak by to bylo možné implementovat:

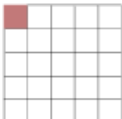
1. Je možné nastavit jasné pozice a velikosti grafů, ale to by omezilo možnost nastavení pro uživatele a nesplnilo by původně stanovený cíl, aby si každý uživatel mohl nastavit rozměry a vzhled toho, co potřebuje.
2. Další způsob implementace byl ve vývoji komponenty typu Canvas, ve kterém mohl uživatel vytvářet okna, která by zobrazovala grafy. To by vyřešilo hlavní problém nastavení, protože každý uživatel si mohl vytvořit potřebný počet grafů a umístit tak, jak potřebuje.
3. Rozdělení dashboardu na takzvanou síť. Tato síť rozděluje Dashboard na mřížku čtverců. Tyto čtverce umožňují přizpůsobit velikost grafů. Každý uživatel tak může nastavit rozměry pro zobrazené grafy.

Obrázek 6.2 představuje design komponenty, která uživateli umožňuje vybrat si typ grafu, data a jeho velikost:

Select type of graph

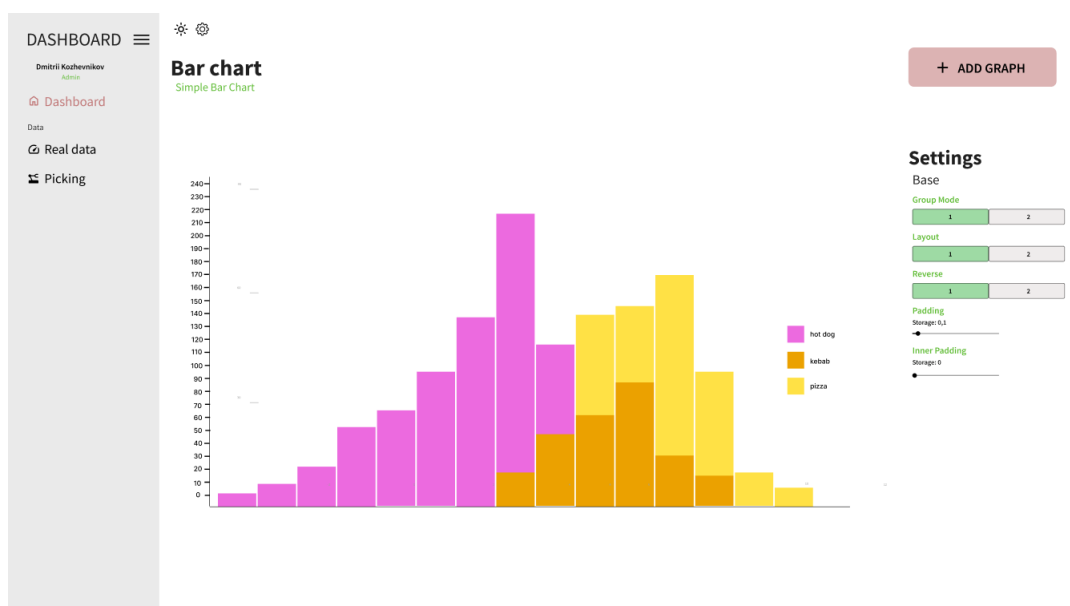
Line Chart
 Bar Chart
 Pie Chart

Width in columns: One ▾
Height in rows: One ▾



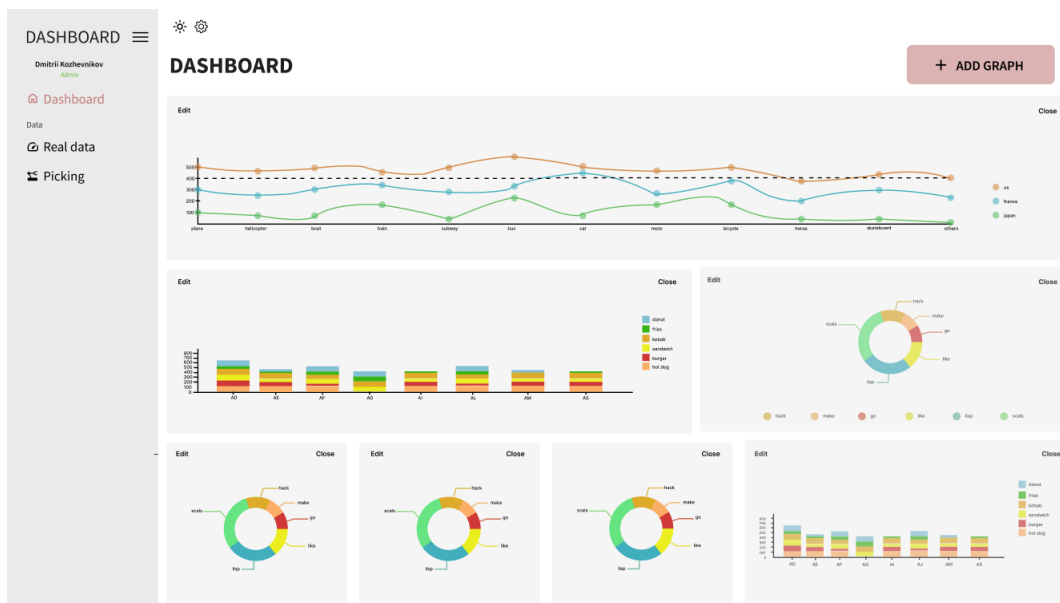
Obrázek 6.2: Nastavení grafu pro dashboard

Dále bylo nutné přidat návrh grafů, které by byly vytvořeny z dat tabulek. Společnost Sanezoo původně vytvořila návrh pouze pro jeden typ grafu pro provoz operátora. Bylo tedy nutné přidat nové typy grafů, které by umožnily vizualizaci dat různými způsoby, a také přidat uživatelům možnost přizpůsobit editace a nastavení grafů 6.3. Tyto grafy představují jednotlivé komponenty.



Obrázek 6.3: Design editace grafu

Po vytvoření grafů a řešení jejich umístění na dashboardu byl vytvořen design samotného dashboardu. Vzhled dashboardu se může změnit v závislosti na nastavení a počtu grafů, které zobrazuje.



Obrázek 6.4: Design Dashboard

Na obrázku 6.4 můžete vidět navržený design tohoto dashboardu, když na něj přidáte jiný typ grafů. Tento design byl navržen s ohledem na požadavky společnosti a také zohledňoval základní pravidla pro vývoj dashboardu.

Na prezentovaném dashboardu 6.4 zabírá hlavní graf většinu obrazovky. Typ hlavního grafu určí každý uživatel podle úkolu, který chce splnit. Pod tento graf bude možné přidat další typy grafů s daty, které bude potřebovat uživatel. Například tam mohou být zobrazeny obecné údaje o konkrétní šarži výrobků za určité období.

Uživatel může také v případě potřeby zvážit každý graf individuálně a nakonfigurovat a zobrazit v samostatném okně, které uživateli poskytne více informací.

6.2 Existující řešení

Po vývoji návrhu bylo nutné vyřešit jeden z hlavních úkolů, jak budou data zobrazena. Otázka ukládání a zpracování dat už byla vyřešena společností Sanezoo. Hlavním úkolem tedy bylo vytvoření komponent a vizualizace dat na frontendu.

Aktuální návrh umožňoval práci pouze s jedním typem grafu - histogramem, což omezovalo možnost práce s daty. Navržená aplikace umožňovala získat data v formátu vhodném pouze pro operátora, ale neumožňovala pracovat s komponentou různými pracovníky. Také každý vytvořený graf by měl možnost poskytnout uživateli přizpůsobit zobrazená data podle jeho požadavků. Tuto možnost zahrnuje i návrh od firmy Sanezoo, který můžeme vidět na obrázku 6.5. Ale ten návrh je jen pro histogram a umožňuje pouze základní změny, jako rozměr. Zcela chybí možnost nastavení os a barev pro zobrazovaná data, proto je třeba rozšířit možnost nastavení.



Obrázek 6.5: Histogram podle návrhu firmy Saneezo

Tento graf je také zobrazen jako samostatná součást, ze které nelze pochopit, jak probíhá výběr dat, která budou vizualizována. Proto je nutné provést vylepšení komponent, které uživatelům poskytnou možnost přizpůsobit si vizualizaci dle svých potřeb. Je také nutné vyřešit řadu problémů souvisejících se zobrazováním dat a nastavením grafů, aby se zlepšila kvalita aplikace pro různé pracovníky, kteří s ní budou v budoucnu pracovat.

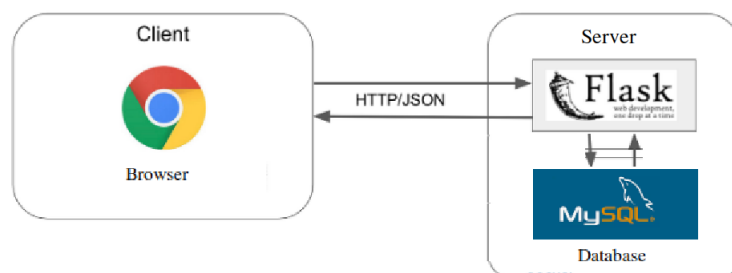
Za tímto účelem je potřeba například přidat možnost výběru dat z tabulek pro každou osu grafu, takže každý uživatel si bude moci vybrat data, která potřebuje. To lze provést pomocí nastavení podle sloupců. Je také nutné přidat možnost získat data za určité období, například za měsíc nebo za rok. To umožní vytvářet grafy, které mohou analytici použít, například k získání údajů o tom, kolik vadných dílů bylo vyrobeno v konkrétním měsíci.

Dále je třeba rozšířit možnosti nastavení grafů: přidat možnost nastavení barvy, velikosti a možnost přidání poznámek o tom, jaká data uživatel vidí. To umožní uživatelům, kteří mohou mít problémy se zrakem konfigurovat grafy tak, aby s nimi byli spokojeni a také přidá možnost rychleji pochopit vyobrazené informace.

Kapitola 7

Implementace

Tato kapitola se zabývá implementací systému. Frontend je postaven na knihovně React v jazyce JavaScript a komunikuje asynchronně s backendem pomocí požadavků. Implementační jazyk pro backend je Python. Pro uložení dat slouží MySQL databáze. Schéma technologií v návaznosti na části systému je na obrázku 7.1. Pro vizualizaci dat a vytváření grafů se používá knihovna Nivo.



Obrázek 7.1: Architektura aplikace

7.1 Backend

Tato aplikace pracuje s daty poskytnutými firmou Sanezoo. Během vývoje bylo plánováno použít backend poskytnutý firmou, ale při vývoji se ukázalo, že hodně koncových bodů ještě nebylo implementováno pro tyto komponenty statistik a grafů. Ve firmě se navíc provádí změna verze backendu, takže v současné implementaci dashboardu bylo rozhodnuto použít lokální backend pro demonstraci. Tato aplikace simuluje práci se firemním backendem. Byly analyzovány použité technologie nové verze s cílem zajištění kompatibility a integrace v budoucnu. Pro realizaci se používá jazyk Python s využitím frameworku Flask. Flask je mikro webový framework napsaný v programovacím jazyce Python. Je klasifikován jako mikro webový framework, protože nevyžaduje konkrétní nástroje ani další vnitřní knihovny. Pro demonstrace je možné spustit backend z složky „server“ pomocí příkazu `flask run`.

7.2 Frontend

Aplikace pro frontend využívá knihovnu React. Knihovnu React jsem zvolil, protože je moderní technologií pro implementaci webových aplikací s obrovskou podporou komunity.

Také je používána ve společnosti Sanezoo. Design aplikace vytvořen s využitím Material-UI frameworku. Při realizaci této práce bylo vyvinuto několik komponent, které odpovídají návrhům v kapitole 6. Spuštění frontendu je možné z složky „client“ pomocí příkazu *npm run start*.

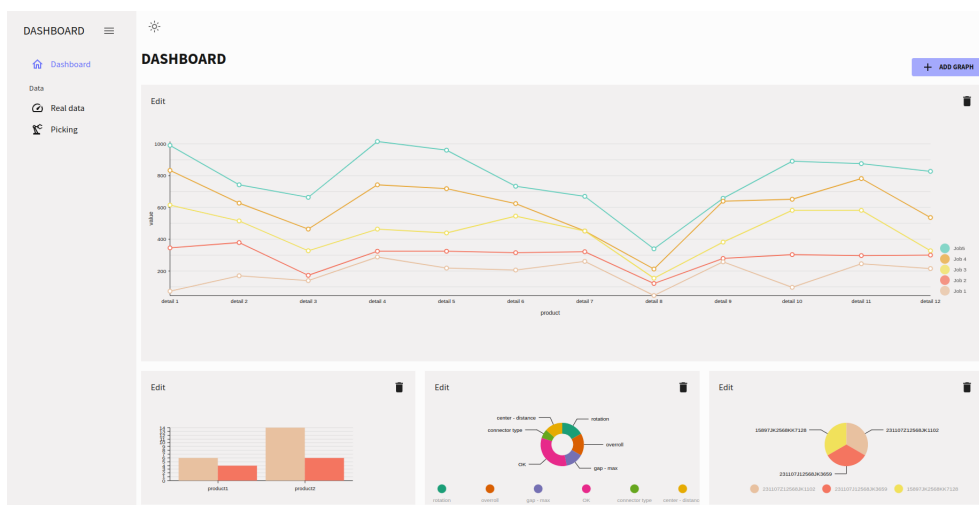
Byly vytvořeny 2 tabulky. Tabulka „Real Data“ zobrazuje data od robotů z výrobní linky, druhá tabulka „Picking“ zobrazuje data z linky kontroly kvality. Hlavním rozdílem oproti tabulkám firmy, které bylo nutné implementovat, bylo zobrazení dat získaných od robotů z testovací a kontrolní linky. Poskytovaná data byla logického typu, proto byla tabulkám přidána pohodlnější prezentace pomocí barevného zobrazení, aby uživatel mohl okamžitě pochopit, v jakém testu byla chyba. Tuto tabulku je možné vidět na obrázku 7.2.

ID	Serial Number	Ground Truth	Detail ID	Result 1	Result 2	Result 3	Result 4	Repeatability	Ground Truth Result
1	220737J098T54	OK	1835	PASS	FAIL	PASS	PASS	False	75
2	380737J098464	NOK	1835	PASS	FAIL	FAIL	FAIL	False	25
3	8689792502478Y1	NOK	1835	FAIL	FAIL	FAIL	FAIL	True	0
4	V68979250247811	NOK	1835	FAIL	PASS	FAIL	FAIL	True	25
5	868U79250247811	OK	1835	PASS	PASS	PASS	FAIL	True	75
6	M689792502478...	NOK	1835	PASS	FAIL	FAIL	PASS	False	50
7	U68979250247811	OK	1835	PASS	PASS	PASS	PASS	True	100
8	868J79250247811	NOK	1835	PASS	FAIL	FAIL	FAIL	False	25
9	L68979250247811	NOK	1835	FAIL	FAIL	PASS	FAIL	False	25
10	868B79250247811	OK	1835	PASS	FAIL	PASS	PASS	False	75

Obrázek 7.2: Picking table

Komponenta Dashboard

Komponenta Dashboard představuje uživateli hlavní stránku, na které budou zobrazeny vytvořené grafy. Tato komponenta využívá další komponenty grafu: *LineChart*, *BarChart*, *PieChart* a *BulletChart*. Příklad implementované komponenty je vidět na obrázku 7.3.



Obrázek 7.3: Implementovaný dashboard

Při vývoji dashboardu bylo důležité poskytnout možnost nastavení zobrazených grafů a rozhodnout, jakým způsobem bude každý uživatel konfigurovat dashboard pro své použití. Možné varianty implementace tady 6. Nakonec bylo zvoleno řešení 3. Nastavení jasné pozice a velikosti grafů by omezilo možnost a nesplnilo by původně stanovený cíl, že každý uživatel si sám může nastavit rozměry a vzhled toho, co potřebuje. Způsob implementace pomocí Canvas, ve kterém mohl uživatel vytvářet okna, která by zobrazovala grafy, by vyřešil hlavní problém nastavení, protože každý uživatel si mohl vytvořit potřebný počet grafů a umístit tak, jak potřebuje. Ale v tomto případě se ztratil základní koncept dashboardu¹ a před uživatelem by bylo běžné plátno, na kterém by uživatel mohl vytvořit nesrozumitelné množství grafů na dashboardu. To může vést k problémům s volným umístěním a ke kolizím ve vizualizaci. Aby bylo možné zkombinovat možnosti nastavení zobrazených grafů a zachovat koncept dashboardu, byla provedena implementace pomocí rozdělení dashboardu na síť. Tento způsob implementace omezuje uživatele ve vytváření velkého počtu grafů, aby nedošlo k přetížení dashboardu, a také poskytuje všechna data na jedné obrazovce. Počet čtverců v síti omezuje uživatele v počtu grafů, které může zobrazit, ale také toto omezení umožňuje omezit uživatele na nadbytek dat na obrazovce a ztrátu kontextu toho, co se děje. Při této realizaci vzniká nový problém konceptu dashboardu, že hlavní graf musí zabírat většinu centrálního prostoru, ale zároveň umožňuje plně přizpůsobit dashboard potřebám uživatele.

Select type of graph

Graph Type

- Bar Chart
- Bullet Chart
- Pie Chart
- Line Chart

Data

Real Data

- Result
- Products
- Batch Number

Picking

- Repeatability
- Results

Width in columns: One ▾

Height in rows: One ▾

CREATE

Obrázek 7.4: Modální okno pro nastavení grafu v dashboardu

¹Hlavní podstatou dashboardu je zobrazování dat pohodlně na jedné obrazovce.

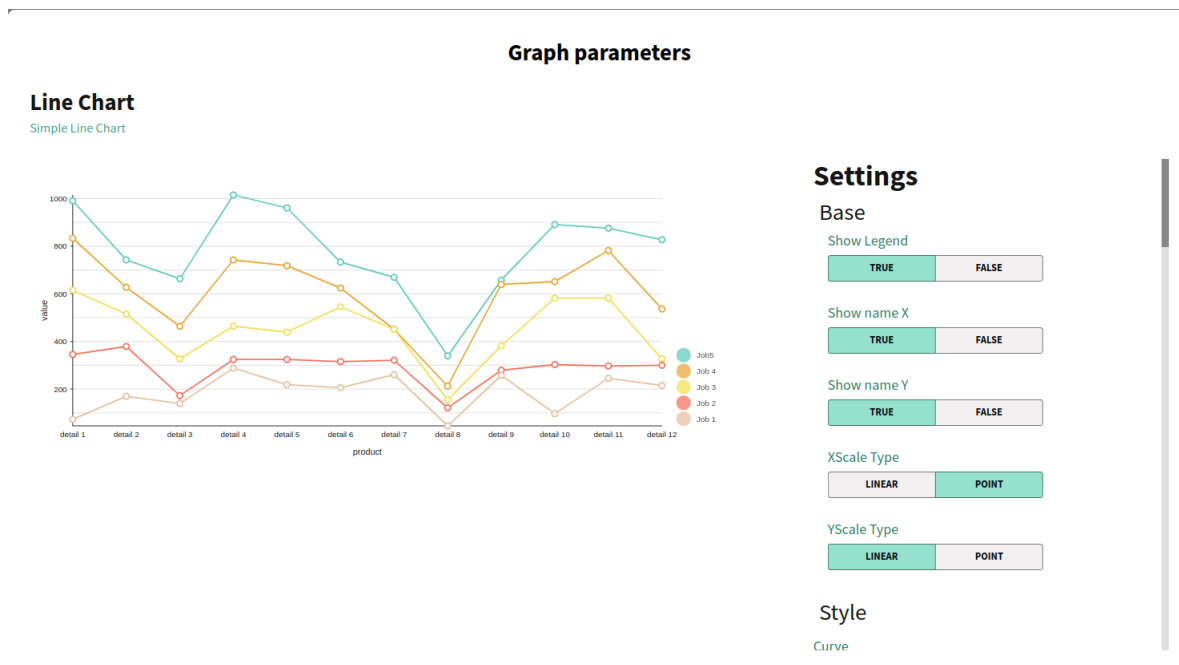
Nastavení dashboardu probíhá v hlavním okně v komponentě *Dashboard*. Na začátku uživatel vidí pouze tlačítko pro vytvoření prvního grafu. Po kliknutí se dostane do komponenty *Modal*, která zobrazuje informace pro nastavení v modálním okně 7.4. Tato komponenta přijímá vstupní data, která budou vizualizována uvnitř daného modálního okna. To umožňuje používat tuto komponentu v různých grafech a nastaveních. Modální okno umožní vybrat typ grafu, který chce uživatel zobrazit, a také nastavit jeho velikost, kterou bude daný graf zabírat. Po výběru typu grafu a úrovně velikosti musí uživatel vybrat, jaká data chce zobrazit pro příslušný typ grafu. Každý graf zobrazuje určité typy dat podle možností. Je to proto, aby uživatel získal smysluplná data v pohodlné podobě. V opačném případě by měl uživatel vědět, jaká data lze zkombinovat s ostatními.

Po výběru dat bude vytvořena komponenta *Graph*, která zobrazí graf na dashboardu podle zvolených dat. Každému grafu jsou předána počáteční nastavení pro zobrazení, například: barva, animace, interaktivita atd. Ale uživatel však může při použití tlačítka editace přejít k podrobnějšímu nastavení každého grafu.

Grafy

Analýza dat poskytovaných roboty ukázala, že ne všechny grafy jsou vhodné pro vizualizaci různých kombinací dat. V tomto ohledu byly vybrány, navrženy a implementovány 4 typy grafů v komponentách: *BarChart*, *LineChart*, *PieChart* a *BulletChart*. Každá z těchto komponent se skládá z odpovídající komponenty grafu: *Bar*, *Line*, *Pie*, *Bullet* a panelu podrobných nastavení pro každý graf. Tyto typy grafů umožňují nejrozšířenější vizualizaci dat pro různé typy uživatelů.

Graf je původně vytvořen na stránce Dashboard. Pro každý graf byly vybrány možnosti zobrazení dat tak, aby pro uživatele dávaly smysl. Toto umožňuje uživateli okamžitě získat data v pohodlné podobě pro konkrétní graf.



Obrázek 7.5: Modální okno pro nastavení parametru vizualizace grafu

Komponenty grafů používají vstupy, které jsou zodpovědné za zobrazená data. Každý vstup odpovídá konkrétnímu parametru grafu. Při vytváření grafu se vytvoří základní nastavení pro zobrazení. Toto umožňuje uživateli okamžitě zobrazit výsledek, bez nutnosti nastavovat velké množství parametrů a potřeboval by čas na pochopení. Nic méně, s každým grafem však může uživatel pracovat individuálně a přizpůsobit si ho podle sebe. Při přechodu na nastavení grafu se uživatel dostane do modálního okna, které umožňuje přizpůsobení daného grafu 7.5. Každý typ grafu má svá vlastní nastavení, která jsou obsažena v komponentách příslušných typů grafů: *Bar*, *Line*, *Pie* a *Bullet*. Tyto komponenty umožňují měnit parametry pro zobrazení dat podle potřeby uživatele. Uživatel tak může změnit rozměry, barvu značky, animace, polohu a další možnosti.

Každý graf má své vlastní parametry, které jsou rozděleny do konkrétních sekcí. Každá sekce poskytuje pokročilá nastavení určité možnosti. Pro nezkušeného uživatele se však počet parametrů u některých grafů může zdát nepochopitelný a nadbytečný. Za tímto účelem se na začátku zobrazují základní parametry, které uživatel nejčastěji používá, aniž by musel rozumět ostatním možnostem. Nastavení také používá malé komponenty, které umožňují uživateli provést nastavení: *SettingBlock*, *NonLinearSlider*, *ColorInput*, *Selector*. Každá komponenta umožňuje nastavování:

- *SettingBlock* - umožňuje nastavit parametry s výběrem možnosti
- *NonLinearSlider* - představuje sebou scroll pro nastavení velikosti
- *ColorInput* - umožňuje nastavit jakoukoli barvu pro parametr
- *Selector* - umožňuje vybrat z několika možností

Jakékoli změny, které uživatel provede v panelu nastavení, se okamžitě zobrazí na grafu, který se zobrazuje vlevo od daného panelu. Uživatel tak okamžitě vidí výsledek a může rychleji reagovat na změny.

Kapitola 8

Testování

Implementované části musí být vždy nějakým způsobem testovány. V tomto případě byl pro testování vybrán Cypress framework 4.2, který umožnil automatizaci testování frontendu a provádění E2E testování.

8.1 Automatické testování

Pro tuto práci byly napsány automatické testy, které umožňují ověřit správnost zobrazených dat při vytváření, zobrazování a změně. Tyto testy také simulují chování koncového uživatele, čímž dávají možnost rychle zkontrolovat, zda skutečný uživatel při používání aplikace narazí na nějaké problémy. Také umožňují vizualizovat každý krok. Tímto způsobem se kontroluje funkčnost tlačítek, přechody na odkazech, správnost zobrazení dat a možnosti přizpůsobení.

Každý test provádí kontrolu základních věcí pro vytvoření grafu, kontrolu dat nebo změnu nastavení. Existují však také testy, které umožňují projít kroky přizpůsobení všech dat a zkontrolovat výsledek, jako by je provedl skutečný uživatel. Testy lze spustit přímo prostřednictvím konzole pomocí příkazu `npm cypress run`. V takovém případě budou testy spuštěny bez vizualizace a po dokončení práce bude poskytnuta tabulka 8.1 všech testů s výsledky. V případě, že některý test skončí s chybou, bude v systému uloženo chybové logování do složky `/client/cypress/logs`, stejně jako obrázek do složky `/client/cypress/screenshots` a video pro test, který skončil neúspěchem do `/client/cypress/videos`. Díky tomu je možné nejen zjistit, který test skončil chybou, ale také vidět kroky testu a chybu, která se objevila v procesu testování. Při spuštění tímto způsobem budou spuštěny všechny soubory s testy.

```

Running: 02_dataTables.spec.cy.js (2 of 3)
Warning: The following browser launch options were provided but are not supported by electron
- args

Real data table tests
✓ Check sorting in Real data table (7309ms)
✓ Check pagination in Real data table (4090ms)

Picking table tests
✓ Check sorting in Picking table (7910ms)
✓ Check pagination in Picking table (4129ms)

4 passing (24s)

(Results)

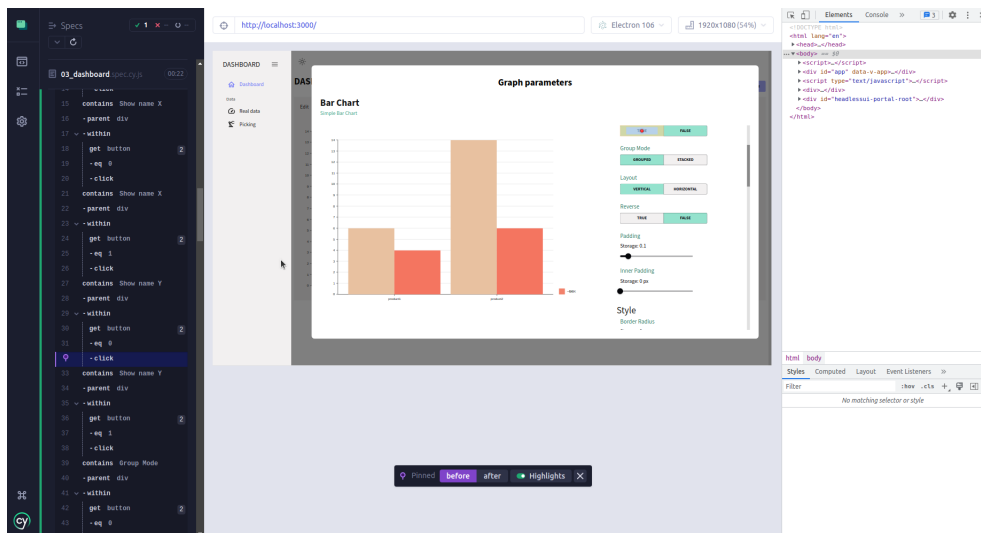
Tests:      4
Passing:    4
Failing:    0
Pending:    0
Skipped:    0
Screenshots: 0
Video:      true
Duration:   23 seconds
Spec Ran:   02_dataTables.spec.cy.js

(Video)
- Started processing: Compressing to 32 CRF
- Finished processing: /home/dmitrii/Documents/my_folder/dashboard/client/cypress/
  videos/02_dataTables.spec.cy.js.mp4 (2 seconds)

```

Obrázek 8.1: Tabulka s výsledky testů

Testy lze navíc spustit pomocí příkazu `npm run cypress`. Tento příkaz zahájí pokročilé testování s možností vizualizace, ale to může vyžadovat více systémové paměti. Při tomto spuštění bude uživateli poskytnut seznam všech dostupných testů a uživatel může spustit kterýkoli z testů kliknutím. Poté začne tento test běžet s vizualizací 8.2. Uživatel uvidí všechny kroky testu od začátku do konce s popisem všech kroků v levé části, které se provádějí při testování. Toto umožňuje rychle vyhodnotit, co se děje, a také pochopit, co by skutečný uživatel v systému dělal. Navíc to urychluje možnost testování systému, protože testy Cypress provádějí obrovské množství nezbytných kroků pro test, jako by je provedl skutečný uživatel. Při spuštění tímto způsobem si uživatel může vybrat, který test bude spuštěn.



Obrázek 8.2: Příklad běhu testu

Při napsání testu je nutné vytvořit postup kroků, který bude simulovat chování uživatele. Je důležité, že každý test musí ověřit jednu konkrétní věc, ale existují také komplexní testy, které umožňují simulovat různé scénáře použití aplikace. Cypress testy jsou ve složce `/client/cypress/e2e`. V této složce jsou tři soubory s testy. Soubory `01_navigation.spec.cy.js` a `02_dataTables.spec.cy.js` kontrolují jednoduché scénáře, jako je navigační přechod a filtrování v datových tabulkách. Soubor `03_dashboard.spec.cy.js` obsahuje jak jednoduché testy pro tvorbu grafů, tak komplexní testy, které přepínají možnosti nastavení vytvořeného grafu.

8.2 Testování s uživateli

Další důležitou součástí testování je testování s uživateli. Automatické testy umožňují urychlit testování a ověřit funkčnost aplikace, nicméně neřeší jeden z klíčových úkolů provedené práce: posoudit použitelnost aplikace a správnost vizualizace dat. Systém před sebou vidí graf s daty a kontroluje jeho funkčnost, ale nedokáže odhadnout, jak kvalitativně aplikace funguje z hlediska lidského vnímání.

Při tomto testování mi pomohli pracovníci firmy Sanezoo, kteří ověřili použití a možnost nastavení těchto komponent v aplikaci. Byla ohodnocena použitelnost aplikace a také soulad dat, která se používají v aplikaci Sanezoo. Pracovníci ocenili snadnost práce s aplikací i soulad s daty, která se v aplikaci používají.

Pozitivními prvky ze strany použitelnosti se stala možnost velkého přizpůsobení pro zobrazovaná data a přítomnost několika grafů pro vizualizaci. A také obecně přítomnost dashboardu, který původně v rozvrhu firmy nebyl. Bylo zdůrazněno, že byla poskytnuta velká příležitost pro přizpůsobení grafů, což je pohodlné a flexibilní při práci.

Z pohledu uživatele bylo zaznamenáno uživatelsky přívětivé rozhraní a intuitivnost a srozumitelnost komponent, se kterými bylo nutné pracovat. Konkrétní výběr dat byl hodnocen pozitivně z tohoto pohledu, že v reálném provozu s tímto systémem může pracovat člověk, který si nebude vědět, co si musí vybrat pro nějaký graf. Tato možnost urychluje práci pro běžného uživatele.

Nicméně byl také zaznamenán nedostatek možnosti vybrat data za určité období. Je například možné, že někteří uživatelé budou chtít zobrazit data za konkrétní měsíc nebo rok. Zpočátku analýza dat získaných z linky kontroly kvality od robotů nevyžadovala tuto funkcionalitu, protože byl omezen okruh pracovníků, kteří budou s touto aplikací pracovat. Tato možnost v aplikaci chybí také z důvodu, že byla poskytnuta omezená data, a také proto, že zpočátku nebyla vyžadována a nebyla promyšlena. Také někteří uživatelé by rádi viděli více možností pro změnu nastavení grafů, které jim nestačily.

Provedená práce však ukázala, že jsou splněny základní požadavky firmy, které byly ukázány na začátku. Další úkoly v rámci rozšíření funkčnosti a vylepšení aplikace budou realizovány při integraci do existující aplikace firmy Sanezoo.

Kapitola 9

Závěr

V rámci této práce byla provedena analýza, navrhování dashboardu a jeho implementace a testování vytvořených komponent. Cílem této práce je vytvořit dashboard s možností integrace do aplikací pro kontrolu kvality produktů. Je také nutné, aby byl tento dashboard snadno použitelný pro různé účely a aby s ním mohlo pracovat co nejvíce různých pracovníků firmy, která bude tento dashboard používat. Je třeba přidat možnosti výběru různých dat a možnost vytvářet a konfigurovat různé typy grafů, které budou tato data zobrazovat. V této práci byla provedena analýza existujícího řešení společnosti Sanezoo a byly identifikovány hlavní problémy, které byly opraveny. Byl vyvinut nový design a komponenty, které budou flexibilnější v nastavení a umožní práci s různými daty. Pak byla provedena implementace: byly vytvořeny nové komponenty pro vizualizaci dat a také pro přizpůsobení zobrazovaných dat. Byl také vytvořen dashboard, který může uživatel upravit a přizpůsobit podle svých potřeb. Po implementaci bylo provedeno testování, které se skládá z automatických testů, které umožnily urychlit proces testování funkcionality, a také z testování zaměřeného na skupinu, která poskytla zpětnou vazbu o kvalitě aplikace z uživatelského hlediska.

Dále byl proveden odhad použitelnosti výsledné aplikace. Podle posudků byly splněny základní požadavky a práce s aplikací se ukázala jako jednoduchá a srozumitelná. Ale někteří uživatelé chtěli mít více možností přizpůsobení grafů v různých parametrech a datech a také přidat podporu barev pro zrakově postižené uživatele. V dalším procesu integrace a rozšíření aplikace se plánuje přidání nových způsobů nastavení vizualizace dat na základě informací z výsledků testování a také zvýšení počtu typů grafů pro vizualizaci dat a možnost manipulace se získanými daty.

Literatura

- [1] ANSCOMBE, F. J. *Graphs in Statistical Analysis* [journal]. 1973. Dostupné z: <https://www.sjsu.edu/faculty/gerstman/StatPrimer/anscombe1973.pdf>.
- [2] *Automated Quality Control Systems* [online]. 2021. Dostupné z: <https://www.creaform3d.com/blog/automated-quality-control-systems/>.
- [3] VOJÁČEK, A. *Automatizace.HW.cz* [online]. 2021. Dostupné z: <https://automatizace.hw.cz/roboticke-3d-skenovani-a-kontrola-kvality-gom-scancobot.html>.
- [4] DAVID, M. *How to design dashboard*.
- [5] ECKERSON, W. W. *Performance Dashboards: Measuring, Monitoring, and Managing Your Business*. 2nd edition. Wiley, 2010. ISBN 9780470589830.
- [6] FEW, S. *Data Visualization for Human Perception*. Dostupné z: <https://www.interaction-design.org/literature/book/the-encyclopedia-of-human-computer-interaction-2nd-ed/data-visualization-for-human-perception>.
- [7] FEW, S. *Dashboard Confusion*. 2004. Dostupné z: https://www.perceptualedge.com/articles/ie/dashboard_confusion.pdf.
- [8] FEW, S. *Information Dashboard Design: The Effective Visual Communication of Data 1st Edition*. 1st edition. O'Reilly Media, 2006. ISBN 0596100167.
- [9] FEW, S. *Information Dashboard Design: Displaying Data for At-a-Glance Monitoring*. 2nd edition. Analytics Press, 2013. ISBN 1938377001.
- [10] HABIB, M. K. *Advanced Robotics and Intelligent Automation in Manufacturing*. 1. vyd. IGI Global, 2019. ISBN 1799813827.
- [11] MOHAN, M. *Creating dashboards for the color blind* [online]. 2019. Dostupné z: <https://uxdesign.cc/creating-dashboards-for-the-color-blind-74d62d67547>.
- [12] PANASIUK, J. a KACZMAREK, W. *Robotization of production processes*. Wydawnictwo Naukowe PWN, 2019. ISBN 830120415X.
- [13] PARMENTER, D. *Key Performance Indicators: Developing, Implementing, and Using Winning KPIs*. 3rd edition. Wiley, 2015. ISBN 9781118925102.
- [14] STOUT, K. *Quality Control in Automation Softcover*. Springer; Softcover reprint of the original 1st ed. 1985 edition, 2012. ISBN 1468475010.

- [15] TUFTE, E. *Sparkline theory and practice Edward Tufte*. Dostupné z: <https://aimeeknight.files.wordpress.com/2016/07/tuftesparklines.pdf>.
- [16] WARE, C. *Visual Thinking for Design (Morgan Kaufmann Series in Interactive Technologies)*. 1st edition. Morgan Kaufmann, 2008. ISBN 0123708966.
- [17] WARE, C. *Information Visualization: Perception for Design (Interactive Technologies)*. 3rd edition. Morgan Kaufmann, 2012. ISBN 0123814642.
- [18] WEXLER, S., SHAFFER, J. a COTGREAVE, A. *The Big Book of Dashboards: Visualizing Your Data Using Real-World Business Scenarios*. 1st edition. Wiley, 2017. ISBN 1119282713.