



Zadání diplomové práce

Název:	Analýza a návrh UX pro semiautomatické sledování času projektových aktivit
Student:	Bc. Vojtěch Kraus
Vedoucí:	Ing. Martin Čížek, MBA
Studijní program:	Informatika
Obor / specializace:	Webové a softwarové inženýrství, zaměření Webové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Pro poskytovatele konzultačních služeb je klíčová schopnost přesně a efektivně účtovat čas a aktivity jednotlivých konzultantů. To se obvykle děje na základě výkazů vytvářených konzultanty. Pro ekonomický výsledek je podstatná jak přesnost těchto výkazů, tak náklady na čas konzultantů strávený vykazováním. Pro jakýkoliv nástroj pro sledování času je tak naprosto kritické UX. Nástroje pro sledování času již existují, nicméně buď modelují pouze typ aktivity bez vazby na obchodní případy nebo jsou nedostatečné právě z hlediska UX.

Cílem práce je navrhnout řešení pro účinné sledování času stráveného nad konkrétními úlohami.

Díčí kroky:

- Popište slabiny existujících řešení.
- Analyzujte nástroje zadavatele a zhodnoťte nedostatky jejich UX.
- Definujte kroky pro odstranění nalezených nedostatků. Předpokládá se spolupráce s konzultanty z firmy zadavatele a průběžná tvorba prototypů.
- Navrhněte a následně použijte metriky pro vyhodnocení přínosu řešení.
- Připravte výsledný prototyp.



**FAKULTA
INFORMAČNÍCH
TECHNOLGIÍ
ČVUT V PRAZE**

Diplomová práce

Analýza a návrh UX pro semiautomatické sledování času projektových aktivit

Bc. Vojtěch Kraus

Katedra softwarového inženýrství

Vedoucí práce: Ing. Martin Čížek, MBA

6. května 2021

Poděkování

Rád bych poděkoval vedoucímu této práce Ing. Martinu Čížkovi, MBA, za jeho pomoc a odborné rady. Chtěl bych také poděkovat svým dalším kolegům z firmy Orchitech Solutions za všechny cenné rady, které mi během psaní této práce poskytli. V neposlední řadě bych také chtěl poděkovat své rodině a přítelkyni za podporu během studia.

Prohlášení

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

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

V Praze dne 6. května 2021

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2021 Vojtěch Kraus. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Kraus, Vojtěch. *Analýza a návrh UX pro semiautomatické sledování času projektových aktivit*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Tato diplomová práce se zabývá návrhem řešení pro efektivní a pohodlné sledování a vykazování času stráveného projektovými aktivitami ve firmě Orchitech Solutions. Návrh řešení vychází ze zhodnocení UX současného nástroje používaného ve firmě a podrobně analyzovaných požadavků a potřeb zaměstnanců firmy. Čtenář je nejprve stručně seznámen s problematikou návrhu UX se zaměřením na UX výzkum, tvorbu prototypů a testování UX. Dále je čtenáři stručně představena firma zadavatele práce spolu s rolí, kterou ve firmě hrají výkazy zaměstnanců. Následně se práce věnuje analýze existujících nástrojů pro sledování času se zaměřením na identifikaci klíčových nedostatků těchto nástrojů. Práce dále popisuje podrobnou analýzu UX nástroje zadavatele, jenž autor vykonal nejdříve sám a poté se zaměstnanci firmy zadavatele formou hloubkových rozhovorů. Dále následuje návrh nového řešení, které eliminuje identifikované nedostatky a naplňuje požadavky a potřeby zaměstnanců zjištěné při hloubkových rozhovorech. Návrh je zaměřen především na UX celkového řešení a zahrnuje popis funkcionalit spolu s grafickými návrhy uživatelských rozhraní jednotlivých aplikací, ze kterých se řešení skládá. Pro vybrané komponenty řešení byly následně implementovány interaktivní prototypy, které byly otestovány se zaměstnanci firmy zadavatele jakožto potenciálními uživateli. Na závěr se práce věnuje zhodnocení přínosu navrženého řešení pomocí stanovených metrik.

Klíčová slova UX výzkum, návrh UX, návrh UI, sledování času, vykazování času, Redmine, ActivityWatch, interaktivní prototyp, Electron

Abstract

This diploma thesis deals with the design of a solution for effective and convenient tracking and logging of time spent on project activities in company Orchitech Solutions. The design of the solution is based on UX evaluation of a tool that is currently used in the company and on a detailed analysis of the requirements and needs of the company's employees. The reader is first briefly acquainted with the topic of UX design with a focus on UX research, prototyping and UX testing. Furthermore, the reader is briefly introduced to this thesis supervisor's company together with the importance of time logged by the company's employees. Subsequently, the thesis deals with the analysis of existing time tracking tools with a focus on identifying the key shortcomings of these tools. The thesis then describes a detailed UX analysis of the client's tool, which the author first performed himself and then with employees of the client's company in the form of in-depth interviews. Then follows the proposal of a new solution that eliminates the identified shortcomings and meets the requirements and needs of employees identified during in-depth interviews. The design is mainly focused on the overall solution's UX and includes a description of the functionalities together with graphical designs of user interfaces of the individual applications that make up the solution. Interactive prototypes were subsequently implemented for selected components of the solution and properly tested with the company's employees as potential users. The last chapter of this thesis focuses on evaluation of the benefits of the proposed solution using established metrics.

Keywords UX research, UX design, UI design, time tracking, time logging, Redmine, ActivityWatch, interactive prototype, Electron

Obsah

Úvod	1
1 Teorie návrhu UX	3
1.1 Základní termíny	3
1.2 Návrh UX	3
1.3 UX výzkum	4
1.4 Příprava prototypů	6
1.5 Testování UX bez uživatelů	7
2 Představení firmy Orchitech Solutions	11
2.1 Nástroje používané ve firmě	11
2.2 Vykazování ve firmě	12
3 Rešerše a analýza existujících řešení pro sledování času	15
3.1 Přístupy ke sledování času	15
3.2 Různé přístupy k integraci se systémem Redmine	15
3.3 Nástroje pro manuální měření času	16
3.4 Nástroje pro automatické sledování aktivity	18
3.5 Nástroje určené primárně pro Redmine	19
3.6 TTRebel	20
4 Metriky pro zhodnocení přínosu navrhovaného řešení	23
4.1 Hlavní cíle navrhovaného řešení	23
4.2 Návrh metrik	23
4.3 Výběr reprezentativního vzorku zaměstnanců	24
5 Analýza a zhodnocení UX aplikace TTRebel	25
5.1 Vlastní zhodnocení UX	25
5.2 Analýza běžného užívání aplikace s uživateli	30
5.3 Zhodnocení UX aplikace se zaměstnanci	35

5.4	Shrnutí klíčových nedostatků UX	37
5.5	Obecné problémy zaměstnanců se sledováním a vykazováním odpracovaného času	38
6	Návrh řešení pro sledování a vykazování času	39
6.1	Požadavky na navrhované řešení	39
6.2	Architektura navrhovaného řešení	40
6.3	Webová aplikace TTRebel	42
6.4	Desktopový klient aplikace TTRebel	59
6.5	Rozšíření prohlížeče	63
6.6	ActivityWatch	66
6.7	Synchronizace jednotlivých komponent navrhovaného řešení	69
7	Implementace a testování interaktivních prototypů	71
7.1	Implementace prototypu desktopového klienta aplikace TTRebel	71
7.2	Testování prototypu desktopového klienta aplikace TTRebel s uživateli	76
8	Zhodnocení přínosu navrženého řešení	81
8.1	Připomenutí metrik	81
8.2	Výběr reprezentativního vzorku zaměstnanců	81
8.3	Vyhodnocení metrik před nasazením navrženého řešení	82
8.4	Vyhodnocení metrik po nasazení navrženého řešení	84
8.5	Výsledné zhodnocení	86
	Závěr	87
	A Seznam použitých zkratk	89
	B Obsah příloženého média	91
	C Grafické návrhy a snímky obrazovky	93

Seznam obrázků

6.1	Diagram nasazení komponent navrženého řešení	41
6.2	Grafické návrhy modálního okna s vyhledáváním Redmine úloh a výběrem Redmine úloh přiřazených na uživatele ve webové aplikaci TTRebel	46
6.3	Grafický návrh hlavní stránky webové aplikace TTRebel se spuštěným měřením	47
6.4	Grafický návrh modálního okna s detailem měření ve webové aplikaci TTRebel	49
6.5	Grafický návrh modálního okna se seznamem oblíbených měření ve webové aplikaci TTRebel	50
6.6	Grafické návrhy modálního okna s přehledem vykázanosti v aktuálním měsíci a aktuálním týdnem ve webové aplikaci TTRebel . . .	51
6.7	Grafický návrh stránky s exportem měření webové aplikace TTRebel s kurzorem namířeným na měření	53
6.8	Grafický návrh modálního okna s obecným nastavením ve webové aplikaci TTRebel	54
6.9	Grafický návrh hlavní stránky webové aplikace TTRebel na mobilním telefonu	56
6.10	Grafický návrh modálního okna s detailem měření ve webové aplikaci TTRebel na mobilním telefonu	57
6.11	Grafický návrh obrazovky pro vytvoření nového měření v desktopové aplikaci TTRebel	61
6.12	Grafický návrh obrazovky pro vytvoření nového měření v desktopové aplikaci TTRebel se spuštěným měřením	61
6.13	Grafický návrh obrazovky se seznamem uživatelových měření v desktopové aplikaci TTRebel	62
6.14	Grafický návrh obrazovky se seznamem uživatelových měření v desktopové aplikaci TTRebel se spuštěným měřením	62

6.15 Grafické návrhy nabídky pro vytvoření měření v aplikaci TTRebel a tlačítek k vyvolání nabídky	65
7.1 Snímek obrazovky s první verzí prototypu desktopové aplikace TTRebel používané na systému Windows	78
7.2 Snímek obrazovky s druhou verzí prototypu desktopové aplikace TTRebel používané na systému Windows	79
C.1 Hlavní stránka původní webové aplikace TTRebel	94
C.2 Správa časových intervalů v původní webové aplikaci TTRebel	95
C.3 Stránka s exportem původní webové aplikace TTRebel	96
C.4 Grafický návrh hlavní stránky webové aplikace TTRebel s ukotvením informací o měřeních při posunutí níže na stránku	97
C.5 Grafický návrh modálního okna s vyhledáváním a výběrem Redmine úloh otevřeného z modálního okna s detailem měření ve webové aplikaci TTRebel	98
C.6 Grafický návrh stránky s exportem měření webové aplikace TTRebel s kurzorem namířeným na časový interval na časové ose	99
C.7 Grafický návrh stránky s exportem měření webové aplikace TTRebel s kurzorem namířeným na prázdný interval na časové ose	100
C.8 Grafický návrh modálního okna s nastavením notifikací ve webové aplikaci TTRebel	101
C.9 Grafický návrh hlavní stránky webové aplikace TTRebel na mobilním telefonu s ukotvením informací o měřeních při posunutí níže na stránku	102
C.10 Grafický návrh exportní stránky webové aplikace TTRebel na mobilním telefonu	103
C.11 Grafický návrh exportní stránky webové aplikace TTRebel na mobilním telefonu s ukotvením souhrnných informací při posunutí níže na stránku	104
C.12 Grafický návrh uživatelského rozhraní desktopové aplikace TTRebel zasazený do popředí systému Windows spolu s ikonou signalizující aktivní měření v notifikační oblasti systému	105
C.13 Grafický návrh zakomponování nabídky pro vytvoření měření a tlačítka k vyvolání této nabídky na detail úlohy v systému Redmine	106
C.14 Grafický návrh zakomponování nabídky pro vytvoření měření a tlačítka k vyvolání této nabídky na detail merge requestu v systému GitLab	107

Seznam tabulek

Úvod

V mnoha firmách se dnes již běžně setkáme s povinností zaměstnanců vykazovat odpracovanou dobu, ať už za účelem evidence docházky, proplácení mezd či fakturace klientů. Významnou roli hrají výkazy především pro poskytovatele profesionálních konzultačních služeb, jejichž ekonomický zisk se přímo odvíjí od času jednotlivých konzultantů účtovaného klientům poskytovatele služeb. Pro takové firmy je tedy klíčové, aby konzultanti byli schopni přesně a efektivně sledovat a následně vykazovat čas strávený prací na různých projektech klientů firmy. Ekonomický výsledek firmy však také ovlivňují náklady na čas konzultantů strávený sledováním a vykazováním času, proto je pro jakýkoliv nástroj pro sledování času kritická uživatelská zkušenost (UX).

Část zaměstnanců firmy zadavatele Orchitech Solutions má již dlouhodobě negativní zkušenosti se sledováním a vykazováním času. Ve firmě bylo vyzkoušeno několik různých nástrojů pro sledování času, zaměstnanci dokonce i jeden takový nástroj sami vyvinuli, zatím však stále nebylo nalezeno řešení, které by každému vyhovovalo.

Hlavním cílem této práce je navrhnout řešení pro přesné, efektivní a pohodlné sledování a následné vykazování času stráveného projektovými aktivitami. Výstupem práce bude podrobná analýza a zhodnocení UX nástrojů zadavatele, analýza požadavků a potřeb zaměstnanců firmy zadavatele a návrh řešení primárně zaměřený na kvalitní UX.

Kapitola 1 této práce se věnuje teorii návrhu UX s bližším zaměřením na UX výzkum, tvorbu prototypů a testování UX. Kapitola 2 čtenáře stručně seznamuje s firmou zadavatele a popisuje roli a důležitost výkazů odpracovaného času zaměstnanců firmy. Kapitola 3 obsahuje autorovu rešerši a analýzu existujících řešení pro sledování času zaměřenou na vhodnost řešení pro zaměstnance firmy zadavatele.

Praktická část práce začíná kapitolou 4, která obsahuje návrh metrik vhodných pro zhodnocení přínosu řešení navrženého v rámci této diplomové práce. Kapitola 5 se věnuje podrobné analýze a následnému zhodnocení UX nástrojů

zadavatele, jenž autor vykonal nejdříve sám a poté s jednotlivými zaměstnanci firmy zadavatele, se kterými byly provedeny hloubkové rozhovory. Výstupem této kapitoly je seznam klíčových nedostatků UX nástrojů zadavatele, které jsou dále využity při návrhu řešení v kapitole 6. Návrh řešení je zaměřen především na kvalitní UX splňující požadavky a potřeby zaměstnanců firmy zadavatele, které byly stanoveny na základě proběhlých hloubkových rozhovorů. Kapitola dále obsahuje podrobný popis komponent navrženého řešení, popis navržených funkcionalit a grafické návrhy uživatelských rozhraní jednotlivých komponent řešení. Kapitola 7 se věnuje implementaci interaktivních prototypů vybraných komponent navrženého řešení a následnému testování prototypů se zaměstnanci jakožto budoucími uživateli. Přínos navrženého řešení je následně zhodnocen pomocí stanovených metrik v kapitole 8.

Teorie návrhu UX

1.1 Základní termíny

Termín „user experience“ (dále pouze „UX“) zapouzdřuje celkovou zkušenost, kterou uživatel prožívá při interakci s produktem či službou a jejím poskytovatelem. Ideální UX by mělo uživateli nabízet přesně to, co potřebuje, v jednoduché a elegantní podobě tak, aby uživatel produkt rád používal. UX je často mylně zaměňováno s uživatelským rozhraním (z anglického termínu „user interface“ či běžně zkráceně „UI“), které je ve skutečnosti pouze součástí UX. [1]

UX se také často zaměřuje s „použitelností“ produktu (z anglického termínu „usability“), která je však definována jako vlastnost UI určující, jak snadná je pro uživatele interakce s UI. Použitelnost UI je určena následujícími kritérii [2]:

- jak snadné je naučit se pracovat s rozhraním,
- jak efektivně lze rozhraní používat ke splnění zamýšlených cílů,
- jak snadné je zapamatovat si jak se s rozhraním pracuje,
- jak je rozhraní náchylné ke způsobení chyb a chybnému použití,
- jak příjemné je rozhraní používat.

1.2 Návrh UX

Proces návrhu UX lze rozdělit na následující kroky:

- porozumění cílovým uživatelům a pochopení jejich problémů,
- definice požadavků uživatelů,

- návrh řešení,
- sestavení prototypů pro navržené řešení,
- testování navrženého řešení pomocí prototypů,
- implementace navrženého řešení.

Pořadí plnění těchto kroků není pevně dané a běžně se některé kroky několikrát opakují, aby bylo dosaženo co nejlepšího výsledku. [3]

1.3 UX výzkum

Návrh UX je již z principu zaměřen primárně na (potenciální) uživatele produktu. Všechny kroky návrhu UX zahrnující práci s uživateli spadají do oblasti zvané „UX výzkum“. Aby UX návrhář (či tým návrhářů) mohl navrhnout ideální UX pro své uživatele, v rámci UX výzkumu se snaží pochopit jejich problémy, potřeby, motivaci, preference a porozumět jejich chování. Zpětná vazba od uživatelů, ať přímá či nepřímá, je pro návrháře UX cenná kdykoliv během návrhu UX. Před přípravou prvního návrhu je nutné znát požadavky uživatelů a každý vzniklý grafický návrh či prototyp je vhodné otestovat s uživateli. [4]

1.3.1 Metody UX výzkumu

Existuje mnoho různých metod pro zkoumání potřeb uživatelů či sbírání zpětné vazby na navržené UX. Tyto metody ve většině případů nelze libovolně zaměňovat, jejich použití se liší dle možností a dostupných prostředků návrháře, dle požadované podoby výstupů či dle toho, zda je k dispozici alespoň prototyp navrhovaného produktu. Výzkumné metody lze kategorizovat podle tří faktorů, které budou následně podrobně popsány [5]:

- zda-li zkoumáme postoj, nebo skutečné chování uživatelů,
- zda-li je výzkum kvalitativní, nebo kvantitativní,
- jak a zda vůbec uživatelé interagují s produktem během výzkumu.

Zkoumání postoje, nebo skutečného chování uživatelů

Pokud nás zajímá, jaký je postoj uživatelů k jejich problémům, potřebám a chování, využijeme výzkumné metody, které dají uživatelům prostor se vyjádřit. Díky těmto metodám můžeme získat přehled o mentálním modelu uživatelů, o způsobu, jakým vnímají informace, se kterými pracují a o jejich vlastních názorech na navrhovaný produkt. Mezi tyto výzkumné metody řadíme např. rozhovory s uživateli či různé dotazníky.

Při návrhu UX informačních systémů a aplikací jsou však častěji využívány metody, které zkoumají skutečné chování uživatelů. Tyto metody nám pomáhají pochopit, jak uživatelé produkt používají a pochopitelně tedy vyžadují, aby uživatelé měli přístup k danému produktu nebo jeho prototypu. Chování uživatele může sledovat a zaznamenávat jak samotný výzkumník, tak specializovaná technika, jako je např. kamera, která snímá pohyb očí uživatele během používání aplikace. Mezi tyto výzkumné metody řadíme např. A/B testování, snímání pohybu očí (tzv. „eyetracking“) a testování ve specializovaných UX laboratořích. [5]

Kvalitativní, nebo kvantitativní výzkum

Při kvalitativním výzkumu si výzkumník klade za cíl hluboce porozumět každému jednotlivému uživateli. Výzkumníci běžně pozorují uživatele napřímo, komunikují s nimi osobně a doptávají se na různé podrobnosti, pokud je potřeba. Typickou metodou kvalitativního výzkumu jsou hloubkové rozhovory s více či méně definovanou osnovou, které jsou běžně vedeny pouze mezi výzkumníkem a jediným uživatelem. Další běžně užívaná metoda zvaná „focus groups“ spočívá ve skupinových rozhovorech, kterých se účastní více uživatelů najednou a s výzkumníkem diskutují o předem určených tématech a plní různá cvičení.

Kvantitativní výzkumné metody na rozdíl od těch kvalitativních spočívají v matematickém vyhodnocování získaných dat a vyvozování závěrů na základě statistik. Získávání dat pro kvantitativní výzkumy je zpravidla časově méně náročné a umožňuje tak zkoumat větší množství uživatelů. Typickými zástupci kvantitativních výzkumných metod je vyplňování dotazníků či sledování chování uživatelů pomocí softwarových analytických nástrojů, jako je např. Google Analytics [6]. [5]

Interakce s produktem

Různé metody UX výzkumu se liší také tím, jak a zda-li vůbec zkoumaní uživatelé interagují s produktem. Běžně rozlišujeme tři případy:

- uživatelé používají produkt přirozeně, běžným způsobem,
- uživatelé používají produkt ke splnění předem definovaných cílů či scénářů,
- uživatelé s produktem vůbec neinteragují.

Při zkoumání chování uživatelů, kteří používají produkt běžným způsobem, se výzkumník snaží nijak nezasahovat a sledovat uživatele ve svém „přirozeném prostředí“. Ničím neovlivněné chování uživatelů představuje velice cenné informace, avšak za cenu toho, že se výzkumník nemusí vždy dozvědět

vše, co potřebuje. Velice rozšířená metoda takového typu se nazývá „etnografický výzkum“.

Výzkumník může pro uživatele stanovit cíle, kterých má použitím produktu dosáhnout, či scénáře, které uživatelům s jistou mírou neurčitosti říkají, co mají dělat. Tím, že uživatelé plní předem stanovené úkoly, sice výzkumník do jisté míry přichází o možnost pozorovat běžné chování uživatelů, na druhou stranu však získá zpětnou vazbu na užší část UX. [5]

1.4 Příprava prototypů

Prototypy nám umožňují rychle a levně převést návrh UX do hmatatelné podoby, nad kterou můžeme návrhy konzultovat v rámci týmu i se samotnými uživateli a snadno zapracovávat případné změny. Prototypy mohou nabývat různých podob, od statických grafických návrhů, přes interaktivní PDF dokumenty až po naprogramované uživatelské rozhraní. Záleží převážně na tom, jak daleko v procesu návrhu UX se nacházíme, zda se chceme pouze s týmem či zákazníkem domluvit na kostře uživatelského rozhraní nebo zda chceme otestovat (téměř) finální verzi návrhu. Díky prototypům můžeme brzy obdržet zpětnou vazbu na návrh UX a učinit potřebná rozhodnutí za menší náklady, než kdybychom vytvořili celou aplikaci.

Prototypy můžeme rozdělit podle úrovně věrnosti, tedy podle toho, jak moc se podobají výsledné aplikaci. Prototypy s nízkou úrovní věrnosti se nazývají „low-fidelity“ prototypy (někdy zkráceně „lo-fi“ prototypy nebo také „wireframy“), prototypy s vysokou úrovní věrnosti se nazývají „high-fidelity“ prototypy (někdy zkráceně „hi-fi“ prototypy) [7]. Některé zdroje uvádí také prototypy se střední úrovní věrnosti, které se svou věrností nachází někde mezi low-fidelity prototypy a high-fidelity prototypy. Pochopitelně běžně platí, že čím vyšší úroveň věrnosti prototyp má, tím více času návrhář stráví jeho přípravou. [8]

1.4.1 Low-fidelity prototypy

Low-fidelity prototypy znázorňují pouze kostru uživatelského rozhraní, ze které je patrné rozmístění prvků a obsah jednotlivých obrazovek/stránek aplikace. Používají se ve chvíli, kdy chceme co nejjednodušeji a nejrychleji vizualizovat nějaký návrh a odladit úplně základy. Low-fidelity prototypy záměrně neobsahují téměř žádnou grafiku a často jsou pouze černobílé. K vytvoření low-fidelity prototypu nám může postačit i obyčejná tužka a papír, existují však nástroje, pomocí kterých můžeme prototypy vytvořit v digitální podobě, např. Balsamiq [9], Adobe XD [10] či Pencil [11]. [7]

1.4.2 High-fidelity prototypy

High-fidelity prototypy by se měly co nejvíce podobat finálnímu produktu, jak v rozložení prvků a celkovém vzhledu uživatelského rozhraní, tak i předpokládaném obsahu, možných uživatelských akcích a chování aplikace. Tyto prototypy používáme spíše později v procesu návrhu UX, když chceme vytvořit co nejpřesnější reprezentaci návrhu, abychom ji mohli prezentovat zákazníkovi či otestovat s uživateli. High-fidelity prototypy jsou zpravidla interaktivní, ať už mají podobu PDF dokumentu či naprogramovaného uživatelského rozhraní. Pokud prototyp nechceme programovat, můžeme použít již zmíněné nástroje pro low-fidelity prototypy nebo např. nástroje Figma [12], InVision [13] či Sketch [14]. [7]

1.5 Testování UX bez uživatelů

Pro UX výzkum jsou uživatelé nesmírně cenní, v jistých situacích je však vhodnější nechat navržené UX otestovat experty, kteří mají s UX zkušenosti a zároveň nejsou přímou součástí návrhu UX. Zpětná vazba od experta je užitečná téměř kdykoliv, především v rané fázi návrhu UX, kdy provádění úprav není ještě tak časově náročné. Pokud se plánují velké změny v návrhu UX, je vhodné přizvat experta na otestování poslední verze aplikace, aby identifikoval silné i slabé stránky aktuálního UX a návrháři se měli od čeho odrazit při práci na novém UX. Běžně se také UX nechává otestovat expertem před testováním použitelnosti s uživateli, aby se k uživatelům dostala co nejvíce odladěná verze produktu.

UX experti při testování využívají svých zkušeností a znalostí jak z testování UX a použitelnosti UI, tak např. z oboru kognitivní psychologie. Existují však také různé návody a vodítka, kterými se expert může řídit při testování UX. Tento princip využívá metoda testování UX bez uživatelů zvaná „heuristická evaluace“, které se podrobněji věnuje následující sekce. [15]

1.5.1 Heuristická evaluace

Heuristická evaluace je metoda testování UX zaměřená na použitelnost UI. Jejím autorem je Jakob Nielsen, který v roce 1990 stanovil deset obecných heuristik, které by mělo splňovat každé uživatelské rozhraní. Jakob Nielsen, dodnes pracující v oblasti návrhu UX, se v roce 2020 vyjádřil, že tyto heuristiky jsou nadále relevantní i v dnešní době.

Dané heuristiky jsou poměrně obecné, je tedy lepší, když heuristickou evaluaci provádí více evaluátorů, aby se zvýšila šance odhalení více nedostatků. Neplatí však, že by se s přibývajícím počtem evaluátorů úměrně zvyšovala šance odhalení více nedostatků, příbytek je spíše logaritmický. Vhodný počet evaluátorů se může lišit v závislosti na velikosti projektu, komplexnosti pro-

duktu apod., z výzkumů však plyne, že pět evaluátorů odhalí v průměru až 75 % nedostatků v použitelnosti UI.

Během heuristické evaluace každý evaluátor samostatně prochází aplikaci a zkoumá, zda UI splňuje všechny heuristiky. Výstupem evaluace je seznam identifikovaných nedostatků s odkazy na konkrétní nedodržené heuristiky či jiné běžné praktiky použitelnosti UI a závažnost každého nedostatku. Doporučená či možná řešení jednotlivých nedostatků mohou také být součástí výstupu, řešení však často plyne z popisu nedostatku a porušené heuristiky. [16]

1.5.1.1 Vyhodnocování závažnosti odhalených nedostatků

Správně určit priority pro vyřešení odhalených nedostatků není snadný úkol, především v případě, kdy heuristickou evaluaci provádí více evaluátorů. Evaluátoři by měli odhaleným nedostatkům určit závažnost, na základě které budou následně určeny priority. Závažnost jednoho nedostatku však může každý evaluátor vnímat jinak, proto je vhodné obeznámit evaluátory s kritérii, které by při určování závažnosti měli zvážit [17]:

- frekvence, se kterou se nedostatek v aplikaci projevuje,
- dopad nedostatku na uživatele,
- přetrvávání nedostatku – omezuje nedostatek uživatele pokaždé, když na něj narazí, nebo jej dokáže uživatel překonat?

Závažnost odhalených nedostatků použitelnosti aplikace mohou evaluátoři stanovit pomocí hodnot 0 až 4 Nielsenovy škály [17]:

- 0 – nejedná se o nedostatek použitelnosti aplikace,
- 1 – jedná se pouze o kosmetický nedostatek, měl by být vyřešen pouze v případě, že na projektu přebývá čas,
- 2 – drobný nedostatek, měl by být vyřešen s nízkou prioritou,
- 3 – velký nedostatek, měl by být vyřešen s vysokou prioritou,
- 4 – kritický nedostatek, musí být vyřešen před spuštěním aplikace do ostrého provozu.

1.5.1.2 Deset heuristik Jakoba Nielsena

1. **Viditelnost stavu systému** – Uživatel by měl být informován o stavu systému vhodnou cestou s dostatečnou rychlostí.
2. **Shoda mezi systémem a skutečným světem** – Systém by měl používat jazyk, terminologii a koncepty, které uživatel dobře zná.

3. **Uživatelská kontrola a svoboda** – Uživatel by měl mít pocit, že má systém plně pod kontrolou a může zvrátit nechtěně provedené akce.
4. **Konsistence a standardy** – Systém by měl následovat zavedené standardy a konvence, aby uživatel nemusel zbytečně přemýšlet nad svými akcemi.
5. **Prevence chyb** – Systém by měl uživateli dostatečně bránit ve vyvolání systémových chyb a provedení nechtěných akcí, např. pomocí validací a potvrzovacích dialogů.
6. **Rozpoznání raději než vzpomínání** – Systém by měl být navržený tak, aby minimalizoval nutnost uživatele pamatovat si informace pro dosažení požadovaných cílů.
7. **Flexibilní a efektivní použití** – Zkušenější uživatel by měl mít možnost dosáhnout požadovaných cílů efektivněji, např. pomocí klávesových zkratk či přizpůsobení systému vlastním potřebám.
8. **Estetický a minimalistický design** – Systém by měl uživateli dostatečně jasně a přehledně prezentovat důležité informace a nerozptylovat jej od svých cílů nepodstatnými informacemi či rušivými vizuálními prvky.
9. **Pomoc uživateli rozpoznat a pochopit chybu a vzpamatovat se z ní** – Nastane-li v systému chyba, měla by být uživateli prezentována v dostatečně výstižné a pochopitelné podobě. Systém by měl také uživateli nabídnout možná řešení.
10. **Nápověda a dokumentace** – Stručná a přehledná nápověda k používání systému a další dokumentace by měly být uživateli na blízku vždy, kdy by je mohl potřebovat. [18]

Představení firmy Orchitech Solutions

Orchitech Solutions je menší konzultantská společnost figurující v odvětví vývoje software. Firma má několik klientů, pro které různě velké týmy zaměstnanců tvoří a spravují softwarová řešení na míru. Kromě klientských projektů má firma také několik vlastních projektů, na kterých běžně pracují menší týmy zaměstnanců.

Orchitech Solutions čítá v době psaní této práce 12 zaměstnanců. Někteří zaměstnanci zastávají na různých projektech různé role, např. testerů figurují na některých projektech jako projektoví manažeři, projektoví manažeři se věnují marketingu firmy apod.

2.1 Nástroje používané ve firmě

Hlavním nástrojem používaným v Orchitech Solutions je issue tracker Redmine [19]. Tento open-source nástroj pro firmu představuje „hlavní zdroj pravdy“, který je využíván k projektovému řízení, evidenci informací o projektech, klientech a jiných interních záležitostech, sdílení vědomostí a zkušeností a v neposlední řadě také evidenci výkazů. Veškerá činnost zaměstnanců by se měla vztahovat k nějaké úloze v Redmine, veškerá jakkoliv důležitá rozhodnutí a kroky, které k nim vedly, by měly být evidovány v Redmine.

Kromě několika vlastních rozšíření, které mají zaměstnancům práci s Redmine příjemnit, využívá firma také vlastní nástroje integrované s Redmine. Jedním z těchto nástrojů je Time Tracking Rebel (dále jen TTRebel), který zaměstnancům slouží ke sledování a vykazování času stráveného řešením konkrétních Redmine úloh.

Dalšími nástroji, které se ve firmě hojně používají, jsou nástroj pro komunikaci Slack [20], aplikace platformy Google Workspace (Gmail, Disk, Kalendář apod.) [21] a systém pro správu zdrojových kódů GitLab [22].

2.2 Vykazování ve firmě

Každý zaměstnanec firmy je odpovědný za vykazování vlastního času, který stráví řešením nějaké úlohy. Za řešení úlohy je považováno plnění zadání úlohy, asistování hlavnímu řešiteli, různé kontroly, konzultace apod. Od všech zaměstnanců se očekává, že jejich výkazy pokryjí veškerý čas, který měli dle smlouvy v uplynulém měsíci odpracovat.

2.2.1 Vytváření výkazů

Jak již bylo zmíněno, výkazy se ve firmě Orchitech Solutions evidují na úrovni úloh v systému Redmine. Práce s výkazy v Redmine je velice variabilní, uživatelé si mohou zobrazovat součet časů vykázaných u konkrétní úlohy, u podúloh dané úlohy, u konkrétního projektu či uživatele. Redmine také umožňuje vytvářet a exportovat komplexní přehledy a reporty, např. vykázaný čas konkrétních uživatelů na konkrétním projektu za dané časové období.

2.2.1.1 Náležitosti výkazu v systému Redmine

Výkaz v systému Redmine obsahuje následující atributy:

- úloha, ke které se výkaz vztahuje,
- uživatel,
- datum,
- čas k vykázání,
- aktivita,
- doplňující komentář.

Všechny zmíněné atributy jsou při vytváření nového výkazu povinné. Redmine dále umožňuje definovat vlastní atributy, u kterých je možné nastavit povinnost atributu a očekávaný typ vstupu. V Orchitech Solutions aktuálně existují dva takové atributy, v tuto chvíli však nejsou aktivně používány.

Na formulář pro vytvoření výkazu se uživatel nejčastěji dostane přes tlačítko dostupné na konkrétní úloze, která je v takovém případě ve formuláři již předvyplněna. Pole s uživatelem je vždy předvyplněno aktuálně přihlášeným uživatelem.

Seznam aktivit, ze kterých uživatel může vybrat, je možné nadefinovat v administraci systému. Orchitech Solutions používá aktivity pokrývající běžné činnosti při vývoji software, jako je analýza a návrh, vývoj, revize kódu, testování, konzultace a řízení projektu.

2.2.2 K čemu jsou výkazy používány

Běžným a očekávaným využitím výkazů v konzultantských společnostech je tvorba podkladů pro vyúčtování klientů a proplácení vlastních zaměstnanců. Po hlubší analýze s vedením firmy však vyšlo najevo, že výkazy mají pro chod firmy větší hodnotu, než by se mohlo na první pohled zdát. Následuje popis všech oblastí, ve kterých výkazy hrají důležitou roli.

Účetnictví

Orchitech Solutions je konzultantská společnost, její výdělek tedy závisí především na fakturaci klientů. Čas, který zaměstnanci stráví běžným vývojem či podporou na projektu klienta, musí být danému klientovi pochopitelně náúčtován.

Na základě výkazů jsou také vypláceni zaměstnanci, jejichž výplata se odvíjí od počtu odpracovaných hodin. Jedná se o vlastní zaměstnance i externí dodavatele.

Projektové řízení

Aktuálnost výkazů pomáhá projektovým manažerům při plánování řešení úloh a alokaci zaměstnanců na různých projektech. Dle typu spolupráce je na některých projektech také potřeba průběžně hlídat, že nebyl překročen počet hodin, které si klient objednal nebo které byly přiděleny internímu projektu. Dalším běžným případem jsou klientské požadavky s dopředu stanoveným počtem hodin, které budou klientem zaplacené. U takových požadavků je nutné průběžně hlídat vykázaný čas, aby v případě hrozícího překonání stanoveného času mohla být učiněna potřebná rozhodnutí.

Hospodaření firmy

Výkazy jsou důležité i pro hospodaření firmy, především z dlouhodobého hlediska. Díky výkazům může firma sledovat a porovnávat výnosy a náklady na jednotlivé projekty, jak klientské, tak interní. Na základě toho vedení firmy vyhodnocuje obraty a dělá různá obchodní rozhodnutí, jako je financování interních projektů, změny ve spolupráci s klienty apod.

Podklady pro analýzu

Při odhadování pracnosti nového požadavku se mohou vývojáři inspirovat vykázaným časem u podobných úloh, které již byly vyřešeny. Po dokončení úlohy výkazy také poskytují srovnání původně odhadovaného času se skutečně stráveným časem. Stane-li se, že byl odhadovaný čas překonán, projektový tým může zpětně analyzovat případné problémy a poučit se z nich.

2.2.3 Důsledky nesprávného vykazování

Z předchozí sekce můžeme odvodit mnoho negativních důsledků, které má nesprávné vykazování jednotlivých zaměstnanců – finanční ztráta na klientských projektech, nesrovnalosti v účetnictví, obtížné řízení projektů, obchodní rozhodnutí založená na nepravdivých datech apod. Je však také potřeba zmínit další důsledek, kterým je frustrace zaměstnanců, kteří s výkazy pracují (např. při vytváření faktur či různých reportů).

Rešerše a analýza existujících řešení pro sledování času

3.1 Přístupy ke sledování času

Pro sledování času při práci (nejen) na počítači dnes existuje velké množství nástrojů, ať už s komerční, či s open-source licencí. Tyto nástroje se liší především v tom, jakým způsobem umožňují uživateli sledovat čas strávený nějakou činností. Některá řešení spočívají v manuálním měření času, které lze přirovnat k běžným stopkám, některá řešení automaticky sbírají informace o uživatelské aktivitě na počítači a na základě toho vyvozují čas strávený danými činnostmi. Existují také řešení, která uživateli nabízí obě zmíněné možnosti, často jsou však zaměřena primárně na jednu z možností a tu druhou poskytují navíc jako doplněk.

3.2 Různé přístupy k integraci se systémem Redmine

Integrace se systémem Redmine je pro zadavatele práce klíčovým požadavkem na nástroj pro sledování času. U dostupných nástrojů se můžeme zpravidla setkat s dvěma typy integrace s Redmine (či jinými systémy pro řízení projektů), které se rozlišují tím, který ze dvou integrovaných systémů je považován za „zdrojový“ a který za „cílový“.

Pokud je za cílový systém považován nástroj pro měření času, běžně se setkáme s řešením, které na vybraná místa ve zdrojovém systému přidá nějaký ovládací prvek pro interakci s nástrojem na sledování času. Spuštění měření času často také automaticky převezme nějaké klíčové informace ze zdrojového systému, aby je uživatel nemusel vyplňovat sám. V případě systému Redmine je takový ovládací prvek přidáván na detailní stránku úlohy, ze které jsou převzaty informace jako je identifikátor, předmět či typ úlohy. I když se může

zprvu zdát, že takováto integrace nemá moc velký přínos, přidaný ovládací prvek může zpříjemnit ovládání nástroje pro měření času a především také uživateli připomenout, že by si měl měřit čas strávený prací na dané úloze. Nástroje, které poskytují tento typ integrace, jsou např. Toggl [23], TMetric [24] a Clockify [25].

V případě, kdy je za cílový systém považován Redmine (či jiný integrovaný systém), nástroje pro sledování času pak umožňují uživateli ve svém uživatelském rozhraní měřit čas ke konkrétním úlohám a následně k nim naměřený čas exportovat. Tento druh integrace není ve srovnání s prvním zmíněným druhem tak častý a poskytují jej především nástroje, jejichž účelem je primárně právě měření a vykazování času stráveného nad Redmine úlohami – těmto nástrojům se více věnuje sekce 3.5. Nástroje, které se nezaměřují pouze na Redmine a také poskytují tento druh integrace, jsou např. TMetric a Time Doctor [26].

3.3 Nástroje pro manuální měření času

Nástrojů pro manuální měření času existuje více než nástrojů pro automatické sledování aktivity a většina z nich si jsou principiálně velice podobné. Uživatel může měřit čas pomocí mechanismu podobného běžným stopkám s prostými poznámkami či vybranými činnostmi nebo zadávat naměřený čas manuálně. Naměřený čas si pak uživatelé mohou prohlížet v časových osách či grafech, některé nástroje umožňují export do různých formátů či integrovaných systémů.

Z této skupiny nástrojů se práce blíže věnuje pouze nástroji Toggl, který se zdá být jedním z nejužívanějších nástrojů pro sledování času vůbec, a nástroji Timeular [27]. Autor práce prozkoumal a vyzkoušel i jiné nástroje – např. Clockify, TMetric, Kimai [28] – vyhodnotil však, že oproti zmíněným dvěma nástrojům nepřináší navíc nic podstatného pro tuto práci.

3.3.1 Toggl

Toggl je nástroj, který se objevuje ve většině článků o nástrojích pro sledování času a zdá se, že je jedním z těch nejužívanějších. Nástroj je poskytovaný jako SaaS v několika placených variantách a i jedné neplacené variantě. Webová aplikace s minimalistickým rozhráním umožňuje uživateli manuálně spouštět či zastavovat měření času s různými popisky. Na základě měření vznikají časové intervaly s danými popisky, kterým je dále možné přiřazovat projekty či vlastní štítky. Uživatelé si mohou svá měření procházet v detailních statistikách a časových osách, které je možné různě filtrovat, řadit a shlukovat podle projektů, štítků, popisků či časového období. [29]

Aplikace nabízí velké množství integrací s dalšími systémy, převážně se systémy pro řízení projektů a týmů, správu úkolů apod. Většina integrací

spočívá v tom, že si uživatel do prohlížeče nainstaluje rozšíření, které do integrované aplikace přidá ovládací prvek na měření času v Toggl.[30] Uživatelům je kromě webové aplikace také k dispozici mobilní aplikace a desktopový klient, jenž navíc nabízí možnost plánování upomínek či detekci uživateli aktivity na daném zařízení.

UX samotné aplikace je na dobré úrovni – hlavní stránka s měřením času je přehledná a minimalistická, časové intervaly jsou kompaktně zobrazeny, ale i přes to jsou všechny potřebné informace snadno dostupné. Ovládání aplikace je jednoduché na pochopení a uživatel se s aplikací rychle naučí pracovat.

3.3.1.1 Klíčové nedostatky

Integrace Toggl s Redmine přidává na stránky s Redmine úlohami tlačítko pro měření času stráveného danou úlohou. Integrace ve druhém směru, tedy vytvoření výkazů u úloh na základě měření času v Toggl, dostupná není. Absence této funkcionality je zásadní nevýhodou tohoto řešení. Dalším nedostatkem pro uživatele Redmine je absence nějakého našeptávače, který by uživatelům pomáhal při výběru úlohy, ke které chtějí měřit čas. [31]

3.3.2 Timeular

Timeular je placený multiplatformní nástroj s minimalistickým uživatelským rozhraním zaměřený na pohodlné sledování času při práci. Uživatelům je k dispozici v podobě mobilní a desktopové aplikace. Pro efektivnější a méně rozptylující ovládání desktopové aplikace nabízí Timeular funkcionalitu zvanou „QuickTrack“, která uživatelům umožňuje pomocí zvolené klávesové zkratky vyvolat jednoduché modální okno, ve kterém si mohou rychle přidat poznámku k měřené činnosti nebo vybrat z nabídky historických činností. [32]

Klíčovým aspektem, kterým se Timeular odlišuje od ostatních nástrojů, je hardwarové zařízení pro ovládání sledování času. Jedná se o osmistěnnou kostku, jejímž otáčením uživatel přepíná mezi dopředu nastavenými aktivitami. Kostka je připojena přes bluetooth k zařízení s aplikací Timeular, ve které se přiřazují aktivity k jednotlivým stěnám kostky.[33] Autor práce neměl příležitost vyzkoušet si kostku osobně, avšak na základě mnoha recenzí se zdá, že ovládání sledování času pomocí kostky je pro uživatele zábavné a jednoduché. Oceňují především to, že nemusí na svém počítači či mobilním telefonu hledat aplikaci Timeular vždy, když chtějí přepnout svou aktuální činnost. Samotná kostka na pracovním stole také uživatelům připomíná, že si mají měřit čas při práci či jiných činnostech. [34]

3.3.2.1 Klíčové nedostatky

Timeular v době psaní této práce nenabízí žádnou integraci se systémem Redmine¹. I kdyby integrace existovala v takové podobě, že by se otáčením kostky přepínala práce na jednotlivých Redmine úlohách, uživatelé by museli úlohy znát dopředu, což pro některé zaměstnance není dobře možné už z povahy jejich práce.

3.4 Nástroje pro automatické sledování aktivity

Nástroje pro automatické sledování aktivity zpravidla vyžadují instalaci desktopového klienta, který má přístup k informacím o spuštěných aplikacích v operačním systému. Tento klient pak sleduje aktivitu uživatele a zaznamenává čas strávený používáním jednotlivých aplikací a v případě prohlížečů i čas strávený na konkrétních webových stránkách. Některé ze skupiny těchto nástrojů aktivitu rozdělují do kategorií, čemuž mohou uživatelé napomoci vlastními pravidly (např. interakce s příkazovým řádkem spadá do kategorie „programování“). Uživatelům je jejich aktivita často prezentována v podobě časových os nebo grafů s procentuálním zastoupením jednotlivých aplikací či kategorizovaných činností.

U většiny prozkoumaných řešení poskytovaných jako SaaS jsou data o aktivitě uživatele ukládána u poskytovatele služby, díky čemuž může služba uživateli nabídnout sledování aktivity a její synchronizaci napříč několika zařízeními. Pro některé uživatele však může aktivita na počítači představovat velice citlivá data, která by neměla opustit jejich vlastní zařízení. V takovém případě musí uživatelé sáhnout spíš po open-source řešeních, avšak za cenu vlastního provozování.

Práce se dále podrobněji věnuje nástrojům RescueTime a ActivityWatch. RescueTime reprezentuje nástroje poskytované jako SaaS, mezi které patří např. Timely [36] a Time Doctor [26], ActivityWatch reprezentuje open-source nástroje, mezi které patří např. Selfspy [37] a ulogme [38].

3.4.1 RescueTime

RescueTime je placený nástroj, který se specializuje na automatické sledování času stráveného u počítače. Hlavní myšlenkou nástroje je poskytnout uživateli přehled o tom, jak tráví svůj čas na počítači a kolik času tráví produktivními činnostmi ve srovnání s těmi neproduktivními. Nástroj se snaží uživatele motivovat k větší produktivitě pomocí notifikací upozorňujících na dlouhou neaktivitu, dále také nabízí možnost blokovat používání rozptylujících aplikací či potlačovat rozptylující systémové notifikace. [39]

¹Návrh na tuto integraci můžeme nalézt na stránkách s požadavky uživatelů, avšak ve srovnání s oblíbeností ostatních požadavků se dá usoudit, že tato integrace v blízké budoucnosti přidána nebude. [35]

3.4.1.1 Klíčové nedostatky

Aplikace nenabízí žádnou integraci se systémem Redmine. Zaznamenaná aktivita může sice posloužit jako podklad pro manuální tvorbu výkazů, to ale jako samostatné řešení pro sledování a vykazování času nepostačí. Jelikož je aplikace opatřena komerční licenci, není možné ji nijak rozšířit o integraci s Redmine.

Pro některé uživatele může také být velkým nedostatkem tohoto řešení již zmíněné ukládání informací o jejich aktivitě u poskytovatele služby.

3.4.2 ActivityWatch

Zřejmě nejznámější open-source nástroj pro automatické sledování aktivity na počítači. Je vybaven již zmíněnými funkcionalitami, jako je podrobné sledování používání aplikací, grafické uživatelské rozhraní s časovými osami, statistikami a možnostmi exportu. Autoři nástroje zastávají názor, že aktivita na počítači jsou velice citlivá data, která by uživatelé měli mít plně pod kontrolou a ActivityWatch je i takto navržen. Veškerá naměřená data se ukládají do databáze na zařízení uživatele. ActivityWatch je multiplatformní a je možné jej používat i na systému Android, avšak plnohodnotná synchronizace mezi zařízeními ještě nebyla implementována. [40]

ActivityWatch je navržen tak, aby jej bylo možné rozšířit dle potřeby uživatelů. Uživatelé si mohou implementovat vlastní komponentu pro sbírání dat či vlastní dotazy nad nasbíranými daty, vše pomocí knihoven pro jazyk Python [41] a REST API tohoto nástroje. Členové komunity okolo ActivityWatch již vytvořili několik volně dostupných rozšíření, např. pro sběr aktivity při používání prohlížeče či různých IDE. [42]

3.4.2.1 Klíčové nedostatky

Hlavním nedostatkem nástroje ActivityWatch je absence jakékoliv integrace se systémem Redmine. Avšak díky tomu, že uživatel má sbíraná data plně pod kontrolou a aplikace je snadno rozšiřitelná a přizpůsobitelná potřebám uživatele, je ActivityWatch vhodným adeptem pro vlastní řešení využívající automatické sledování aktivity na počítači.

3.5 Nástroje určené primárně pro Redmine

Existuje také několik nástrojů zaměřených speciálně na sledování času stráveného nad Redmine úlohami či projekty. Minimum z těchto nástrojů však lze považovat za plnohodnotná řešení s kvalitním UX, která by uživatelé mohli pohodlně používat každý den při práci. Práce se dále zabývá pouze dvěma nástroji, které autor práce vyhodnotil jako nejlepší zástupce z této skupiny –

komerční Redmine Time Tracker od Lerus LTD. [43] a open-source projekt na GitHub „ujiro99/RedmineTimeTracker“ [44].

3.5.1 Redmine Time Tracker od Lerus LTD.

Desktopová aplikace s minimalistickým rozhraním působí, že obsahuje vše, co by si mohli uživatelé přát – našeptávač úloh, jednoduché ovládání měření času, seznam úloh, na kterých aktuálně pracují ostatní uživatelé či možnost automatického exportu naměřeného času do Redmine. Spolu s aplikací je také nabízeno rozšíření do Redmine, které na detailní stránku úloh přidá jednoduché ovládání měření času stráveného danou úlohou. Řešení je multiplatformní a podporuje nejnovější verzi Redmine, bohužel je opatřeno komerční licencí a autor práce si jej nemohl osobně vyzkoušet. [45]

3.5.1.1 Klíčové nedostatky

Aplikace není dostupná v podobě nativní mobilní aplikace či webové aplikace, která by mohla být používána na mobilních telefonech a tabletech. Dalším nedostatkem také může být cena za komerční licenci, která však není veřejně známá a je nutné kvůli ní kontaktovat dodavatele.

3.5.2 GitHub projekt ujiro99/RedmineTimeTracker

Tento open-source nástroj je dostupný v podobě rozšíření prohlížeče či desktopové aplikace. Uživatelské rozhraní je velice jednoduché a aplikace obsahuje pouze nezbytné funkcionality, jako je vyhledávání Redmine úloh, měření času a export výkazů do Redmine. I když je nabídka funkcionalit o něco menší ve srovnání s předchozím nástrojem (viz sekce 3.5.1), tato aplikace obsahuje navíc režim měření času zvaný „Pomodoro time“, který uživateli v pravidelných intervalech připomíná, aby si dal od práce krátkou pauzu. [46]

3.5.2.1 Klíčové nedostatky

Stejně jako předchozí nástroj (viz sekce 3.5.1), ani tato aplikace není dostupná v podobě nativní mobilní aplikace či webové aplikace, která by mohla být používána na mobilních telefonech a tabletech. Přestože desktopová aplikace poslouží svému účelu, UX není na moc dobré úrovni.

3.6 TTRebel

TTRebel je webová aplikace pro manuální sledování a vykazování času stráveného řešením konkrétních Redmine úloh. Tato aplikace je dílem Orchitech Solutions a její zaměstnanci ji využívají již přes pět let. Důvodem vzniku aplikace byla absence open-source řešení, které by splňovalo potřeby zaměstnanců.

Většina nástrojů pro sledování času integrovaných s Redmine nenabízela možnost exportovat naměřený čas do Redmine v podobě výkazů a ty, které to nabízely, byly closed-source či placené, což znemožňovalo jejich snadné rozšíření o další požadované funkcionality.

3.6.1 Popis aplikace

Hlavním prvkem aplikace TTRebel je tzv. „tracking session“, v rámci které si uživatelé stopují čas strávený řešením konkrétní Redmine úlohy. Tracking session obsahuje všechny náležitosti výkazu v Redmine a spolu s naměřeným časem ji uživatelé mohou do Redmine exportovat jako výkaz času.

Aplikaci tvoří dvě stránky – hlavní stránka, na které si uživatelé spravují své tracking sessions a stopují čas (viz obrázek C.1) a stránka s exportem tracking session do Redmine (viz obrázek C.3). Na hlavní stránce jsou tracking sessions shlukovány po dnech, po kterých jsou také exportovány. Stránka s exportem poskytuje uživateli přehled s časem již vykázaným v Redmine pro konkrétní den, časem k exportu a celkovým časem, který bude po exportu vykázan.

U tracking session uživatelé vybírají Redmine úlohu, vyplňují aktivitu a popis práce. Tracking session je možné založit buď s konkrétní Redmine úlohou vybranou z našeptávače nebo pouze s popisem práce – taková tracking session je v aplikaci označována jako „anonymní tracking session“. Tracking session musí obsahovat všechny zmíněné údaje, aby ji bylo možné exportovat do Redmine. Tyto údaje mohou uživatelé upravovat jak na hlavní stránce, tak na stránce s exportem. Pokud si uživatel nepřeje exportovat některou tracking session, může ji z exportu vynechat odškrtnutím checkboxu, jak je vidět na obrázku C.3.

Stopování času u konkrétní tracking session je reprezentováno časovými intervaly. Tyto intervaly mohou uživatelé v aplikaci libovolně upravovat, odstraňovat či manuálně přidávat nové, jak je vidět na obrázku C.2. Na stránce s exportem mohou navíc uživatelé u jednotlivých tracking session upravovat čas, který bude exportován do Redmine, bez nutnosti upravovat časové intervaly. Pokud uživatel čas k exportu upraví, zobrazí se tlačítko pro obnovení času k exportu na původní naměřený čas.

3.6.2 Technická stránka aplikace

TTRebel má architekturu běžné moderní webové aplikace – frontend (klient) a backend (server). Frontend je implementován v JavaScriptovém frameworku, konkrétně Vue.js [47], backend je implementován v jazyce Java s frameworkem Spring Boot [48]. Komunikace mezi oběma částmi aplikace i mezi backendem a Redmine probíhá přes REST API. K persistenci dat používá aplikace databázový stroj PostgreSQL [49].

3.6.3 Uživatelské rozhraní

Uživatelské rozhraní je značně inspirované Material designem [50], souborem prvků a nejrůznějších zásad pro uživatelská rozhraní. Material design je nejvíce používán na mobilních aplikacích pro Android, několik posledních let však roste jeho oblíbenost i u webových aplikací. Pro frontendové frameworky běžně existují knihovny implementující prvky Material designu dle oficiálních specifik, které mohou vývojáři používat, aby nemuseli prvky vytvářet sami. Oficiální knihovnou pro framework Vue.js je Vuetify [51], které se používá právě v aplikaci TTRebel.

3.6.4 Podpora více systémů

V Orchitech Solutions se kromě hlavního firemního systému Redmine používá ještě jedna instance Redmine dedikovaná jedinému projektu, který bude pro účely této práce označován jako „Support projekt“. Tato instance Redmine slouží primárně pro zákaznickou podporu a je tomu přizpůsobena svým nastavením a různými rozšířeními.

Původně byla aplikace TTRebel navržena tak, aby ji bylo možné používat s více různými instancemi systému Redmine, relativně nově implementovaný frontend tomu však nebyl zcela přizpůsoben. Vzhledem k tomu, že na Support projektu pracuje pouze zlomek zaměstnanců, dokončení této funkcionality zatím není hlavní prioritou vývojového týmu.

3.6.5 Rozšíření aplikace

V rámci diplomové práce [52] vznikl prototyp rozšíření pro aplikaci TTRebel v podobě tlačítka přidaného do Redmine. Toto tlačítko se nachází na detailní stránce každé úlohy a umožňuje uživateli spustit či zastavit měření času dané úlohy. Tlačítko je pouze zodpovědné za odesílání instrukcí aplikaci TTRebel (vytvořit tracking session pro danou úlohu a spustit či zastavit měření) a komunikování stavu měření uživateli.

Myšlenka tohoto rozšíření – poskytnout uživateli možnost spustit měření času stráveného na konkrétní úloze již ve chvíli, kdy se nachází na detailní stránce této úlohy – byla uživateli přijata pozitivně. Vzniklý prototyp však vyžadoval značné úpravy, kterých se mu dodnes nedostalo a rozšíření se ve firmě aktuálně nepoužívá.

3.6.6 Klíčové nedostatky

Klíčové nedostatky aplikace TTRebel budou podrobně analyzovány v kapitole 5.

Metriky pro zhodnocení přínosu navrhovaného řešení

4.1 Hlavní cíle navrhovaného řešení

V sekci 2.2.3 již bylo popsáno, jaký dopad má na firmu Orchitech Solutions nesprávné vykazování zaměstnanců. Ve firmě je vykazování dlouhodobě rozebírané téma a dle zadavatele práce dosud nebylo nalezeno dostatečně efektivní řešení, které by vyhovovalo všem zaměstnancům.

Na základě domluvy se zadavatelem práce bude hlavním cílem navrhovaného řešení zpříjemnit uživatelům sledování a vykazování času stráveného prací na Redmine úlohách natolik, aby byli schopni si efektivně a pravidelně vykazovat bez zbytečného rozptylování od jejich práce. Dílčí cíle navrhovaného řešení je tedy možné stanovit následovně:

- minimalizovat čas strávený sledováním a vykazováním času,
- minimalizovat rozptýlení od současné práce způsobené nutností sledovat a vykazovat čas,
- minimalizovat dobu, za kterou se výkazy dostanou k odpovídajícím Redmine úlohám.

4.2 Návrh metrik

Za účelem zhodnocení přínosu navrhovaného řešení byly stanoveny následující metriky, které se přímo odvíjí od dílčích cílů navrhovaného řešení:

- Čas strávený vykazováním – poměr času stráveného sledováním a vykazováním odpracovaného času ku skutečně odpracovanému času

4. METRIKY PRO ZHODNOCENÍ PŘÍNOSU NAVRHOVANÉHO ŘEŠENÍ

- Úroveň rozptýlení – úroveň rozptýlení od současné práce způsobeného nutností sledovat a vykazovat čas
- Zpoždění výkazů – doba, za kterou se výkazy dostanou k odpovídající Redmine úloze po dokončení vykazované činnosti

Jelikož úroveň rozptýlení je složité měřit, byly stanoveny následující čtyři možnosti, ze kterých budou dotazovaní zaměstnanci vybírat dokončení věty „Sledování a vykazování času stráveného prací mě ...“:

- „vůbec nerozptyluje.“
- „mírně rozptyluje, ale neovlivňuje to moji efektivitu při práci.“
- „rozptyluje a ovlivňuje to moji efektivitu při práci.“
- „velice rozptyluje a výrazně to ovlivňuje moji efektivitu při práci.“

Metriky by měly být vyhodnoceny na reprezentativním vzorku zaměstnanců před a po nasazení navrženého řešení a následně porovnány. U všech definovaných metrik platí, že menší hodnoty jsou lepší. Jelikož se náplň běžného pracovního dne může u každého zaměstnance výrazně lišit, měly by být vždy porovnávány hodnoty pocházející od stejného zaměstnance, aby bylo dosaženo co nejpřesnějšího porovnání.

4.3 Výběr reprezentativního vzorku zaměstnanců

Jak již bylo zmíněno, průběh běžného pracovního dne se může u každého zaměstnance lišit. Aby byl pro vyhodnocení stanovených metrik vybrán reprezentativní vzorek zaměstnanců, musí vybraní jedinci pokrývat následující kategorie:

- aktivní uživatelé aplikace TTRebel, kteří používají pouze hlavní Redmine,
- aktivní uživatelé aplikace TTRebel, kteří používají hlavní Redmine i Redmine pro Support projekt,
- zaměstnanci, kteří využívají jiný nástroj pro sledování času,
- zaměstnanci, kteří nevyužívají žádný nástroj pro sledování času, ale čas vykazují,
- zaměstnanci, kteří čas strávený prací nijak nesledují a sami vykazují pouze výjimečně.

Analýza a zhodnocení UX aplikace TRebel

5.1 Vlastní zhodnocení UX

V rámci této sekce provede autor práce vlastní zhodnocení UX aktuální verze aplikace TRebel s produkčními daty. Je nutné podotknout, že autor práce je zároveň běžným uživatelem aplikace, proto je možné, že vlastní zhodnocení nemusí být ve všech ohledech zcela objektivní. Po dlouhodobém užívání aplikace je pravděpodobné, že autor práce snáze přehlédne nedostatky, kterých by si nezaujatý hodnotitel všiml.

Nejprve budou uvedeny pozitivní a negativní aspekty UX, které autor práce vnímá na základě svých zkušeností s aplikací. Poté bude uveden výstup zhodnocení použitelnosti UI pomocí heuristické evaluace, která by měla pomoci s objektivnějším přístupem.

5.1.1 Pozitivní aspekty UX

Aplikace TRebel je snadno použitelná a značně zjednodušuje a zpříjemňuje vykazování stráveného času do Redmine. Manuální měření času je pohodlné a převážně spolehlivé, uživatel může dokonce poznat podle favicony záložky v prohlížeči, že má spuštěné měření. Integrace s Redmine funguje dobře, pomocí našeptávače úloh lze většinou snadno nalézt hledanou úlohu a export tracking session do Redmine funguje také v pořádku.

Na exportní stránce autor práce oceňuje především souhrn naměřeného času, vykázaného času a součet obou časů po následném exportu tracking session. Možnost rychle a snadno upravit čas k exportu u každé tracking session autor práce používá poměrně často a je to o mnoho snazší, než upravovat jednotlivé intervaly.

Vzhled aplikace je až na pár detailů pěkný, komponenty Material designu aplikaci dodávají minimalistický a moderní vzhled. Většina aplikace je až na

pár detailů plně responzivní a na menších zařízeních je vcelku dobře použitelná ve „chvíli nouze“, kdy uživatel není u počítače a potřebuje nutně přepnout měření času.

5.1.2 Negativní aspekty UX

Karty s tracking sessions (jak na hlavní stránce, tak na stránce s exportem) zabírají hodně místa, proto se jich na stránku vejde málo a to je poměrně nešikovné, především při používání aplikace na mobilním telefonu. Co se týče karet s tracking sessions na stránce s exportem, kvůli jejich světle modrému pozadí, které mají, když jsou vybrány k exportu, nevypadá stránka s exportem dobře. Dále je v aplikaci několik prvků, které nejsou dobře responzivní, např. zástupné texty v polích pro přidání nového intervalu či pro vyhledávání Redmine úlohy.

Vyhledávání Redmine úloh umožňuje najednou zobrazit pouze pět výsledků, což při vyhledávání podle názvu občas nestačí a uživatel musí úlohu hledat podle identifikátoru nebo spíše častěji přímo v Redmine, protože identifikátor nezná. Občas se také stane, že uživatel hledá úlohu dle části názvu, našeptávač mu napoví méně než pět úloh, ale hledaná úloha mezi nimi není.

V aplikaci zcela chybí možnost vlastního přizpůsobení, které by uživateli mohlo zpříjemnit a zefektivnit používání aplikace. Přizpůsobení by mohlo obsahovat např. možnost nastavit výchozí aktivitu, seznam úloh, na kterých uživatel často pracuje nebo automatické spuštění měření času na právě přidání tracking session. Také zcela chybí možnost používat klávesové zkratky, které by uživateli ušetřili alespoň jedno kliknutí, např. pro inicializaci vytvoření nové tracking session.

S chybějící možností přizpůsobit si aplikaci tak, aby se měření tracking session spustilo ihned po jejím přidání, souvisí také to, že uživatel může současně měřit čas na více tracking sessions, což je pro podobné nástroje poměrně nezvyklé. Možná existují situace, kdy uživatel chce měřit čas na více tracking sessions současně, autor práce je však přesvědčen, že ve většině případů uživatelé chtějí spíše měřit čas pouze na jedné tracking session a pokud měří na více současně, tak pouze zapomněli zastavit starší měření. Pokud uživatel nevědomky měří více tracking sessions najednou, je později donucen měření opravit, což je zbytečná starost navíc. V ideálním případě by měla aplikace uživateli umožnit nastavit si, zda chce spuštěním měření jedné tracking session automaticky zastavit jiné měření či nikoliv – opět se tedy jedná o přizpůsobování aplikace jednotlivým uživatelům.

5.1.3 Heuristická evaluace

Výstup heuristické evaluace je prezentován v podobě seznamu jednotlivých heuristik (viz sekce 1.5.1.2) spolu s odhalenými nedostatky, které je porušují. U nedostatků je dále uvedena závažnost a u některých i navrhané řešení.

Nedostatky bez navrženého řešení jsou komplexnějšího charakteru a jejich náprava by vyžadovala hlubší analýzu a pravděpodobně širší úpravy uživatelského rozhraní. Závažnost je udávána dle Nielsenovy škály, viz sekce 1.5.1.1.

5.1.3.1 Výstup heuristické evaluace

1. Viditelnost stavu systému

1.1. Vytvoření nové tracking session je nepatrné, nová tracking session je přidána na začátek seznamu bez jakékoliv animace. V dolní části stránky se sice zobrazí notifikace informující o vytvoření tracking session, ta je však snadno přehlédnutelná, především na mobilním telefonu. Uživatel se tedy musí po vytvoření ujistit, že byla tracking session vytvořena.

- *Závažnost:* 2
- *Řešení:* doprovázet vytvoření tracking session vhodnou animací nebo tracking session na krátký čas nějakým způsobem zvýraznit

1.2. Naměřený čas za konkrétní den není vidět na hlavní stránce, ale pouze na stránce s exportem. Pokud uživatel chce pouze odpracovat určitý čas daný den, musí buď naměřený čas sečíst sám nebo jej zkontrolovat na stránce s exportem.

- *Závažnost:* 3
- *Řešení:* přidat na hlavní stránku součet naměřeného času

1.3. V případě, že má uživatel více tracking session, než se vejde na stránku, a posune se níže na stránku tak, že aktuálně měřená tracking session není na obrazovce, informace o aktuálním měření je ztracena.

- *Závažnost:* 2
- *Řešení:* ukotvit informace o aktuálním měření na hlavní stránce tak, aby byly vždy viditelné

2. Shoda mezi systémem a skutečným světem

2.1. Význam výrazu „tracking session“ v nemusí být českým uživatelům zřejmý.

- *Závažnost:* 2
- *Řešení:* zvolit vhodnější český výraz, např. „měření“

3. Uživatelská kontrola a svoboda

3.1. Chybí možnost zvrátit odstranění tracking session.

- *Závažnost:* 3

- *Řešení*: přidat standardní snackbar² s možností zvrátit odstranění

3.2. Chybí možnost zvrátit odstranění časového intervalu.

- *Závažnost*: 3
- *Řešení*: přidat standardní snackbar³ s možností zvrátit odstranění

3.3. V případě anonymní tracking session je pole s aktivitou deaktivováno a uživateli není prezentován důvod.

- *Závažnost*: 3
- *Řešení*: přidat na pole s aktivitou tooltip s vysvětlením

4. Konsistence a standardy

4.1. Přidání tracking session se iniciuje přes autocomplete seznam, který běžně neimplikuje vytvoření nového záznamu.

- *Závažnost*: 2
- *Řešení*: použít běžné tlačítko pro vytvoření nového záznamu, např. floating action button⁴

4.2. Tracking sessions na stránce s exportem, které nejsou vybrané k exportu, vypadají téměř identicky jako tracking sessions na hlavní stránce, uživatel je dokonce může i upravovat běžným způsobem. Uživateli se tedy může zdát, že tracking sessions s tímto vzhledem představují „běžný stav“, kdežto na stránce s exportem chce uživatel tracking sessions převážně exportovat, takže by „běžnému stavu“ měly odpovídat spíše tracking sessions vybrané k exportu. Vzhled nevybraných tracking session sice je kontrastní ke vzhledu vybraných tracking session, ten však také není ideální, viz nedostatky 8.4.

- *Závažnost*: 2

5. Prevence chyb

5.1. Pole pro zadání počátečního a koncového času při vytváření časového intervalu nijak neindikují formát času, který má uživatel zadat. Uživatel může správný formát poznat pouze podle toho, že zmizí validační hláška nebo podle již existujících intervalů.

- *Závažnost*: 3

²Komponenta běžně používaná v Material designu pro zobrazování informací uživateli či nabízení okamžitých akcí, viz [53].

³Viz předchozí poznámku.

⁴Komponenta běžně používaná v Material designu pro primární akce, jako např. vytvoření nového záznamu, viz [54].

- *Řešení*: nastavit pole tak, aby uživateli stačilo pouze doplňovat číslice a vstup se vhodně naformátoval automaticky

6. Rozpoznání raději než vzpomínání

6.1. Viz nedostatek 1.2

6.2. V případě, že má v rámci jednoho dne uživatel více tracking session, než se vejde na stránku, může se mu snadno stát, že se na stránce posune dostatečně nízko na to, aby již nebylo viditelné datum, do kterého tracking sessions patří.

- *Závažnost*: 3
- *Řešení*: umístit datum do ukotvené lišty tak, aby bylo vždy viditelné

7. Flexibilní a efektivní použití

7.1. Systém nenabízí možnost iniciovat vytvoření nové tracking session pomocí klávesové zkratky.

- *Závažnost*: 3
- *Řešení*: přidat klávesovou zkratku, např. na klávesu „+“

7.2. Systém nenabízí možnost zastavit spuštěnou tracking session pomocí klávesové zkratky.

- *Závažnost*: 2 (uživatelé tuto možnost pravděpodobně nebudou potřebovat moc často)
- *Řešení*: přidat klávesovou zkratku

8. Estetický a minimalistický design

8.1. Tracking sessions zabírají mnoho místa a důsledkem toho se jich na stránku vejde málo, především na mobilním telefonu.

- *Závažnost*: 3

8.2. Součástí časových intervalů je počáteční a koncové datum. Jelikož uživatel nejčastěji měří čas v rámci jednoho dne, tyto datумы jsou v převážné většině případů stejné a tudíž redundantní.

- *Závažnost*: 1

8.3. Tracking session je možné vytvořit i na stránce s exportem. Stránku s exportem by uživatelé měli navštěvovat jen když si chtějí exportovat tracking sessions, tato možnost je tedy redundantní.

- *Závažnost*: 2
- *Řešení*: odstranit možnost vytvoření tracking session na stránce s exportem

8.4. Tracking sessions vybrané k exportu na stránce s exportem mají světle modré pozadí. Když uživatel přijde na stránku s exportem, ve výchozím stavu jsou všechny tracking sessions vybrané k exportu (a většinou pravděpodobně zůstanou vybrané až do exportu), světle modrá barva pak zabírá podstatnou část stránky, především na mobilním telefonu. Tato barva je velice výrazná a s ohledem na barevnou paletu uživatelského rozhraní je její použití v takovém měřítku nepřírozané.

- *Závažnost:* 1
- *Řešení:* bylo by vhodné vybrat jiný způsob, jak odlišit vybrané tracking sessions od nevybraných

9. Pomoc uživateli rozpoznat a pochopit chybu a vzpamatovat se z ní

9.1. Pokud uživatel upraví časové intervaly tak, že se vzájemně překrývají, u jednoho intervalu se objeví ikona varování, která uživatele o této skutečnosti obeznámí. Uživatel však musí sám zjistit, s jakým intervalem se označený interval překrývá.

- *Závažnost:* 1 (k této situaci pravděpodobně dochází velice zřídka)
- *Řešení:* zobrazit varování u všech intervalů, které se navzájem překrývají

10. Návod a dokumentace

10.1. Systém neobsahuje žádnou sekci s návodem či návody k použití.

- *Závažnost:* 1 (aplikace je jednoduchá k pochopení)

5.2 Analýza běžného užívání aplikace s uživateli

Aplikace TTRebel má šest aktivních uživatelů a z neformálních rozhovorů vyplynulo, že ne všichni užívají aplikaci stejným způsobem. Na této skutečnosti nemusí být ve své podstatě nic špatného, je však možné, že způsob užívání aplikace některými uživateli je důsledkem nedostatků UX, které si uživatelé nemusí nutně uvědomovat. Případné nedostatky by bylo vhodné vyřešit, aby se všem uživatelům s aplikací dobře pracovalo.

5.2.1 Výzkumné otázky

Ve spolupráci s vývojovým týmem aplikace TTRebel bylo sestaveno několik výzkumných otázek ohledně užívání aplikace, které se bude práce snažit zodpovědět dále v této kapitole. Některé z těchto otázek jsou založeny na neověřených teoriích vývojového týmu či autora práce, jiné mají pomoci ucelit si představu o různých způsobech užívání aplikace.

1. Používají uživatelé aplikaci ke sledování času stráveného prací na úlohách z Redmine pro Support projekt?
2. Když uživatelé během dne pracují na jedné konkrétní úloze s časovými odstupy, měří si strávený čas vždy na stejné tracking session nebo zakládají nové?
3. Používají uživatelé možnost upravit tracking session na stránce s exportem?
4. Používají uživatelé na stránce s exportem možnost neexportovat konkrétní tracking session odškrtnutím checkboxu? Pokud ano, v jakých případech?
5. Kolik času uživatel stráví na stránce s exportem v porovnání s hlavní stránkou?
6. Navštěvují uživatelé stránku s exportem za jinými účely než exportovat tracking sessions?
7. Vytváří si uživatelé tracking sessions v předstihu (např. na začátku pracovního dne) nebo průběžně během dne?
8. Jak často vytváří uživatelé novou tracking session s konkrétní Redmine úlohou v porovnání s anonymní tracking session (bez Redmine úlohy)?
9. Využívají uživatelé možnost manuálně přidat nový časový interval?

5.2.2 Automatizovaný sběr informací o chování uživatelů

Za účelem co nejpřesnějšího zodpovězení některých výzkumných otázek byl do aplikace TTRebel implementován sběr informací o chování uživatelů pomocí nástrojů služby Google Analytics. Tato služba byla vybrána z následujících důvodů:

- podpora sběru vlastních událostí s libovolnými parametry,
- snadné zakomponování do aplikace,
- možnost neplaceného použití služby,
- propracované uživatelské rozhraní, které nabízí různé statistiky a přehledy o událostech a uživateli.

Zakomponování Google Analytics do aplikace vyžadovalo úpravy pouze ve frontendu. Pro účely této práce není podstatné implementaci podrobně rozebírat, bude tedy shrnuta do následujících kroků:

1. přidání knihovny „analytics.js“ [55] jako závislosti do projektu,

2. definice vlastních událostí, které mají být sbírány,
3. implementace sběru jednotlivých událostí na vhodných místech.

5.2.2.1 Sbírané události

Odpovídající událost spolu s doplňujícími parametry je odeslána do Google Analytics vždy, když uživatel:

- změní stav své aktivity (uživatel je považován za neaktivního, pokud 60 sekund nijak neinteraguje s aplikací),
- přejde na jinou stránku v aplikaci,
- manuálně vytvoří časový interval,
- vytvoří tracking session s konkrétní Redmine úlohou,
- vytvoří tracking session bez Redmine úlohy,
- označí či odznačí tracking session pro export na stránce s exportem,
- modifikuje časový interval tracking session na stránce s exportem,
- modifikuje Redmine úlohu tracking session na stránce s exportem,
- modifikuje popis práce tracking session na stránce s exportem,
- modifikuje aktivitu tracking session na stránce s exportem,
- exportuje tracking sessions (jako parametr je počet exportovaných tracking session).

Každá událost navíc také obsahuje identifikátor uživatele a stránku aplikace, na které se uživatel v době vyvolání události nacházel.

5.2.2.2 Průběh sbírání informací

Události uvedené v předchozí sekci byly se souhlasem všech uživatelů aplikace sbírány po dobu 12 pracovních dní. Služba Google Analytics byla dále integrována s datovým skladem Google BigQuery [56], do kterého byly sbírané události automaticky jednou denně exportovány. Google BigQuery podporuje dotazování pomocí standardního SQL, díky čemuž mohla být nasbíraná data analyzována podrobněji, než by to bylo možné v uživatelském rozhraní služby Google Analytics.

5.2.2.3 Analýza nasbíraných dat

Z analýzy nasbíraných dat vzešly následující poznatky:

- Úpravu údajů tracking session na exportní stránce uživatelé využívají, i když někteří více, než jiní.
- Možnost neexportovat konkrétní tracking session používají uživatelé minimálně.
- Uživatelé průměrně stráví na hlavní stránce aplikace přibližně šestinásobně více času než na stránce s exportem.
- Většina uživatelů častěji vytváří tracking session jako anonymní, pouze u několika málo jedinců převládá vytváření tracking session s konkrétní Redmine úlohou.
- Většina uživatelů vytváří časové intervaly manuálně v průměru třikrát denně.

5.2.3 Rozhovory s uživateli

Z nasbíraných dat nebylo možné zodpovědět všechny výzkumné otázky, proto byly se všemi uživateli udělány rozhovory na téma používání aplikace. I přes to, že rozhovory původně měly pevně danou strukturu, autor si kladl za cíl využít rozhovory k hlubšímu porozumění chování různých uživatelů a proto struktura v některých případech nebyla striktně dodržena a sloužila spíš jako osnova témat k diskusi.

Díky rozhovorům s uživateli bylo možné potvrdit některé závěry vyvozené při analýze automaticky nasbíraných dat a zodpovědět zbylé výzkumné otázky. Následujících několik bodů shrnuje poznatky relevantní k zodpovězení výzkumných otázek:

- Uživatelé používají TTRebel i k měření času stráveného na úlohách ze Support Redmine, i když není s aplikací integrovaný. Běžně si založí anonymní tracking session, do jejíhož popisku vyplní název či identifikátor úlohy ze Support Redmine a po dopracování na úloze ji většinou okamžitě ručně přenesou do Support Redmine.
- Všichni uživatelé se vyjádřili, že i v případě, kdy přes den pracují na té samé úloze vícekrát s nějakým časovým odstupem, veškerý čas měří na jediné tracking session s danou úlohou a do popisu práce uvádí různé činnosti, kterým se věnovali. Jedinou výjimkou jsou případy, kdy uživatelé pracují na Support projektu, jelikož se většinou snaží vykázat si k úlohám hned, a když se k úloze později musí vrátit, vytvoří si novou tracking session a další práci si vykáží zvlášť.

- Většina uživatelů doplňuje potřebné údaje ke všem tracking sessions již na hlavní stránce, aby mohli na stránce s exportem pouze zkontrolovat a případně upravit čas k exportu a následně exportovat. Někteří uživatelé využívají validaci na stránce s exportem k tomu, aby je upozornila na tracking sessions, u kterých jim nějaký údaj chybí a podle toho jej doplní, to však dělají převážně v případě, kdy mají větší množství tracking session a nechtějí je všechny kontrolovat bez pomoci.
- Uživatelé používají možnost neexportovat konkrétní tracking session pouze minimálně. Uživatelé, kteří častěji pracují na stejných úlohách pravidelně (např. zpracovávání podkladů pro faktury na konci měsíce), si běžně tyto úlohy připraví v aplikaci TTRebel na začátku dne a pokud se během dne nedostanou ke všem úlohám, při exportu tyto úlohy neexportují, aby jim v aplikaci zůstaly na další den jako připomenutí. Tuto možnost také využívají uživatelé pracující na Support projektu, kteří mají vytvořené tracking sessions pro úlohy ze Support Redmine, ale exportovat je nemohou, proto je při exportu odznačí. Jeden uživatel se také vyjádřil, že tuto možnost občas používá ve chvíli, kdy chce během dne exportovat pouze některé tracking sessions a u jiných ještě nemá doplněné všechny údaje.
- Polovina uživatelů se vyjádřila, že občas navštěvují stránku s exportem čistě za účelem zjistit naměřený čas, aby měli přehled o tom, kolik času již ten den napracovali.
- Většina uživatelů se vyjádřila, že i když na začátku pracovního dne vědí, jakým úlohám by se měli přes den věnovat, tracking session v aplikaci vytváří až ve chvíli, kdy se odpovídající úloze začínají věnovat. Pouze uživatelé, kteří pravidelně pracují na stejných úlohách (např. zpracovávání podkladů pro faktury na konci měsíce), si běžně tracking sessions s těmito úlohami připraví již na začátku dne. Tracking sessions pro ně prý potom jistým způsobem představují seznam úkolů.
- Většina uživatelů se vyjádřila, že častěji vytváří anonymní tracking session (bez konkrétní Redmine úlohy) v případě, kdy si nepamatují či neznají název či identifikátor Redmine úlohy. Pokud název či identifikátor úlohy znají, snaží se ji vyhledat již při zakládání tracking session, ale když se jim to z nějakého důvodu nepodaří, většinou spíše založí anonymní tracking session a dohledávání úlohy nechají na později.
- Všichni uživatelé se vyjádřili, že přidávají časové intervaly manuálně (někteří méně často než jiní). Často však tuto možnost využívají pouze k navýšení naměřeného času, např. o 10 minut, spíše než k doplnění konkrétního časového intervalu. Dva uživatelé se také vyjádřili, že místo manuálního přidávání časového intervalu raději spustí a ihned zastaví

měření času, čímž se vytvoří nový časový interval, kterému pak pouze upraví počáteční čas a ušetří si zadávání koncového času.

5.2.4 Nedostatky UX vyvozené z různých vzorců chování uživatelů

Jak je patrné z poznatků výše, rozdíly mezi vzorci chování jednotlivých uživatelů nejsou tak veliké, jak se původně předpokládalo. Největší rozdíly v chování, které můžeme pozorovat, závisí na tom, zda uživatel pracuje na Support projektu či nikoliv. Tyto rozdíly by však teoreticky neměly být přítomné, kdyby Redmine pro Support projekt byl integrovaný s aplikací TTRebel, což bude pravděpodobně zásadní nedostatek aplikace pro některé uživatele.

Na základě vyjádření některých uživatelů, že občas navštěvují stránku s exportem čistě za účelem zjistit naměřený čas, lze usoudit, že tento souhrn by se měl nacházet také na hlavní stránce nebo v nějakém samostatném modálním okně, které by uživatelé mohli otevřít z hlavní stránky aplikace.

Spuštění a následné zastavení měření času na tracking session za účelem vytvoření časového intervalu svědčí o tom, že manuální vytváření časových intervalů některým uživatelům nevyhovuje.

5.3 Zhodnocení UX aplikace se zaměstnanci

Se všemi zaměstnanci firmy byly provedeny hloubkové rozhovory, v rámci kterých se zaměstnanci vyjádřili k pozitivním i negativním aspektům UX aplikace TTRebel. V této sekci jsou dále uvedeny pouze poznatky, na kterých se shodlo více zaměstnanců nebo nejsou čistě subjektivní (jako by např. mohl být negativní názor na odstín zelené barvy použité v aplikaci).

5.3.1 Pozitivní aspekty UX

Za největší přínos aplikace TTRebel by většina zaměstnanců označila integraci s Redmine, která výrazně usnadňuje vyhledávání úloh a vytváření výkazů. Také manuální měření času hodnotí většina zaměstnanců pozitivně a ovládání měření času v aplikaci jim je pohodlné.

Další pozitivní názory pochází především od dlouhodobých uživatelů. Většina uživatelů je spokojená s celkovým vzhledem uživatelského rozhraní a oceňují především uživatelsky přívětivou „inline“ editaci tracking session, která úpravy ukládá průběžně a nevyžaduje potvrzování formuláře. Mezi další klíčové funkcionality aplikace uživatelé zařadili možnost vytvářet anonymní tracking session a shrnutí naměřeného a vykázaného času na stránce s exportem.

5.3.2 Negativní aspekty UX

Většina zaměstnanců se shodla, že tracking sessions v uživatelském rozhraní zabírají zbytečně moc místa a z toho důvodu se jich na stránku vejde poměrně málo (jak na hlavní stránce aplikace, tak na stránce s exportem). Několik málo uživatelů se také vyjádřilo, že se jim nelíbí vzhled stránky s exportem a že oddělení dnů na hlavní stránce často snadno přehlédnou.

Co se týče použitelnosti aplikace, několik zaměstnanců se shodlo v názoru, že aplikace nelze pohodlně ovládat klávesnicí – zcela chybí možnost používat klávesové zkratky a většinu formulářů nelze potvrdit klávesou enter, jak je na počítači již v dnešní době zvykem. Několik málo zaměstnanců označilo vytváření nových časových intervalů za nepohodlné, především kvůli nutnosti ručně zadávat čas v požadovaném formátu a nutnosti dávat pozor na datum u tracking session vytvořených v některém z předchozích dní.

Několik uživatelů zmínilo negativní aspekty našeptávače Redmine úloh, pomocí kterého se vytvářejí nové tracking sessions a přiřazují úlohy k existujícím tracking sessions. Některým uživatelům se prý často stává, že nenajdou požadovanou úlohu v seznamu navržených úloh, jelikož se hledaná část názvu úlohy opakuje u více úloh a konkrétní hledaná úloha se na seznam již nevešla. Jeden uživatel se také vyjádřil, že by našeptávač měl být při vyhledávání tolerantnější k překlepům a nepřesnostem v názvu úlohy.

Několika zaměstnancům nevyhovuje, že má aplikace pouze webového klienta, jelikož to od nich vyžaduje, aby měli přehled o tom, kde se v jejich webovém prohlížeči právě nachází záložka s aplikací. To, že se záložka s aplikací uživateli „ztratí“ mezi ostatními záložkami, kterých mají někteří zaměstnanci ve svých webových prohlížečích velké množství, může mít za následek i to, že uživatelé zapomenou aktualizovat měření, jelikož nemají aplikaci dostatečně na očích.

I když je aplikace až na pár detailů plně responzivní, na mobilních telefonech ji uživatelé nepoužívají moc rádi a uvítali by, kdyby bylo uživatelské rozhraní lépe přizpůsobené menším obrazovkám. Jeden zaměstnanec se vyjádřil, že by aplikaci na mobilu používal pouze v nativní podobě. Další nepříjemností při používání aplikace na mobilním či jiném dalším zařízení je dle jednoho uživatele fakt, že data nejsou synchronizována napříč instancemi aplikace – např. pokud má uživatel aplikaci otevřenou ve webovém prohlížeči na počítači i na mobilním telefonu a na počítači spustí měření času u nějaké tracking session, na mobilním telefonu se tato akce nijak neprojeví, dokud uživatel neobnoví stránku s aplikací.

Souhrn naměřeného a již vykázaného času, jenž se nachází na stránce s exportem, většina zaměstnanců chválí, ale chtěli by jej i na hlavní stránce aplikace. Dále několika zaměstnancům chybí nějaká vizualizace časových intervalů, ve které by mohli na první pohled identifikovat chybějící naměřenou činnost a celkově si lépe představit, jak jejich naměřené časové intervaly pokrývají celý pracovní den.

Dalším klíčovým nedostatkem, na kterém se shoduje několik zaměstnanců, je absence jakéhokoli dohlížení na chování uživatele. Některým zaměstnancům se již několikrát stalo, že zapomněli zastavit měření času a po několika dnech zjistili, že mají naměřené desítky hodin a nedokázali zpětně určit, kdy s činností skutečně přestali. Opačným a také častým případem je, když uživatel zapomene spustit měření času a není schopný zpětně určit, kdy s činností začal. V obou zmíněných případech by prý uživatelé uvítali nějakou formu upozornění od aplikace.

Absence integrace aplikace TTRebel se systémem Redmine pro Support projekt je posledním důležitým nedostatkem aplikace dle zaměstnanců, kteří na Support projektu pracují.

5.4 Shrnutí klíčových nedostatků UX

V této sekci budou stručně shrnuty klíčové nedostatky UX aplikace převzaté z vlastní analýzy UX autorem práce a analýzy UX se zaměstnanci firmy. Pro zestručnění seznamu jsou zde zahrnuty pouze takové nedostatky odhalené během heuristické evaluace, které byly ohodnoceny závažností 3 a výše.

Byly identifikovány následující klíčové nedostatky UX aplikace TTRebel:

- karty s tracking sessions zabírají zbytečně moc místa a na stránku se jich vejde moc málo,
- exportní stránka je kvůli rozměrům a modrému pozadí tracking session vybraných k exportu nevzhledná,
- oddělení dnů na hlavní stránce je snadno přehlédnutelné, samotné datum není nijak ukotvené, uživatel tedy může snadno ztratit přehled o datu, kdy byly vytvořeny tracking sessions, které má právě na obrazovce,
- vytváření nových intervalů je nepohodlné kvůli nutnosti zadat ručně čas v požadovaném formátu, který uživatel musí „odkoukat“ od existujících intervalů,
- pole s aktivitou je v některých případech deaktivováno a uživateli není prezentován důvod,
- chybí možnost zvrátit odstranění tracking session či časového intervalu,
- našeptávač Redmine úloh zobrazuje málo výsledků a netoleruje překlepy,
- souhrn naměřeného času a již vykázaného času se nachází pouze na stránce s exportem, na hlavní stránce chybí,
- aplikace nenabízí žádnou vizualizaci časových intervalů,
- aplikace má pouze webového klienta,

- i přes téměř kompletní responsivitu uživatelského rozhraní je používání aplikace na mobilních telefonech nepohodlné,
- data aplikace nejsou synchronizována napříč zařízeními, na kterých je aplikace používána,
- aplikace nedohlíží na chování uživatele – umožňuje mu měřit čas po nesmyslně dlouhou dobu a naopak se uživateli nijak nepřipomíná, takže je pro uživatele snazší zapomenout měřit si čas,
- chybí integrace se systémem Redmine pro Support projekt,
- práci s aplikací není možné zefektivnit používáním klávesových zkratk,
- chybí možnost jakéhokoliv vlastního přizpůsobení aplikace.

5.5 Obecné problémy zaměstnanců se sledováním a vykazováním odpracovaného času

Z hloubkových rozhovorů se zaměstnanci vyplynulo, že přesné a průběžné sledování času stráveného na konkrétních úlohách je problematické především pro zaměstnance, kteří musí během dne často přepínat kontext své práce. Jedná se většinou o seniorní vývojáře zapojené do více projektů, kteří přes den pracují na několika různých úlohách, pomáhají ostatním vývojářům s jejich úkoly, konzultují detaily se zákazníky apod. Pro tyto zaměstnance je náročné se dostatečně efektivně věnovat své práci a mezi tím ještě myslet na průběžné sledování odpracovaného času, často na to tedy buď zapomenou nebo to odloží na později, až budou mít volnou chvíli.

Avšak jak popisuje [57], pokud zaměstnanci nesledují svůj odpracovaný čas průběžně, tak čím později se dostanou k vykazování, tím náročnější (časově i psychicky) pro ně bude vykázat si strávený čas s dostatečnou přesností. Nutnost pamatovat si co nejvíce detailů o času stráveném na různých úlohách může navíc ovlivnit efektivitu zaměstnanců při práci. Tento jev také potvrzují hloubkové rozhovory, ve kterých se několik zaměstnanců vyjádřilo, že vykazování práce z několika předchozích dnů jim vždy zabere nezanedbatelné množství času. Z toho důvodu zaměstnanci často vykazování oddalují, čímž se problém bohužel ještě více prohlubuje. O něco lépe jsou na tom uživatelé aplikace TTRebel, avšak i těm vykazování zabere nezanedbatelné množství času, především když jejich tracking sessions neobsahují dostatečně podrobné informace, které musí zpětně dohledávat.

Návrh řešení pro sledování a vykazování času

6.1 Požadavky na navrhované řešení

Cíle, kterých by mělo být navrhovaným řešením dosaženo, již byly stanoveny v sekci 4.1, nyní budou následovat konkrétnější požadavky na navrhované řešení. Tyto požadavky vychází z hloubkových rozhovorů se všemi zaměstnanci a již stanovených předpokladů, např. že zaměstnanci pracují převážně na počítači, jsou počítačově gramotní, vykazují k Redmine úlohám apod. V rozhovorech autor práce od zaměstnanců zjišťoval, jak vypadá jejich běžný pracovní den, jaký je jejich přístup k sledování času a vykazování a jak si představují ideální řešení, které by jim v tomto ohledu pomáhalo.

Obecné požadavky na navrhované řešení jsou následující:

- Řešení bude uživateli pomáhat sledovat a následně vykazovat čas strávený prací na konkrétních úlohách, které mohou pocházet z více různých instancí Redmine.
- Řešení bude uživateli pomáhat udržovat si přehled o tom, kdy a na čem pracoval.
- Řešení bude uživatele motivovat k pravidelnému a včasnému vykazování.
- Řešení bude uživateli dostatečně na očích, aby nezapomínal na sledování času stráveného prací.
- Řešení bude pomáhat uživateli s přiřazováním odpracovaného času ke konkrétním Redmine úlohám.
- Řešení bude uživateli nabízet možnost přizpůsobení, aby se mu s ním pracovalo dostatečně efektivně a pohodlně.

- Řešení bude uživateli umožňovat snadno a rychle si poznamenat alespoň minimální detaily o tom, na čem právě pracuje – např. v situacích, kdy uživatel obdrží nečekaný telefonát.
- Řešení dokáže automaticky či s drobnými manuálními zásahy alespoň hrubě rekonstruovat průběh pracovního dne uživatele na úrovni času stráveného prací na konkrétních Redmine úlohách či projektech nebo alespoň na úrovni času stráveného kategorizovanými činnostmi.

Za zmínku stojí analogie, kterou jeden dotázaný zaměstnanec (zároveň člen vývojového týmu aplikace TTRebel) popsal ideální řešení pro sledování a vykazování času. Podle něj by řešení mělo fungovat jako osobní asistent, který má na starosti sledovat, kdy a na čem zaměstnanec pracuje a následně vytvářet odpovídající výkazy. Za tímto účelem by asistent čas od času vyžadoval od uživatele alespoň minimální interakci. Asistent by měl být např. dostatečně inteligentní na to, aby mohl uživateli pomoci přiřadit jeho činnost ke konkrétní úloze. Asistent by také měl uživatele jistým způsobem hlídat a upozorňovat jej např. na to, že sice na něčem pracuje, ale asistent neví na čem a nemůže tak měřit čas. Díky takovému asistentovi by se mohl uživatel přes den dostatečně soustředit na svou práci, na konci dne by s asistentem pouze v rychlosti zkontroloval záznamy a asistent by za něj vytvořil odpovídající výkazy.

6.2 Architektura navrhovaného řešení

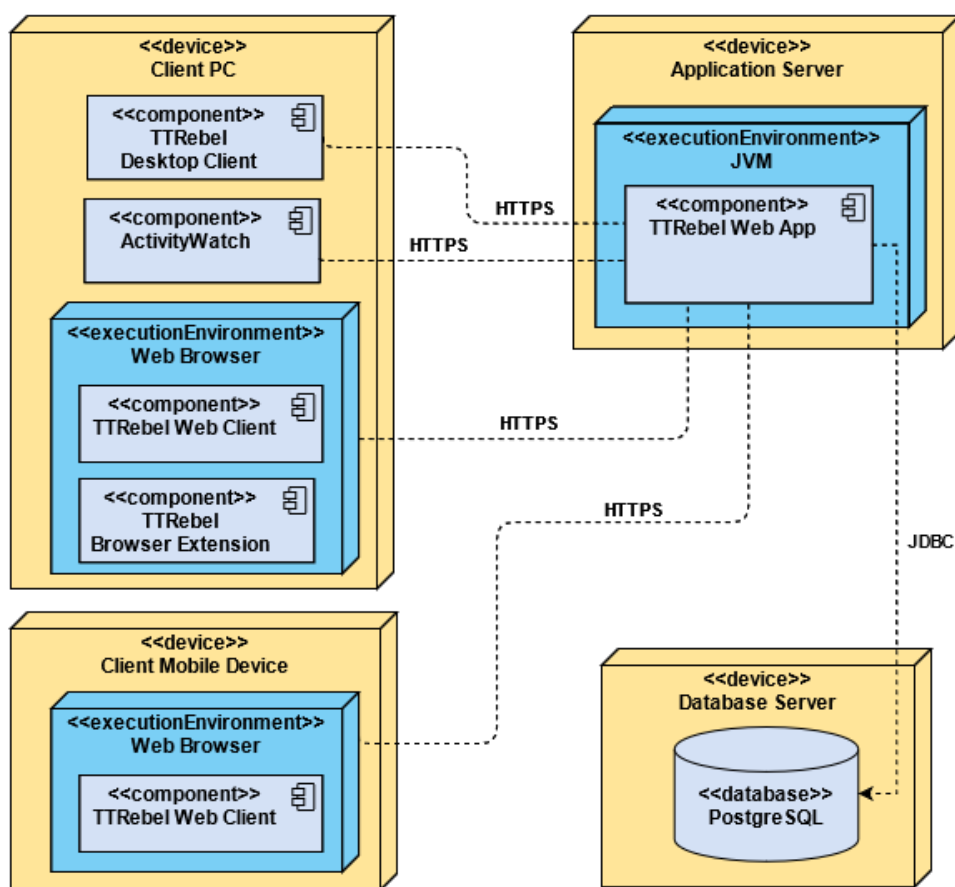
Aby bylo možné pokrýt co nejvíce vytyčených požadavků, architektura navrženého řešení se skládá z několika komponent:

- webová aplikace TTRebel,
- desktopový klient aplikace TTRebel,
- rozšíření prohlížeče,
- ActivityWatch.

Webová aplikace TTRebel je hlavní komponentou celého řešení, se kterou by si měla vystačit většina uživatelů. Ostatní komponenty slouží převážně k zefektivnění a zpříjemnění sledování odpracovaného času. Autor práce neočekává, že by všichni zaměstnanci aktivně využívali všechny komponenty řešení a je to tak v pořádku – každému zaměstnanci vyhovuje něco trochu jiného, pracovní den každého zaměstnance probíhá trochu jinak. Navržené řešení jako celek se snaží pokrýt požadavky všech zaměstnanců tak, aby každému dokázalo zpříjemnit neoblíbenou, avšak nutnou činnost, kterou vykazování odpracovaného času je.

Návrhu UX jednotlivých komponent se dále podrobněji věnují sekce 6.3, 6.4, 6.5 a 6.6. Všechny grafické návrhy uživatelských rozhraní byly vytvořeny pomocí nástroje Pencil a balíčků s komponentami Material designu, díky kterým mají návrhy vysokou úroveň věrnosti, jelikož se předpokládá následování Material designu i při implementaci výsledných aplikací. Veškerá data použitá v návrzích jsou fiktivní, byla však vybrána tak, aby dobře reprezentovala skutečná data co se týče délky a formátu vstupů, časových údajů apod.

Nyní následuje krátký popis účelu každé komponenty.



Obrázek 6.1: Diagram nasazení komponent navrženého řešení

Webová aplikace TTRebel

S nově navrženým UX, které se bude snažit vyřešit identifikované nedostatky a zároveň naplnit požadavky uživatelů (viz sekce 6.3.1), bude webová aplikace TTRebel svým uživatelům lépe sloužit k manuálnímu měření a vykazování času stráveného na Redmine úlohách. Webová aplikace na rozdíl od ostatních komponent nevyžaduje instalaci a bude tak pro uživatele snadno dostupná a

použitelná v každém moderním prohlížeči, jak na počítači, tak na mobilním telefonu či tabletu.

Desktopový klient aplikace TTRebel

Desktopový klient aplikace TTRebel bude samostatná aplikace s omezenou nabídkou funkcionalit, jejímž hlavním účelem bude poskytnout uživateli možnost snadno a rychle ovládat měření času a vytvářet nové tracking sessions. Uživatelé budou moci aplikaci vyvolat pomocí systémových klávesových zkratk a UI aplikace bude primárně přizpůsobeno efektivnímu ovládní pouze pomocí klávesnice.

Rozšíření prohlížeče

Jednoduché rozšíření prohlížeče, které bude na vybrané webové stránky přidávat tlačítko, které vyvolá nabídku pro zadání údajů, se kterými bude po potvrzení nabídky vytvořena nová tracking session v aplikaci TTRebel. V závislosti na konkrétní webové stránce bude nabídka obsahovat již předvyplněné údaje, které bude možné z obsahu stránky vyvodit, jako je např. konkrétní Redmine úloha nebo popis identifikující interakci s danou stránkou. Smyslem těchto tlačítek je usnadnit uživateli vytvoření nové tracking session alespoň s informacemi, díky kterým bude uživatel schopen zpětně určit, na čem pracoval. Tlačítka na vybraných stránkách budou také uživateli připomínat, že by si měl měřit čas, který stráví prací spojenou s danou stránkou, jako je např. diskuze v komentářích u Redmine úlohy, revize kódu v GitLabu nebo úprava obsahu Google dokumentu.

ActivityWatch

Nástroj ActivityWatch bude sbírat informace o uživatelově aktivitě na počítači z různých zdrojů a ve svém uživatelském rozhraní bude nabízet podrobný přehled o aktivitě na časové ose spolu s dalšími statistikami. Takový přehled poslouží uživatelům především ve chvílích, kdy si chtějí vykázat, ale tracking sessions v aplikaci TTRebel nepokrývají požadovaný počet hodin a uživatelé potřebují zpětně dohledat informace o tom, na čem daný den pracovali a jak dlouho.

6.3 Webová aplikace TTRebel

6.3.1 Požadavky na úpravy aplikace

Shromažďování požadavků na úpravy aplikace TTRebel bylo součástí hloubkových rozhovorů se zaměstnanci, kteří se vyjadřovali k tomu, co za funkcionality jim v aplikaci chybí a co by aplikace měla umět, aby jim co nejvíce usnadnila

sledování a vykazování času. Některé požadavky také vzešly z identifikovaných nedostatků UX aplikace a již v minulosti vyžádaných úprav, které byly pro účely této práce poskytnuty vývojovým týmem aplikace. Vyřešení identifikovaných klíčových nedostatků UX aplikace se předpokládá a v seznamu požadavků není uvedeno.

Funkční požadavky

- Omezit počet současně spuštěných tracking sessions na jednu.
- Možnost zvolit si, zda se při vytvoření nové tracking session má okamžitě spustit měření či nikoliv.
- Možnost nastavit si preferovanou výchozí aktivitu.
- Možnost spravovat seznam uživatelových oblíbených úloh, ke kterým často vykazuje.
- Možnost rychlého spuštění měření práce na oblíbené úloze.
- Možnost nechat si vypsat úlohy, ke kterým si uživatel měřil čas v nedávné době (např. max. měsíc zpět), za účelem vytvoření nové tracking session s danou úlohou pro aktuální den.
- Možnost rychlého vyhledání Redmine úloh přiřazených na uživatele.
- Možnost navýšit naměřený čas u tracking session (např. o pět minut) bez nutnosti vytváření či úpravy časových intervalů.
- Vizualizace naměřeného času u každého dne na hlavní stránce, ze které uživatel rychle pozná, kolik času má naměřeno daný den a kolik mu ještě zbývá k odpracování požadované doby.
- Vizualizace naměřených časových intervalů pro daný den na exportní stránce s možností rychlého přidání tracking session pokrývající mezeru mezi intervaly.
- Přehled vykázaného času za aktuální měsíc či týden. Volitelně i s časem zbývajícím k odpracování.
- Upozornování uživatele na to, že má pravděpodobně omylem spuštěné měření času.
- Upozornování uživatele na to, že během pracovní doby nemá zapnuté měření času.
- Dotazování se uživatele na to, zda se stále věnuje měřené práci v případě, že je měření spuštěné bez přerušení nezvykle dlouho.

- Možnost deaktivovat a volitelně podrobněji nastavovat zmíněná upozornění.
- Zobrazení jména a fotografie přihlášeného uživatele na hlavní stránce (informace převzaty z Redmine).
- Možnost iniciovat vytvoření nové tracking session pomocí klávesové zkratky. Volitelně i možnost vlastního nastavení odpovídající klávesové zkratky.

Nefunkční požadavky

- Šifrování citlivých informací uživatelů v databázi, jako jsou tracking sessions a časové intervaly, aby k nim měl přístup pouze sám uživatel.
- Synchronizace dat mezi klienty aplikace (např. mezi webovou aplikací v prohlížeči na počítači a v prohlížeči na mobilním telefonu).
- Responzivní uživatelské rozhraní, které bude možné efektivně a pohodlně používat na zařízeních s menšími displeji (tablet, mobilní telefon).

6.3.2 Návrh uživatelského rozhraní aplikace

Základní princip webové aplikace TTRebel zůstane stejný jako v původní verzi aplikace – uživatelé budou nadále moct vytvářet tracking sessions, u kterých si budou evidovat Redmine úlohu, popis prováděné práce a vybírat Redmine aktivitu. Stejně zůstane také měření času v rámci tracking session po časových intervalech a export tracking sessions do Redmine. Navrhované změny převážně spočívají v přidání nových funkcionalit, optimalizaci uživatelského rozhraní a jeho přizpůsobení novým funkcionalitám.

Změnou, která se promítne na velké části uživatelského rozhraní, je nahrazení výrazu „tracking session“ za „měření“. I přes to, že zaměstnanci firmy nemají problémy s angličtinou, nové pojmenování by mohlo především novým uživatelům pomoci pochopit koncept aplikace a lépe se s ním sžít. Výraz „měření“ bude dále v práci používán tomto významu.

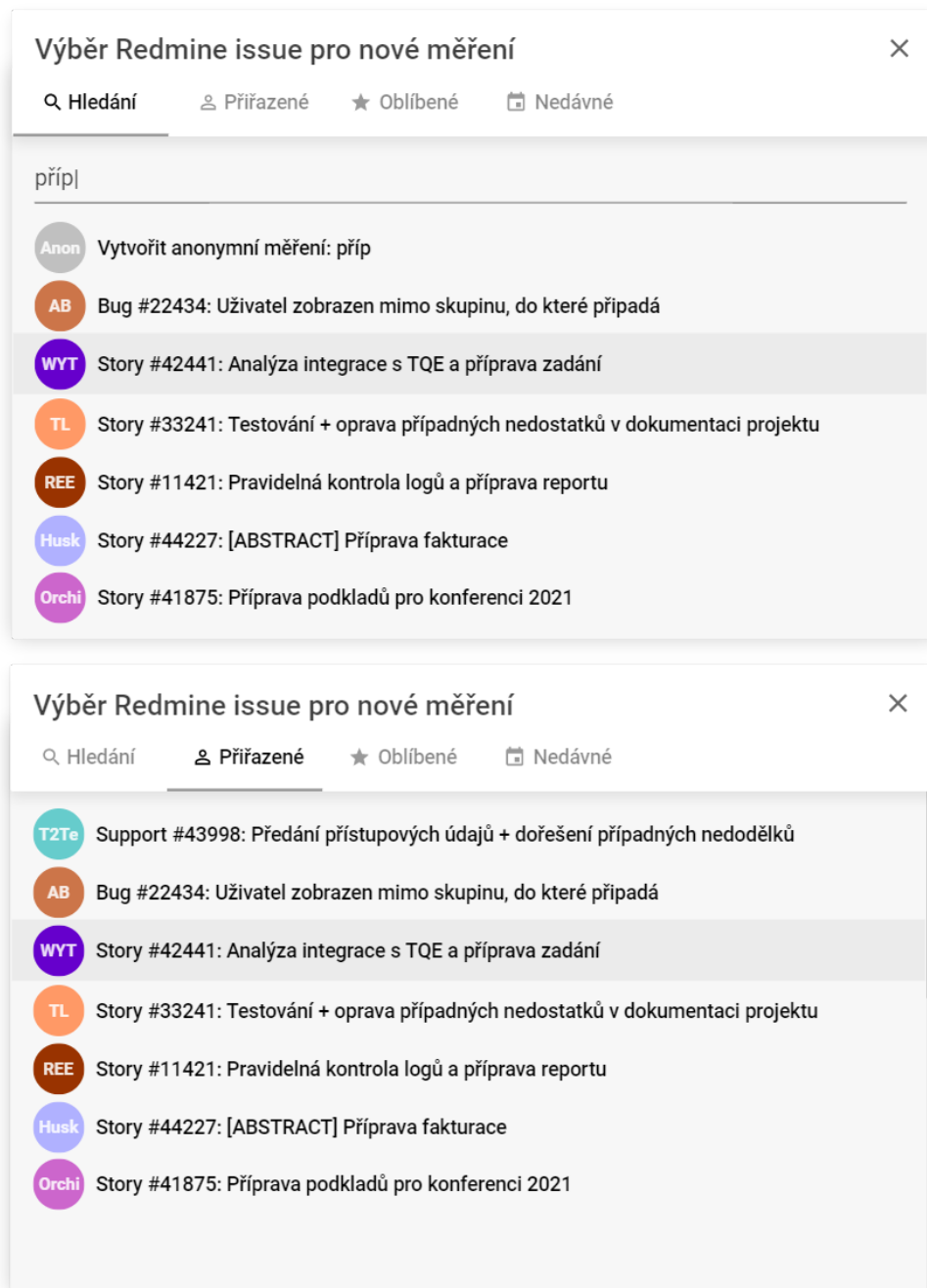
6.3.2.1 Hlavní stránka

Hlavní stránka aplikace (viz obrázek 6.3) nyní obsahuje tlačítko pro vytvoření nového měření namísto vyhledávacího pole. Po kliknutí na tlačítko či stisknutí nastavené klávesové zkratky (viz sekce 6.3.2.6) se otevře modální okno s pokročilým výběrem Redmine úlohy pro nové měření, které obsahuje několik záložek. Pod první záložkou se skrývá vyhledávač Redmine úloh, ve kterém mohou uživatelé stejně jako u původní aplikace vyhledávat úlohy dle identifikátoru či názvu a také mají možnost vytvořit anonymní měření, tedy pouze s popisem práce a bez Redmine úlohy. Další záložky postupně obsahují Redmine úlohy přiřazené na uživatele, úlohy z oblíbených měření a úlohy, ke

kterým uživatel měřil čas v nedávné době. Po seznamu úloh na každé záložce se bude možné pohybovat šipkami, mezi záložkami se bude možné přepínat pomocí tabulátoru a výběr měření bude možné potvrdit klávesou enter. Po potvrzení výběru se modální okno zavře a bude vytvořeno nové měření. Toto okno pro výběr Redmine úlohy je používáno všude v aplikaci, kde má uživatel možnost měnit Redmine úlohu.

Výraznou změnou prošla prezentace uživatelových měření, která nyní více připomínají prvky seznamu, jsou kompaktnější a díky tomu se jich na stránku vejde více, jak je vidět na obrázku 6.3. V levé části měření se nachází barevné kolečko se zkráceným názvem projektu, ze kterého pochází vybraná Redmine úloha (plný název projektu se zobrazí po najetí kurzorem na kolečko). Barva kolečka se bude odvíjet od názvu projektu tak, aby projekty se stejnou předponou (např. „ABC Development“ a „ABC Support“) měly stejnou barvu. Na měření zůstala pohodlná inline editace Redmine úlohy a popisu práce, avšak výběr Redmine aktivity byl přemístěn na detail měření, jelikož většina uživatelů používá pouze výchozí aktivitu a prvek tedy zbytečně zabíral prostor. Tlačítko pro spuštění/zastavení měření zůstalo stejné jako v původní aplikaci.

6. NÁVRH ŘEŠENÍ PRO SLEDOVÁNÍ A VYKAZOVÁNÍ ČASU



Obrázek 6.2: Grafické návrhy modálního okna s vyhledáváním Redmine úloh a výběrem Redmine úloh přiřazených na uživatele ve webové aplikaci TTRebel

The screenshot displays the TTRebel web application interface. At the top, there is a navigation bar with a green button labeled '+ Nové měření', a profile picture, and icons for a chart, stars, and settings. Below the navigation bar, the main content area is divided into two sections: 'Pátek 25. 03.' and 'Čtvrtek 24. 03.'. Each section has a date and time indicator and an 'Exportovat' button. The tasks are listed with their respective team icons (AB, TTR, Anon, SY, WYT) and descriptions. The durations are shown in colored boxes (orange, green, grey).

Day	Time	Team	Task Description	Duration
Pátek 25. 03.	03:34:15	AB	Story #42441: Analýza integrace s TQE a příprava zadání Implementace a konzultace s týmem	00:01:10
	08:00:00	TTR	Story #43245: Integrace Redmine s Jira Testování, aktualizace dokumentace	02:10:00
		Anon	Anonymní měření telefonát s panem Novotným	00:04:20
		SY	Story #42246: Přidání možnosti upravit klávesové zkratky Popis prováděné práce	01:18:45
Čtvrtek 24. 03.	07:04:36	WYT	Support #11128: Chyba "Null Pointer Exception: r242 (E3)" v logu aplikacího serveru Analýza chyby, konzultace	00:26:12
		AB	Story #44227: [ABSTRACT] Příprava fakturace Sbírání podkladů	01:10:32
		OTP	Bug #33343: Modální okno s nápovědou se nezobrazuje Implementace a konzultace s KŽ	02:30:51
		TTR	Bug #22434: Uživatel zobrazen mimo skupinu, do které připadá Popis prováděné práce	02:44:23
		Anon	Anonymní měření konzultace s Vojtou před schůzkou	00:12:38

Obrázek 6.3: Grafický návrh hlavní stránky webové aplikace TTRebel se spuštěným měřením

Po najetí kurzorem na měření jsou odhalena tlačítka, která jsou jinak skryta, aby nepůsobila rušivě. Konkrétně se jedná o tlačítko pro odstranění měření, nové tlačítko pro navýšení naměřeného času o pět minut, tlačítko pro vytvoření kopie měření, tlačítko pro označení měření za oblíbené a tlačítko pro zobrazení detailu měření. Odstranění měření bude nutné potvrdit v dialogovém okně a po potvrzení se ve spodní části obrazovky objeví snackbar s tlačítkem pro zvrácení odstranění měření. Navýšení naměřeného času o pět minut bude fungovat tak, že k danému měření bude přidán nový časový interval dlouhý pět minut se začátkem nastaveným na čas v době použití tlačítka. Vytvoření kopie měření způsobí to, že se pro aktuální den vytvoří nové měření, které bude mít nastavenou stejnou Redmine úlohu, popis práce a Redmine aktivitu jako kopírované měření. Detailu měření a oblíbeným měřením se blíže věnují sekce 6.3.2.2 a 6.3.2.3.

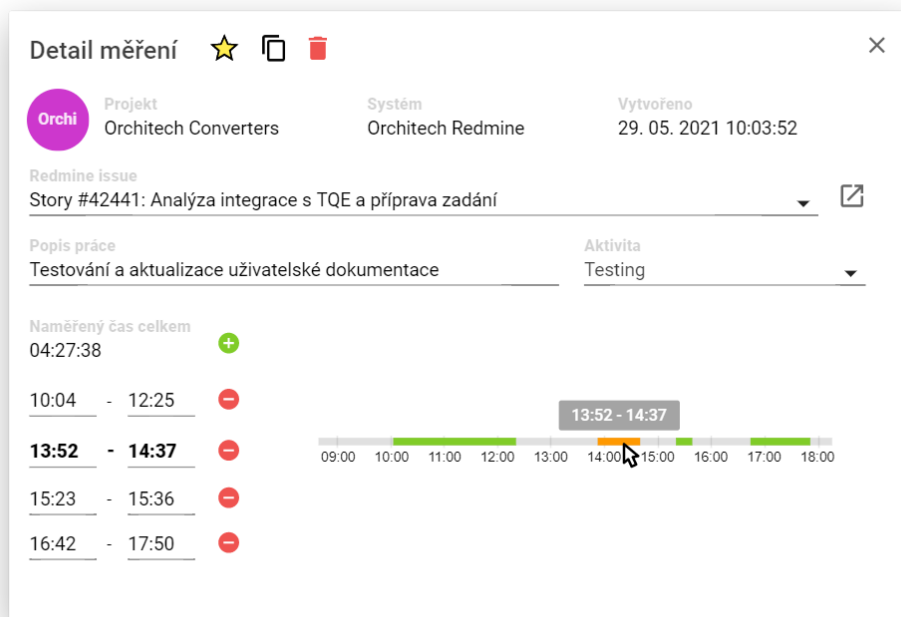
Měření jsou na hlavní stránce stále seskupovány po dnech, kdy byly vytvořeny. Nově se u každého dne nachází progress bar, který znázorňuje poměr naměřeného času ku požadovanému vykázanému času pro daný den. Výchozí hodnota požadovaného vykázaného času bude převzata z uživatelského nastavení běžné pracovní doby (viz sekce 6.3.2.6), pro každý den však bude možné hodnotu upravit inline editací, což mohou využít především zaměstnanci na částečný úvazek. Jak je vidět na obrázku C.4, pokud se uživatel posune níže na stránku, den, progress bar a tlačítko pro export vztahující se k měřením zobrazeným v horní části stránky budou ukotveny na samostatné liště, aby uživateli byly tyto informace stále dostupné.

Po měřeních se uživatel bude moci pohybovat pomocí šipek nahoru a dolů na klávesnici. Aktuálně vybrané měření může uživatel spustit/zastavit klávesou enter a vyvolat dialog pro odstranění měření klávesou delete.

6.3.2.2 Detail měření

Detail měření bude zobrazen v podobě modálního okna (viz obrázek 6.4) a bude nabízet podrobnější informace o měření. Pokud má měření nastavenou Redmine úlohu, budou zde zobrazeny informace o projektu a systému, ze kterých úloha pochází a také odkaz na detail úlohy v Redmine. Stejně jako na hlavní stránce je zde možné pomocí inline editace změnit Redmine úlohu, popis práce a navíc také Redmine aktivitu. Pro výběr Redmine úlohy bude opět použito modální okno jako při vytvoření nového měření (viz první odstavec sekce 6.3.2.1), nebude však zobrazeno nad modálním oknem s detailem měření, ale nahradí jej. Okno s výběrem Redmine úlohy nebude umožňovat vytvoření anonymního měření a bude navíc obsahovat tlačítko pro přechod zpět na detail měření, jak je možné vidět na obrázku C.5.

Na detailu měření je dále uveden celkový naměřený čas spolu s možností podrobně spravovat časové intervaly. Pole se začátky a konci časových intervalů budou mít již předvyplněnou dvojtečku mezi hodinami a minutami tak, aby uživatel mohl zadávat pouze číslice a nemusel dvojtečku zadávat sám. Po-



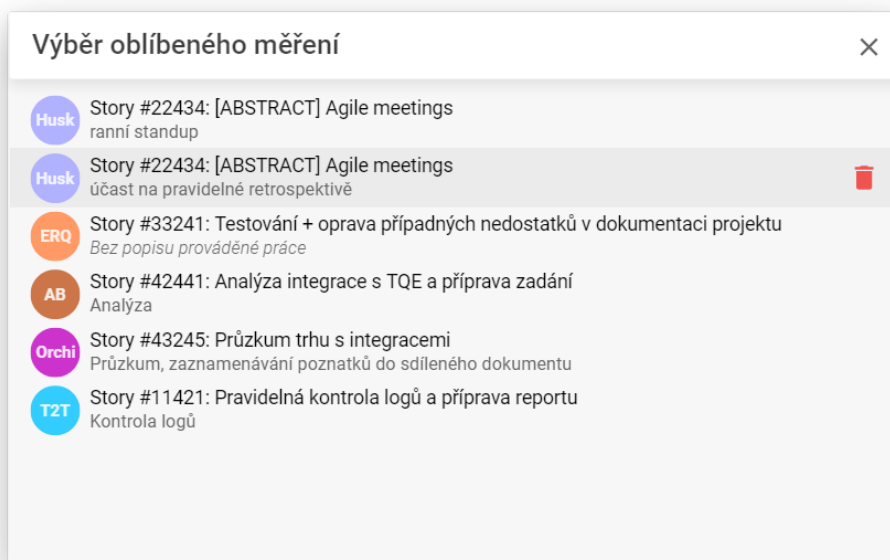
Obrázek 6.4: Grafický návrh modálního okna s detailem měření ve webové aplikaci TTRebel

kud však dvojtečku zadá na správné pozici, bude ignorována a uživatel bude moci pokračovat se zadáváním minut. Vedle sloupce s časovými intervaly se nachází jejich vizualizace v podobě časové osy, jejíž počátek a konec se odvíjí od prvního a posledního časového intervalu ze všech měření daného dne. Časové intervaly měření jsou na časové ose vyobrazeny zelenou barvou, po najetí kurzorem se barva změní na oranžovou, objeví se informace o časovém intervalu a daný interval bude zvýrazněn v seznamu intervalů.

V horní části modálního okna se také nachází tlačítko pro přidání či odebrání měření ze seznamu oblíbených měření, vytvoření kopie měření a odstranění měření. Odstranění měření bude také možné vyvolat pomocí klávesy delete.

6.3.2.3 Oblíbená měření

Z rozhovorů se zaměstnanci vyplynulo, že většina zaměstnanců by v aplikaci využila možnost spravovat si seznam oblíbených Redmine úloh, které by bylo možné přidat k měření snazším způsobem než běžným vyhledáváním. Mezi tyto úlohy by zaměstnanci zařadili takové úlohy, ke kterým si vykazují často či pravidelně, jako jsou např. úlohy zastřešující meetingy na projektu či průběžná správa dokumentace. Jelikož se v těchto případech často opakuje i prováděná práce, koncept oblíbených úloh byl upraven na oblíbená měření.

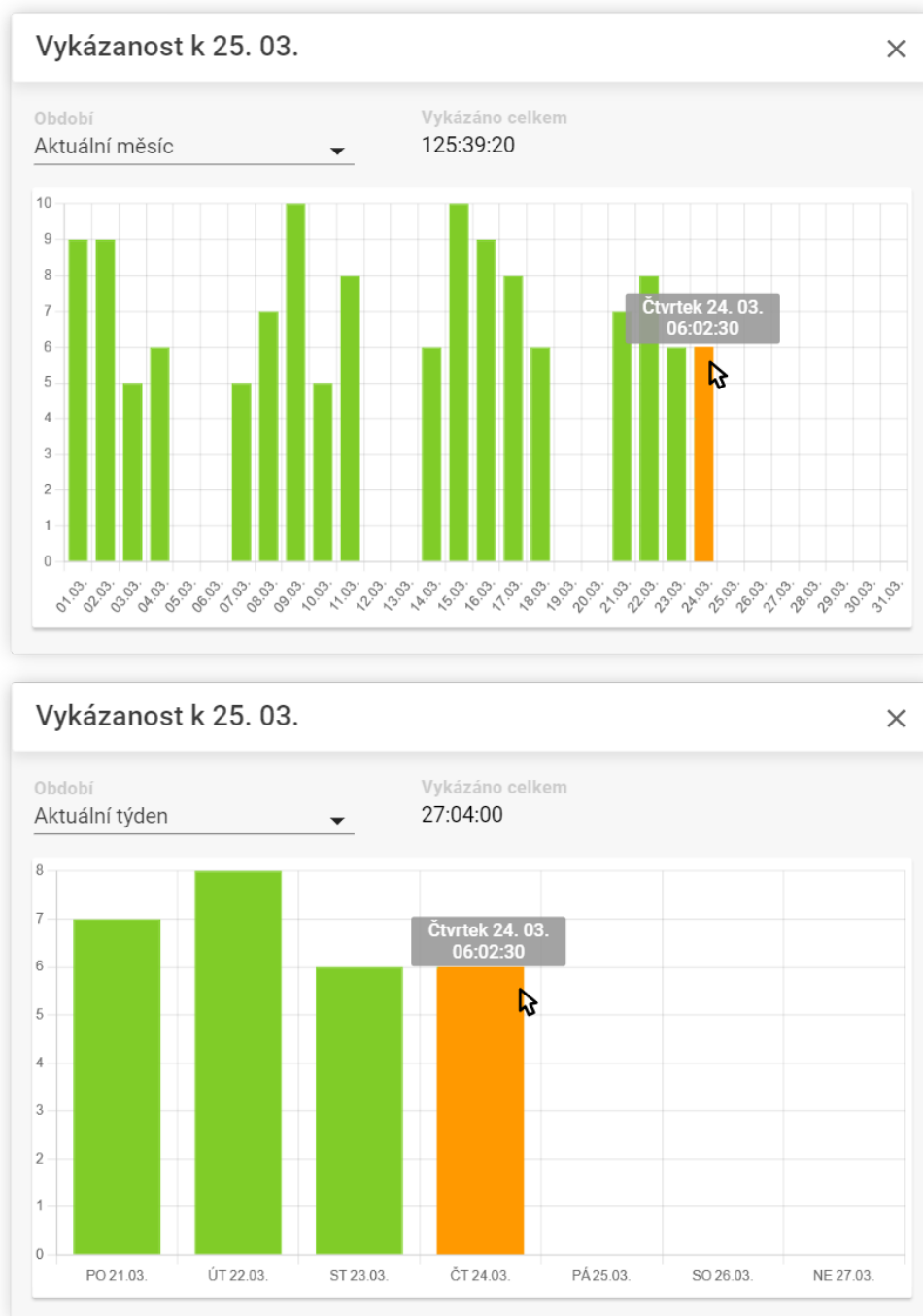


Obrázek 6.5: Grafický návrh modálního okna se seznamem oblíbených měření ve webové aplikaci TTRebel

Měření lze označit za oblíbené na hlavní stránce nebo na detailu daného měření. Seznam oblíbených měření v podobě modálního okna (viz obrázek 6.5) je možné zobrazit pomocí tlačítka ve tvaru hvězdy v horní liště aplikace. Na seznamu je možné vybrat konkrétní měření, čímž se vytvoří jeho kopie pro aktuální den, nebo jej odstranit.

6.3.2.4 Přehled vykázanosti

Aby uživatelé měli větší přehled o své vykázanosti a mohli se např. rozhodnout odejít z práce dříve, protože mají napracované hodiny z předchozích dní, byla do aplikace přidána sekce s přehledem vykázanosti. Tlačítko pro zobrazení modálního okna s přehledem se nachází v horní liště aplikace. Uživatelé si mohou vybrat, zda je zajímavá přehled vykázanosti v aktuálním měsíci či aktuálním týdnu (viz obrázek 6.6). Pro vybrané období bude vypsán celkový naměřený čas a vykreslen interaktivní sloupcový graf zachycující počet hodin vykázaných v jednotlivých dnech daného období. Grafy v grafických návrzích byly vytvořeny pomocí knihovny Chart.js [58], která by mohla být použita i ve finální implementaci.



Obrázek 6.6: Grafické návrhy modálního okna s přehledem vykázanosti v aktuálním měsíci a aktuálním týdnu ve webové aplikaci TTRebel

6.3.2.5 Export měření

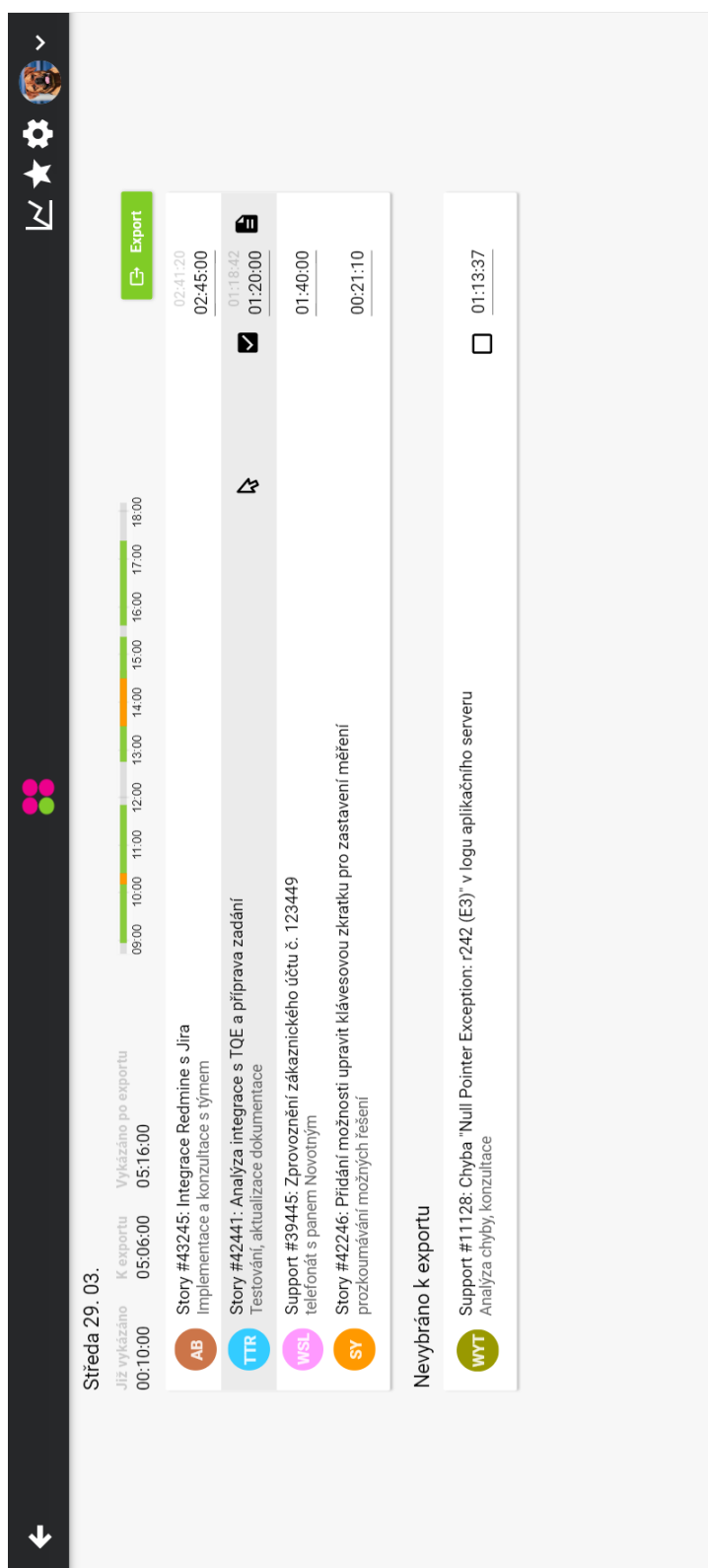
Stránka s exportem měření (viz obrázek 6.7) i nadále obsahuje údaje o již vykázaném času, času k exportu a výsledném času, který bude vykázaný po exportu. Měření na stránce vypadají stejně jako na hlavní stránce, obsahují tedy barevné kolečko se zkratkou projektu, Redmine úlohu a popis práce, které však není možné upravovat inline, ale pouze na detailu měření. Tlačítko pro zobrazení detailu měření a checkbox pro výběr měření k exportu se zobrazí po najetí kurzorem na měření. Měření nevybraná k exportu se nově nachází v samostatném seznamu na stránce pod seznamem vybraných měření. Tato měření mají vždy viditelné checkboxy pro vybrání k exportu, které pomohou uživateli okamžitě rozpoznat, že dané měření není vybrané k exportu. U každého měření je nadále možné upravovat čas k exportu – pokud tak uživatel učiní, původní naměřený čas bude zobrazen jako label daného pole.

Na stránce s exportem se nově nachází vyobrazení časových intervalů všech měření vybraných k exportu v podobě časové osy, velice podobné té na detailu měření. Po najetí kurzorem na měření budou na časové ose oranžovou barvou zvýrazněny všechny časové intervaly daného měření (viz obrázek 6.7), po najetí na jeden interval budou zvýrazněny i ostatní intervaly daného měření a samotné měření na seznamu (viz obrázek C.6). Časová osa navíc nabízí možnost vytvořit nové měření pokrývající vybraný interval, který není pokrytý žádným vybraným měřením (viz obrázek C.7) – v tomto případě bude otevřeno modální okno pro výběr Redmine úlohy jako při přidávání nového měření.

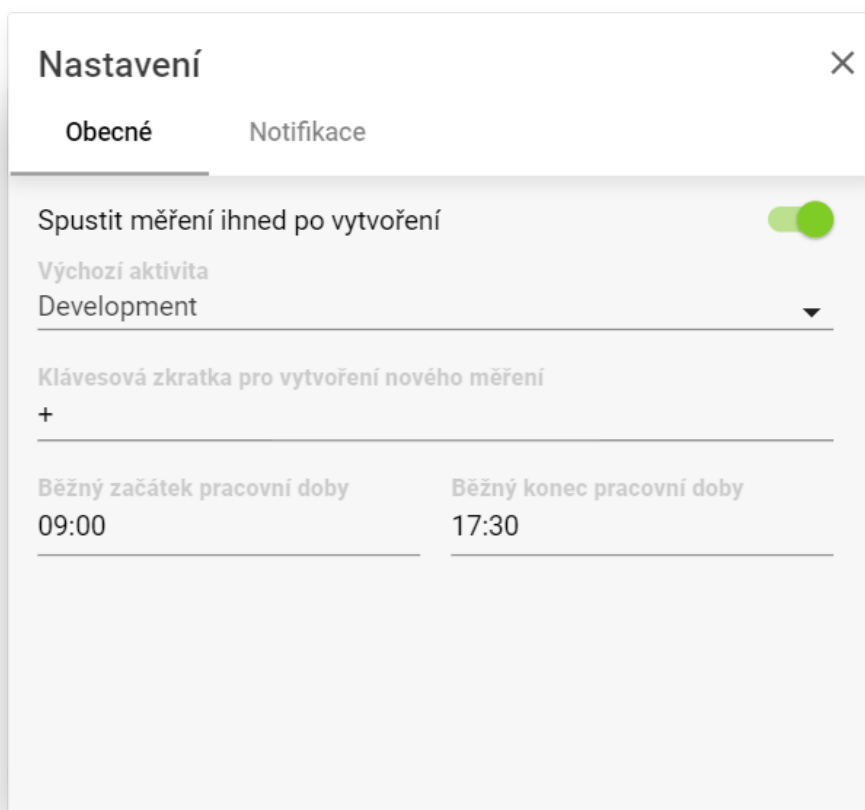
6.3.2.6 Přizpůsobení aplikace

Aplikace nově obsahuje sekci s přizpůsobením aplikace pro přihlášeného uživatele, která se v podobě modálního okna otevře po kliknutí na tlačítko ozubeného kola v horní liště aplikace. Okno obsahuje několik záložek, první z nich (viz obrázek 6.8) se věnuje obecným nastavením, jako je výchozí Redmine aktivita, klávesová zkratka pro vytvoření nového měření a nastavení běžného začátku a konce pracovní doby, kterými se řídí notifikace a progress bary na hlavní stránce.

Druhá záložka okna (viz obrázek C.8) obsahuje nastavení notifikací. Pokud si uživatel nepřeje být rušen, notifikace může zcela zakázat. Na výběr je také místo, kam mají být notifikace zasílány, resp. způsob upozorňování uživatele. Jednou možností jsou systémové notifikace, o které se postará prohlížeč s otevřenou aplikací, druhou možností je zaslání zpráv na Slack. Notifikacím se blíže věnuje sekce 6.3.4.



Obrázek 6.7: Grafický návrh stránky s exportem měření webové aplikace TTRebel s kurzorem naměřeným na měření



Obrázek 6.8: Grafický návrh modálního okna s obecným nastavením ve webové aplikaci TTRebel

6.3.2.7 Ovládání aplikace pomocí klávesnice

Aplikace bude podporovat pohyb po seznamech pomocí šipek, výběr ze seznamů pomocí klávesy enter a odstraňování záznamů pomocí klávesy delete. Mezi tyto seznamy patří např. modální okno s výběrem Redmine úlohy či seznam oblíbených měření. Všechna modální okna bude možné zavřít pomocí klávesy escape. Jak bylo popsáno v sekci 6.3.2.1, pomocí klávesové zkratky bude také možné vytvořit nové měření, nachází-li se uživatel na hlavní stránce aplikace.

6.3.3 Rozložení uživatelského rozhraní na menších zařízeních

Navrhované uživatelské rozhraní je plně responzivní a je navrženo tak, aby jeho používání na menších zařízeních bylo pohodlné. V rámci této práce nebyly vytvořeny návrhy všech stránek a oken pro menší zařízení, ale pouze těch, jejichž rozložení vyžadovalo složitější úpravy než pouhé zmenšení obsažených prvků.

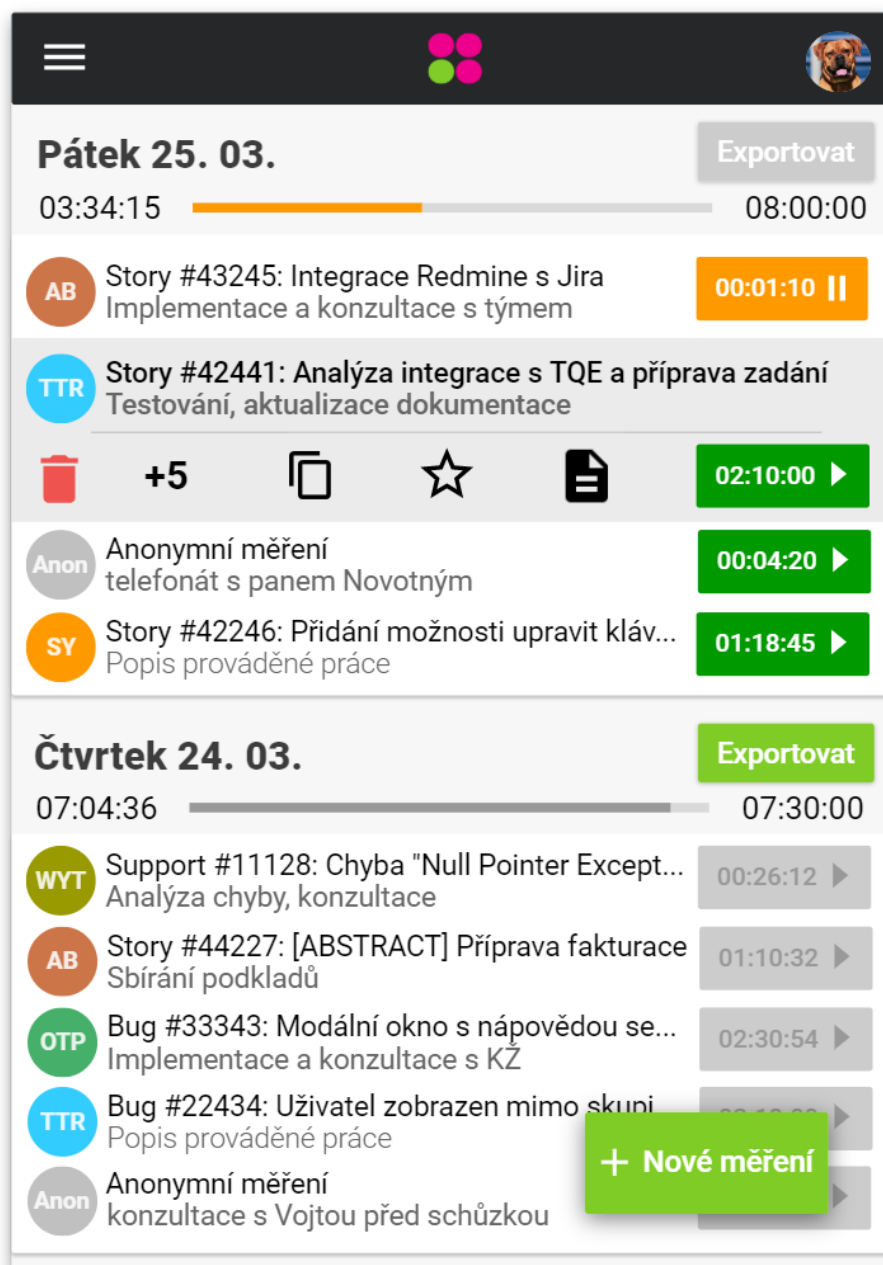
Co se týče horní lišty aplikace, tlačítka pro zobrazení seznamu oblíbených měření, vykázanosti a nastavení jsou přesunuta z pravého rohu lišty pod tzv. „hamburger menu“ umístěné do levého rohu lišty, jak je vidět na obrázku 6.9.

Nejvýraznější změnou na hlavní stránce (viz obrázek 6.9) je přesunutí tlačítek u měření, která jsou na mobilních zařízeních umístěna na lištu, která vyjede zpoza měření po kliknutí na něj. Je-li tato lišta zobrazena, přesune se na ni také tlačítka pro spuštění/zastavení měření, čímž se uvolní místo pro vypsání více znaků z názvu Redmine úlohy a popisu práce. Tlačítka pro vytvoření nového měření je na mobilních telefonech plovoucí a nachází se v pravém dolním rohu stránky. Stejně jako na velkých obrazovkách se po posunutí níže na stránce ukotví den, progress bar s naměřeným časem a tlačítka pro export měření v samostatné liště (viz obrázek C.9).

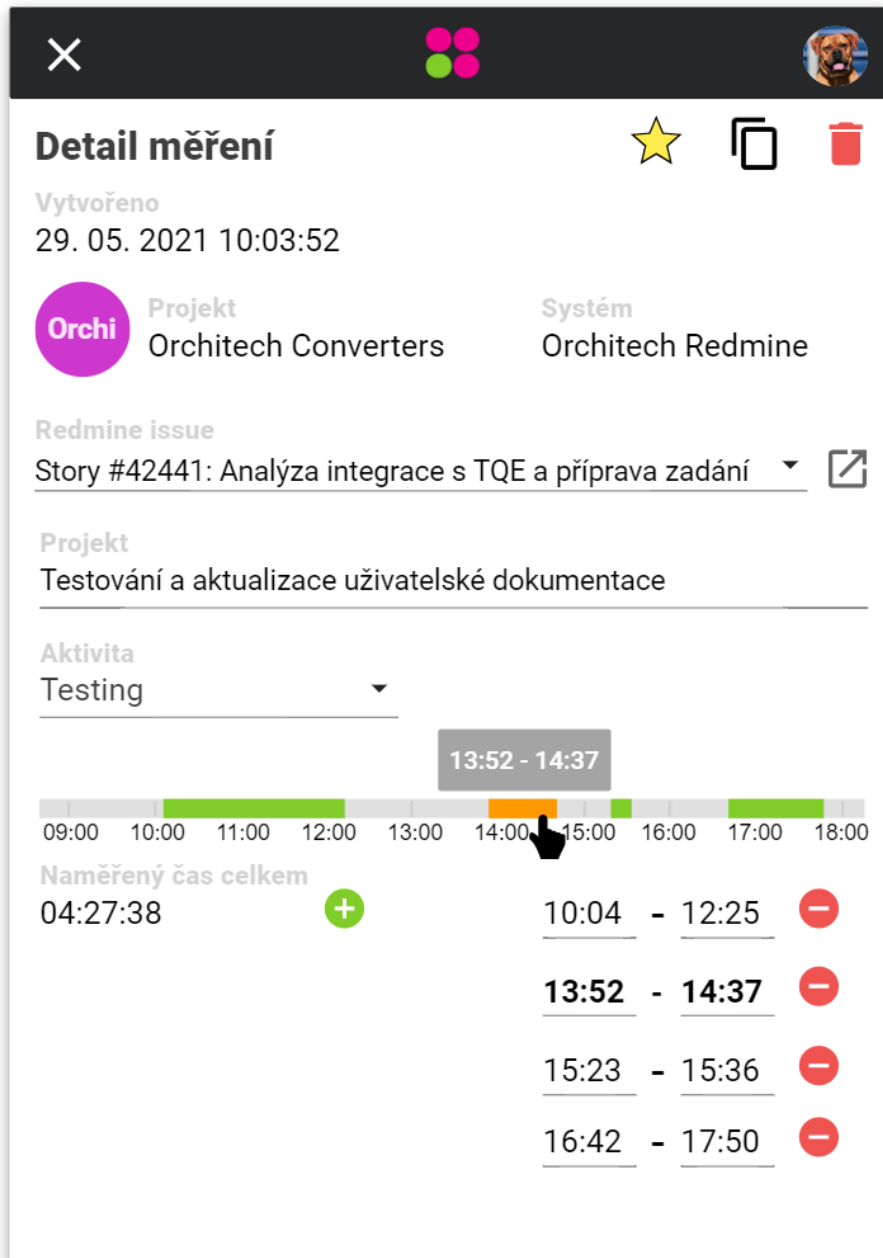
Všechna modální okna v aplikaci se na mobilním zařízení budou zobrazovat přes celou stránku, zůstane nad nimi však horní lišta aplikace, jak je vidět např. na obrázku s návrhem detailu měření 6.10. Tlačítka pro zavření modálního okna či tlačítka pro přechod zpět na předchozí modální okno se v těchto případech bude zobrazovat v levém rohu horní lišty aplikace a bude nahrazovat zmíněné hamburger menu.

Rozložení exportní stránky se mobilním telefonům přizpůsobuje stejným způsobem, jako hlavní stránka (viz obrázek C.10). Nejvýraznější změnou oproti rozložení na velkých zařízeních je opět přemístění tlačítek ze samotného měření na samostatnou lištu, která vyjede zpoza měření po kliknutí na něj. Stejně jako na hlavní stránce se při posunutí níže na stránce ukotví informace z horní části obrazovky v samostatné liště, jak je vidět na obrázku C.11.

6. NÁVRH ŘEŠENÍ PRO SLEDOVÁNÍ A VYKAZOVÁNÍ ČASU



Obrázek 6.9: Grafický návrh hlavní stránky webové aplikace TTR rebel na mobilním telefonu



Obrázek 6.10: Grafický návrh modálního okna s detailem měření ve webové aplikaci TTRebel na mobilním telefonu

6.3.4 Notifikace

Aplikace TTRebel bude nově umět komunikovat uživatelům informace pomocí notifikací. Bude se jednat buď o systémové notifikace, které bude vytvářet prohlížeč, ve kterém má uživatel webovou aplikaci otevřenou, nebo o zprávy zasláné uživateli na aplikaci Slack, která se v Orchitech Solutions aktivně používá. V nastavení aplikace si uživatelé budou moci notifikace zcela deaktivovat a vybrat si, jakým způsobem chtějí být upozorňováni.

Zasílání obou druhů notifikací bude iniciovat backendová část aplikace. V případě zasílání zpráv na Slack bude využívat Slack API [59], v případě systémových notifikací bude využívat technologii „Server Sent Events“ [60], která serverům umožňuje iniciovat komunikaci s klienty. Pomocí plánovače bude aplikace pravidelně každou minutu kontrolovat situace popsané níže. Tyto situace jsou pouze návrhy a před skutečným použitím vyžadují řádné otestování, aby uživatele zbytečně neotravovaly a neznepříjemňovaly jim používání navrženého řešení.

Uživatel zapomněl zastavit měření

Pokud má uživatel nepřerušeně zapnuté jedno měření po dobu osmi hodin, lze předpokládat, že si ve většině případů nechal měření omylem zapnuté v době, kdy už nepracuje. V takovém případě bude uživateli zaslána notifikace s účelem upozornit jej na tuto skutečnost, aby mohl měření zastavit a upravit si časové intervaly.

Uživatel zapomíná přepínat měření

Tato situace je velice podobná té předchozí, nastává však dříve, konkrétně po pěti hodinách souvisle spuštěného měření. V tomto případě je vhodné zaslat uživateli notifikaci ze dvou důvodů:

- je dost možné, že uživatel se během posledních pěti hodin věnoval i jiné práci, ale z nějakého důvodu si nepřepnul měření, notifikace mu tedy připomene, že si má hlídat svá měření,
- notifikace uživateli připomene, na čem by měl podle spuštěného měření pracovat, což mu může pomoci vrátit se k práci v případě, že se jeho soustředěnost upnula na něco jiného.

Uživatel zapomněl spustit měření během pracovní doby

Uživatelům se může snadno stát, že si zapomenou měřit čas, protože jsou příliš zabraní do své práce nebo musejí často přepínat mezi různými činnostmi. Pokud nastane situace, že během uživatelovi nastavené pracovní doby uběhne více jak dvě hodiny bez spuštěného měření, bude uživateli zaslána notifikace, jejímž účelem bude připomenout mu, že by si měl svou práci měřit.

6.3.5 Integrace s více systémy Redmine

V práci již bylo zmíněno, že někteří zaměstnanci pracují na projektech, které mají vlastní dedikovaný systém Redmine, a že původní aplikace TTRebel integraci s více systémy nepodporuje. Absence této podpory je zapříčiněna především tím, že na to frontendová část aplikace není připravena, jak implementačně, tak z hlediska UX.

Navrhovaná verze aplikace v této práci alespoň z hlediska UX s více systémy počítá. Existence více systémů je uživateli převážně zatajena, s výjimkou detailu měření, kde je uveden systém vybrané Redmine úlohy. Při vyhledávání Redmine úloh by uživateli měly stačit název projektu a identifikátor a název úlohy k tomu, aby s jistotou našel hledanou úlohu. Minimálně v Orchitech Solutions platí, že názvy projektů se mezi jednotlivými Redmine systémy liší a tedy šance, že by uživatel v aplikaci TTRebel narazil na dvě Redmine úlohy se stejným projektem, stejným identifikátorem a stejným názvem, ale z jiného systému, je mizivá.

Během běžného vývoje aplikace TTRebel byla párkrát vznešena otázka, zda by při exportu měření do více systémů Redmine měly být souhrnné časy na stránce s exportem rozděleny dle systému. Po hloubkových rozhovorech se zaměstnanci dospěl autor práce k názoru, že uživatelům nezáleží na tom, kolik budou mít v kterém systému vykázáno, zajímá je pouze celkový součet, v tomto návrhu aplikace jsou tedy i na stránce s exportem souhrnné informace zobrazeny dohromady.

6.4 Desktopový klient aplikace TTRebel

Desktopový klient bude zaměřen na snadné, rychlé a nerozptylující ovládání aplikace TTRebel během pracovního dne. Aplikace bude nabízet pouze takové funkcionality webové aplikace TTRebel, které uživatelé používají nejčastěji – vytváření nových měření (anonymních i s Redmine úlohami), spuštění a zastavování měření a zadávání popisu práce. Ke všemu ostatnímu, jako je exportování měření do Redmine, vizualizace naměřeného a odpracovaného času, spravování seznamů oblíbených úloh aj., bude nadále sloužit webová aplikace TTRebel.

Podobně jako běžně používané desktopové aplikace (např. Microsoft Teams, Slack, Skype apod.) bude tato aplikace spuštěna na pozadí a bude dostupná v notifikační oblasti uživatelova systému. Díky tomu bude mít uživatel aplikaci stále na očích a nebude ji muset hledat ve svém prohlížeči. Aplikace v notifikační oblasti systému bude mít také možnost komunikovat uživateli informaci, zda má spuštěné nějaké měření či nikoliv, pouze pomocí své ikony, s čímž se uživatelé běžně setkávají např. u aplikací pro komunikaci.

Další podstatnou výhodou desktopové aplikace spuštěné na pozadí je možnost reagovat na klávesové zkratky na úrovni celého systému. Webové aplikace mohou běžně reagovat na klávesové zkratky pouze ve chvíli, kdy se uživatel

nachází v okně s prohlížečem, ve kterém má právě zobrazenou záložku s aplikací. Desktopové aplikace naproti tomu mohou na klávesové zkratky reagovat v podstatě kdykoliv, s výjimkou případů, kdy na použitou klávesovou zkratku zareaguje dřív jiná aplikace. Díky tomu budou uživatelé moci používat aplikaci bez zbytečného rozptýlení způsobeného hledáním záložky s aplikací ve webovém prohlížeči, ať už při běžné změně činnosti, tak při náhlé situaci (např. při obdržení nečekaného telefonátu).

6.4.1 Návrh uživatelského rozhraní

Uživatelské rozhraní aplikace bude jednoduché a bude obsahovat pouze prvky nutné k vytváření nových měření a spouštění/zastavování měření. Okno s aplikací bude zobrazeno ve středu primární obrazovky a bude poměrně malé, aby uživateli nepřekrývalo moc aktuálního obsahu na obrazovce a nerozptylovalo jej od práce. Navržené okno s aplikací zasazené do popředí systému Windows je vidět na obrázku C.12.

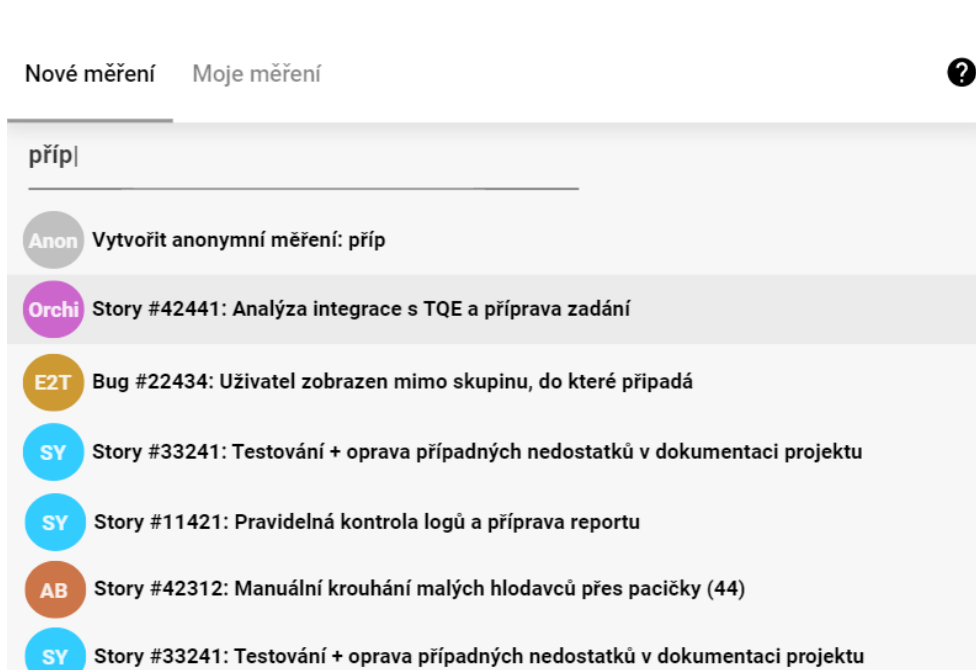
V horní části aplikace se bude nacházet lišta se záložkami a pomocnými tlačítky, ve spodní části obrazovky se bude nacházet lišta s informacemi o aktuálně spuštěném měření s tlačítkem pro jeho zastavení. Za záložkami na horní liště se budou skrývat dvě obrazovky: obrazovka pro přidání nového měření a seznam uživatelských měření.

Přidání nového měření bude spočívat pouze ve výběru Redmine úlohy z našeptávače podobného tomu ve webové aplikaci TTRebel. I nadále bude možné vytvořit anonymní měření pouze s popisem práce.

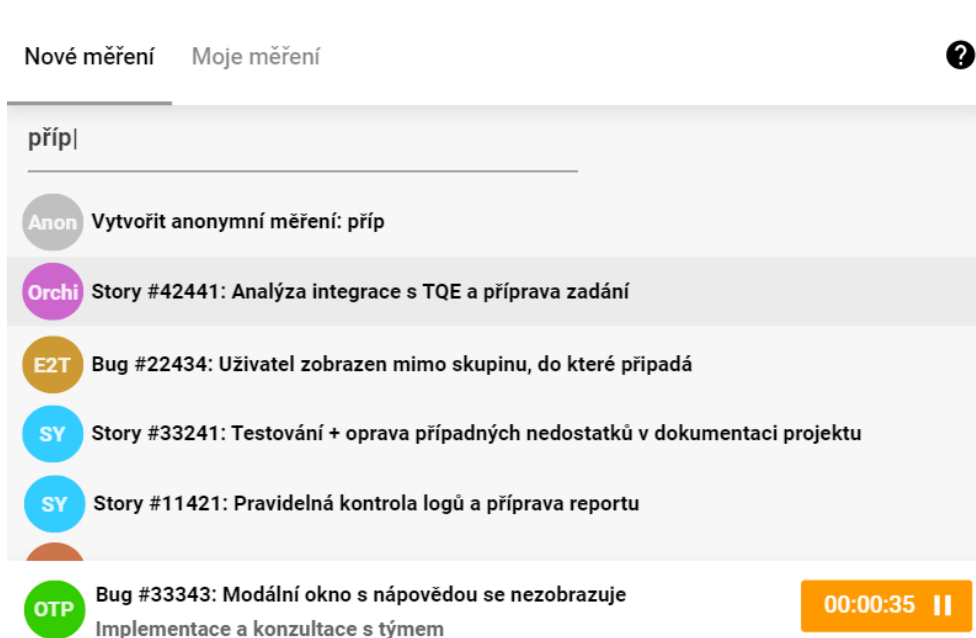
Obrazovka se seznamem uživatelských měření bude obsahovat pouze měření z aktuálního dne, protože u historických měření již nelze měřit čas a tudíž by v aplikaci byly zbytečné. U každého měření se bude nacházet pole pro zadání popisu práce a tlačítko pro spuštění/zastavení měření spolu s doposud naměřeným časem. Podrobnější informace o naměřeném čase v podobě časových intervalů ani nastavování aktivity v této aplikaci dostupné nebudou.

V liště v horní části aplikace se bude nacházet tlačítko pro zobrazení nápovědy. Obsahem nápovědy bude pouze stručné vysvětlení ovládání aplikace, mělo by tedy stačit zobrazit nápovědu v modálním okně.

Pro uživatelské rozhraní budou použity některé prvky z webové aplikace TTRebel, např. reprezentace měření či tlačítko pro spuštění/zastavení měření. Uživatelské rozhraní se dále bude řídit specifiky Material designu stejně jako webová aplikace.

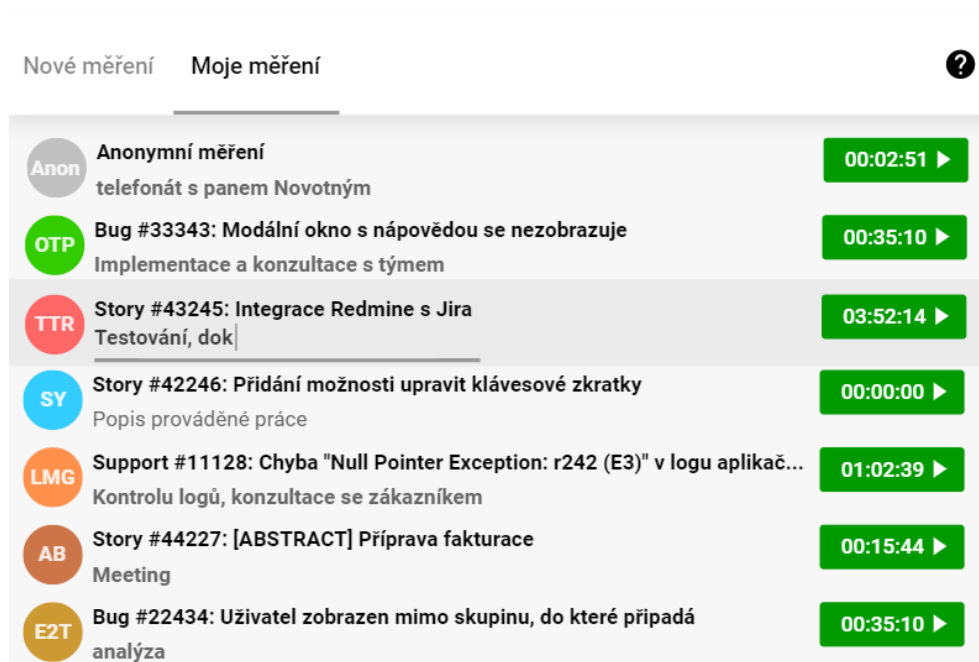


Obrázek 6.11: Grafický návrh obrazovky pro vytvoření nového měření v desktopové aplikaci TTRebel

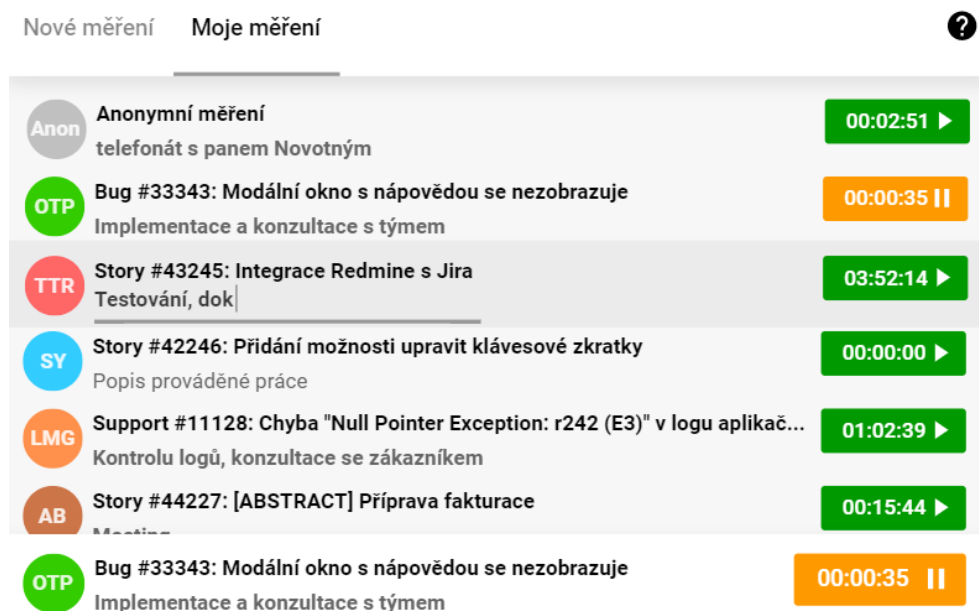


Obrázek 6.12: Grafický návrh obrazovky pro vytvoření nového měření v desktopové aplikaci TTRebel se spuštěným měřením

6. NÁVRH ŘEŠENÍ PRO SLEDOVÁNÍ A VYKAZOVÁNÍ ČASU



Obrázek 6.13: Grafický návrh obrazovky se seznamem uživatelských měření v desktopové aplikaci TTRebel



Obrázek 6.14: Grafický návrh obrazovky se seznamem uživatelských měření v desktopové aplikaci TTRebel se spuštěným měřením

6.4.2 Ovládání aplikace

Jak již bylo řečeno, aplikace je navržena tak, aby ji bylo možné efektivně ovládat pomocí klávesnice bez nutnosti používat myš. Pomocí systémových klávesových zkratk budou uživatelé moci vyvolat okno s aplikací nebo zastavit spuštěná měření, což se hodí především v situacích, kdy uživatel musí náhle odejít od počítače (např. kvůli zvonku u dveří či odchodu na oběd) nebo se začít věnovat nepracovním záležitostem (např. odpovědět na osobní telefonát). Vyvolání okna s aplikací či zastavení spuštěných měření bude možné i z kontextové nabídky, která se objeví po kliknutí na ikonu aplikace v systémové liště.

V aplikaci se bude uživatel moci přepínat mezi oběma obrazovkami pomocí klávesové zkratky. Pohyb po seznamu Redmine úloh i měření bude možný pomocí šipek nahoru a dolů, k potvrzení výběru Redmine úlohy či spuštění/zastavení vybraného měření bude sloužit klávesa enter. Při pohybu po seznamu měření budou automaticky aktivní vstupy pro popis práce, aby ji uživatel mohl snadno vyplnit, opět bez nutnosti používat myš.

Vždy, když uživatel přejde na jednu z obrazovek, předem určené pole bude automaticky aktivní – na obrazovce pro přidání nového měření to bude pole pro vyhledávání Redmine úloh, na seznamu měření to bude pole s popisem aktuálně spuštěného (nebo prvního) měření v seznamu. Díky tomu se po přechodu na danou obrazovku bude uživatel moci ihned soustředit na to, co právě potřebuje.

6.4.3 Přizpůsobení aplikace

Aplikace bude nabízet následující možnosti přizpůsobení:

- určení vlastních klávesových zkratk pro vyvolání okna s aplikací a zastavení spuštěných měření,
- výběr obrazovky, na které bude vyvoláváno okno s aplikací,
- možnost automatického spuštění aplikace po spuštění systému.

Bez zpětné vazby od uživatelů není snadné dopředu určit, jakými dalšími způsoby si uživatelé budou přát přizpůsobit si aplikaci, zmíněné možnosti však budou jistě užitečné. Samostatnou obrazovku s přizpůsobením aplikace budou uživatelé moci vyvolat volbou v kontextové nabídce, která se objeví po kliknutí na ikonu aplikace v systémové liště.

6.5 Rozšíření prohlížeče

Rozšíření prohlížeče bude obstarávat přidání tlačítka pro vytvoření nového měření na vybrané stránky, na kterých zaměstnanci vykonávají nějakou pracovní činnost. Mezi takové stránky patří např. detail úlohy firemního Redmine,

detail merge requestu v systému GitLab nebo stránka s otevřeným dokumentem či tabulkou v prostředí Google Drive. Tlačítko zaměstnancům umožní si zaznamenat práci na dané stránce v podobě měření v aplikaci TTRebel, v některých případech jim ušetří vyplňování informací, podle kterých by práci mohli zpětně identifikovat a také jim bude pasivně připomínat, že si mají svou práci spjatou s danou stránkou měřit.

S podobným přístupem, tedy využitím rozšíření prohlížeče pro přidání nějakého ovládacího prvku na vybrané stránky, se můžeme setkat např. u nástroje pro měření času s názvem Toggl. U tohoto nástroje je přes samotné rozšíření, které je v moderních webových prohlížečích uživatelům dostupné na liště s rozšířeními, možné také ovládat měření času a k naměřenému času přidávat různé informace. Navrhované rozšíření pro aplikaci TTRebel tyto funkcionality nabízet nebude a jeho jediným účelem bude zmíněné přidávání tlačítek na vybrané stránky. Pokud uživatelé kliknou na ikonu rozšíření, budou přemístěni na záložku s webovou aplikací TTRebel nebo bude aplikace otevřena v nové záložce, pokud ji uživatel ještě nebude mít otevřenou.

V sekci 3.6.5 je zmíněn prototyp obdobného tlačítka pro vytváření a spouštění/zastavování měření, které je umístěné na detailu Redmine úlohy. Tento prototyp byl navržen a implementován jako rozšíření systému Redmine, zaměstnanci by si tedy nemuseli nic instalovat do svého prohlížeče a tlačítko by navíc bylo automaticky dostupné i při navštěvování detailu Redmine úlohy na mobilním telefonu. Tento způsob implementace však není možný u systémů, které firma sama neprovozuje, jako jsou např. často užívané aplikace v prostředí Google Workspace (Disk, Meet, Gmail aj.) – rozšíření webového prohlížeče naproti tomu umožní přidávat tlačítko na téměř kteroukoliv webovou stránku.

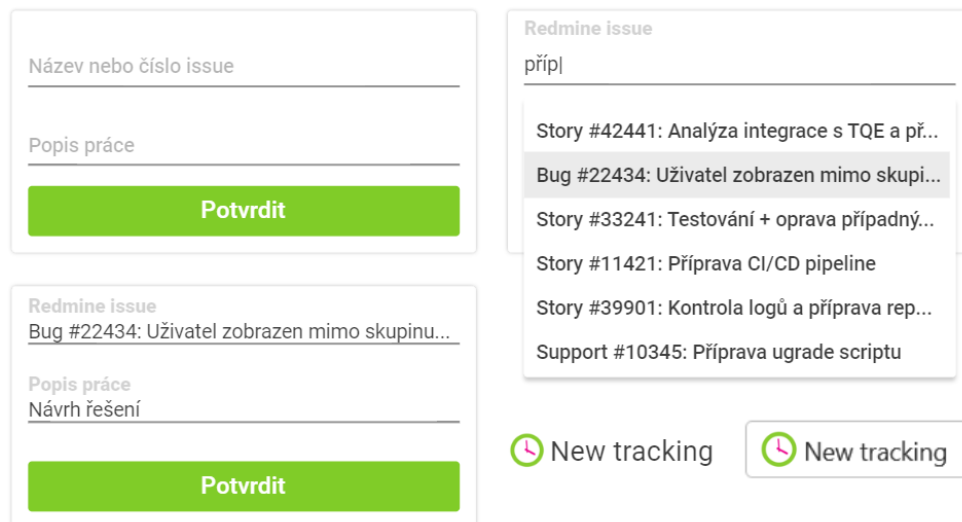
6.5.1 Návrh uživatelského rozhraní

Pro každou vybranou stránku budou vzhled i umístění tlačítka přizpůsobeny danému uživatelskému rozhraní, aby do něj tlačítko vhodně zapadalo. Po kliknutí na tlačítko se na vybrané stránce objeví nabídka, ve které budou pole pro zadání Redmine úlohy a popisu práce a tlačítko pro potvrzení. Pole s Redmine úlohou bude fungovat jako našeptávač úloh, podobně jako ve webové aplikaci TTRebel. Vzhled nabídky bude na všech stránkách stejný. Po potvrzení nabídky bude v aplikaci TTRebel vytvořeno nové měření se zadanými údaji. Zda bude vytvořené měření rovnou i spuštěno, bude záviset na uživatelově nastavení aplikace TTRebel. Pokud nebude spuštěno a uživatel k němu bude chtít čas doplnit zpětně, datum a čas vytvoření nalezne na detailu měření ve webové aplikaci TTRebel.

Aby rozšíření ušetřilo uživatelům zadávání informací, které jsou dostupné z obsahu dané stránky, budou dle dostupných informací obě pole nabídky předem vyplněna. Prvky stránky a způsob získávání informací z nich budou dopředu definovány v implementaci rozšíření, např. na detailu Redmine úlohy

bude její identifikátor moci být získáván z URL nebo prvku s názvem úlohy. Ve firmě Orchitech Solutions je běžnou konvencí vkládat identifikátor úlohy do názvů relevantních objektů, jako jsou např. Google Docs dokumenty nebo merge requesty v GitLabu – na jejich detailních stránkách bude tedy také možné v některých případech jednoznačně určit Redmine úlohu. Pokud nebude možné na stránce nalézt identifikátor odpovídající Redmine úlohy, budou do pole s popisem práce vyplněny detaily, pomocí kterých bude moci uživatel práci na stránce zpětně identifikovat, např. zmíněné názvy Google Docs dokumentu či merge requestu v GitLabu.

Zasazení návrhu nabídky pro vytvoření měření a tlačítka k vyvolání nabídky na detailu Redmine úlohy a na detailu merge requestu v systému GitLab jsou vidět na obrázcích C.13 a C.14.



Obrázek 6.15: Grafické návrhy nabídky pro vytvoření měření v aplikaci TTRebel a tlačítek k vyvolání nabídky

6.5.2 Autentizace uživatele

Rozšíření bude komunikovat s backendovou částí aplikace TTRebel přes její REST rozhraní stejně jako frontendová část webové aplikace a k tomu je potřebná autentizace uživatele. K autentizaci uživatele budou použity cookies webové aplikace TTRebel – dokud tedy uživatel nebude přihlášen ve webové aplikaci, nebude moci používat tlačítka pro vytvoření měření na vybraných stránkách.

6.6 ActivityWatch

6.6.1 Stručné seznámení se s architekturou nástroje

Architektura nástroje ActivityWatch se dle [61] skládá z několika komponent, pro účely této práce je však dostačující ji zjednodušit na server a klienty. Server v tomto případě nepředstavuje aplikační server spuštěný na vzdáleném zařízení, ale aplikaci spuštěnou na pozadí uživatelského systému. Kromě toho je však úloha serveru již tradiční, tj. zajišťuje manipulaci s daty, zapouzdřuje hlavní logiku aplikace a komunikuje s klienty. Samotný server nijak nesbírá informace o uživatelské aktivitě, o to se starají právě klienti.

Klienti se v terminologii ActivityWatch rozdělují na tzv. „watchers“ a „importers“. Watchers mají za úkol periodicky sbírat informace o uživatelské aktivitě a předávat je serveru. Watchers jsou běžně zaměřeni na jeden konkrétní zdroj, ze kterého informace sbírají, jako je např. webový prohlížeč nebo textové editory a IDE. Úlohou importers je na požádání importovat informace o uživatelské proběhlé aktivitě do ActivityWatche. Nekomunikují tedy se serverem průběžně, ale pouze když je potřeba. Běžný importer po spuštění sesbírá data z konkrétního zdroje (v požadovaném množství), zpracuje je do vhodné podoby a předá je serveru, který data uloží. Importer je tedy vhodné použít např. ve chvíli, kdy má uživatel nasbíraná data o své aktivitě pomocí jiného nástroje a chce je jednorázově přenést do nástroje ActivityWatch.

Data pocházející od různých klientů jsou běžně ukládána v samostatných jmenných prostorech, které se v terminologii ActivityWatch nazývají „buckets“.

6.6.2 Zdroje informací o uživatelské aktivitě

ActivityWatch již v základu obsahuje watchers, kteří sledují aktivitu na úrovni otevřených oken v systému a zda uživatel aktivně interaguje s počítačem či nikoliv. Tyto watchers samotné tedy mohou uživateli poskytnout přehled o tom, kdy byl aktivní v jakém okně systému.

Práce na počítači zaměstnanců Orchitech Solutions z většiny spočívá v používání webového prohlížeče a nějakého IDE. Pro získávání detailnějších informací o aktivitě v prohlížečích a IDE již existuje několik watchers implementovaných komunitou ActivityWatch, např. pro prohlížeče Google Chrome a Firefox a pro IDE VS Code nebo IntelliJ IDEA [62]. Tyto watchers sbírají informace na úrovni názvu aktivní záložky a URL stránky v prohlížeči či názvu otevřeného souboru v IDE.

V rámci hloubkových rozhovorů byli zaměstnanci dotázáni, kde hledají informace o své aktivitě z předchozích dnů za účelem vykázání. Většina zaměstnanců uvedla jako hlavní zdroj stránku s aktivitou v nástroji Redmine, kde lze nalézt informace o přidávání komentářů k úlohám, vytváření či upravování úloh, vykázání času k úlohám a vytváření či upravování Wiki stránek.

Druhou nejčastější odpovědí bylo vyhledávání uživatelových zpráv podle časového období v komunikačním nástroji Slack. Několik zaměstnanců dále uvedlo stránku s aktivitou v nástroji GitLab, která obsahuje informace o všech změnách v repozitáři, jako je např. pushnutí nové branchy či nového commitu. Pár zaměstnanců také zmínilo, že si zpětně procházejí pracovní e-maily v klientovi Gmail a události v pracovním Google Kalendáři. Dva zaměstnanci, kteří se na projektech zabývají především podporou, uvedli jako zdroj informací seznam hovorů se zákazníky na svých mobilních telefonech.

Posledním zdrojem, který žádný ze zaměstnanců sice nezmínil, ale může také obsahovat relevantní informace o aktivitě uživatele, je aktivita na Google Disku. Firma Orchitech Solutions má na Google Disku několik sdílených disků (každý pro jiné účely), v rámci kterých si lze v uživatelském rozhraní zobrazit historii práce s dokumenty.

6.6.3 Sbíráání informací z jednotlivých zdrojů

Pro zaměstnance by jistě bylo výrazně příjemnější, kdyby jim byla aktivita ze zmíněných zdrojů prezentována v přehledné podobě na jednom místě, místo nutnosti dohledávat svou aktivitu v každém zdroji samostatně. ActivityWatch je pro tento účel ideálním nástrojem, je pouze potřeba pokrýt všechny zmíněné zdroje pomocí watchers a importers. Návrh dále předpokládá, že uživatel potřebuje informace o své aktivitě pouze na vyžádání, tedy ve chvíli, kdy si chce zobrazit svou aktivitu na časové ose, není tedy potřeba data ukládat do ActivityWatch průběžně. Možnosti pokrytí jednotlivých zdrojů jsou popsány níže.

Redmine

K integraci různých nástrojů se systémem Redmine je běžně využíváno REST API, přes které se lze dostat k úlohám, informacím o projektech a uživatelích apod. – aktivitu uživatele, kterou nabízí uživatelské rozhraní, však přes REST API získat nelze. Jediným způsobem, jak je možné získat aktivitu z Redmine ve strojově čitelné podobě bez nutnosti dalších zásahů do implementace je Atom feed.

Ke sledování aktivity v Redmine do jisté míry poslouží i watcher aktivity ve webovém prohlížeči, z jehož záznamů lze poznat, jaké konkrétní Redmine stránky má uživatel otevřené. Nelze z nich však poznat, kdy uživatel upravil nějakou úlohu nebo k ní přidal komentář či přílohu, jelikož tyto činnosti jsou vykonávány na stránce s detailem úlohy a z titulku stránky ani jejího URL tyto podrobnosti nelze vyvodit. Bohužel se jedná o důležité informace, jelikož činnosti jako je úprava úloh a psaní komentářů mohou zabrat nezanedbatelné množství času.

Pro dostatečné pokrytí aktivity v Redmine je tedy potřeba vytvořit ActivityWatch importer, který bude buď zpracovávat Atom feed stránky s uživa-

telovou aktivitou, nebo získávat informace o aktivitě z REST API endpointu, který by musel být do Redmine implementován (pravděpodobně v podobě rozšíření).

Slack

I přes to, že Slack má webové rozhraní, převážná většina zaměstnanců používá nativní aplikaci, aktivitu v této aplikaci tedy nebude možné pokrýt watcherem webového prohlížeče. Slack naštěstí nabízí velice bohaté REST API, které obsahuje i endpoint pro vyhledávání zpráv s možností pokročilého filtrování dle časového období, autora zprávy, kanálu, ve kterém byla zpráva zaslána apod. [63]. Pro získávání aktivity ve Slacku tedy stačí vytvořit ActivityWatch importer, který bude s Slackem komunikovat přes REST API.

GitLab

Aktivitu v systému GitLab je možné procházet jak v uživatelském rozhraní, tak ve strojově čitelné podobě přes REST API endpoint s událostmi [64]. Události je navíc možné filtrovat např. podle časového období nebo typu události (např. pushnutí nového commitu do branch). I když je tento endpoint zmíněn pouze v dokumentaci placené edice GitLabu, autor práce se přesvědčil, že endpoint je dostupný i v komunitní (neplacené) edici. Stejně jako v případě Slacku bude pro získávání aktivity v GitLabu stačit vytvořit ActivityWatch importer, který bude s GitLabem komunikovat přes REST API.

Gmail, Google Disk a Google Kalendář

Pro získávání informací z Google aplikací, jako jsou Gmail, Google Disk a Google Kalendář, je možné využít REST API nebo oficiální klientské knihovny. Tyto knihovny jsou dostupné pro mnoho programovacích jazyků, jako je např. JavaScript, Java, Python nebo Ruby. ActivityWatch importer, který by měl na starosti získávání informací o aktivitě v těchto aplikacích, by tedy mohl využívat zmíněné knihovny nebo používat samotné REST API.

Podobně jako u systému Redmine, i u aplikace Gmail by pro získávání částečné aktivity stačil watcher pro webový prohlížeč, jelikož při čtení či odpovídání na konkrétní e-mail se změní titulek stránky. Avšak při psaní nového e-mailu se titulek ani URL stránky nezmění a tato informace je důležitá, jelikož psaní e-mailu může také zabrat nezanedbatelné množství času. Informace o zaslaných a přijatých e-mailech je proto vhodnější získávat přes REST API [65].

U Google Disku je možné zjišťovat přímo aktivitu vybraného souboru, složky či sdíleného disku [66]. V případě firmy Orchitech Solutions by bylo potřeba zjišťovat informace o aktivitě ve všech sdílených discích, ke kterým má uživatel přístup.

Z Google Kalendáře není důležitá interaktivita s aplikací, ale události v uživatelské kalendáři. Pro získání událostí uživatele opět existuje endpoint umožňující události filtrovat podle časového období či informace, zda uživatel odpověděl, že se události zúčastní či nikoliv [67].

6.6.4 Výsledný návrh komponent ActivityWatch

Aby ActivityWatch mohl uživateli co nejpřesněji prezentovat jeho aktivitu na časové ose, budou zapotřebí následující komponenty:

- watcher systémové aktivity,
- watcher aktivního užívání počítače,
- watcher pro uživatelův prohlížeč,
- watcher pro uživatelské IDE,
- Redmine importer,
- Slack importer,
- GitLab importer,
- importer pro Google aplikace (Gmail, Disk a Kalendář).

Pokud by uživatel chtěl v ActivityWatch porovnávat svou aktivitu se svými měřeními z aplikace TTRebel, mohl by pro tento účel také být vytvořen importer. Prototyp takového importéru vznikl v rámci diplomové práce [52].

Jelikož ActivityWatch z bezpečnostních důvodů blokuje veškerou komunikaci z jiného původu, než je lokální systém, je nutné, aby všechny zmíněné watchery a importéry bylo možné provozovat na zařízení uživatele.

V neposlední řadě bude také zapotřebí přidat do stávajícího uživatelského rozhraní ActivityWatch možnost zadat přihlašovací údaje k jednotlivým systémům a možnost spustit všechny importéry před tím, než si uživatel zobrazí časovou osu. Návrh této změny uživatelského rozhraní v práci z důvodu limitovaného rozsahu uveden nebude.

6.7 Synchronizace jednotlivých komponent navrhovaného řešení

Z technického pohledu na navržené řešení se jedná o architekturu klient-server. Všechny komponenty nějakým způsobem komunikují s backendovou částí aplikace TTRebel, která obsahuje hlavní logiku pro měření času, komunikuje se systémem Redmine a má přístup k datům uživatele. Provede-li uživatel nějakou změnu v datech pomocí jedné komponenty, změny by se měly dostat do všech ostatních komponent, aby uživatel měl vždy k dispozici aktuální data.

Rozšíření prohlížeče bude se serverem komunikovat jen při vyhledávání Redmine úloh a ActivityWatch pouze ve chvíli, kdy si uživatel bude chtít v aplikaci zobrazit svá měření, jejich synchronizace je tedy dostatečná bez nutnosti dalších zásahů. Avšak zbývající dvě komponenty, tedy webový a desktopový klient aplikace TTRebel, nemají v základu žádný mechanismus pro zjišťování změn v datech. Pro práci s těmito komponentami je však naprosto zásadní, aby uživatel měl k dispozici aktuální data a pokud je mít nebude, může být snadno zmaten a nedopatřením může vytvořit nekonzistenci v datech nebo dokonce i data nenávratně přepisovat. Řešením v tomto případě může být buď periodické dotazování serveru na změny v datech nebo přenechání odpovědnosti na samotném serveru, který se bude starat o informování komponent o změnách. Druhá zmíněná možnost je preferovaná a může být implementována pomocí již zmíněné technologie „Server Sent Events“.

Implementace a testování interaktivních prototypů

Navržené řešení popsané v kapitole 6 tvoří čtyři komponenty, jejichž samotný návrh byl pracný a implementace interaktivního prototypu každé komponenty by přesahovala rozsah této práce. Po dohodě se zadavatelem práce byl implementován a otestován pouze interaktivní prototyp desktopového klienta aplikace TTRebel.

7.1 Implementace prototypu desktopového klienta aplikace TTRebel

7.1.1 Výběr technologií

Při výběru technologií pro implementaci interaktivního „high-fidelity“ prototypu bylo nutné učinit rozhodnutí, zda má být využit již existující zdrojový kód webové aplikace, nebo bude aplikace implementována zcela nanovo. Vzhledem k tomu, že velká část uživatelského rozhraní i logiky aplikace bude stejná nebo alespoň podobná webové verzi aplikace, autor práce se rozhodl existující zdrojový kód využít.

V dnešní době existují různé možnosti, jak z webové aplikace udělat multiplatformní desktopovou aplikaci, nejfrekventovanější řešení jsou následující:

- využít placenou službu, např. ToDesktop [68],
- udělat z aplikace tzv. „progresivní webovou aplikaci“ (běžně také zkráceně „PWA“),
- zakomponovat do aplikace Electron [69].

Využití placené služby kvůli implementaci prototypu v rámci diplomové práce bylo zavrženo. Zbývající dvě možnosti připadaly v úvahu a vyžadovaly

blíží prozkoumání. V následujících několika odstavcích budou obě možnosti krátce představeny.

7.1.1.1 Progresivní webové aplikace

Aby webová aplikace mohla být označena za progresivní, její UX musí být na úrovni nativní aplikace pro mobilní telefon či desktop. Takové aplikace se vyznačují několika vlastnostmi, nejdůležitějšími jsou rychlá odezva, schopnost fungovat bez připojení k internetu, využívání funkcionalit nabízených operačním systémem (např. přístup k systémovým notifikacím), responzivní uživatelské rozhraní a možnost nainstalovat aplikaci do systému.

Značnou část zmíněných vlastností mají webové aplikace díky moderním webovým prohlížečům. Pokud webová aplikace obsahuje potřebnou konfiguraci, je možné ji na cílový systém nainstalovat z prostředí webového prohlížeče. Při spuštění nainstalované aplikace se pak spustí samostatné okno (lehce upraveného) webového prohlížeče, ve kterém je daná aplikace načtena. Okno webového prohlížeče neobsahuje adresní řádek, kontextové menu apod., aplikace tedy skutečně působí jako nativní. [70]

Progresivní webové aplikace jsou tedy už ze své podstaty multiplatformní – na cílové platformě je pouze zapotřebí moderní prohlížeč, který takové aplikace podporuje. Dnes již všechny nejpoužívanější moderní prohlížeče progresivní webové aplikace do jisté míry podporují (pouze Safari má několik omezení, jako např. plnou podporu systémových notifikací). [71]

To jediné, co musí vývojáři webové aplikace sami zajistit, aby aplikaci bylo možné nainstalovat na cílový systém, je konfigurační soubor zvaný „Web App Manifest“ a skript psaný v JavaScriptu zvaný „Service Worker“. Web App Manifest obsahuje informace o aplikaci potřebné k jejímu používání na cílovém systému, jako je např. název, ikony pro jednotlivé platformy či úvodní URL. Service Worker primárně poskytuje přístup k systémovým notifikacím a udává, jak se má aplikace chovat bez připojení k internetu. [72]

7.1.1.2 Electron

Electron je open-source framework pro vytváření multiplatformních desktopových aplikací pomocí technologií určených pro webové aplikace – HTML, JavaScript a CSS. Existující webovou aplikaci tedy není nutné přepisovat do nativního jazyka pro každý systém zvlášť, stačí použít Electron a pro každý cílový systém sestavit desktopovou aplikaci.

Obdobně jako zmíněná technologie PWA i Electron využívá funkcionalit webového prohlížeče pro vykreslování webové aplikace, konkrétně se jedná o open-source základ některých moderních webových prohlížečů Chromium. Pro komunikaci s operačním systémem je použito JavaScriptové běhové prostředí Node.js. Obě tyto části jsou spuštěny jako samostatný proces a navzájem

mezi sebou komunikují pomocí mechanismů pro meziprocesovou komunikaci (angl. nazýváno „Inter-Process Communication“, zkráceně „IPC“).

Konverze existující webové aplikace na desktopovou aplikaci pomocí frameworku Electron spočívá v implementaci Node.js skriptu, který se stará především o spuštění webového prohlížeče Chromium spolu s webovou aplikací a o případnou komunikaci s webovou aplikací. Electron nabízí snadno použitelné API s řadou možností, jak si aplikaci přizpůsobit, např. je možné určit rozměry a umístění okna s aplikací na obrazovce, přidat do aplikace vlastní lištu s nabídkou, přizpůsobit chování při zavření okna s aplikací aj. [73]

Pro sestavení Electron aplikace existuje několik nástrojů, nejčastěji zmiňovanými jsou Electron Builder [74] a Electron Forge [75]. Tyto nástroje umožňují sestavit aplikaci pro různé systémy v různých podobách (instalovatelná aplikace, nebo rovnou spustitelná). Aby uživatelé aplikace nebyli nuceni instalovat si další závislosti do svého systému, součástí sestavené Electron aplikace jsou také binární soubory frameworku, které mimo jiné zahrnují i Chromium a kvůli tomu může mít sestavená aplikace velikost několik desítek megabytů. [76]

7.1.1.3 Finální výběr technologie

Úprava webové aplikace tak, aby splňovala požadavky na progresivní webovou aplikaci, se zprvu zdála být vhodnou a především nejméně náročnou možností – základní verze konfigurace Web App Manifest a skriptu Service Worker se ve webové aplikaci TTRebel již nachází a vyžadovaly by pouze menší úpravy. Progresivní webové aplikace však neumožňují reagovat na klávesové zkratky na úrovni celého systému, což je pro navrhovanou aplikaci klíčová funkcionality, bez které se neobejde.

Podle oficiální dokumentace frameworku Electron se zdálo, že vývoj desktopové aplikace bude také poměrně snadný. Framework navíc nabízí řadu velice užitečných přizpůsobení a především podporuje klávesové zkratky na úrovni celého systému, volba technologie tedy nakonec padla na Electron.

7.1.2 Popis implementace

Základem implementace byl již existující zdrojový kód webové aplikace TTRebel, konkrétně frontendového projektu psaného v JavaScriptovém frameworku Vue.js. Backendový projekt nebylo potřeba pro účely prototypu nijak upravovat a byl pouze využit za účelem ladění implementace.

Do frontendového projektu byly přidány závislosti na Electron a nástroj „vue-cli-plugin-electron-builder“ [77]. Tento nástroj využívá nástroje Electron Builder, který slouží k sestavování aplikací pro cílové systémy, a zajišťuje jeho správnou konfiguraci potřebnou pro sestavení aplikace psané ve Vue.js. Jediné, co poté zbývalo manuálně nakonfigurovat, byly detaily, jako název aplikace, cesty k ikonám apod. Tím bylo vyřešeno sestavování původní webové aplikace

a zbývalo implementovat samotné uživatelské rozhraní ve Vue.js a Node.js script, který je při spuštění Electron aplikace spuštěn na pozadí a stará se o načítání uživatelského rozhraní a komunikaci s operačním systémem.

Při implementaci webové aplikace byly využity některé komponenty z původní webové aplikace, především ty, které mají na starosti správu centrálního stavu aplikace s použitím knihovny zvané „Vuex“ [78]. Některé komponenty uživatelského rozhraní byly také využity, např. tlačítko pro spuštění/zastavení měření, většina uživatelského rozhraní se však skládá z nových komponent. I nové komponenty jsou převážně založeny na komponentách poskytovaných knihovnou Vuetify, která implementuje Material Design pro Vue.js.

Implementaci Node.js skriptu výrazně usnadnila oficiální dokumentace a snadno použitelné API frameworku. Primárním účelem tohoto skriptu je spouštění, vyvolávání a zavírání upraveného okna s prohlížečem, ve kterém je následně načtena webová aplikace. Skript dále naslouchá a reaguje na systémové klávesové zkratky, stará se o změnu ikony aplikace zobrazené v notifikační oblasti systému a komunikuje s původní backendovou částí aplikace TTRebel. K těmto zmíněným úkonům je zapotřebí komunikace s webovou aplikací (uživatelským rozhraním), která bude podrobněji popsána v následující sekci.

7.1.2.1 Meziprocesová komunikace

Meziprocesová komunikace v Electron aplikaci je z pohledu vývojáře založena na událostech. Probíhá tak, že jeden proces (Node.js skript, nebo webová aplikace) vyšle specifickou událost a druhý proces na tuto konkrétní událost zareaguje. U implementovaného prototypu je tento způsob komunikace využíván v následujících situacích:

- Node.js skript zaznamená stisknutí klávesové zkratky pro zastavení spuštěného měření a tuto skutečnost komunikuje webové aplikaci, která podle toho překreslí uživatelské rozhraní.
- Uživatel přes uživatelské rozhraní spustí či zastaví nějaké měření, webová aplikace tuto skutečnost komunikuje Node.js skriptu, který změní ikonu aplikace a upraví položky v kontextovém menu.
- Webová aplikace potřebuje zaslat HTTP požadavek na backendovou část aplikace TTRebel, předá tedy potřebné informace k uskutečnění požadavku Node.js skriptu, ten pošle požadavek a obdrženou odpověď komunikuje zpět webové aplikaci.

7.1.2.2 Komunikace s backendovou částí aplikace TTRebel

Původní webová aplikace TTRebel komunikuje s backendovou částí aplikace pomocí modelu AJAX [79] stejným způsobem, jako to webové aplikace dnes

již běžně dělají. V implementovaném prototypu však tento způsob komunikace nebylo možné ponechat kvůli bezpečnostnímu mechanismu v prohlížeči Chromium zvanému „same-origin policy“. Tento mechanismus znemožňuje plnohodnotnou komunikaci mezi JavaScriptovým kódem webové aplikace a servery jiného původu, přičemž původ udává protokol, doména a volitelně i port, který však nemusí být specifikovaný [80]. Původní webová aplikace TTRebel má na produkčním prostředí stejný původ jako backendová část, tam se tedy tento problém nevyskytuje, implementovaný prototyp však bude používán jako samostatná aplikace s odlišným původem.

Zmíněný problém se dá vyřešit úpravou na straně aplikačního serveru, ten by však musel přistoupit na komunikaci s aplikacemi jakéhokoliv původu, což není vhodné řešení. Přijatelnější úpravou by bylo povolení komunikace s aplikacemi z interní sítě firmy, toto řešení je však pro zaměstnance příliš restriktivní. Jelikož zmíněná ochrana není přítomna v komunikaci mezi servery, nejvhodnějším řešením v tomto případě bylo přenechat komunikaci s backendovou částí aplikace na Node.js skriptu. Komunikace je vždy iniciována uživatelským rozhraním, které s Node.js skriptem komunikuje již popsáním způsobem.

7.1.3 Omezení prototypu

Implementovaný prototyp nedokáže automaticky reagovat na akce provedené ve webové aplikaci TTRebel, pokud tedy uživatel např. upraví popis práce u nějakého měření, v prototypu se tato změna neprojeví automaticky. Implementace této funkcionality by vyžadovala značné úpravy backendové části aplikace, proto bylo na seznam uživatelských měření přidáno alespoň tlačítko pro manuální synchronizaci změn do prototypu, které je vidět na obrázku 7.1.

V prototypu také chybí možnost nastavit si vlastní systémové klávesové zkratky, ty byly napevno nastaveny následovně:

- Ctrl + E – vyvolání okna s aplikací
- Ctrl + Shift + S – zastavení spuštěného měření

Do prototypu také nebyla implementována možnost vybrat si obrazovku, na které bude vyvoláváno okno s aplikací – to je vždy vyvoláváno na té obrazovce, která je v systému označena jako primární. Za tímto rozhodnutím stojí předpoklad, že uživatel chce mít okno okamžitě na očích, aby mohl co nejrychleji vykonat požadované akce a vrátit se zpět ke své práci.

7.2 Testování prototypu desktopového klienta aplikace TTRebel s uživateli

Po dokončení implementace první verze prototypu byl prototyp sestaven pro systémy Windows a Linux a byla otestována funkčnost na obou systémech.

Poté byl prototyp poskytnut několika zaměstnancům, aby jej vyzkoušeli během svého pracovního dne a následně na něj poskytli zpětnou vazbu. Po získání zpětné vazby na první verzi prototypu bylo v prototypu provedeno několik málo úprav a zaměstnancům byla na otestování poskytnuta druhá a zároveň finální verze. Níže budou podrobněji popsány zpětné vazby zaměstnanců na obě verze.

7.2.1 První verze prototypu

Reakce na první verzi prototypu byly z většiny pozitivní, zaměstnancům vyhovovalo uživatelské rozhraní a příjemně se jim aplikace ovládala, především těm, kteří se v rámci původních hloubkových rozhovorů vyjádřili, že preferují ovládání aplikací klávesnicí. Za největší přínos zaměstnanci označili možnost rychlého vyvolání okna pomocí klávesové zkratky, díky kterému mohou téměř okamžitě spravovat svá měření. Velký přínos pro zaměstnance měla také ikona aplikace zobrazená v notifikační oblasti systému, díky které mohou pouze letmým pohledem zjistit, zda si náhodou nezapomněli zapnout měření.

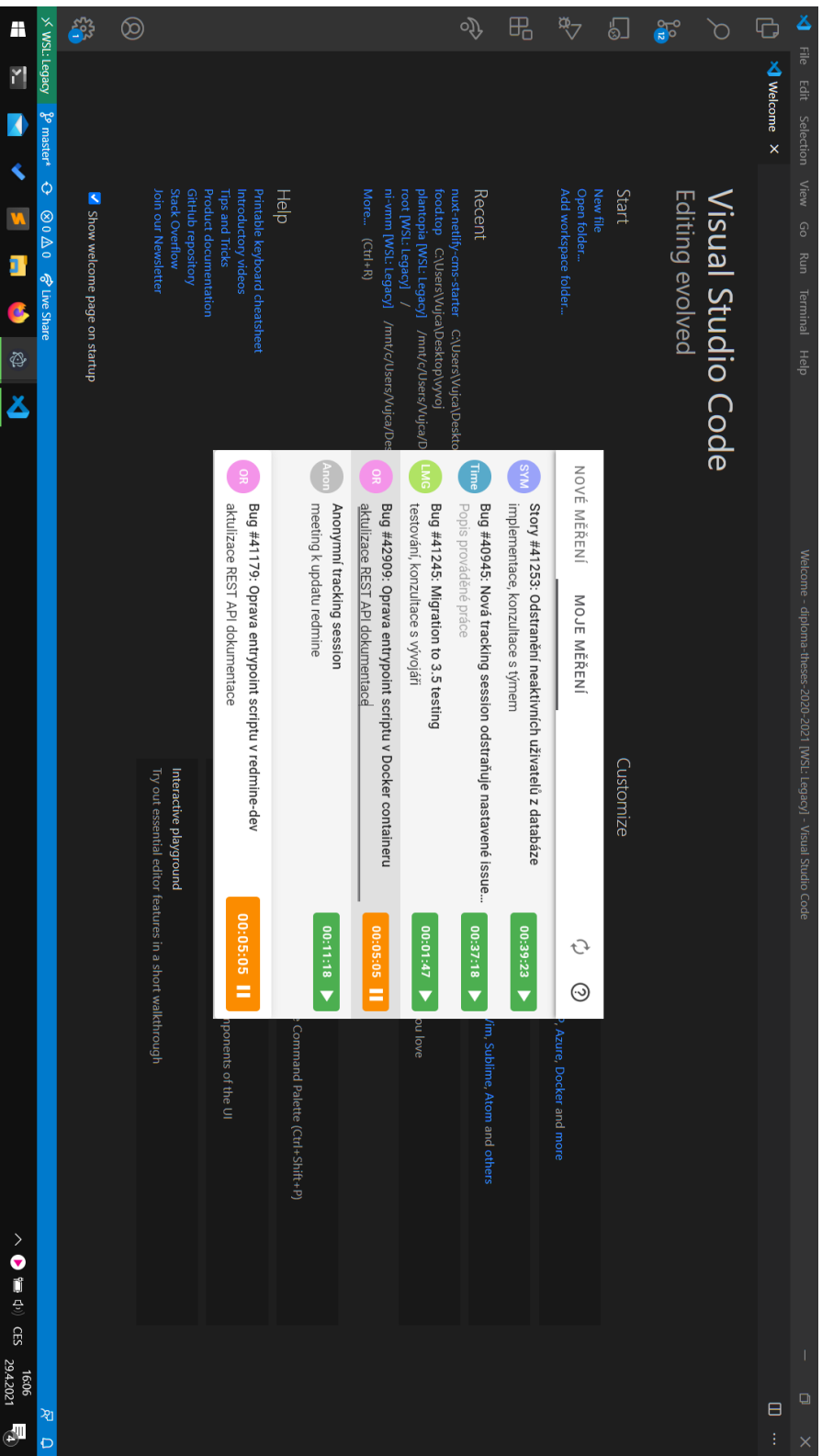
Co se negativních aspektů týče, polovině zaměstnanců nevyhovovalo, že se aplikace vždy otevírá na primárním monitoru a toto umístění není možné nijak změnit. Zaměstnanci také narazili na chybu, kvůli které bylo potřeba aplikaci restartovat každý den před používáním, jelikož stále zobrazovala měření z předchozího dne. Jednomu zaměstnanci přišlo písmo naměřeného času na tlačítkách pro spuštění/zastavení měření příliš malé a prý měl občas s jeho přečtením problémy. Jeden ze zaměstnanců se vyjádřil, že ikona nepoutá dostatečně jeho pozornost a bylo by vhodnější, kdyby byla nějakým způsobem animovaná.

7.2.2 Druhá verze prototypu

Druhá verze prototypu se snažila opravit nedostatky identifikované u první verze. Aby si uživatelé mohli sami určit, na jaké obrazovce se bude okno s aplikací otevírat, byl oknu přidán rámeček, který je pro desktopové aplikace běžný. Za tento rámeček je možné okno s aplikací přesunout a při dalším vyvolání se okno otevře na stejném místě. Nalezená chyba, kvůli které bylo potřeba aplikaci každý den před používáním restartovat, byla opravena. Písmo naměřeného času na tlačítkách pro spuštění/zastavení měření bylo i se samotnými tlačítky zvětšeno a aby kvůli tomu neubyl prostor pro údaje o měřeních, bylo i okno aplikace trochu rozšířeno. Animace ikony byla odložena z časových důvodů a kvůli tomu, že chyběla pouze jedinému zaměstnanci.

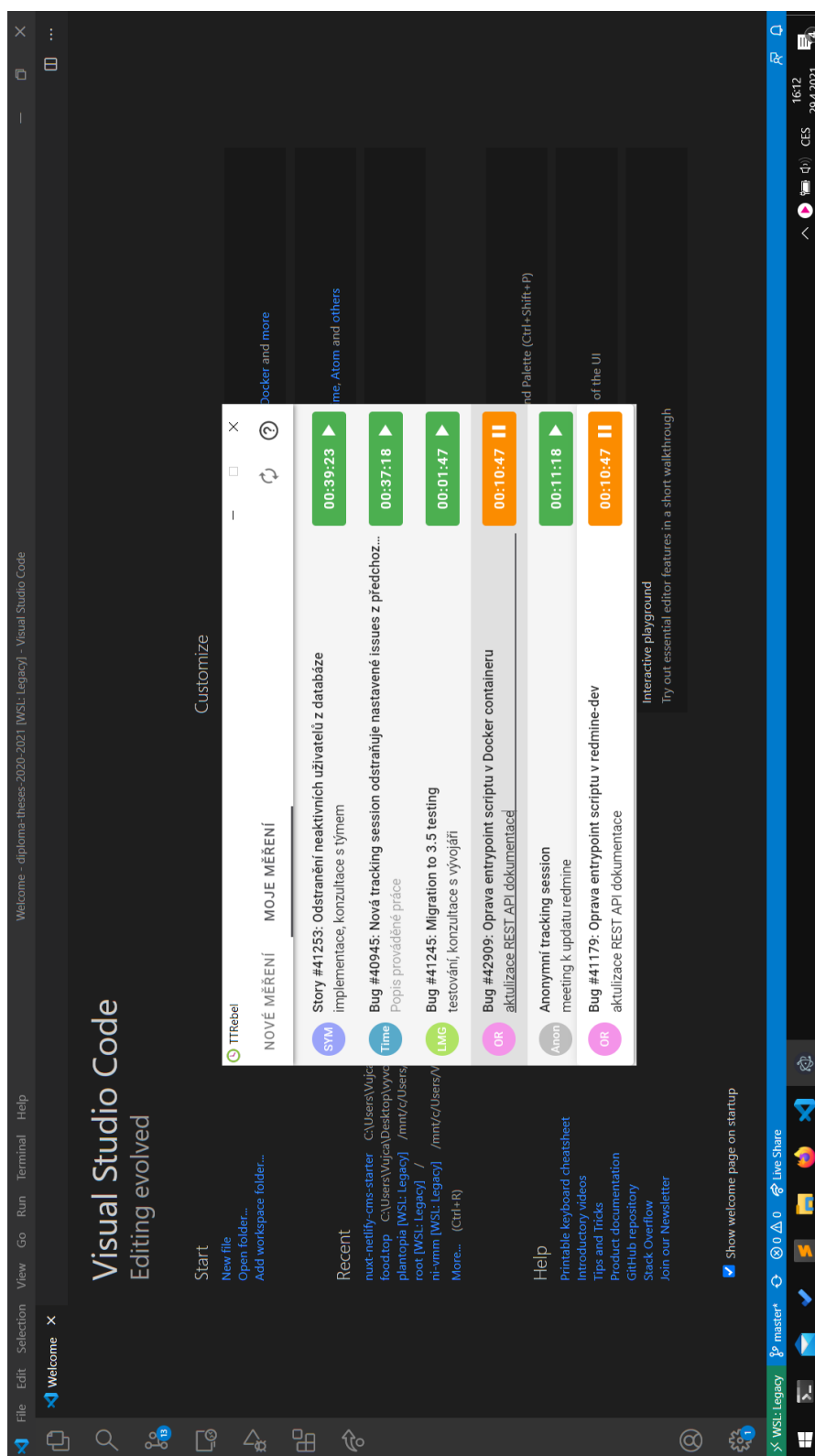
Testování druhé verze prototypu zaměstnanci nepřineslo žádné nové poznatky, s úpravami byli spokojeni a tak byla druhá verze prototypu prohlášena za finální.

7.2. Testování prototypu desktopového klienta aplikace TTRebel s uživateli



Obrázek 7.1: Snímek obrazovky s první verzí prototypu desktopové aplikace TTRebel používané na systému Windows

7. IMPLEMENTACE A TESTOVÁNÍ INTERAKTIVNÍCH PROTOTYPŮ



Obrázek 7.2: Snímek obrazovky s druhou verzí prototypu desktopové aplikace TTRebel používané na systému Windows

Zhodnocení přínosu navrženého řešení

8.1 Připomenutí metrik

V sekci 4.2 byly definovány metriky, jejichž vyhodnocením a následným srovnáním hodnot před a po nasazení navrženého řešení by mělo být možné zhodnotit přínos navrženého řešení. Metriky jsou následovné:

- Čas strávený vykazováním – poměr času stráveného sledováním a vykazováním odpracovaného času ku skutečně odpracovanému času
- Úroveň rozptýlení – úroveň rozptýlení od současné práce způsobeného nutností sledovat a vykazovat čas
- Zpoždění výkazů – doba, za kterou se výkazy dostanou k odpovídající Redmine úloze po dokončení vykazované činnosti

8.2 Výběr reprezentativního vzorku zaměstnanců

V sekci 4.3 byly stanoveny kategorie, do kterých je možné všechny zaměstnance firmy roztrždit. Pro vyhodnocování metrik by tedy bylo ideální mít dostupná data o zaměstnancích z každé kategorie, aby vzorek vybraných zaměstnanců dostatečně reprezentoval většinu zaměstnanců firmy. Avšak z důvodu časové nedostupnosti měla pouze polovina (tedy šest) zaměstnanců možnost vyzkoušet implementovaný prototyp desktopové aplikace TTRebel a tito zaměstnanci bohužel nepokrývají všechny stanovené kategorie. Nelze tedy bohužel prohlásit, že metriky budou vyhodnoceny na zcela reprezentativním vzorku zaměstnanců. Jelikož je důležité porovnávat vyhodnocené metriky vždy u stejných zaměstnanců, budou uvedeny pouze hodnoty metrik týkající se těchto šesti zaměstnanců.

Co se týče pokrytých kategorií, tři zaměstnanci spadají do kategorie aktivních uživatelů aplikace TTRebel vykazujících pouze do hlavního systému Redmine, jeden zaměstnanec spadá do kategorie aktivních uživatelů aplikace TTRebel vykazující do obou systémů Redmine a dva zaměstnanci spadají do kategorie zaměstnanců užívajících jiný nástroj pro sledování času (konkrétně Toggl).

8.3 Vyhodnocení metrik před nasazením navrženého řešení

Za účelem získat potřebná data pro vyhodnocení stanovených metrik před nasazením navrženého řešení bylo třeba využít několik zdrojů informací. Prvním zdrojem byla data nasbíraná v rámci automatizovaného sběru informací o chování uživatelů aplikace TTRebel, které je blíže popsáno v sekci 5.2.2. Z nasbíraných dat bylo možné vypočítat čas strávený aktivním používáním aplikace. Druhým zdrojem informací byly hloubkové rozhovory se zaměstnanci, během kterých měli zaměstnanci sami odhadnout čas, který stráví sledováním a vykazováním času, dále měli určit úroveň rozptýlení a odhadnout medián zpoždění svých výkazů. Pro účely této práce byl autorovi práce poskytnut přístup k datům aplikace TTRebel, ze kterých bylo možné určit zpoždění, se kterým se výkazy uživatelů aplikace dostávají do systému Redmine. Posledním zdrojem informací byl samotný systém Redmine, ze kterého bylo možné zjistit podrobnější informace o výkazech zaměstnanců, jako je vykázaný čas za konkrétní období.

Následující vyhodnocení jsou založena na datech pocházejících z výše zmíněných zdrojů a období jedenácti pracovních dnů. Zmíněných šest zaměstnanců, kterých se data týkají, bude dále označováno písmeny A až F a každé dané písmeno bude vždy označovat stejného zaměstnance. Zaměstnanci A, B, C, a D jsou aktivní uživatelé aplikace TTRebel, zaměstnanci E a F nejsou uživatelé aplikace TTRebel a aktivně používají aplikaci Toggl.

8.3.1 Čas strávený vykazováním

Čas strávený sledováním a vykazováním odpracovaného času byl u zaměstnanců A, B, C a D získán automatickým sběrem informací o chování uživatelů aplikace TTRebel. U uživatelů E a F jsou tyto hodnoty založeny na odhadech, které zaměstnanci poskytli během hloubkových rozhovorů. Skutečně odpracovaný čas byl u všech zaměstnanců získán z výkazů v systému Redmine za zmíněné období jedenácti pracovních dnů.

U zaměstnanců A, B, C a D byl poměr času stráveného používáním aplikace TTRebel ku skutečně odpracovanému času vypočítán pro každý den zvoleného období a z těchto poměrů byl vypočítán průměr. U zaměstnanců E a F byly poměr pro každý den a výsledný průměr vypočítány stejně, akorát čas

8.3. Vyhodnocení metrik před nasazením navrženého řešení

strávený sledováním a vykazováním času byl založen na odhadu poskytnutém zaměstnanci.

Označení zaměstnance	Čas strávený vykazováním [min]
A	1,1
B	2,5
C	1,2
D	0,8
E	0,6
F	0,3

Tabulka 8.1: Přehled hodnot metriky „čas strávený vykazováním“ u jednotlivých zaměstnanců před použitím navrženého řešení

8.3.2 Úroveň rozptýlení

Úroveň rozptýlení byla od všech zaměstnanců zjišťována v rámci hloubkových rozhovorů. Na výběr měli ze čtyř možností, které jsou popsány v sekci 4.2. Pro větší přehlednost byly tyto možnosti očíslovány, význam hodnot v tabulce 8.2 je tedy následovný:

- 0 – „Sledování a vykazování času stráveného prací mě vůbec nerozptyluje.“
- 1 – „Sledování a vykazování času stráveného prací mě mírně rozptyluje, ale neovlivňuje to moji efektivitu při práci.“
- 2 – „Sledování a vykazování času stráveného prací mě rozptyluje a ovlivňuje to moji efektivitu při práci.“
- 3 – „Sledování a vykazování času stráveného prací mě velice rozptyluje a výrazně to ovlivňuje moji efektivitu při práci.“

Označení zaměstnance	Vybraná úroveň rozptýlení
A	1
B	0
C	1
D	0
E	1
F	1

Tabulka 8.2: Přehled hodnot metriky „úroveň rozptýlení“ u jednotlivých zaměstnanců před použitím navrženého řešení.

8.3.3 Zpoždění výkazů

U zaměstnanců A, B, C a D bylo zpoždění výkazů vypočítáno pomocí dat z aplikace TTRebel, konkrétně jako medián rozdílů data exportování a data vytvoření každého měření vytvořeného ve zkoumaném období. U zaměstnanců E a F nebylo možné tuto informaci zjistit jinak než odhadem poskytnutým samotnými zaměstnanci.

Označení zaměstnance	Zpoždění výkazů [hod]
A	40.5
B	6.3
C	65.4
D	191.8
E	48
F	720

Tabulka 8.3: Přehled hodnot metriky „zpoždění výkazů“ u jednotlivých zaměstnanců před použitím navrženého řešení

8.4 Vyhodnocení metrik po nasazení navrženého řešení

Hned na začátek této sekce je nutné zmínit, že zaměstnanci mohli vyzkoušet pouze část navrženého řešení, kterou je interaktivní prototyp desktopového klienta aplikace TTRebel. Vyhodnocení metrik po používání tohoto prototypu tedy nevyovídá o přínosu kompletního navrženého řešení, ale spíše o potenciálním přínosu desktopového klienta aplikace TTRebel.

Zmíněných šest zaměstnanců zapojených do tohoto vyhodnocování dobrovolně souhlasilo se zapojením prototypu do svého běžného pracovního režimu po dobu jednoho týdnu. Po týdnu byly se zaměstnanci provedeny hloubkové rozhovory, v rámci kterých poskytovali zpětnou vazbu na prototyp a odhadovali, jak používání prototypu ovlivnilo jejich metriky. Jelikož se u všech zaměstnanců jedná pouze o odhady změn, vyhodnocování metrik způsobem jako před nasazením navrženého řešení by nebylo příliš vypovídající, proto budou dále uvedeny pouze odhady zaměstnanců s kontextem z hloubkových rozhovorů.

8.4.1 Čas strávený vykazováním

Zaměstnanci A, B, C a D byli seznámeni s časy používání aplikace získanými pomocí automatického sbírání informací o používání aplikace TTRebel a byli dotázáni, zda si myslí, že se používáním prototypu desktopového klienta aplikace TTRebel tyto časy nějak změní.

Zaměstnanci A, B a D téměř shodně uvedli, že mají webovou aplikaci běžně otevřenou na samostatné obrazovce nebo v samostatném okně prohlížeče, kde jim je relativně dobře dostupná. Nativní aplikace tudíž pro tyto zaměstnance nepřináší takovou výhodu co se týče dostupnosti, uvedli však, že možnost vyvolávat okno s aplikací pomocí klávesové zkratky by jim mohlo ušetřit přibližně jednu minutu denně.

Zaměstnanec C uvedl, že mu prototyp výrazně usnadňuje správu svých měření v aplikaci TTRebel během dne a myslí si, že čas strávený sledováním odpracovaného času se mu zkrátil o polovinu. Jako hlavní důvody uvedl možnost vyvolat okno s aplikací pomocí klávesové zkratky a ikonu aplikace v notifikační oblasti systému, díky které si na spravování svých měření vzpomene častěji a tím pádem později neztrácí čas se vzpomínáním a zpětným upravováním svých měření.

Zaměstnanec E také chválil především možnost vyvolat okno s aplikací pomocí klávesové zkratky a odhadl, že kdyby měl více času zvyknout si na tuto aplikaci a také webovou aplikaci TTRebel, jeho čas strávený sledováním času by se mohl snížit přibližně o jednu a půl minuty.

Protipólem k ostatním zaměstnancům je zaměstnanec F, který uvedl, že mu velice vyhovuje ovládání aplikace klávesnicí a že kdyby měl více času zvyknout si na tuto aplikaci a více motivace přestat používat aplikaci Toggl, jeho čas strávený sledováním odpracovaného času by se „snad nemusel o mnoho zvýšit“. Zaměstnanec však uznal, že používáním webové aplikace TTRebel by ušetřil nezanedbatelné množství času při vytváření výkazů v Redmine, ve výsledku by mu tedy navržené řešení nějaký čas určitě ušetřilo.

8.4.2 Úroveň rozptýlení

I přes to, že se většina zaměstnanců vyjádřila, že jim u prototypu vyhovuje možnost téměř okamžitě vyvolat okno s aplikací pomocí klávesové zkratky, již při vyhodnocování metrik před nasazením navrženého měření uváděli, že je jejich dosavadní způsob sledování a vykazování času téměř nerozptyluje a jejich odpovědi se ani po používání prototypu nezměnily.

Je možné, že toto je důsledkem nevhodně zvolené reprezentace rozptýlení od práce způsobené nutností sledovat čas strávený prací. Kdyby např. bylo na výběr více odpovědí nebo pouze číselná škála většího rozsahu bez konkrétních odpovědí, odpovědi uživatelů by možná byly odlišné.

8.4.3 Zpoždění výkazů

Vzhledem k tomu, že desktopový klient aplikace TTRebel má uživatelům pomáhat především se sledováním času během dne a nenabízí možnost exportovat měření do Redmine, je nepravděpodobné, že by samotný prototyp zpoždění výkazů nějakým způsobem ovlivnil. Také většina samotných zaměstnanců v rozhovorech uvedla, že si nemyslí, že by jim desktopový klient pomohl zkrátit

čas, za který se dostanou výkazy s pracovními činnostmi do Redmine. Pouze zaměstnanec C se vyjádřil, že díky desktopovému klientovi si svůj čas strávený prací měří mnohem častěji a přesněji než při používání původní webové aplikace TTRebel a věří, že důsledkem toho mohou občas nastávat situace, kdy měření do Redmine exportuje dříve, protože jejich finální úpravy před exportem nebudou zabírat tolik času.

8.5 Výsledné zhodnocení

Jak již bylo avizováno, pomocí navržených metrik není zcela možné zhodnotit přínos navrženého řešení jako celku, ale spíše přínos pouze jeho části, kterou je desktopový klient aplikace TTRebel. Z hloubkových rozhovorů se zaměstnanci, kteří měli možnost vyzkoušet prototyp aplikace, vyplynulo, že tato část řešení jistě dokáže některým zaměstnancům ušetřit čas strávený sledováním času během běžného pracovního dne. Rozptýlení zaměstnanců od jejich práce způsobené nutností sledovat svůj odpracovaný čas tato samotná část řešení pravděpodobně nesníží nebo pouze minimálně. Zpoždění, se kterým se výkazy dostávají do systému Redmine, tato samotná část řešení zřejmě také nedokáže snížit, alespoň u většiny zaměstnanců. Dle odpovědí dotázaných zaměstnanců však lze usoudit, že tato samotná část řešení rozptýlení zaměstnanců ani zpoždění výkazů naopak ani nezvýší.

Závěr

Hlavním cílem této práce bylo navrhnout řešení pro přesné, efektivní a pohodlné sledování a následné vykazování času stráveného projektovými aktivitami. Návrh měl být založen na podrobné analýze a zhodnocení UX současných nástrojů zadavatele a na požadavcích a potřebách zaměstnanců firmy zadavatele Orchitech Solutions. Návrh řešení měl dále být primárně zaměřený na kvalitní UX, jenž je hlavním nedostatkem u mnoha nástrojů pro sledování času, které zaměstnanci firmy vyzkoušeli.

Všech vytyčených cílů práce se podařilo dosáhnout. Autor práce osobně analyzoval a zhodnotil UX současného nástroje zadavatele TTRebel, nejprve na základě volného průchodu aplikací a následně metodou heuristické evaluace zaměřenou na použitelnost aplikace. Za účelem hlubšího pochopení potřeb a požadavků zaměstnanců firmy byly se všemi zaměstnanci provedeny kvalitativní hloubkové rozhovory. V rámci rozhovorů byly odhaleny problémy současného nástroje, které jednotliví zaměstnanci vnímají. Díky těmto rozhovorům byl autor práce schopný identifikovat klíčové nedostatky UX současného nástroje a navrhnout řešení, které tyto nedostatky eliminuje. Návrh řešení obsahuje návrh úprav současného nástroje a návrh dalších samostatných aplikací, díky kterým bylo možné pokrýt většinu požadavků zaměstnanců. Návrh řešení se primárně zaměřuje na efektivní a pohodlné UX celkového řešení, zahrnuje podrobný popis funkcionalit jednotlivých komponent a grafické návrhy uživatelských rozhraní s vysokou úrovní věrnosti. Dále byl implementován interaktivní prototyp vybrané komponenty navrženého řešení, konkrétně desktopového klienta aplikace TTRebel, který byl řádně otestován spolu s několika zaměstnanci firmy zadavatele. Pro účely této práce byly dále definovány metriky, pomocí kterých měl původně být zhodnocen přínos celého navrženého řešení, jelikož však v rozsahu této práce nebylo možné vytvořit interaktivní prototypy všech komponent řešení, metriky byly použity pouze pro zhodnocení přínosu desktopového klienta aplikace TTRebel. Dle tohoto zhodnocení by se používáním této komponenty mohl čas strávený sledováním odpracovaného času během dne u některých zaměstnanců snížit až o 50 %.

V návaznosti na tuto diplomovou práci bude vývojový tým aplikace TTRebel seznámen s výstupy analýzy a navrženým řešením. Grafické návrhy webové aplikace TTRebel jistě poslouží při úpravách uživatelského rozhraní aplikace, které už jsou dlouhou dobu plánovány. Autor práce dále předpokládá, že na základě zpětné vazby na implementovaný interaktivní prototyp bude naplánován další vývoj minimálně této komponenty řešení.

Bibliografie

1. NORMAN, Don; NIELSEN, Jakob. The Definition of User Experience (UX) [online] [cit. 2021-04-14]. Dostupné z: <https://www.nngroup.com/articles/definition-user-experience/>.
2. NIELSEN, Jakob. Usability 101: Introduction to Usability [online] [cit. 2021-04-14]. Dostupné z: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>.
3. GIBBONS, Sarah. Design Thinking 101 [online] [cit. 2021-04-14]. Dostupné z: <https://www.nngroup.com/articles/design-thinking/>.
4. FOUNDATION, Interaction Design. UX Research [online] [cit. 2021-04-16]. Dostupné z: <https://www.interaction-design.org/literature/topics/ux-research>.
5. ROHRER, Christian. When to Use Which User-Experience Research Methods [online] [cit. 2021-04-14]. Dostupné z: <https://www.nngroup.com/articles/which-ux-research-methods/>.
6. GOOGLE. *Google Analytics* [software] [cit. 2021-04-15]. Dostupné z: <https://marketingplatform.google.com/about/analytics/>. Nástroj pro analýzu chování uživatelů od společnosti Google.
7. PERNICE, Kara. UX Prototypes: Low Fidelity vs. High Fidelity [online] [cit. 2021-04-15]. Dostupné z: <https://www.nngroup.com/articles/ux-prototype-hi-lo-fidelity/>.
8. MYHILL, Chris. How detailed should wireframes be? A guide to wireframe fidelity [online] [cit. 2021-04-15]. Dostupné z: <https://pixelfridge.digital/how-detailed-should-wireframes-be-a-guide-to-wireframe-fidelity/>.
9. STUDIOS, Balsamiq. *Balsamiq* [software] [cit. 2021-04-15]. Dostupné z: <https://balsamiq.com/>.

10. INC., Adobe. *Adobe XD* [software] [cit. 2021-04-15]. Dostupné z: <https://www.adobe.com/cz/products/xd.html>.
11. EVOLUS. *Pencil* [software] [cit. 2021-04-15]. Dostupné z: <https://pencil.evolus.vn/>. Open source projekt.
12. FIGMA. *Figma* [software] [cit. 2021-04-15]. Dostupné z: <https://www.figma.com/>.
13. INC., InVisionApp. *InVision* [software] [cit. 2021-04-15]. Dostupné z: <https://www.invisionapp.com/>.
14. B.V., Sketch. *Sketch* [software] [cit. 2021-04-15]. Dostupné z: <https://sketch.com/>.
15. HARLEY, Aurora. UX Expert Reviews [online] [cit. 2021-04-16]. Dostupné z: <https://www.nngroup.com/articles/ux-expert-reviews/>.
16. NIELSEN, Jakob. How to Conduct a Heuristic Evaluation [online] [cit. 2021-04-16]. Dostupné z: <https://www.nngroup.com/articles/how-to-conduct-a-heuristic-evaluation/>.
17. NIELSEN, Jakob. Severity Ratings for Usability Problems [online] [cit. 2021-04-14]. Dostupné z: <https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/>.
18. NIELSEN, Jakob. 10 Usability Heuristics for User Interface Design [online] [cit. 2021-04-14]. Dostupné z: <https://www.nngroup.com/articles/ten-usability-heuristics/>.
19. *Redmine* [software] [cit. 2021-04-15]. Dostupné z: www.redmine.org. Open-source nástroj pro řízení projektů.
20. TECHNOLOGIES, Slack. *Slack* [software] [cit. 2021-04-15]. Dostupné z: <https://slack.com/>.
21. GOOGLE. *Google Workspace* [software] [cit. 2021-04-15]. Dostupné z: <https://workspace.google.com>. Sada aplikací od Google.
22. GITLAB. *GitLab* [software] [cit. 2021-04-15]. Dostupné z: <https://about.gitlab.com/>.
23. TOGGL. *Toggl* [software] [cit. 2021-04-04]. Dostupné z: <https://toggl.com/>. Placený nástroj pro sledování času.
24. DEVART. *TMetric* [software] [cit. 2021-04-04]. Dostupné z: <https://tmetric.com/>. Placený nástroj pro sledování času.
25. CLOCKIFY. *Clockify* [software] [cit. 2021-04-04]. Dostupné z: <https://www.clockify.me/>. Nástroj pro sledování času.
26. TIME DOCTOR, LLC. *Time Doctor* [software] [cit. 2021-04-04]. Dostupné z: <https://www.timedoctor.com/>. Placený nástroj pro sledování času.

27. TIMEULAR. *Timeular* [software] [cit. 2021-04-04]. Dostupné z: <https://timeular.com/>. Placený nástroj pro sledování času.
28. PAPST, Kevin. *Kimai* [software] [cit. 2021-04-04]. Dostupné z: <https://www.kimai.org/>. Open source projekt.
29. Toggl [online] [cit. 2021-04-04]. Dostupné z: <https://toggl.com/>.
30. Toggl Integrations List [online] [cit. 2021-04-04]. Dostupné z: <https://toggl.com/track/integrations/>.
31. Toggl Integration with Redmine [online] [cit. 2021-04-04]. Dostupné z: <https://toggl.com/track/redmine-time-tracking/>.
32. Timeular [online] [cit. 2021-04-04]. Dostupné z: <https://timeular.com/>.
33. Timeular Tracker [online] [cit. 2021-04-04]. Dostupné z: <https://timeular.com/product/tracker/>.
34. Timeular Reviews 2021 [online] [cit. 2021-04-04]. Dostupné z: <https://www.capterra.com/p/176609/ZEI-by-Timeular/reviews/>.
35. Timeular Roadmap Feedback [online] [cit. 2021-04-04]. Dostupné z: <https://portal.productboard.com/hcqkznnlljkuyqw9xunagicg/c/39-integration-redmine>.
36. AS, Memory. *Timely* [software] [cit. 2021-04-04]. Dostupné z: <https://memory.ai/timely>. Placený nástroj pro automatické sledování času.
37. *Selfspy* [software] [cit. 2021-04-04]. Dostupné z: <https://github.com/selfspy/selfspy>. Open source projekt.
38. *uologme* [software] [cit. 2021-04-04]. Dostupné z: <https://github.com/karpathy/uologme>. Open source projekt.
39. RESCUETIME. *RescueTime* [software] [cit. 2021-04-15]. Dostupné z: <https://www.rescuetime.com/>. Placený nástroj pro automatické sledování času stráveného u počítače.
40. ActivityWatch [online] [cit. 2021-04-04]. Dostupné z: <https://activitywatch.net/>.
41. FOUNDATION, Python Software. *Python* [software] [cit. 2020-04-04]. Dostupné z: <https://python.org/>.
42. ActivityWatch Watchers [online] [cit. 2021-04-04]. Dostupné z: <https://docs.activitywatch.net/en/latest/watchers.html>.
43. LTD., Lerus. *Redmine Time Tracker app* [software] [cit. 2021-04-04]. Dostupné z: <https://www.ukrsolution.com/Redmine-Time-Tracker-Software-Time-Logger/>. Aplikace pro sledování a vykazování času do Redmine opatřená komerční licencí.
44. *RedmineTimeTracker* [software] [cit. 2021-04-04]. Dostupné z: <https://github.com/ujiro99/RedmineTimeTracker>. Open source projekt.

45. LTD., Lerus. Redmine Time Tracker app [online] [cit. 2021-04-04]. Dostupné z: <https://www.ukrsolution.com/Redmine-Time-Tracker-Software-Time-Logger/>.
46. GitHub projekt RedmineTimeTracker [online] [cit. 2021-04-04]. Dostupné z: <https://github.com/ujiro99/RedmineTimeTracker>. Open source projekt na GitHub.
47. *Vue.js* [software] [cit. 2020-04-04]. Dostupné z: <https://vuejs.org/>. Open source framework pro vývoj v JavaScriptu.
48. PIVOTAL SOFTWARE, Inc. *Spring Boot* [software] [cit. 2019-05-15]. Dostupné z: <https://spring.io/projects/spring-boot>. Open source projekt.
49. *PostgreSQL* [software] [cit. 2019-05-05]. Dostupné z: <https://www.postgresql.org/>. Open source projekt.
50. GOOGLE. *Material Design* [online] [cit. 2021-04-15]. Dostupné z: <https://material.io/>.
51. *Vuetify* [software] [cit. 2020-04-14]. Dostupné z: <https://vuetifyjs.com/>. Open source knihovna implementující prvky Material design pro Vue.js.
52. KROČÁK, Vladimír. *Automatické sledování času pracovníků v oblasti profesionálních služeb*. České Budějovice 2019, 2019. Diplomová práce. Jihočeská univerzita v Českých Budějovicích, Přírodovědecká fakulta. Vedoucí práce Martin Čížek.
53. Snackbars - Material design [online] [cit. 2021-04-14]. Dostupné z: <https://material.io/components/snackbars>. Dokumentace Material design komponenty Snackbar.
54. Buttons: floating action button - Material design [online] [cit. 2021-04-14]. Dostupné z: <https://material.io/components/buttons-floating-action-button>. Dokumentace Material design komponenty Floating action button.
55. GOOGLE. *Analytics for Web (analytics.js)* [software] [cit. 2021-04-15]. Dostupné z: <https://developers.google.com/analytics/devguides/collection/analyticsjs>. Knihovna pro sbírání informací o chování uživatelů webových aplikací.
56. GOOGLE. *Google BigQuery* [software] [cit. 2021-04-15]. Dostupné z: <https://cloud.google.com/bigquery>. Cloudový datový sklad od společnosti Google.
57. ČÍŽEK, Martin. *IS/IT Strategy in Orchitech Solutions*. Nottingham, Velká Británie, 2014. Disertační práce. Nottingham Trent University.
58. *Chart.js* [software] [cit. 2021-04-04]. Dostupné z: <https://www.chartjs.org/>. Open source projekt.

59. Sending messages [online] [cit. 2021-04-20]. Dostupné z: <https://api.slack.com/messaging/sending>. Zaslání zpráv v dokumentaci Slack API.
60. MOZILLA. Using server-sent events [online] [cit. 2021-04-20]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/API/Server-sent_events/Using_server-sent_events.
61. ActivityWatch documentation - Architecture [online] [cit. 2021-04-20]. Dostupné z: <https://docs.activitywatch.net/en/latest/architecture.html>. Sekce dokumentace nástroje ActivityWatch se popisem architektury nástroje.
62. Watchers [online] [cit. 2021-04-20]. Dostupné z: <https://docs.activitywatch.net/en/latest/watchers.html>. Sekce dokumentace nástroje ActivityWatch se seznamem implementovaných watchers.
63. Slack API - search messages [online] [cit. 2021-04-14]. Dostupné z: <https://api.slack.com/methods/search.messages>. Dokumentace REST API nástroje Slack.
64. GitLab - Events API [online] [cit. 2021-04-14]. Dostupné z: <https://docs.gitlab.com/ee/api/events.html>. Dokumentace REST API nástroje GitLab.
65. Gmail API - Searching for Messages [online] [cit. 2021-04-14]. Dostupné z: <https://developers.google.com/gmail/api/guides/filtering>. Dokumentace REST API aplikace Gmail.
66. Google Drive Activity API Overview [online] [cit. 2021-04-14]. Dostupné z: <https://developers.google.com/drive/activity/v2>. Dokumentace REST API aplikace Google Disk.
67. Google Calendar API - Events list [online] [cit. 2021-04-14]. Dostupné z: <https://developers.google.com/calendar/v3/reference/events/list>. Dokumentace REST API aplikace Google Kalendář.
68. *ToDesktop* [software] [cit. 2021-04-15]. Dostupné z: <https://www.todesktop.com/>. Placený nástroj pro konverzi webových aplikací na desktopové aplikace.
69. FOUNDATION, OpenJS. *Electron* [software] [cit. 2021-04-15]. Dostupné z: <https://www.electronjs.org/>. Open-source nástroj pro vývoj desktopových aplikací.
70. Progressive Web Apps (PWA) [online] [cit. 2021-04-14]. Dostupné z: <https://techradar.softwareag.com/technology/progressive-web-apps>. Vysvětlení konceptu progresivních webových aplikací.
71. Introduction to progressive web apps [online] [cit. 2021-04-14]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction. Představení progresivních webových aplikací.

72. Introduction to progressive web apps [online] [cit. 2021-04-14]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps. Představení nároků na progresivních webové aplikace.
73. Electron Documentation - Quick start guide [online] [cit. 2021-04-14]. Dostupné z: <https://www.electronjs.org/docs/tutorial/quick-start>. Oficiální návod pro použití knihovny Electron.
74. *Electron Builder* [software] [cit. 2021-04-15]. Dostupné z: <https://www.electron.build/>. Open-source nástroj pro sestavování Electron aplikací.
75. *Electron Forge* [software] [cit. 2021-04-15]. Dostupné z: <https://www.electronforge.io/>. Open-source nástroj pro sestavování Electron aplikací.
76. Electron on GitHub issue - Expected app bundle size? [online] [cit. 2021-04-14]. Dostupné z: <https://github.com/electron/electron/issues/2003>. Nahlášený problém s velikostí sestavené Electron aplikace s vyjádřením autorů knihovny Electron.
77. *Vue Electron Builder* [software] [cit. 2021-04-04]. Dostupné z: <https://github.com/nklayman/vue-cli-plugin-electron-builder>. Open source projekt.
78. *Vuex* [software] [cit. 2021-04-04]. Dostupné z: <https://vuex.vuejs.org/>. Open source projekt.
79. MOZILLA. AJAX [online] [cit. 2021-04-15]. Dostupné z: <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>.
80. RUDERMAN, Jesse. Same-origin policy [online] [cit. 2021-04-15]. Dostupné z: https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy.

Seznam použitých zkratk

API Application Programming Interface

IDE Integrated Development Environment

PDF Portable Document Format

UX User Experience

UI User Interface

PWA Progressive Web Application

AJAX Asynchronous JavaScript and XML

URL Uniform Resource Locator

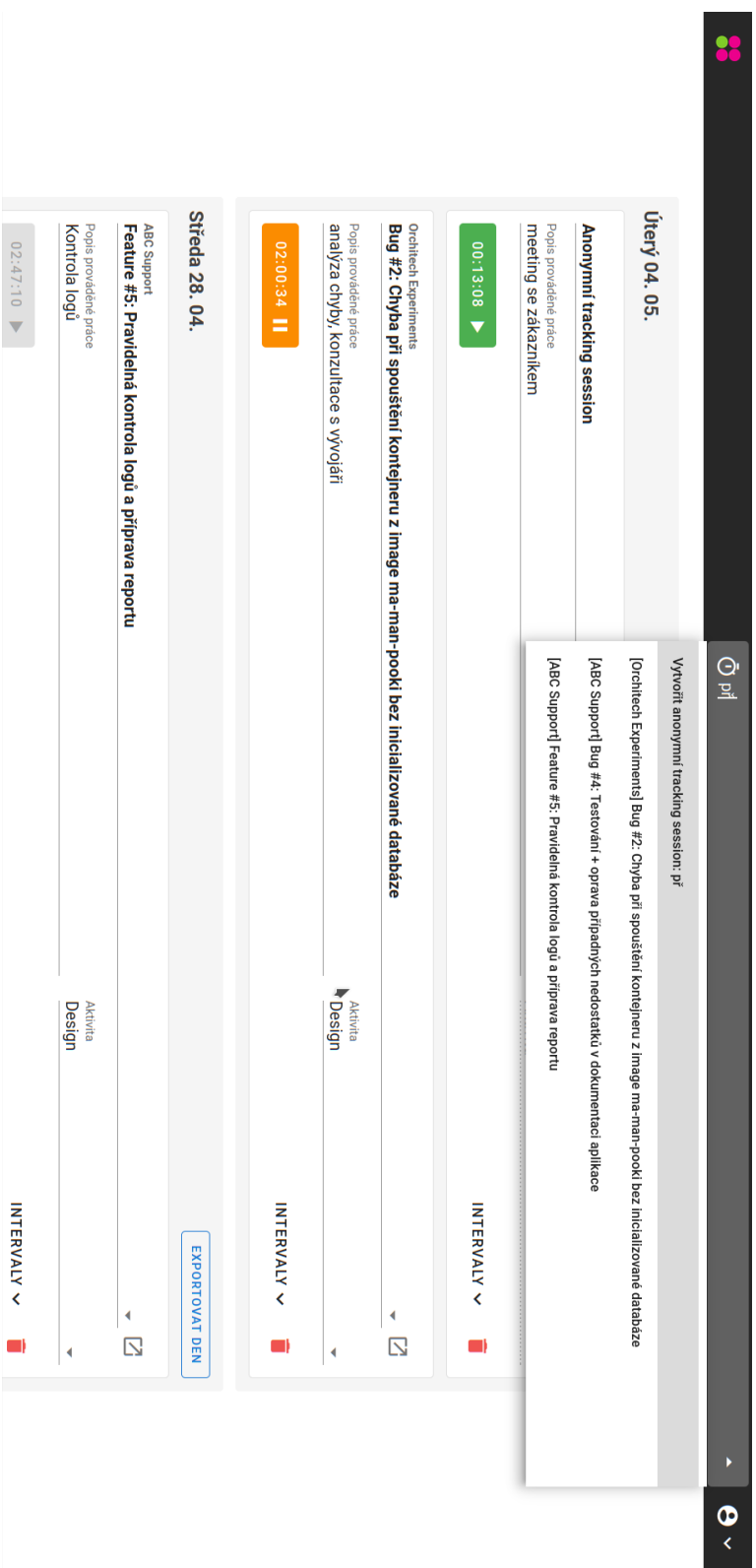
REST Representational State Transfer

SaaS Software as a Service

Obsah přiloženého média

README.md.....	stručný popis obsahu média
src	zdrojová forma práce ve formátu \LaTeX
text	
thesis.pdf.....	text práce ve formátu PDF
designs.....	grafické návrhy vytvořené v rámci práce
screenshots	snímky obrazovky s implementovaným prototypem a původní webovou aplikací TRebel

Grafické návrhy a snímky obrazovky



Obrázek C.1: Hlavní stránka původní webové aplikace TTRebel

Název, číslo issue nebo popis práce

00:13:08

INTERVALY

Ochitech Experiments
Bug #2: Chyba při spouštění kontejneru z image ma-man-pooki bez inicializované databáze
 Popis prováděné práce
 analýza chyby, konzultace s vývojáři

Aktivita
 Design

01:38:00

INTERVALY

út 4. 5. 2021	Začátek	Konec	út 4. 5. 2021
út 4. 5. 2021	15:04	16:10	út 4. 5. 2021
út 4. 5. 2021	12:43	12:50	út 4. 5. 2021
út 4. 5. 2021	10:20	10:45	út 4. 5. 2021

Středa 28. 04.

ABC Support
Feature #5: Pravidelná kontrola logů a příprava reportu
 Popis prováděné práce

Aktivita

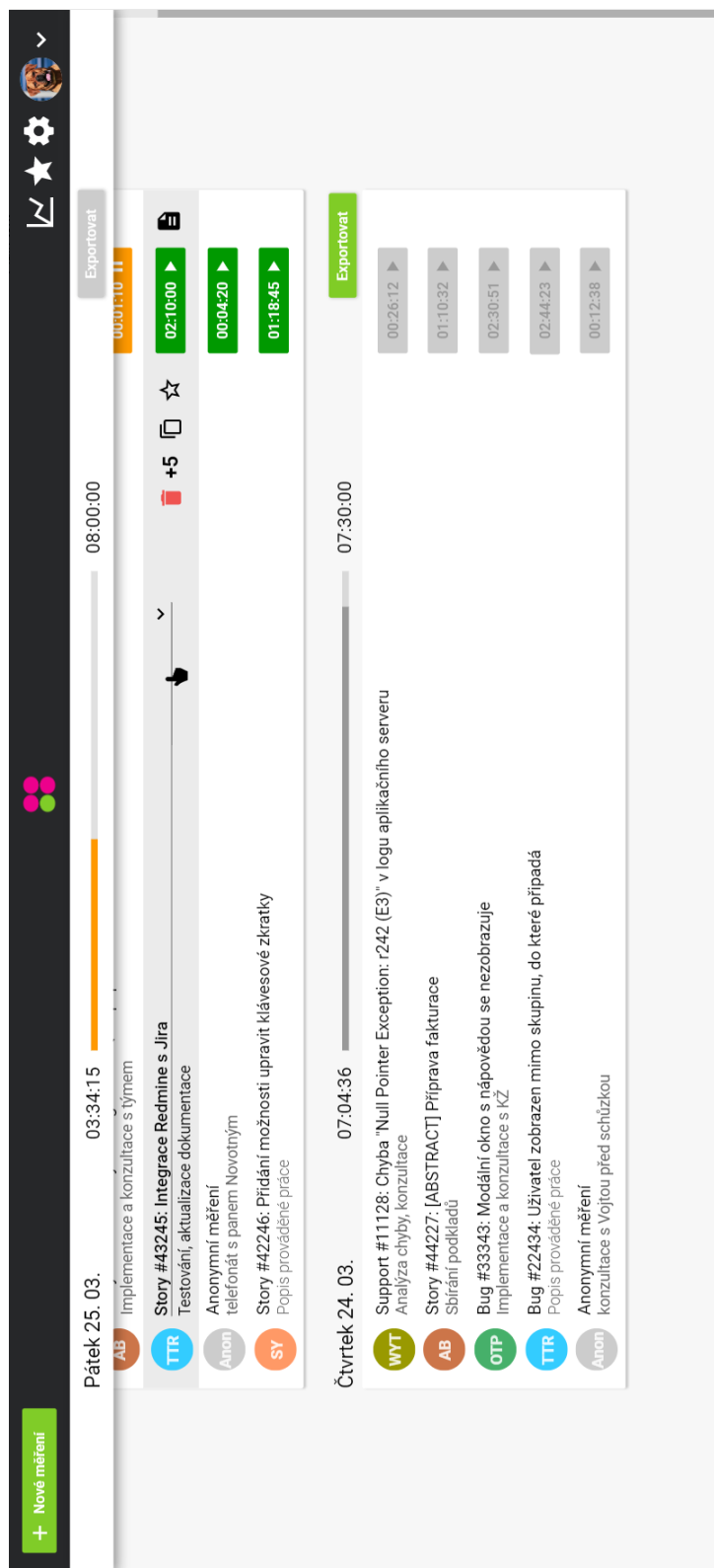
EXPORTOVAT DEN

Obrázek C.2: Správa časových intervalů v původní webové aplikaci TTRebel

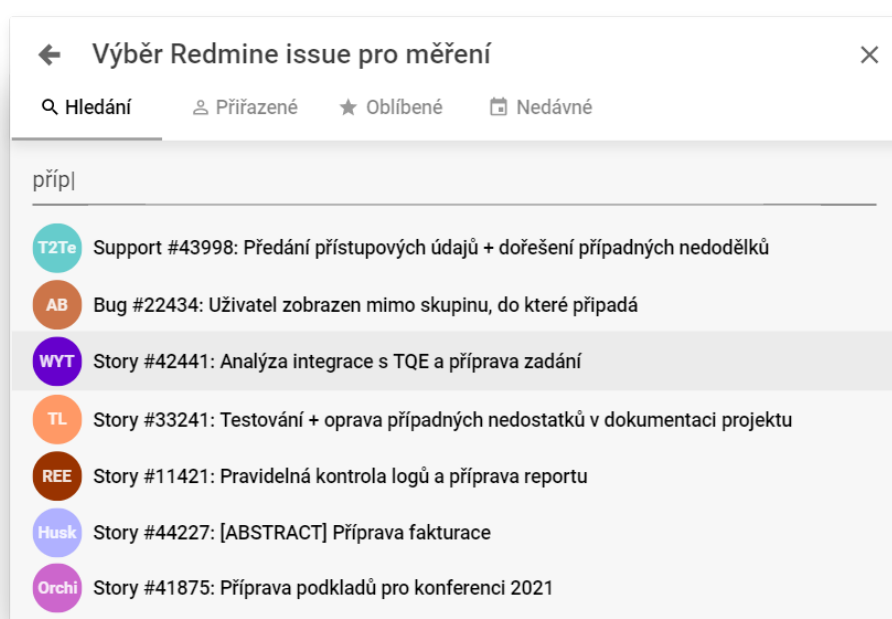
The screenshot displays a project tracking application. At the top, there is a navigation bar with a 'TRACKING' button and a search icon. Below this, a summary for 'Úterý 04. 05.' shows task status: 'Již vykázáno' (00:00:00) and 'K exportu' (00:13:08). A blue 'EXPORTOVAT 00:13:08' button is visible. The main content area lists two tasks under 'Orchitect Experiments':

- Bug #7: Cron jobs should call admin app directly** (Aktivita Design): 'Popis provedené práce meeting se zákazníkem'. It has a 'K exportu' checkbox checked and an 'INTERVALY 00:13:08' dropdown.
- Bug #2: Chyba při spuštění kontejneru z image ma-man-pooki bez inicializované databáze** (Aktivita Design): 'Popis provedené práce analýza chyby, konzultace s vývojáři'. It has a 'K exportu' checkbox unchecked and an 'INTERVALY 01:38:00' dropdown.

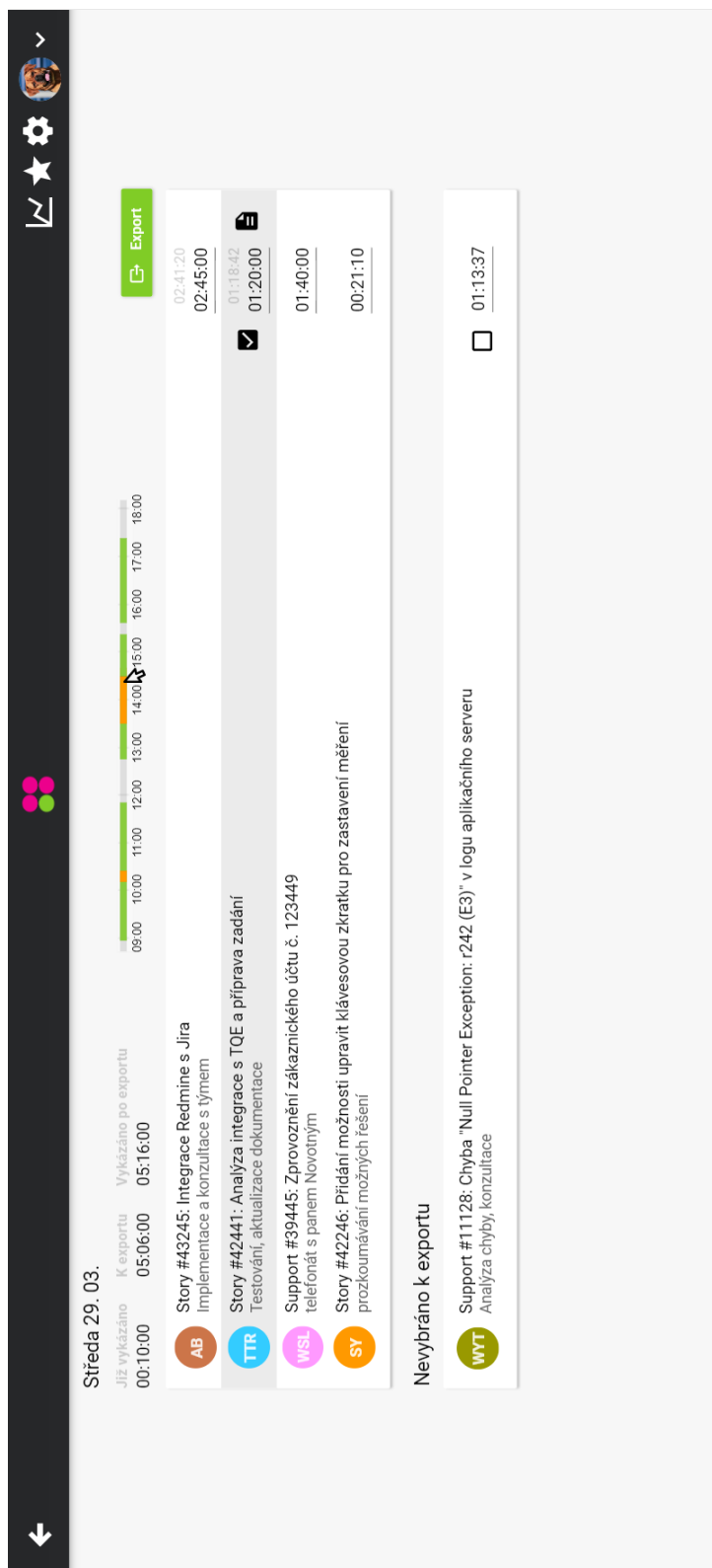
Obrázek C.3: Stránka s exportem původní webové aplikace TTRebel



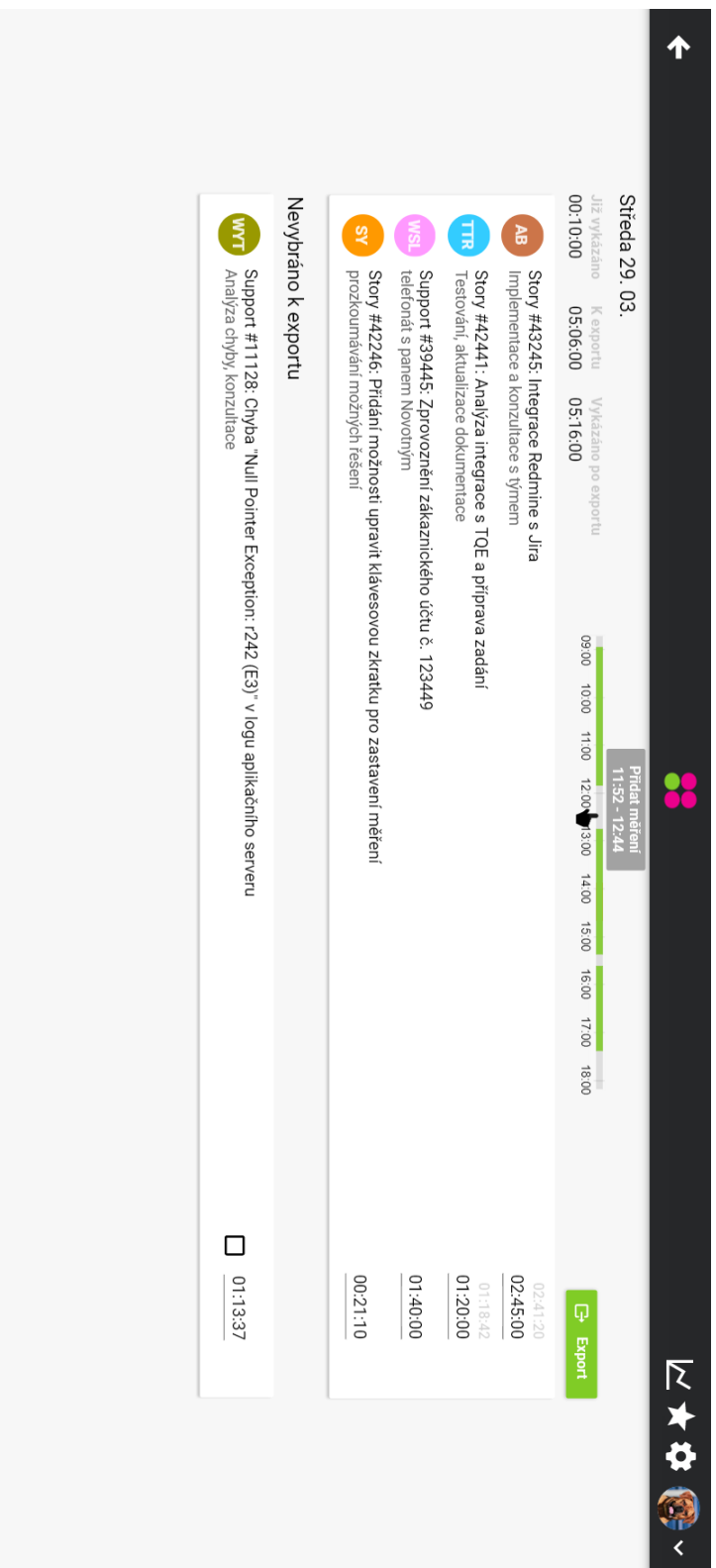
Obrázek C.4: Grafický návrh hlavní stránky webové aplikace TTRebel s ukotvením informací o měřeních při posunutí níže na stránku



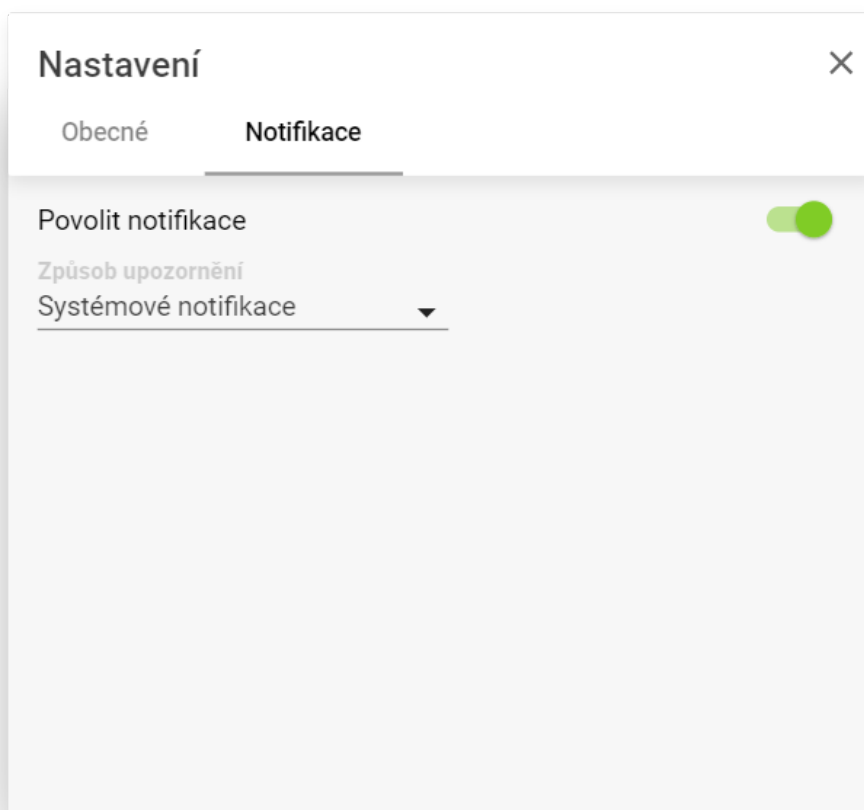
Obrázek C.5: Grafický návrh modálního okna s vyhledáváním a výběrem Redmine úloh otevřeného z modálního okna s detailem měření ve webové aplikaci TTRebel



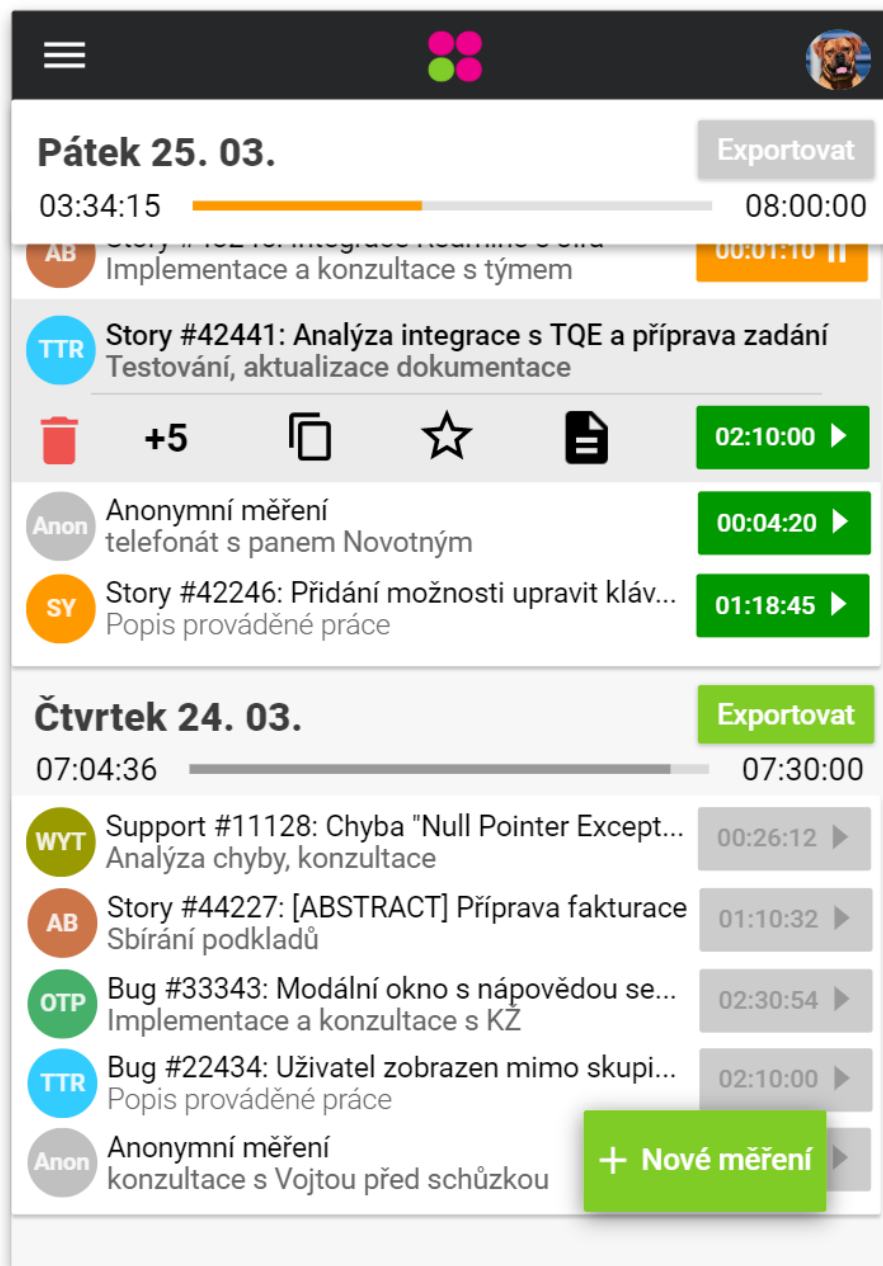
Obrázek C.6: Grafický návrh stránky s exportem měření webové aplikace TTRRebel s kurzorem namířeným na časový interval na časové ose



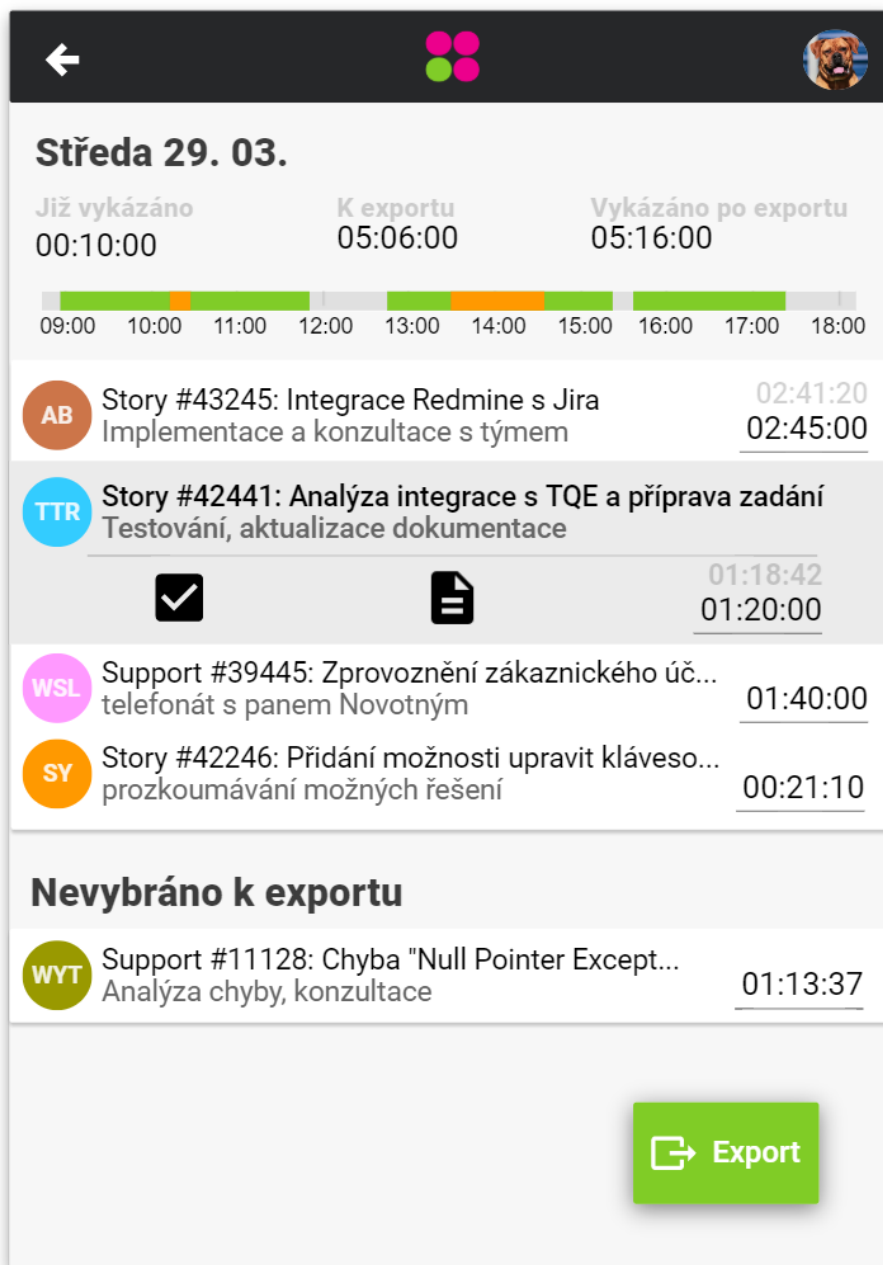
Obrázek C.7: Grafický návrh stránky s exportem měření webové aplikace TTRebel s kurzorem naměřeným na prázdný interval na časové ose



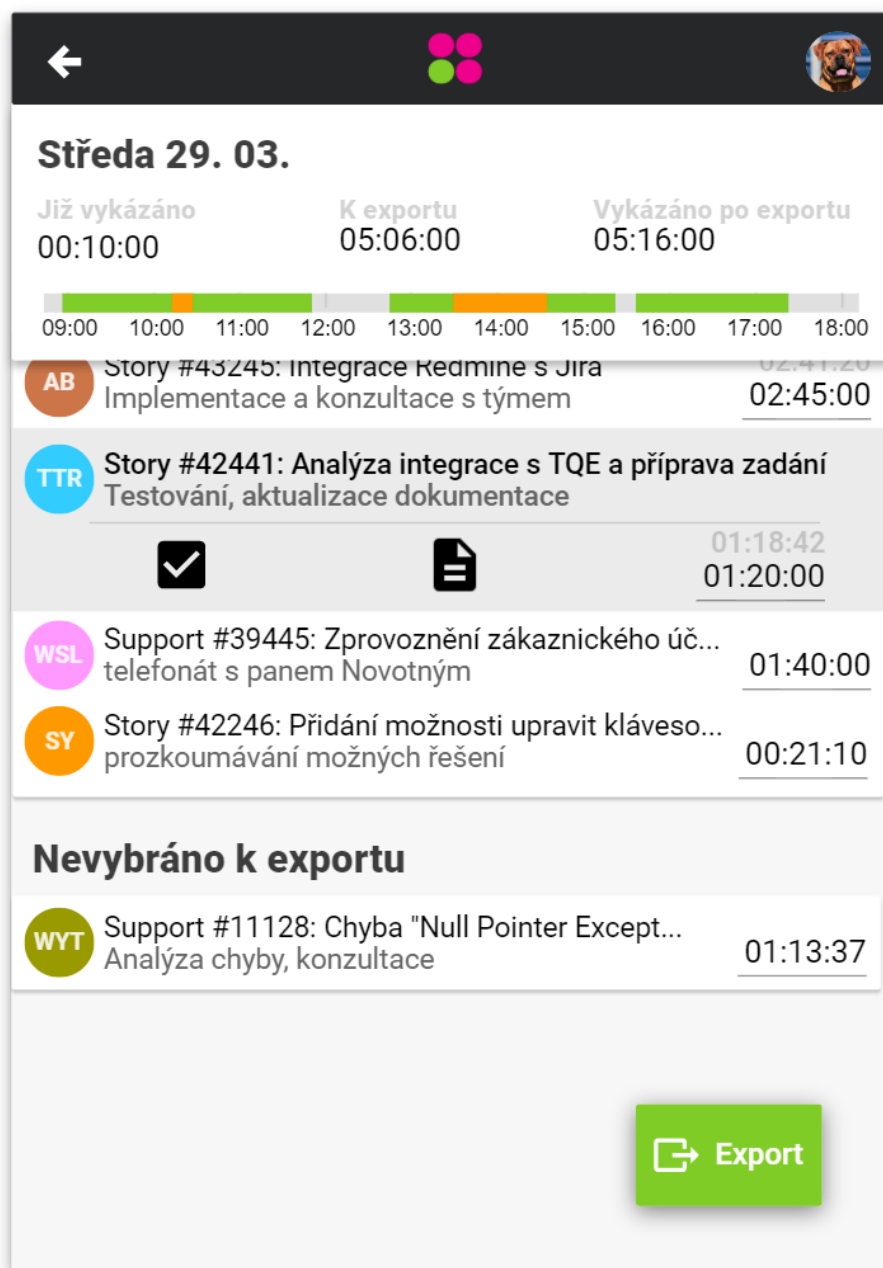
Obrázek C.8: Grafický návrh modálního okna s nastavením notifikací ve webové aplikaci TTRebel



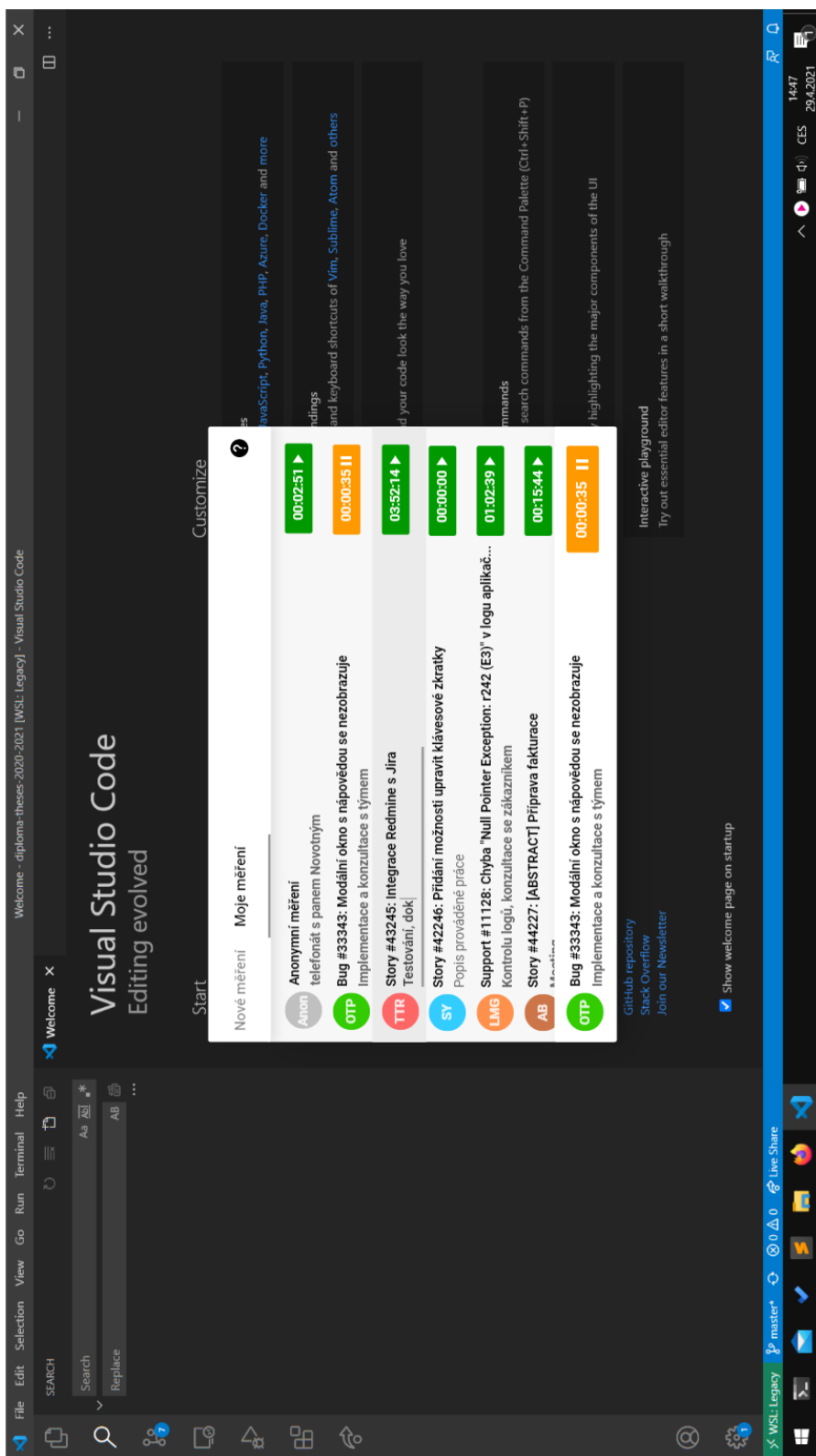
Obrázek C.9: Grafický návrh hlavní stránky webové aplikace TTRebel na mobilním telefonu s ukotvením informací o měřeních při posunutí níže na stránku



Obrázek C.10: Grafický návrh exportní stránky webové aplikace TTRebel na mobilním telefonu

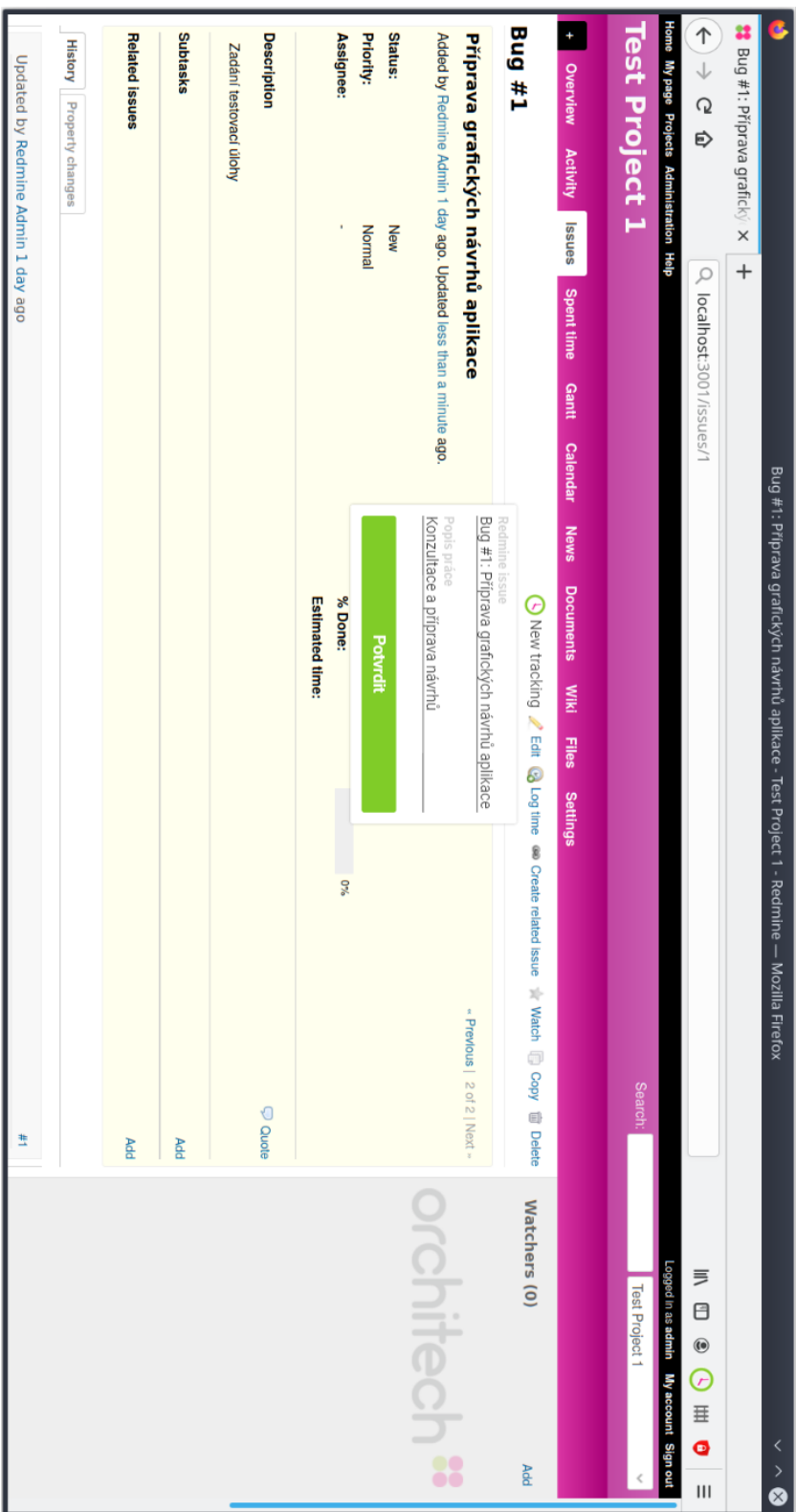


Obrázek C.11: Grafický návrh exportní stránky webové aplikace TTRebel na mobilním telefonu s ukotvením souhrnných informací při posunutí níže na stránku

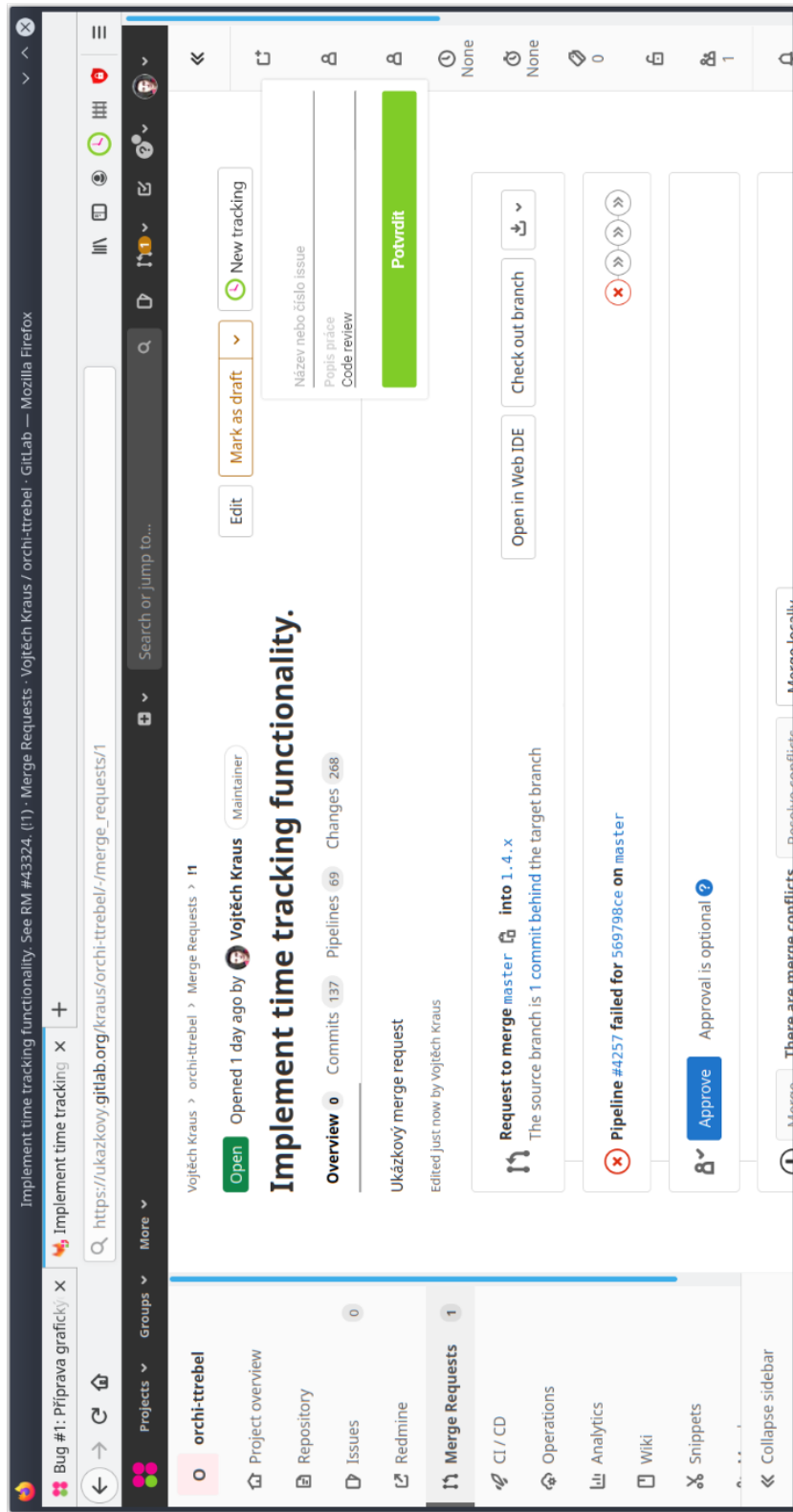


Obrázek C.12: Grafický návrh uživatelského rozhraní desktopové aplikace TTRebel zasazený do popředí systému Windows spolu s ikonou signalizující aktivní měření v notifikační oblasti systému

C. GRAFICKÉ NÁVRHY A SNÍMKY OBRAZOVKY



Obrázek C.13: Grafický návrh zakomponování nabídky pro vytvoření měření a tlačítka k vyvolání této nabídky na detail úlohy v systému Redmine



Obrázek C.14: Grafický návrh zakomponování nabídky pro vytvoření měření a tlačítka k vyvolání této nabídky na detail merge requestu v systému GitLab