

Sem vložte zadání Vaší práce.

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



Bakalářská práce

Návrh a implementace systému podpory oběhu elektronického obsahu firmy

Zdeněk Haták

Vedoucí práce: Ing. David Buchtela, Ph.D.

11. května 2015

Poděkování

Chtěl bych poděkovat své rodině za podporu po celou dobu studia, společnosti FollowBubble s.r.o, že mi poskytla podnět k práci a umožnila aplikaci otestovat v reálném prostředí. Dále děkuji vedoucímu práce, že jsem u něj mohl práci vypracovat a za jeho cenné rady a Lucii Bajgartové za konzultace při tvorbě uživatelského rozhraní.

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 mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 11. května 2015

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2015 Zdeněk Haták. Všechna práva vyhrazena.

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

Odkaz na tuto práci

Haták, Zdeněk. *Návrh a implementace systému podpory oběhu elektronického obsahu firmy*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2015.

Abstrakt

Tato práce se zabývá návrhem a implementací nástroje pro podporu schvalování elektronického obsahu, který se firmě tvoří a předává mezi zaměstnanci a klienty. Práce obsahuje analýzu požadavků na nový nástroj, který má tento proces zefektivnit, jeho návrh a implementaci pomocí Google Drive API. Výsledek je poté pilotně nasazen ve firmě a otestován zaměstnanci firmy a jsou zhodnoceny přínosy pro firemní praxi.

Klíčová slova Digitální agentura, schvalování obsahu, webová aplikace, Nette, Google Drive, workflow

Abstract

This thesis deals with the design and implementation of a tool that supports approval process of electronic content, which is created in a company and transferred between employees and clients. Thesis also contains the analysis of the requirements for the new tool, its design and implementation using Google Drive API. Afterwards, the result is deployed and tested by employees. In the end there is the evaluation of the benefits for the company.

Keywords Digital agency, content approving, web application, Nette, Google Drive, workflow

Obsah

Úvod	1
Cíl	1
Struktura práce	1
1 Současný stav	3
1.1 Struktura firmy	3
1.2 Procesy ve firmě	4
1.3 ICT zázemí	5
1.4 Problém	5
2 Analýza	7
2.1 Zvolené řešení	7
2.2 Existující řešení	7
2.3 Požadavky	13
2.4 Shrnutí	16
3 Návrh aplikace	17
3.1 Uživatelé	17
3.2 Uživatelské prostředí	19
3.3 Databáze	22
3.4 Notifikace	25
3.5 Google Drive	25
3.6 Shrnutí	26
4 Implementace	27
4.1 Programy využívané při tvorbě	27
4.2 Volba programovacích jazyků a nástrojů	28
4.3 Realizace	31
4.4 Shrnutí	36

5 Nasazení a výsledky v praxi	37
5.1 Nasazení	37
5.2 Náklady	39
5.3 Přínosy	40
5.4 Budoucnost	40
Závěr	43
Literatura	45
A Průchod aplikací	49
A.1 Úvodní stránka	49
A.2 Založení procesu	50
A.3 Práce s procesem z pohledu dohlížejícího	51
B Seznam použitých zkratk	53
C Obsah příloženého CD	55

Seznam obrázků

2.1	Ukázka použití rozšíření	9
2.2	Ukázka použití rozšíření 2	9
2.3	Průchod rozšířením Workflows	11
2.4	ApproveForMe	12
2.5	ApproveForMe	12
3.1	Uživatelé systému	19
3.2	Hlavní obrazovka	20
3.3	Začátek procesu	21
3.4	Detail procesu	22
3.5	Databázový model	23
3.6	Ukázka souboru z Google Drive	26
3.7	Ukázka souboru z Google Drive	26
4.1	Developer Konzole	32
4.2	Google scopes	33
A.1	Hlavní stránka aplikace	50
A.2	Založení procesu	51
A.3	Označení souboru na Drive	51
A.4	Detail procesu z pohledu dohlížejícího	52
A.5	Označení souboru na Drive - hovoo	52

Seznam tabulek

3.1	Popis tabulky Users	23
3.2	Popis tabulky Process	24
3.3	Popis tabulky Files	24
3.4	Popis tabulky Comments	24
3.5	Popis tabulky Notifications	25
5.1	Náklady na vývoj aplikace	39

Úvod

Práce popisuje tvorbu nástroje mající za úkol zefektivnit proces oběhu vytvářeného obsahu v digitální agentuře FollowBubble s.r.o.

Cíl

V dnešní uspěchané době je automatizace a efektivita firemních procesů důležitou součástí každé moderní úspěšné společnosti. Svět Internetu a digitálních technologií stále roste a je třeba s ním držet krok co možná nejvíce. Cílem této bakalářské práce je vytvořit aplikaci, která pokud možno urychlí a usnadní proces oběhu obsahových dokumentů mezi zaměstnanci agentury a klienty. Není však záměrem stvořit hotový produkt, nýbrž škálovatelnou aplikaci, která půjde snadno rozšířit.

Struktura práce

Práce je rozdělena do několika hlavních kapitol, které podle svých názvů kopírují průběh tvorby software. V kapitole 1 řeším současný stav procesu oběhu digitálního obsahu ve firmě a identifikuji problém, který se při něm vyskytuje. V kapitole 2 se zaměřím na analýzu existujících řešení a funkčních i nefunkčních požadavků nové aplikace, u které provedu návrh v kapitole 3. Zde navrhnu uživatelské prostředí, představím uživatelské role systému a provedu návrh databáze. V kapitole 4 se budu věnovat realizaci navrženého řešení, popíši nástroje, které k tvorbě využíval, přiblížím frontendovou i backendovou část aplikace. V 5. kapitole proberu nasazení a testování. Nakonec zhodnotím, zda práce měla pro firmu reálný přínos.

Současný stav

V úvodní kapitole lehce přiblížím současný stav společnosti *FollowBubble s.r.o.* co se týče zázemí, zaměstnanců a procesů tvorby obsahu, přidělování úkolů, komunikace mezi zaměstnanci i klienty a problémů, které se při těchto procesech vyskytují.

Digitální agentura FollowBubble s.r.o. se zabývá tvorbou a správou digitálního obsahu ostatních společností. Vymýšlí online kampaně pro firmy jako jsou například Sazka a.s., T-Mobile a.s či Karlovarské minerální vody, a.s. a zároveň spravuje jejich digitální život na Internetu včetně jejich sociálních sítí. Vytváří i webové stránky od návrhu obsahové strategie, focení materiálů, tvorby videí přes grafický návrh až po implementaci a následnou údržbu.

1.1 Struktura firmy

Společnost čítá přibližně dvacet stálých zaměstnanců a přibližně deset externistů. Počet externistů je proměnlivý, najímají se většinou pro atypické zakázky, na které se nevyplatí mít v kanceláři interního člověka nebo z důvodu vytížení stálých zaměstnanců. Z hlediska struktury lze firmu rozdělit do těchto kategorií:

Copywriteři

Copywriteři se starají o veškerý tvůrčí text. Tvoří prezentace, články a texty na webové stránky či překládají. Vymýšlí obsahové strategie pro různé kampaně jak na weby, tak sociální sítě. Nemalou část tvoří tzv. *Comunnity manažeréři*, kteří mají na starost správu sociálních sítí klientských společností, nejčastěji Facebook[1]. Vymýšlí na ně statusy a nebo odpovídají fanouškům.

Developeři

Jakýkoliv kód, který se ve firmě naprogramuje, jde přes tuto sekci a to jak od interních zaměstnanců, tak externistů. Zde se programují webové stránky a webové i mobilní aplikace.

Designéři

O všechnen grafický obsah se stará oddělení designérů. Tvoří se zde grafické návrhy webových stránek, aplikací, 3D modely, ale i retuše fotografií či grafická podoba prezentací.

Management

V tomto oddělení se řeší vše ostatní. Jsou zde například *account manažeři*, jejichž úkolem je pečovat o klíčové zákazníky. Navazují a udržují dlouhodobé vztahy s klienty, vyjednávají podmínky spolupráce či marketingovou podporu. Důležitou součástí firmy jsou i projektoví manažeři, kteří se starají o rozdělování práce, píší projektové dokumentace či reporty a mají celkový přehled o všech aktuálních, budoucích i hotových projektech.

1.2 Procesy ve firmě

Je logické, že se při těchto činnostech vytvoří spousta dokumentů, tabulek, prezentací, obrázků a návrhů. Drtivá většina takového obsahu musí projít různým procesem schvalování a to v rámci jedné, ale i více firem. Ať už jde o projektovou dokumentaci, návrh smlouvy či článek na web, kvůli vytížení zainteresovaných lidí je někdy těžké získat od všech odezvu a zároveň splnit daný úkol v časovém termínu.

Úložiště pro vytvořený obsah se liší podle jeho typu. Textové dokumenty se většinou nacházejí na Google Drive[2], pro uložení grafického typu obsahu se využívá vlastní server a nebo, pokud je třeba konzultovat věc s klientem, služba Dropbox[3]. Charakter úloh, které se ve firmě řeší je velice rozmanitý, takže se těžce hledá nějaký pevný vzorec, podle kterého se procesy rozdělování, delegování a komunikace nad nimi řídí.

V rámci společnosti se však k zadávání úkolů či komunikaci mezi zaměstnanci používají tyto nástroje:

Google Apps

Každý zaměstnanec má v rámci Google Apps[4] svůj emailový účet a úložiště Google Drive. Dále společnost jako celek využívá Google Kalendář k plánování meetingů, schůzek s klienty či interních akcí v rámci firmy.

TeamWork

Teamwork[5] slouží k zadávání úkolů. Typicky, projektový manažer, nebo někdo, kdo potřebuje úkol předelegovat na někoho jiného založí v aplikaci *task* (anglicky úkol) u vybraného projektu a přiřadí k němu osoby, jež na něm budou pracovat. Dále může připojit obrázkové či textové přílohy, nastavit ter-

mín požadovaného dokončení či prioritu s jakou by měl řešitel k problému přistupovat, pokud je zaneprázdněný.

Slack

Slack[6] je relativně mladý projekt, jež slouží jako náhrada klasického chatu. Jelikož je mezi zaměstnanci i několik externistů, nejde vše potřebné řešit osobně v kanceláři. Pomocí Slacku tedy probíhá veškerá instantní komunikace. Uživatelé mohou komunikovat jeden na jednoho či ve veřejných místnostech, kde jsou zprávy viditelné všem v týmu. Navíc umožňuje vytvoření separátních chatovacích místností viditelné jen pro určité zaměstnance, například pro každý projekt zvlášť. Řeší se i zde menší úlohy, které jde vyřešit zpravidla hned a není k nim třeba nějaká zvláštní průprava.

1.3 ICT zázemí

Firma má k dispozici vlastní server, na kterém běží operační systém Debian[7] (v době psaní bakalářské práce byl na serveru nainstalován Debian ve verzi 7.8). Nainstalované PHP je ve verzi 5.5.24 a MySQL verze 5.6.23. Se serverem pracují výhradně developeri, běžné úlohy zahrnují například práci s verzovacím systémem, či se zálohami databází webových projektů.

V řadách zaměstnanců firmy jsou nejvíce zastoupeny operační systémy iOS od společnosti Apple[8], v menší míře pak Windows od Microsoft[9]. Jiné operační systémy zastoupeny nejsou a vzhledem k charakteru společnosti se dá předpokládat, že se tento fakt v blízké době nezmění.

Počítačová gramotnost

Úroveň počítačové gramotnosti se u zaměstnanců liší pouze drobně. Jsou to převážně mladí lidé, kteří s počítačovými technologiemi vyrůstali a mají k nim velice kladný vztah. Obor ve kterém se firma pohybuje je proměnlivý a je třeba se mu přispůsobovat. Je tedy zřejmé, že na jakékoliv změny reagují zaměstnanci poměrně rychle a nezdržují chod firmy. Příkladem může být přechod na již zmíněný chatovací nástroj *Slack*, který v rámci celé firmy proběhl za méně než 24 hodin.

1.4 Problém

Schvalovací proces a výměna připomínek nad vytvořeným obsahem mezi klienty a firmou probíhá většinou přes emaily, které se při delších komunikacích stávají nepřehledné, těžce se v nich vyhledává a tím se celá věc ještě více komplikuje. Při komunikaci s více lidmi se tento problém ještě vystupňuje. V emailech se posílají odkazy na právě řešený obsah a občas se využívá komentářů, které nabízí Google Drive ve spojení s rozšířením Workflows[10].

1. SOUČASNÝ STAV

Toto rozšíření je poměrně kvalitní, avšak má několik nedostatků (blíže o nich v kapitole 2). Jelikož v této společnosti pracuji jako front-end developer, tak se kvůli náplni mé práce k procesům týkajících se schvalování dostanu zřídka, ale nemohl jsem si problému s emaily nevšimnout.

Shrnutí nedostatků:

- Přeplněnost emailových schránek
- Neschopnost se efektivně ke komunikacím vracet
- Riziko přehlédnutí důležitých faktů
- Nereprezentativnost

Řešením těchto problémů by mohla být nová aplikace, která by byla postavena přesně na míru dle požadavků. Je třeba hledět i na ekonomickou stránku věci, najít kompromis mezi pracností řešení a výsledným efektem pro firmu. Pokud by byla nová aplikace příliš náročná časově, nemusí se vyplatit do vývoje investovat.

Analýza

První fází při vývoje softwaru je zpravidla analýza. Dá se díky ní přijít na chyby, které by mohly v dalších fázích brzdit vývoj a pomáhá získat přehled o tom, co se od aplikace očekává. V první části této kapitoly se nachází analýza již existujících řešení a v druhé jsou podle přání zadavatelské firmy vysloveny funkční a nefunkční požadavky.

2.1 Zvolené řešení

Vzhledem k tomu, že se ve firmě na velkou část obsahů používá systém *Google Drive*, bylo rozhodnuto v tomto trendu pokračovat a použít *Drive* jako stavební kámen pro novou aplikaci. Nástroj využívající v rozumné míře *Google Drive API* by mohl zlepšit kolaboraci různých lidí a firem na stejném obsahu a proces schvalování zefektivnit. Navíc se upevní jakási konzistence v nástrojích používaných napříč celou firmou. Se zadavatelskou firmou jsem se dohodl na tom, že se nemusí vytvořit hotový produkt dotažený do konce, připravený k nasazení do produkčního prostředí, ale solidní základ vhodný k pilotnímu nasazení a otestování a úzpůsobený k možnému rozšiřování.

I z ekonomického hlediska bude lepší nestavět řešení „na zelené louce“, vývoj nezabere tolik času, navíc, při volbě správných nástrojů v průběhu implementace se zkrátí i potřebná doba při údržbě a při již zmíněném rozšiřování o další funkce. Pokud by se aplikace ujala, firma se nebrání přechodu z *Dropboxu* na *Drive* jako hlavního úložiště i pro grafický obsah.

2.2 Existující řešení

Jelikož byl zvolen *Google Drive* jako hlavní platforma vývoje tak se analýza existujících řešení bude zabývat hlavně aplikací a rozšíření týkající se právě *Google Drive*. Čtenář se zde dozví jakým způsobem řešení fungují a bude mít

tak možnost srovnání. Snažil jsem se je probrat z obecného hlediska a nehodnotit jejich žádný konkrétní aspekt.

Po průzkumu trhu jsem zjistil, že existují v podstatě pouze dva použitelné nástroje určené k zefektivnění schvalovacího procesu skrze *Drive*, které jsou na Internetu dohledatelné a je možné je využít.

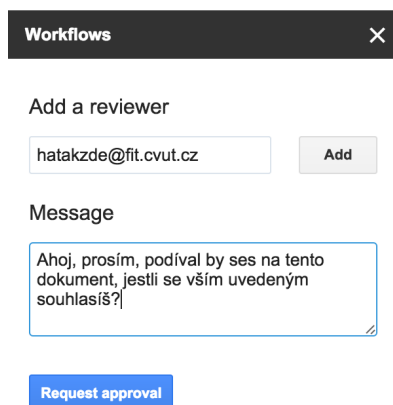
2.2.1 Workflows

Workflows je v této oblasti nejspíše nejpobulárnějším řešením. Jedná o nástroj zabudovaný přímo do prostředí Google Drive ve formě rozšíření. Lze si ho přidat a začít používat velmi jednoduše (předpokládá se, že je uživatel přihlášený a jako hlavní jazyk má nastavenou češtinu):

1. Otevřít libovolný dokument vytvořený v prostředí Google Drive
2. V menu kliknout na volbu *Doplňky*, následně na *Získat doplňky*
3. Ve vyskakovacích okně vyhledat řetězec „Workflows“ a kliknout na tlačítko pro přidání doplňku *Workflows*

Workflows se následně uživatele dotáže, zda souhlasí s oprávněními, které potřebuje ke svému provozu. Jde například o oprávnění umožňující rozšíření posílat emaily uživatelovým jménem, či o možnost zobrazovat a spravovat soubory na Drive. Nyní stačí kliknout znovu na *Doplňky*, a v možnosti *Workflows*, která se objevila zvolit *Show Workflows*.

Pokud chce uživatel požádat o shlednutí (*Žadatel*), či schválení jiného uživatele (*Schvalovatel*), musí jeho emailu přiřadit taková práva na daný dokument, aby jej mohl upravovat. Poté může do *Workflow* zadat jeho email a poslat žádost. Jak lze vidět na obrázku 2.1, žadatel může přidat krátký úvodní text. „Schvalovateli“ přijde email s odkazem na dokument a s již zmíněným textem. Schvalovatelů může být samozřejmě libovolné množství.



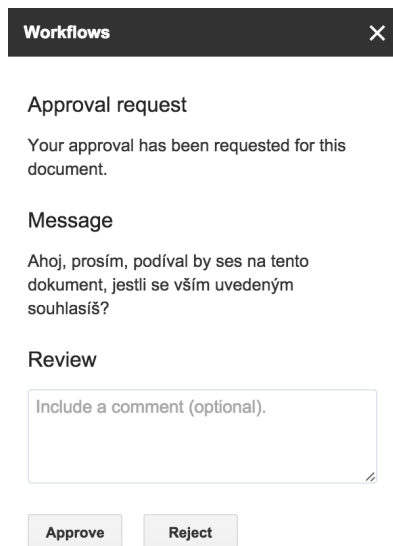
Workflows ✕

Add a reviewer

Message

Obrázek 2.1: Ukázka použití rozšíření Workflows[10]

Na obrázku 2.2 se nachází strana schvalovatele. Má zde možnost přidat komentář a následně výběrem z tlačítek *Approve* nebo *Reject* dokument schválit, respektive zamítnout. V obou případech přijde s touto informací žadateli email. Pokud schvaloval zamítne, žadatel má možnost svou žádost resetovat a tím spustit celou akci znovu. Takto to může pokračovat do doby, než žadatel proces zruší, nebo schvalovatel soubor schválí.



Workflows ✕

Approval request

Your approval has been requested for this document.

Message

Ahoj, prosím, podíval by ses na tento dokument, jestli se vším uvedeným souhlasíš?

Review

Obrázek 2.2: Workflows na straně schvalovatele[10]

2.2.1.1 Záporné stránky

Toto rozšíření již zadavatelská firma používá, avšak není s ním úplně spokojena a to hlavně z těchto důvodů:

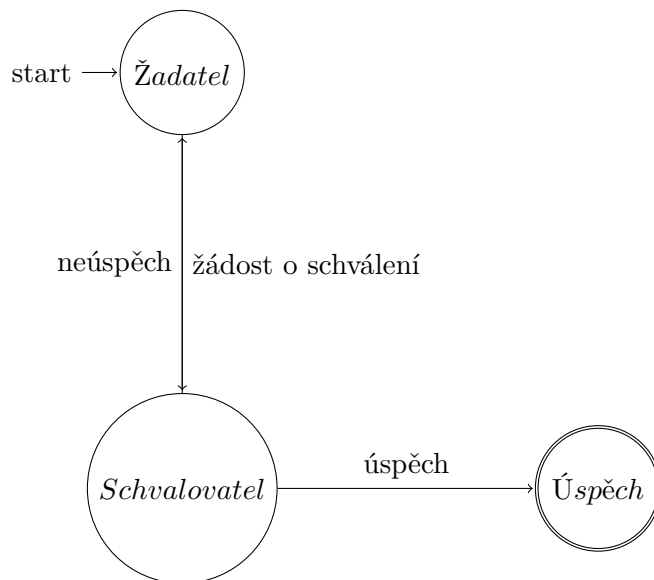
- Neexistuje zde žádná hlavní obrazovka, kde by se nacházely všechny dokumenty, které je třeba schválit
- Nutnost vlastnit Google účet pro obě strany
- V prostředí *Drive* nejde poznat, že je dokument schvalován
- Rozšíření se omezuje pouze na textové (Dokumenty Google) a tabulkové (Tabulky Google) dokumenty
- Jak žadatel, tak schvalovatel musí mít rozšíření nainstalované
- Žadatel musí ručně přiřadit schvalovateli práva editace souboru

2.2.1.2 Kladné stránky

Na druhou stranu má Workflows i kladné stránky:

- Velice jednoduché a intuitivní UI
- Rozšíření je integrováno přímo v Google Drive, takže není třeba zvláštní účet nebo aplikace
- Žadatel má přehled například i o tom, zda si schvalovatel soubor otevřel
- Je zdarma

Celý proces je přehledně znázorněn na následujícím diagramu:



Obrázek 2.3: Průchod rozšířením Workflows

2.2.2 ApproveForMe

Nástroj ApproveForMe[11] není tak populární jako ten výše zmíněný. Nejedná se o rozšíření Google Drive, nýbrž o speciální webovou aplikaci. Vše, co potřebuje uživatel k jejímu používání je Google účet. Nástroj má velice jednoduché uživatelské prostředí. Na hlavní obrazovce lze vidět kromě dokumentů, které uživatel zařadil do schvalování také tlačítko, které přesměrovává na stránku, kde se může schvalovací proces zahájit. Na rozdíl od *Workflows* funguje pro všechny typy souborů, které lze uložit na *Drive*. Nevýhodou je, že lze pro jeden proces vždy vybrat pouze jeden soubor. Výběr kýženého souboru probíhá přes Google Picker[12], což je jednoduchý nástroj od Google ve formě otevíracího dialogu, přes který se dá například přidat soubor z *Drive* či Internetu.

2. ANALÝZA

New Approval

File* Select File
 Allow reviewers to edit this document.

Title*

Description

Deadline*

Approvers*

Name	Email Address	Type
<input type="text" value="Name"/>	<input type="text" value="Email address"/>	Required
<input type="text" value="Name"/>	<input type="text" value="Email address"/>	Required
<input type="text" value="Name"/>	<input type="text" value="Email address"/>	Required

Add Another Approver

Submit Document for Approval

Obrázek 2.4: Založení nového *Approval* v aplikaci ApproveForMe[11]

Po vyplnění termínu, kdy má být daná věc uzavřena, volitelného popisu dokumentu a jmen a emailů schvalovatelů, u kterých je možnost navíc nastavit *Optional* a *Required*, respektive *Volitelné* a *Požadované*(2.4), se na zadané emailové adresy zašle úvodní email s odkazem do aplikace, kde je stejně jako v případě již zmíněného nástroje Workflows odkaz do aplikace s možností soubor zamítnout či schválit (2.5).

Projektová dokumentace k Projektu X
Popis dokumentu

Zdeněk Haták has requested your approval. [Click here to respond.](#)

Project Documentation to Project X
Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

Creator: Zdeněk Haták
Deadline: 05/15/2015, 17 days remaining
Progress: 0/1 required responses

Respond Open File

APPROVERS

hatak No Response
No comments

Obrázek 2.5: ApproveForMe na straně schvalovatele[11]

Verze zdarma je pro firemní používání zcela nepoužitelná, jelikož umožňuje pouze dvě schvalování měsíčně na jeden účet. V úvahu připadá jedině nejvyšší tarif (neomezeně schválení), který je v nabídce za 19 dolarů na měsíc pro jeden účet. Bráno v potaz, že přibližně polovina zaměstnanců firmy by nástroj využívala, v součtu se cena při kurzu 25 Kč za US dolar vyšplhá na částku kolem 4 750 Kč měsíčně, což je při faktu, že uživatel nemá možnost upravovat zdrojové kódy, doprogramovat si nějaké funkce, poměrně dost.

Záporné stránky

- V rámci procesu schvalování lze přiřadit pouze jeden soubor
- Pro reálné nasazení není zdarma
- V prostředí *Drive* nejde poznat, že je dokument v procesu
- Nutnost se přihlašovat do aplikace pro obě strany

Kladné stránky

- Je implementována možnost přiřadit k dokumentu termín, kdy má být dokončený
- Automaticky přidělí schvalovateli práva k souboru - lze zvolit, zda pro editaci, či jen pro čtení
- Na hlavní obrazovce aplikace se zobrazuje seznam souborů, které potřebují od někoho schválit

2.3 Požadavky

Rozhodl jsem se, možná trochu nestandardně, nejdříve zhodnotit existující nástroje a až poté se zabývat požadavky, protože jsem se poučil z chyb, které se v již hotových řešeních nacházejí. Je tedy zřejmé, že bude vynaložena snaha zachovat v nové aplikaci co nejvíce kladných stránek z existujících řešení a zároveň záporných stránek co nejvíce potlačit.

Po dohodě s managementem firmy jsem dospěl k následujícímu seznamu požadavků. Jedná se o klíčovou část celé analýzy, protože z ní vychází jak návrh aplikace, tak její implementace. Funkční požadavky mají označení „F.x“, kde „x“ je číslo požadavku. Nefunkční používají „N.x“. Obě kategorie následuje bližší popis vybraných požadavků.

2.3.1 Funkční požadavky

- F.1 Přidání souborů do procesu schvalování
- F.2 Odebrání souborů z procesu schvalování

2. ANALÝZA

- F.3 Přiřazení k procesu schvalovatele
- F.4 Odebrání schvalovatele od souboru
- F.5 Možnost schválit či neschválit dokument
- F.6 Systém notifikací
 - F.7.1 Notifikace vlastníka dokumentu při změnách
 - F.7.2 Možnost notifikovat schvalovatele, že byl do procesu přidán
- F.7 Komentáře
 - F.7.1 Možnost přidání komentáře k procesu
 - F.7.2 Možnost odebrání komentáře k procesu
 - F.7.3 Možnost nastavení komentáře neviditelného pro klienta
- F.8 Druhá strana nemusí mít Google účet
- F.9 Zvýraznění v prostředí Google Drive
- F.10 Použitelnost na mobilních telefonech

Zde jsou blíže popsány vybrané požadavky:

Požadavky F.1 - F.5

Naprostě klíčové a jasné požadavky, bez kterého by celá aplikace neměla smysl. U F.3 je počítáno s automatickým nastavením práv k danému souboru.

F.6 Systém notifikací

Nelze neustále sledovat nastalé změny v procesu ručně, proto je třeba nějak notifikovat uživatele, že se v procesu, v němž je zainteresovaný, něco změnilo.

F.7. Komentáře

Konkrétně požadavek F.7.3 je možné v produkci využít pro interní záležitosti firmy při zachování konzistence umístění komentářů.

F.8. Druhá strana nemusí vlastnit Google účet

Při používání *Workflows* (2.2.1) a *ApproveForMe* (2.2.2) byly obě strany nuceny vlastnit Google účet. Pro některé klienty toto může být problém, proto jeden z požadavků zní takto.

F.9 Zvýraznění v prostředí Google Drive

Uživatel neměl při používání již existujících řešení a prohlížení svého *Drive* přehled o tom, které dokumenty jsou v procesu schvalování. To by se mělo změnit.

F.10 Použitelnost na mobilních telefonech

Mobilní zařízení jsou čím dál ve větší oblibě. Zaměstnanci jsou leckdy na různých schůzkách, briefincích nebo na cestách mezi klienty a při tom musí být neustále online a schopni řešit případné nové záležitosti či problémy. Při tvorbě aplikace je třeba na její mobilní podporu myslet v návrhu i při implementaci.

2.3.2 Nefunkční požadavky

- N.1 Webová aplikace
- N.3 Zabezpečení
- N.2 Snadná integrovatelnost do produkčního prostředí
- N.3 Jednoduché uživatelské prostředí
- N.4 Anglický jazyk aplikace
- N.5 Komentovaný kód
- N.6 Použití databáze MySQL

N.1 Webová aplikace

Vytvářet v dnešní době nativní aplikaci pro věc tohoto charakteru je jak z hlediska finančního, tak z hlediska pohodlí uživatele naprostý nesmysl. Vše od kancelářských balíčků až po grafické editory se v dnešní době přesouvá na web, na místo, kde má uživatel odkudkoliv a kdykoliv přístup ke svým datům. Navíc zadavatelská firma vlastní server (1.3), kde plánuje aplikaci provozovat a vzhledem k nainstalovaným prostředkům je tedy vhodné, aby byla webová.

N.2 Zabezpečení

Vzhledem k tomu, že spousta projektu a obsahu s nimi spojeným, které se ve firmě vytváří podléhá dohodě o mlčenlivosti (NDA) a je tedy třeba zajistit, aby nepovoláný uživatel neměl k citlivým dokumentům přístup.

N.3 Snadná integrovatelnost do produkčního prostředí

Aplikace by měla být snadno nasaditelná do prostředí, ve kterém poběží (1.3).

N.4 Anglický jazyk

Jelikož spousta klientů, ale i někteří zaměstnanci nejsou z České republiky, či rodilí mluvčí, je třeba, aby výsledná aplikace byla v jazyce, kterému budou všichni rozumět. Nejlogičtější volbou je angličtina. Je to jazyk, u kterého jsou všichni zaměstnanci minimálně na úrovni B1[13], takže by se zde neměla objevit žádná jazyková bariéra.

N.5 Komentovaný kód

Protože chceme zachovat možnost aplikaci rozšiřovat, je třeba mít kód komentovaný, aby další programátor, který kód uvidí poprvé měl pokud možno seznámení co možná nejjednodušší. Šetří se tím čas i finance vynaložené na vývoj.

2.4 Shrnutí

Jak je vidět z výše uvedených požadavků, bylo si z existujících řešení vzato ponaučení. Bez jejich splnění nelze považovat úkol vývoje aplikace za dokončený.

Návrh aplikace

Na základě získaných požadavků bude v této kapitole proveden návrh aplikace, což je další fáze při tvorbě software. Nejprve čtenáři představím typy uživatelů, jež se mohou v systému vyskytnout, následně vytvořím návrh uživatelského prostředí. V druhé části se zaměřím na návrh databáze, popíši jednotlivé tabulky a vztahy mezi entitami. V jednotlivých částech je také uvedeno, jakého požadavku (2.3) se týká a jak ho řeší.

3.1 Uživatelé

Jelikož jsou data, se kterými se bude v aplikaci pracovat leckdy velice citlivá, je třeba aby se k nim nedostali nepovolaní uživatelé. Rozhodl jsem se využít toho, že firma používá Google Apps na širokou škálů věcí, a vyřešit problém autorizace pomocí Google. Navíc tím odpadá nutnost vlastnit další účet, protože každý zaměstnanec ho již má.

V rámci procesu se v aplikaci mohou objevit 3 typy uživatelů. Zvolil jsem pojmenování *Vlastník*, *Dohlízející - zaměstnanec* a *Dohlízející - klient* (dále pouze *Zaměstnanec*, respektive *Klient*). Klient je označení pro dohlízející osobu z jiné firmy. Každá role má jiná oprávnění co se týče práce s *procesem*. Pokud tedy v dalších částech této práci budu hovořit o uživateli, budu uvádět tyto názvy.

Mezi dohlízejícími osobami je třeba takto rozlišovat kvůli požadavku F.8, jež zbavuje klienta povinnosti vlastnit Google účet. Klient musí být schopen dostat se k datům bez přihlášení. Níže je vysvětleno, jak toho bude docíleno. Následuje popis jednotlivých rolí a jejich vlastností. Nakonec je vše znázorněno v diagramu 3.1 vytvořeném v Enterprise Architect[14]

Vlastník

Poté, co uživatel inicializuje proces schvalování, stává se jeho *vlastníkem*. Může tento proces editovat přidáváním či odebráním zainteresovaných osob, sou-

3. NÁVRH APLIKACE

borů z *Drive* a může proces komentovat. V pomyslné hierarchii uživatelů, je tato role nejvýše.

Zaměstnanec

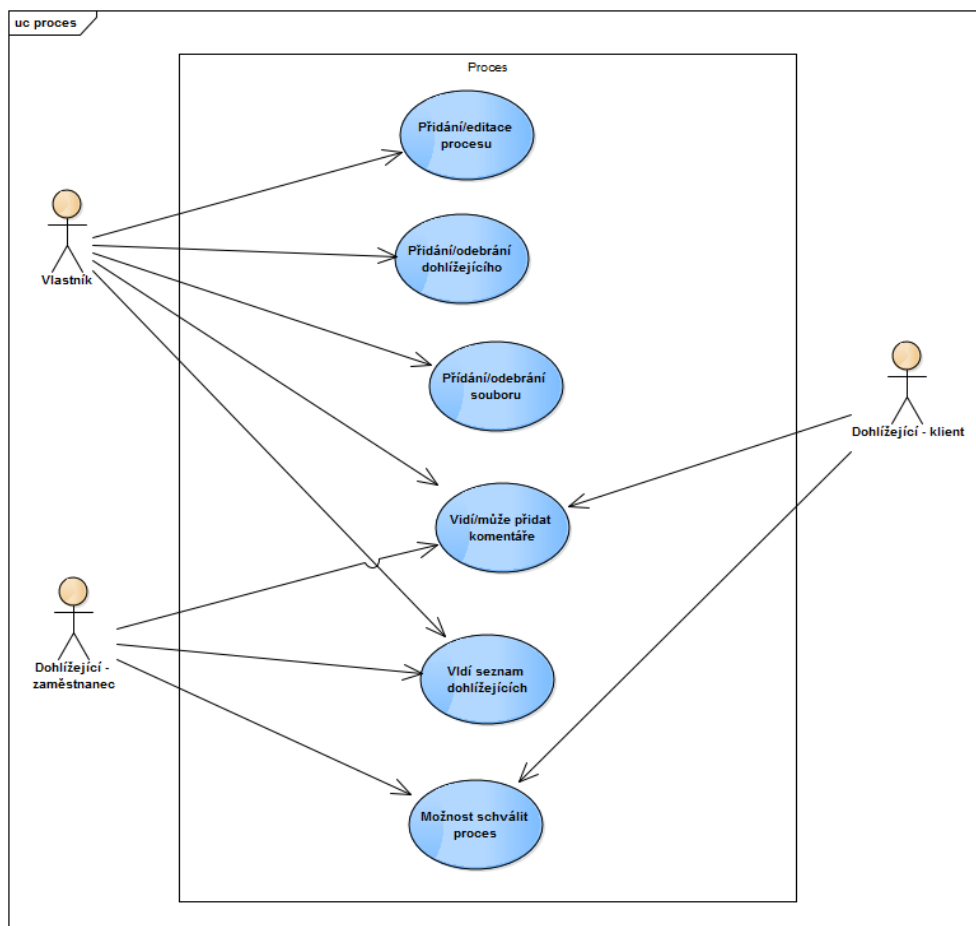
Tuto roli uživatel získá, pokud ho *vlastník* přidá do procesu a zároveň patří do firmy. Narozdíl od *vlastníka* nemůže nijak editovat proces schvalování, ve kterém je přidán, ale může samozřejmě komentovat, prohlížet soubory a schvalovat je.

Klient

Lidé z jiných firem budou z pohledu aplikace *klienti*. Tito uživatelé jsou co se týče možností a práv nejvíce omezeni. Podle funkčního požadavku *F.6.3* se mu například nezobrazí vybrané komentáře. *Klient* se jako jediný typ uživatele nikam nepřihlašuje, ale dostane unikátní vygenerovaný odkaz, pomocí nějž do aplikace dostane.

3.1.1 Grafické znázornění

Diagram případu užití (3.1) znázorňuje interakce všech typů uživatelů se schvalovacím procesem.



Obrázek 3.1: Uživatelé systému v interakci se schvalovacím procesem

3.2 Uživatelské prostředí

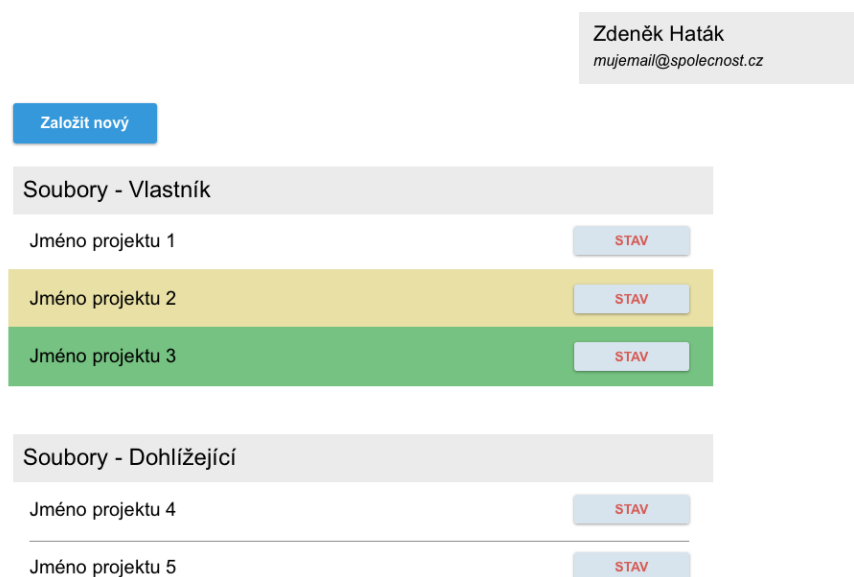
Jedním z požadavků na aplikaci bylo i jednoduché uživatelské rozhraní (2.3).

Návrh UI je podle mého jedním z nejpodceňovanějších, ale přitom klíčových kroků při tvorbě software nebo webové stránky. Sebelepší aplikace s mnoha funkcemi může ztratit svou kvalitu právě kvůli špatnému návrhu uživatelského prostředí. Pokud bychom se na věc podívali z finančního hlediska, špatné UI může způsobit i výrazné ztráty. Když bude uživatel při používání webové stránky tápat, není to dobře.

K lepší představě, jak bude výsledné rozhraní vypadat slouží drátové modely (wireframes). Pomáhají si představit průchod aplikací a odladit očekávaný dojem z aplikace. Výsledný vzhled se samozřejmě od wireframů liší, umístění a skladba prvků by se však lišit téměř neměla.

Na základě funkčních požadavků společnosti na výsledný nástroj jsem tedy vytvořil drátové návrhy celé aplikace. Při jejich tvorbě jsem se snažil mít na

3. NÁVRH APLIKACE



Obrázek 3.2: Drátový model hlavní obrazovky aplikace

mysli to, že UI musí být jednoduché, bez zbytečných rušivých prvků. Všechny modely byly v průběhu tvorby s firmou konzultovány. K tvorbě těchto drátových modelů jsem použil webovou aplikaci UXPin[15], která nabízí širokou škálu předpřipravených webových prvků jako jsou například formuláře, tabulky či ikony.

Níže jsou uvedeny jednotlivé obrazovky aplikace spolu s popisem prvků a funkcionality, jež se na nich nachází. Výsledný vzhled aplikace je možné vidět v příloze A. Byl navrhován s ohledem na požadavek F.10, a sice podporu mobilních zařízení.

3.2.1 Hlavní obrazovka

Hlavní obrazovka (3.2) se uživateli zobrazí ihned po přihlášení a slouží pro něj jako rozcestník pro další kroky. Je zde možnost si prohlédnout jak dokumenty, které přihlašený potřebuje od jiných živateľů schválit, tak ty, které má schválit sám. Dále je zde tlačítko, které uživatele přesune na další obrazovku, kde může založit nový proces od začátku.

Obrázek 3.3: Drátový model obrazovky Začít proces

3.2.2 Začátek procesu

Na této obrazovce (3.3) může kdokoli s přístupem do aplikace (tedy zaměstnanec zadavatelské firmy) odstartovat proces schvalování. Je zde možnost v rámci aplikace proces pojmenovat. K tomu slouží políčko **Jméno projektu**. Aplikace počítá dle požadavku F.1 s možností přidání do procesu více souborů.

Do formulářového prvku *textarea* **Text pro příjemce** lze vepsat text, který se, podle požadavku „F.5.2“ pošle schvalovateli, když je poprvé pozván vlastníkem k zapojení se do procesu schvalování a bude se zobrazovat na jeho detailu. Dále má vlastník procesu možnost zvolit, kteří lidé mají na daný proces dohlížet. Volí mezi klienty, u kterých zadá jejich jméno a email a mezi zaměstnanci firmy.

Nakonec, tlačítko **Začít proces** proces uloží do databáze.

3.2.3 Detail procesu

Tuto obrazovku (3.4) uživatel vidí, pokud si proces rozklikne, v případě zaměstnance firmy, na hlavní stránce a nebo, pokud navštíví vygenerovaný odkaz, když je *klient*. V reálné aplikaci se tato obrazovka bude lehce lišit v

3. NÁVRH APLIKACE

Jméno projektu

[Odkaz na soubor 1](#)

[Odkaz na soubor 2](#)

Úvodní text k procesu

Vlastník procesu:
-- Zdenek Hatak (hatak@foll...)

Klienti:	Firma:
Jméno klienta	-- Lukas
Jméno klienta	-- Terka

Komentáře

Terka 22. únor 2015, 11:11
Ta část ohledně... viditelné

Jméno klienta 22. únor 2015, 12:44
V pořádku

Váš komentář

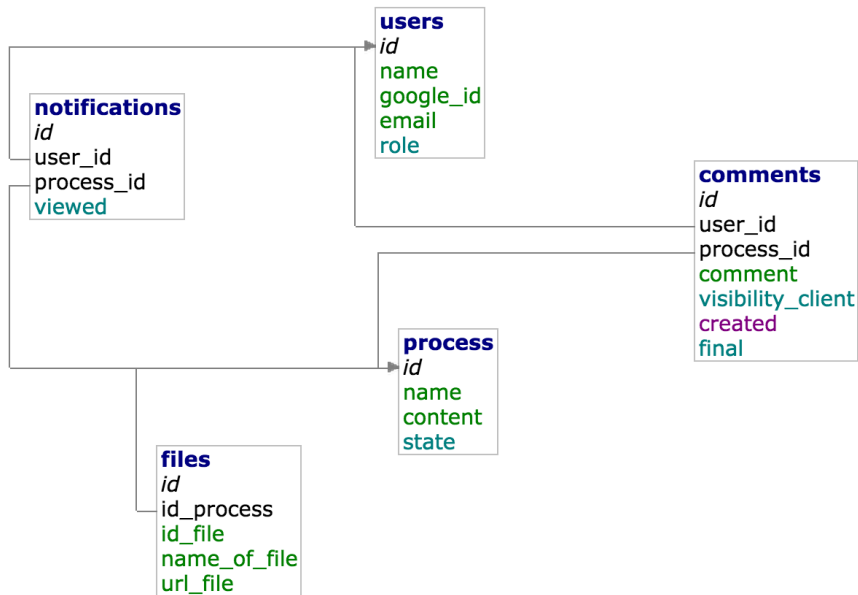
Viditelné klientům

Obrázek 3.4: Drátový model detailu procesu

závislosti na typu uživatele, který si ji zobrazí. Jak jsem již zmínil, typ *vlastník* může proces editovat, zato typ *zaměstnanec* a typ *klient* ne. Ten je navíc ochuzen o některé komentářů označené jako privátní (viditelné pouze firemním zaměstnancům). Je zbytečné zde uvádět všechny typy obrazovek, protože pro představu, jaké prvky se na nich nacházejí stačí tato.

3.3 Databáze

Na obr. 3.5 lze vidět prvky, které byly identifikovány jako entity a u nichž má smysl z databázového hlediska udržovat data.



Obrázek 3.5: Část databázového modelu aplikace

3.3.1 Tabulka Users

Tabulka 3.1: Popis tabulky Users

Sloupec	Typ, popis
ID	Int (UID)
name	Int, ID procesu, u kterého komentář je
user_id	Varchar, jméno uživatele
google_id	Varchar, Google ID uživatele
email	Varchar, email uživatele
role	Enum(admin, client)

Jak již název napovídá, tabulka schraňuje informace o všech uživateli, kteří se do používání aplikace zapojí. Pokud se uživatel poprvé přihlásí přes svůj firemní Google účet, bude přidán do tabulky s rolí typu *admin*. Druhý typ *client* bude sloužit pro klienta.

3.3.2 Tabulka Process

Tabulka procesu má jednoduchou strukturu, ukládáme pouze nejnütnější informace.

3. NÁVRH APLIKACE

Tabulka 3.2: Popis tabulky Process

Sloupec	Typ, popis
ID	Int, UID
name	Int, pojmenování procesu
content	Varchar, úvodní text k procesu
state	Enum(process, completed), stav procesu, po schválení

3.3.3 Tabulka Files

Jelikož chceme mít možnost zařadit do procesu více souborů, je nutná přítomnost této tabulky *Files*.

Tabulka 3.3: Popis tabulky Files

Sloupec	Typ, popis
ID	Int, UID
id_process	Int, ID procesu, ke kterému soubor patří
id_file	Varchar, Google ID souboru na Drive
name_of_file	Varchar, Název souboru na Drive
url	Varchar, sdílená URL souboru

3.3.4 Tabulka Comments

Jedním funkčním požadavkem je i možnost práce s komentáři. K jejich uložení slouží tabulka *Comments*.

Tabulka 3.4: Popis tabulky Comments

Sloupec	Typ, popis
ID	Int (UID)
process_id	Int, ID procesu, u kterého komentář je
user_id	Int, ID uživatele, který komentář poslal
comment	Longtext, text komentáře
visibility_client	Enum(yes, no), viditelnost komentáře klientovi
created	Timestamp, časová značka vytvoření komentáře
final	Enum(yes, no), příznak, zda toto byl schvalovací komentář

3.3.5 Tabulka Notifications

Do tabulky *Notifications* se ukládá změny od poslední návštěvy *Vlastníka*. Pokud se v jeho procesu něco od poslední návštěvy změnilo, založí se v této

tabulce záznam. Jakmile si vlastník změnu prohlédne, záznamu se změní hodnota `viewed`.

Tabulka 3.5: Popis tabulky Notifications

Sloupec	Typ, popis
ID	Int (UID)
useer_id	Int, ID uživatele
process_id	Int, ID procesu, který se změnil
viewed	Enum(yes, no), viditelnost komentáře klientovi

3.4 Notifikace

Podle požadavku *F.6* v 2.3 má aplikace umět notifikovat uživatele při různých událostech. Nejjednodušší a zároveň nejefektivnější cestou jak tohoto docílit je přes email. Může se zdát, že se tímto vracím k původnímu problému, jež způsobil, že se vůbec aplikace začala vyvíjet, avšak není tomu tak. Zde hovořím pouze o jednotkách krátkých zpráv, notifikujících uživatele o hlavních událostech, které v procesu nastaly.

V zásadě budou v rámci aplikace existovat dva typy notifikací:

Emailové notifikace

Každý uživatel systému bude při změně, která se ho nějakým způsobem týká notifikován krátkým emailem. Email se pošle v následujících situacích:

- Uživatel byl požádán o názor na projekt (byl přidán do procesu)
- Byl přidán komentář

V každém emailu bude připojen odkaz do aplikace, která uživatele přivede rovnou na daný proces.

Notifikace v aplikaci


Tyto notifikace bude dostávat každý přihlášený uživatel, tedy všichni, kromě *klienta*. Pokud se od poslední návštěvy něco v jakémkoliv procesu změnilo, tak budou tyto procesy na uživatelově hlavní stránce zvýrazněny. Jakmile si proces přečte, notifikace zmizí.

3.5 Google Drive

Abych jako zaměstnanec firmy poznal i při prohlížení svého Google Disku, že daný soubor je součástí nějakého schvalovacího procesu, je třeba navrhnout


označení takovýchto souborů. Není nutné vymýšlet žádné složitosti. Soubor se přidáním do procesu se jednoduše přejmenuje.

Pro soubor, který je v průběhu schvátování využiji Unicode znak U+1405 (3.6)

 ▷ Ukázkový soubor zařazený do procesu schvátování

Obrázek 3.6: Soubor, jež je v průběhu schvalování

Pro soubor, který je schválen znak U+2714 (3.7)

 ✓ Ukázkový soubor zařazený do procesu schvalování

Obrázek 3.7: Soubor, jež je schválen

Tímto jsem splnil funkční požadavek *F.9* (2.3)

3.6 Shrnutí

Následuje shrnutí nejdůležitějších bodů a poznatků, které se mohly v záplavě informací v průběhu kapitoly ztratit či nemusely být pochopeny.

- V aplikaci se budou vyskytovat tři druhy uživatelů. Vlastník, dohlížející zaměstnanec firmy a dohlížející klient.
- Přihlašovat se zabezpečené části budou přes svůj Google účet. Všichni kromě klientů. Klientovi aplikace vygeneruje unikátní link, pro každý proces zvlášť, přes který bude do aplikace přistupovat.
- Uživatel bude přidán do databáze po prvním přihlášení.
- Notifikace budou dvou typů. Emailové a interní v aplikaci.

Implementace

Tato kapitola se věnuje implementaci celé aplikace. Obsahuje popis volby programovacích jazyků a nástrojů, některým je věnována větší pozornost než jiným. Dále se zde rozebrána implementace modulů, ze kterých se aplikace skládá, a sice frontendu a backendu. Stručně popsána je i část týkající se Google Drive. Při popisu technických pasáží je brána v potaz skutečnost, že uživatel nepochází z oboru a nemusí všem záležitostem v plné míře rozumět. Proto je zvolen popis spíše netechnického charakteru. Zdrojové kódy k aplikaci jsou uloženy na přiloženém CD.

4.1 Programy využívané při tvorbě

Zde jsou popsány všechny hlavní nástroje, které jsem při implementaci používal a stojí za zmínku.

Sublime Text 3

Jako hlavní editor při psaní zdrojového kódu jsem použil oblíbený SublimeText 3[16], který používám již několik let. Při doplnění různých rozšíření, se z něj stává poměrně slušně vybavené IDE (program pro vývoj software), obsahující vše potřebné pro vývoj webů i webových aplikací.

Google Chrome

Při vývoji aplikace jsem využíval webový prohlížeč od společnosti Google, Chrome[17] v aktuální verzi. Obsahuje nástroj Web Developer Tools, který velice usnadňuje ladění. Umožňuje například zobrazit hlavičky, které se posílají v požadavcích, či vypsat chybové hlášky způsobené

Adminer

Adminer[18] je grafický nástroj pro správu různých typů databází. Nabízí přehledné prostředí pro veškerou běžnou práci s databázovými tabulkami jako jsou dotazy nad daty, jejich exporty či tvorba pohledů.

MAMP

Balík MAMP[19] je soubor aplikací obsahující interpret jazyka PHP, MySQL a serveru Apache[20]. Slouží k lokálnímu vývoji aplikací založených na PHP.

4.2 Volba programovacích jazyků a nástrojů

Při vývoji každé aplikace je třeba počítat s charakterem prostředí, ve kterém se aplikace bude používat a přispůsobit tomu i volbu nástrojů či programovacích jazyků. Jelikož jeden z požadavků byl, aby aplikace byla webová, použil jsem k tomu i odpovídající technologie. Ty se samozřejmě liší od toho, zda jsou použity na frontendové nebo backendové části. Budou probrány detailněji již zde, aby v části o realizaci mohla být věnována pozornost hlavním částem aplikace..

4.2.1 Frontend modul

Vzhledem k tomu, že pracuji v zadavatelské firmě jako front-end developer, je tak trochu mým úkolem držet krok s nejmodernějšími technologiemi související s tvorbou webových aplikací. Webový front-end prošel za poslední 3 roky výraznou změnou. Objevily se nové postupy, nespočet frameworků a jazyků kompilujících se do Javascriptu. Stále více se vývoj soustředí na mobilní telefony a responsivní webové stránky.

Nejlogičtější volbou jazyků pro front-end HTML5, CSS3 a Javascriptu. Díky k platformové diverzitě 1.3 se dá očekávat, že zaměstnanci používají všechny běžné prohlížeče.

Sass

Sass[21] je nástavba klasických kaskádových stylů, přidává do CSS prvky programovacích jazyků jako proměnné, jednoduché funkce či podmínky a cykly. Podporuje taktéž vnořování při zápisu CSS stylů, což kód zpřehledňuje a obvlášt při velkých projektech celý vývoj zrychluje. Jelikož jazyk CSS tyto věci nepodporuje, je nutné do něj jazyk Sass kompilovat. Ukázka kódu znázorňuje použití proměnných a mixinů, které hodí obzvlášť při použití CSS3 vlastností4.1.

Ukázka zdrojového kódu 4.1: Jazyk Sass

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  -ms-border-radius: $radius;  
  border-radius: $radius;  
}  
  
.box { @include border-radius(10px); }
```

jQuery

Javascriptový framework usnadňuje převážně operace s DOM stránky. Nabízí pohodlné prostředí pro práci s AJAXem, jež slouží k výměně dat se serverem a k aktualizaci částí stránek bez nutnosti celou stránku znovu načítat.[22] Disponuje velkým množstvím již hotových metod a to jak pro práci s již zmíněným DOM, tak s animacemi. Poskytuje funkce pro obsluhu uživatelských událostí a existuje pro něj také mnoho hotových pluginů (krátký skript rozšiřující funkčnost původní aplikace) od ostatních vývojářů, bohužel mnohdy nevalné kvality.

Při vlastní implementaci budu *jQuery* využívat hlavní pro práci s AJAX technologií, aby byl dojem při používání nástroje co možná nejpříjemnější a samozřejmě pro různé jednoduché animace jako např. vyjždění boxů po kliku na tlačítko.

Grunt.js

Automatizační nástroj Grunt.js[23] řeší velkou škálu problémů související s často opakovanými úkony jako již zmíněná kompilace Sass jazyka do CSS či spojování souborů do jednoho společného. Také umí u souborů sledovat, zda v nich došlo k nějaké změně a spouštět nadefinovanou akci. Nadefinovaných akcí může být spousta. Namátkou:

Spojování souborů

Typická situace k použití této úlohy nastává v případě, když je do HTML stránky třeba vložit spoustu javascriptových souborů, typicky právě jQuery pluginů. Grunt tyto soubory spojí do jednoho a tím při každém novém načtení stránky není nutné stahovat tolik souborů (co soubor, to požadavek).

Mimifikace

Tato úloha odstraní ze zdrojového kódu zbytečné mezery a nové řádky.

Ukázka zdrojového kódu 4.2: Výřez ze souboru Gruntfile.js

```
sass: {
  dist: {
    options: {
      style: 'compressed'
    },
    files: {
      'www/public/css/app.min.css': '
      app-frontend/sass/app.scss'
    }
  }
},
watch: {
  sass: {
    files: [
      'app-frontend/sass/*.scss',
      'app-frontend/sass/**/*.scss'
    ],
    tasks: ['sass:dist']
  }
}
```

Kód je potom sice nečitelný pro běžného uživatele, ale má menší datový objem, takže se načte rychleji.

Kompilace

Kompilace do CSS z *Sass* či do Javascriptu v případě *CoffeeScriptu* je jednou z nejdůležitějších úloh, co Grunt nabízí.

V ukázce (4.2) je možné vidět výřez ze souboru *Gruntfile.js*. Jsou zde nadefinované dvě úlohy, a sice kompilování souborů z jazyka *Sass* do CSS a sledování změn v těchto souborech a spouštění předchozí úlohy. Grunt se spouští z příkazové řádky a úloha *watch* běží neustále.

4.2.2 Backend modul

PHP

PHP je stále nejrozšířenějším jazykem na poli webových aplikací[24]. Jedná se o skriptovací jazyk na straně serveru, jež podporuje objektově orientované programování (OOP). Rozhodl jsem se pro něj také proto, že s ostatními jazyky (Python, Ruby), které se dají použít pro naprogramování webové apli-

Ukázka zdrojového kódu 4.3: Šablonovací systém Latte

```

<ul n:if="$items">
  {foreach $items as $item}
    <li id="item-{$iterator->counter}">{$item|capitalize}
  </li>
{/foreach}
</ul>

```

kace nemám velké zkušenosti. Navíc pro PHP existuje spousta frameworků, které práci značně urychlují.

Nette

Pro rychlejší vývoj jsem tedy zvolil český framework Nette[25], který má v Česku dostatečně velkou komunitu. Navíc s ním mám již malé dřívější zkušenosti, narozdíl od jiných PHP frameworků jako například Symfony 2[26] či Laravel[27]. Obsahuje vrstvy pro pohodlnou práci s databází, při jejichž použití prakticky ani programátor nemusí databázový jazyk znát.

Disponuje též nástrojem Tracy[28]. Ten slouží k odhalení případných chyb, které programátor při práci natropí, k jejich logování či k měření času běhu aplikace. Zvládá přehledně vypisovat veškeré chyby, které v programu nastanou, takže je z něj kvalitní pomocník při programování v PHP.

Framework Nette využívá šablonovací systém Latte[29], pomocí kterého vykresluje uživateli na výstup získaná data z komponent. Není zde obsažena žádná logika, jedná se pouze o jazyk HTML rozšířený o některé funkce jako jsou například cykly a podmínky. Také zajišťuje, že je výstup chráněný proti útokům jako je například XSS. Uvedená ukázka demonstruje použití cyklů a podmínek (4.3)

4.3 Realizace

V této části bude stručně popsána samotná realizace.

Google

Samostatně jsem se rozhodl popsat i část související s Google, jelikož s ním aplikace úzce souvisí. Google poskytuje poměrně kvalitní API ke všem svým hlavním veřejným službám, kvalitně zdokumentované a udržované.

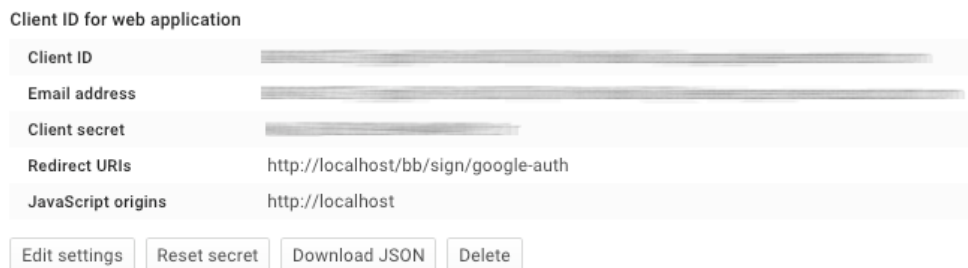
Google po vývojářích vyžaduje, aby si ve *vývojářské konzoli*[30] zaregistrovali každou aplikaci, která bude jeho API využívat. Přesněji řečeno, každou aplikaci, která bude vyžadovat přihlašování ostatních uživatelů. Má tak přehled o tom, kdo, kde a jak služby využívá, při vysoké zátěži (při velkém počtu

4. IMPLEMENTACE

požadavků na API) může uživatelům účtovat poplatky. I vlastník aplikace má možnost sledovat traffic (množství dat, které aplikací projde), logy nebo nastavit název aplikaci či její logo.

K tomu, aby vývojář mohl vyvíjet pomocí Google API stačí obyčejný Google účet. Pro testovací účely jsem na svém účtu ve *vývojářské konzoli* vytvořil aplikaci *BubbleFlow*. Jelikož bude aplikace využívat Drive API, je třeba v *konzoli* toto API povolit. V návrhu v části 3.2.2 je sekce, kde uživatel přidává soubor z Drive a proto se zde nabízí možnost využít *Google Picker*[12], který slouží právě k vybírání všech možných souborů uložených na serverech Google.

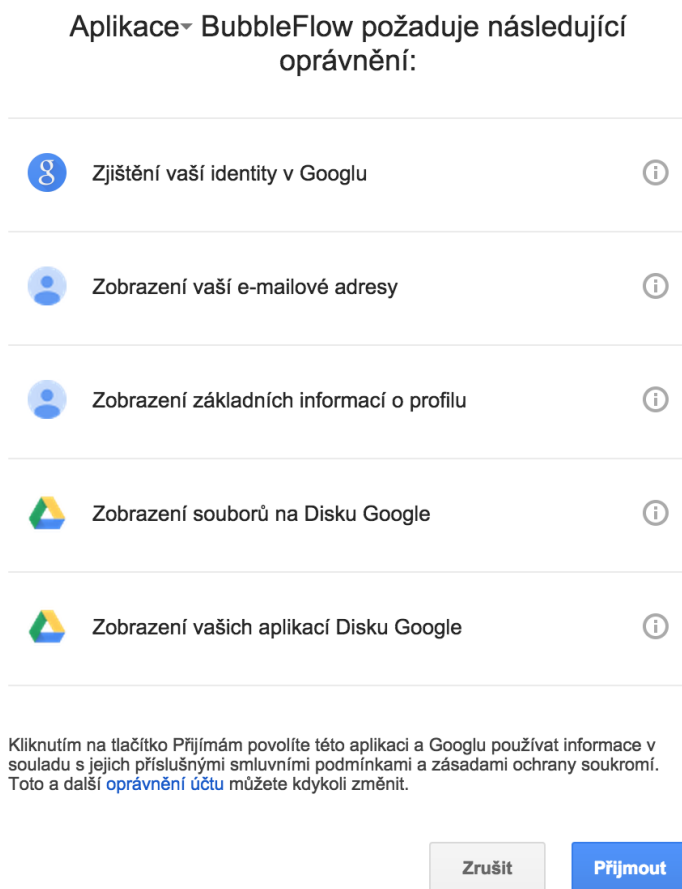
Po povolení vybraných API je třeba v *konzoli* vygenerovat *Client ID* a *Client Secret*, jež slouží k zabezpečení uživatelských údajů skrze OAuth 2.0[31] (zabezpečovací protokol, který Google používá pro autorizaci uživatelů). Dále je třeba nastavit doménu, na které aplikace poběží a bude přístupná z Internetu. Na obrázku 4.1 je vše ukázáno.



Obrázek 4.1: Developerská Google Konzole

Scopes

Pro slovo *scope* nejspíše neexistuje v češtině výraz, který by vystihoval jeho význam v oboru programování. V souvislosti s Google API jde prakticky o rozsah práv, které aplikace využívající API bude po přihlášeném uživateli požadovat. V rámci Drive jsou k dispozici například *scopes* pro zobrazení souborů či pro jejich editaci. V rámci obyčejného Google účtu jsou to práva pro to, že aplikace může zobrazit o uživateli základní informace. Na obrázku 4.2 je ukáзка dialogu, který se objeví po pokusu o přihlášení.



Obrázek 4.2: Aplikace žádající uživatele o oprávnění

Práva uživatele k souborům

Aby Google zamezil přístupu neoprávněných osob k souborům uloženým na Drive, rozlišuje několik úrovní práv.

1. Právo upravovat
2. Právo komentovat
3. Právo prohlížet

Pro potřeby této aplikace bude stačit 3. úroveň, která zajistí, že kdokoliv, kdo bude znát odkaz na soubor, si ho bude moci prohlédnout.

Ukázka zdrojového kódu 4.4: Create skript databáze

```
CREATE TABLE IF NOT EXISTS 'comments' (  
  'id' int(11) NOT NULL,  
  'process_id' int(11) NOT NULL,  
  'user_id' int(11) NOT NULL,  
  'comment' longtext COLLATE utf8_czech_ci NOT NULL,  
  'visibility_client' enum('yes','no') COLLATE  
    utf8_czech_ci NOT NULL DEFAULT 'no',  
  'created' timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP  
  )  
  'final' enum('yes','no') COLLATE utf8_czech_ci NOT  
  NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=  
  utf8_czech_ci
```

Databázové úložiště

Podle návrhu databáze (3) jsem v Admineru vytvořil tabulky s příslušnými sloupci a následně jsem z něj vyexportoval tzv. *create script*, který slouží k inicializaci nové databáze v případě přenášení aplikace do jiného prostředí. Vzhledem k tomu, že na firemním serveru se nachází MySQL, je i databáze aplikace tohoto typu. Následuje ukázka vytvořeného skriptu (4.4), který se nachází také na příloženém CD:

Aplikace

Framework postavený na architektuře Model-View-Presenter[32]. Rozděluje aplikaci na tři části, kde model má za úkol převážně práci s databází. View je na opačné straně, u uživatele. Zobrazuje získaná data model návštěvníkovi. Spojovacím článkem je presenter, který reaguje na změny ve *view* a přenáší je do modelu.

Nejdůležitější část kódu se nachází z hlediska Nette struktury ve složce *app*. Obsahuje složku s konfiguračními soubory, složku s modely, šablonami a presentery.

Následuje popis pouze vybraných souborů a důležitých částí aplikace. Pokud má čtenář zájem si prohlédnout zdrojový kód podrobněji, necht se podívat na příložené CD:

app/config/config.neon

Konfigurační soubor, kde se nastavují přístupy k databázi, a vygenerované API klíče z Vývojářské konzole.

app/model/Comment.php

Jeden z modelů aplikace. Obsahuje převážně metody CRUD (základní metody pro vložení, smazání a vypsání dat z databáze) a zajišťuje, jak již název napovídá ukládání a zobrazování komentářů.

app/model/Google.php

Tento model komunikuje pomocí Google API s *Drive*. Jsou zde metody pro přejmenování souboru, pokud se dostane do procesu či pro nastavení práv. Ukázka kódu 4.5 ukazuje práci s API při nastavování souborových práv.

app/model/Process.php

Kromě metod CRUD pro práci s procesem obsahuje tento model i jednu pro spočítání hash hodnoty pro klienta. Toto je třeba, aby se klient dostal do aplikace i bez Google účtu. (Požadavek F.9) Každý klient je s procesem spojen unikátní hash hodnotou, která je obsažena v odkazu v emailu, jež mu aplikace zašle při přidání do procesu.

app/model/NotificationCenter.php

Model mající pouze jedinou funkci, a sice posílání emailů při přidání uživatele do procesu a přidání komentáře. Využívá při tom objekt *Nette\Mail*.

app/presenters/*Presenter.php

Presentery se starají v aplikaci o komunikaci mezi šablonou a modelem. Například *ProcessPresenter.php* má na starost mimo jiné i sběr dat z formuláře pro přidání komentáře a jeho AJAXové odeslání a uložení do databáze.

app/presenters/templates/*

Složka se šablonami (*views*). Každý stav, každá obrazovka v aplikaci má svou šablonu. Zde se nachází Latte kód (HTML obohacené o prvky programovacích jazyků), tedy vše, co vidí uživatel.

app-frontend/*

Tato složka obsahuje zejména Sass soubory, které se pomocí Gruntfile.js kompilují do jazyka CSS.

Ukázka zdrojového kódu 4.5: Ukázka práce s Google API

```
$newPermission = new \Google_Service_Drive_Permission();
$newPermission->setValue('me');
$newPermission->setType('anyone');
$newPermission->setRole('reader');

try {
    return $service->permissions->insert($fileId,
        $newPermission);
} catch (Exception $e) {
    print "An error occurred: " . $e->getMessage();
}
```

4.4 Shrnutí

Volba PHP frameworku Nette se nakonec ukázala býti správná, z počátku jsem měl obavy, jelikož jsem s ním sice nějakou zkušenost měl, ale myslel jsem si, že se budu spíše trápit než něco reálně vyvíjet. V průběhu implementace se samozřejmě objevily chvíle, kdy jsem si nevěděl rady a musel požádat o pomoc přátele či komunitu, ale nebylo jich mnoho a mohu s klidem říci, že jsem použitím neprohloupil. Navíc díky velkému rozšíření frameworku Nette v České republice je zaručeno, že pokud se aplikace bude dále rozrůstat, je zde velká šance, že programátor, který se toho ujme bude Nette znát, což mu ulehčí práci s cizím kódem a tím se další vývoj zefektivní.

Dále jsem si vyzkoušel práci s Google API, s nímž jsem neměl téměř žádné předešlé zkušenosti. Tato zkušenost se bude jistě v budoucnu hodit. I zde jsem narazil na několik problémů, z nichž největší asi bylo nastavení správných oprávnění pro soubor, aby si ho mohli zobrazit i *klienti*. Problém zde spočíval v nekonzistenci dokumentace k API s realitou při použití jedné z metod pro nastavení práv souborům.

Nasazení a výsledky v praxi

Poslední kapitola se zabývá pilotním nasazením aplikace a následným ekonomickým zhodnocením do firemní praxe.

5.1 Nasazení

5.1.1 Instalační příručka

Prvotním předpokladem správného fungování je účet Google Apps. Ten firma samozřejmě vlastní, ostatně všichni stálí zaměstnanci a i někteří externisté mají email na firemní doméně v rámci Google Apps. Navíc, již v úvodu práce (1.2) jsem se zmínil, že se *Drive*, který je součástí Google Apps využívá jako úložiště pro některý typ obsahu.

Pro bezproblémové nasazení a běh programu je třeba:

- Webový server, například Apache
- Interpreter jazyka PHP
- Databáze MySQL
- Volitelně grafický nástroj pro práci s databází, například Adminer

Vzhledem k tomu, že firma má vlastní server, kde je všechno zmíněné nainstalované, je nasazení aplikace velice přímočaré a jednoduché. Následující úkony je nutné vykonat:

1. Založit aplikaci v Google Developer Konzoli
2. Povolit v ní Drive API a Google Picker API
3. Nechat vygenerovat unikátní *Client ID* a *Client Secret* a vyplnit webovou adresu, na které aplikace poběží

4. Vytvořit v Admineru databázi a spustit nad ní *create script* (je umístěn na příloženém CD)
5. Doplnit DSN záznamy, *Client ID* a *Client Secret* do konfiguračního souboru *app/config/config.neon*
6. Do šablony *@layout.latte* doplnit trojici *Developer Key*, *Client ID* a *ID aplikace*, aby fungovalo vybírání souborů pomocí Google Picker.
7. Nahrát celou aplikaci na server a zpřístupnit ji z Internetu

Jak jsem se zmínil v části 3.6, po prvním uživatelově přihlášení se v databázi v tabulce *users* vytvoří záznam s informacemi o tomto uživateli. Pro další postup je tedy třeba, aby se všichni zaměstnanci alespoň přihlásili, aby se naplnila tabulka s uživateli.

Při nasazování se nevyskytly žádné větší problémy ani zádrhly. Splnil jsem tak nefunkční požadavek *N.2*, ve kterém zadavatelská firma požadovala jednoduché nasazení do produkčního prostředí.

5.1.2 Testování

Po nasazení na firemní server a nastavení všech potřebných věcí bylo uspořádáno menší školení, kde jsem zaměstnancům představil účel aplikace. Zúčastnilo se celkem 14 zaměstnanců. Vzhledem k charakteru aplikace a úkolu, který má plnit se nezúčastnili všichni. Například u externistů či některých developerů je školení zbytečné, když aplikaci používat nebudou.

Poté zaměstnanci, kteří aplikaci reálně využívat budou dostali za úkol nasimulovat mezi sebou typický scénář použití aplikace. Při tomto simulačním testování se nevyskytly žádné problémy co se týče nepochopení aplikace či v neschopnosti se vyznat v prostředí. Rychlé seznámení se dá přisuzovat velice jednoduchému konceptu aplikace a uživatelského prostředí. Navíc zaměstnanci přišly s několika nápady na vylepšení. Uvedeny jsou v 5.4.

Aplikace byla tvořena hlavně pro kolaboraci nad procesem schvalování dokumentů s klienty a proto bylo třeba nějak vyřešit jejich zaškolení. Využije se přitom pravidelných meetingů přímo u klienta, kterých se účastní převážně *projektoví a account manažeři*. Zde proběhne vždy prvotní seznámení klienta a dále bude bezplatná podpora probíhat přes email, popřípadě na dalších osobních schůzkách.

5.1.3 Kontrola požadavků

Funkční požadavky

V průběhu celé práce jsem u jednotlivých částí návrhu či implementace zmiňoval, jakého požadavku se týkají.

Nefunkční požadavky

Všechny nefunkční požadavky byly také splněny.

5.2 Náklady

Při průměrné sazbě 400 Kč/hod jsem vypočítal náklady spojené s vývojem této aplikace. V této částce je započítána i povinná sazba typická pro běžnou OSVČ ve formě daní, sociální a zdravotního pojištění.

5.2.1 Náklady na vývoj

Není to pouze práce programátora, jež je třeba započítat do celkové pracovní doby vývoje. Dále to je analýza problému a požadavků, poté konzultace s vybranými osobami z firmy nad problematikou uživatelského prostředí a v neposlední řadě je nutné počítat se školením zaměstnanců při nasazování aplikace do produkce. Jako programátor jsem se podílel na každém kroku, který vývoj doprovázel a sice analýzy, návrhu, implementace, testování i školení.

Následující tabulka (5.1) znázorňuje pracovní dobu a cenu celé práce vyjádřenou počtem hodin, které jednotlivé části zabraly:

Tabulka 5.1: Náklady na vývoj aplikace

Etapa	Doba	Náklady
Analýza požadavků	10 mhr	4 000 Kč
Návrh aplikace	15 mhr	6 000 Kč
Nastudování Google API	20 mhr	8 000 Kč
Implementace	60 mhr	24 000 Kč
Testování	20 mhr	8 000 Kč
Školení zaměstnanců (14)	2 mhr	11 200 Kč
Celkově	153 mhr	61 200 Kč

Školení klientů nelze rozumně vyčíslit, protože není známé jejich množství a míra spolupráce. Odhadem by měla vždy stačit maximálně hodina. S dalšími náklady se počítá pouze v případě, že by se aplikace měla dále upravovat a doplňovat o nové funkce, což je velice pravděpodobné, jelikož zaměstnanci již navrhli možná rozšíření.

5.2.2 Náklady na údržbu

Náklady na údržbu budou velice nízké. Plánuje se, že výsledný nástroj poběží na subdoméně u firemního webu, takže náklady spojené s webovou adresou budou nulové, pokud věc bereme tak, že by se doména platila i bez existence této aplikace. Je také třeba provádět zálohy databáze, při případném pádu

serveru a ztrátě dat. Zde jsou náklady opět minimální, a spíše časového charakteru. Pravidelné zálohy už se prováděly i předtím, zálohují se totiž databáze všech projektů, které na serveru běží.

5.3 Přínosy

Identifikací problému spojeného s oběhem obsahu a následným vývojem nové aplikace na podporu tohoto procesu získala firma nástroj, jež je prezentovatelný vyššímu managementu u každého klienta, což se o zmatečných emailových konverzacích říci nedalo. Nástroj přehledně znázorňuje celý schvalovací proces od jeho prvotní inicializace přes řešení připomínek až po finální schválení.

Hlavní přínos je tedy také ve zefektivnění kolaborace zaměstnanců a klientů při výměně obsahových dokumentů a řešení jejich schvalování. Hlavní výhody nového nástroje se ukáží spíše při dlouhodobých projektech, než při těch jednorázových, u nichž se na začátku moc problémů nevyskytovalo. Postupem času, jak se budou zaučovat ostatní klienti se dá očekávat zlepšení ještě výraznější.

Další zlepšení se dá očekávat po naimplementování navrhovaných vylepšeníh.

5.4 Budoucnost

Aplikace díky svému univerzálnímu vzhledu není nijak vázána na použití pouze v reklamních či digitálních agenturách a je možné ji používat v široké škále různých firem. Vesměs se dá použít všude, kde je třeba nějak spolupracovat s klienty ve schvalovacím procesu.

Zde jsou uvedeny nápady do budoucna a možná rozšíření výsledné aplikace vycházejících především z zaměstnaneckého otestování. Před reálným nasazením a používáním se plánuje tyto změny implementovat:

Kalendář

- Možnost přiřazení termínu dokončení určitému procesu
- Možnost zobrazení celého kalendáře s termíny dokončení jednotlivých procesů

Seznamy schvalovatelů

Vzhledem k charakteru některých úkolů by bylo dobré mít možnost vytvářet a následně i editovat seznam jmen a emailů lidí, od kterých je žádáno schválení. Toto je možné využít například v případech, kdy se v některých projektech

zajímaví lidé opakují. Šlo by také vytvořit pro každou klientskou firmu vlastní seznam lidí z dané firmy s jejich emaily.

Přiřazení složek

Jeden z nápadů na možné rozšíření bylo i vložení celé složky z Drive. Aplikace by natáhla všechny soubory z této složky a díky tomu by mohly projekty být na Drives lépe rozdělené.

Závěr

V práci jsem probral dle mého nejdůležitější problémy a situace, na které se dá při tvorbě podobné aplikace narazit. Cílem této bakalářské práce bylo navrhnout a implementovat systém, který měl zefektivnit proces oběhu elektronického obsahu v digitální agentuře. Analýzou současného stavu ve společnosti byl identifikován problém vyskytující se při kolaboraci více osob při procesu schvalování a vyměňování připomínek. Pomocí analýzy existujících řešení byl zformován seznam požadavků na nový nástroj, který měl zmíněný problém řešit. Navržená aplikace byla implementována pomocí Google API a jazyka PHP s podporou frameworku Nette. Jako stavební kámen bylo použito úložiště Google Drive. Nakonec proběhlo pilotní nasazení v zadavatelské společnosti.

Zhodnocení přínosů do firemní praxe ukázalo mírné zefektivnění celého procesu týkajícího se oběhu a schvalování obsahu a byl vytvořen nástroj, který může firmu kvalitně reprezentovat při komunikaci s klienty. Cíle tedy byly splněny jak firemní, tak i osobní: naučil jsem základům práce s Google API a prošel si vývojem aplikace od jeho analýzy až po nasazení.

Literatura

- [1] Facebook Inc. [cit. 2015-05-01]. Dostupné z: <https://www.facebook.com/>
- [2] Google, Inc. [cit. 2015-05-01]. Dostupné z: <https://drive.google.com/>
- [3] Dropbox, Inc. [cit. 2015-05-01]. Dostupné z: <https://www.dropbox.com/>
- [4] Google Inc, Google Apps. [cit. 2015-04-26]. Dostupné z: <https://www.google.cz/intx/cs/work/apps/business/>
- [5] Teamwork.com, Ltd. [cit. 2015-05-01]. Dostupné z: <http://www.teamwork.com/>
- [6] Slack Technologies, Inc. [cit. 2015-05-01]. Dostupné z: <http://slack.com/>
- [7] Linuxová distribuce Debian. [cit. 2015-05-01]. Dostupné z: <https://www.debian.org/>
- [8] Apple Inc. [cit. 2015-05-01]. Dostupné z: <http://www.apple.com/>
- [9] Microsoft Corporation. [cit. 2015-05-01]. Dostupné z: <http://www.microsoft.com/cs-cz/default.aspx>
- [10] Letterfeed.com. [cit. 2015-03-13]. Dostupné z: <http://www.letterfeed.com/workflows/>
- [11] Aqility, Inc., ApproveForMe. [cit. 2015-03-20]. Dostupné z: <http://www.approveforme.com/>
- [12] Google, Inc. [cit. 2015-05-01]. Dostupné z: <https://developers.google.com/picker/>
- [13] Společný evropský referenční rámec pro jazyky: *Jak se učíme jazykům, jak je vyučujeme a jak v jazycích hodnotíme*. Olomouc: Univerzita Palackého, první vydání, 2002, ISBN 80-244-0404-4, 267 s.

- [14] Sparx Systems Pty Ltd. [cit. 2015-03-20]. Dostupné z: <http://www.sparxsystems.com.au/products/ea/>
- [15] UXPin Sp. z o.o. [cit. 2015-03-20]. Dostupné z: <http://www.uxpin.com/>
- [16] Sublime HQ Pty Ltd. [cit. 2015-03-20]. Dostupné z: <http://www.sublimetext.com/>
- [17] Google Inc, Prohlížeč Chrome. [cit. 2015-04-28]. Dostupné z: <https://www.google.cz/chrome/browser/desktop/>
- [18] VRÁNA, Jakub, Adminer. [cit. 2015-03-20]. Dostupné z: <http://www.adminer.org/>
- [19] appbsolute GmbH. [cit. 2015-03-20]. Dostupné z: <https://www.mamp.info/en/>
- [20] The Apache Software Foundation. [cit. 2015-04-28]. Dostupné z: <http://httpd.apache.org/>
- [21] Sass: Syntactically Awesome Style Sheets. [cit. 2015-05-02]. Dostupné z: <https://sass-lang.com/>
- [22] W3Schools. [cit. 2015-05-03]. Dostupné z: www.w3schools.com/ajax/
- [23] GRUNT, The JavaScript Task Runner. [cit. 2015-05-03]. Dostupné z: <http://gruntjs.com/>
- [24] W3tech.com, Usage of server-side programming languages for websites. [cit. 2015-05-01]. Dostupné z: http://w3techs.com/technologies/overview/programming_language/all
- [25] Nette Foundation. [cit. 2015-04-26]. Dostupné z: <http://nette.org/>
- [26] SENSIO LABS, Symfony2. [cit. 2015-03-20]. Dostupné z: <https://www.symfony.com/>
- [27] Taylor Otwell, Laravel. [cit. 2015-03-20]. Dostupné z: <http://www.laravel.com/>
- [28] Nette Foundation, Tracy: Ladící nástroj. [cit. 2015-04-26]. Dostupné z: <http://tracy.nette.org/>
- [29] Nette Foundation, Latte šablonovací systém. [cit. 2015-04-26]. Dostupné z: <http://doc.nette.org/cs/2.3/templating/>
- [30] Google Inc. [cit. 2015-04-28]. Dostupné z: <https://developers.google.com/console/>
- [31] OAuth 2.0. [cit. 2015-03-20]. Dostupné z: <http://oauth.net/2/>

- [32] Nette Foundation, Model-View-Presenter (MVP). [cit. 2015-04-20]. Dostupné z: <http://doc.nette.org/cs/0.9/model-view-presenter>

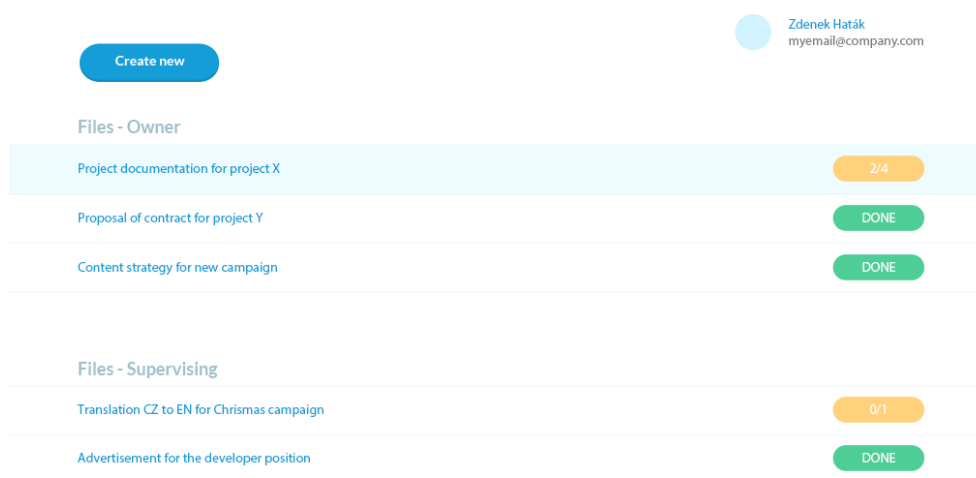
Průchod aplikací

V této kapitole provedu čtenáře celou aplikací ve formě obrazovek (screenshotů) různých scénářů, na které lze při práci s ní narazit. Jelikož je celá aplikace podle požadavku N.1 (2.3) v anglickém jazyce, budou tak uvedeny i ukázky. Předvedu testovací projekt, kde nejsou přítomna žádná reálná data. Slouží pouze k představení fungování aplikace.

A.1 Úvodní stránka

Tuto obrazovku (obrázek A.1) vidí uživatel z firmy ihned po svém přihlášení (uživatel mající roli *klient* se nepřihlašuje). Vidí zde přehledně zobrazené všechny procesy, které založil sám a ty, které vyžadují jeho dohled. U každého procesu je zároveň i jeho aktuální stav. V horní části obrazovky je tlačítko, který může uživatel založit nový proces.

A. PRŮCHOD APLIKACÍ



Obrázek A.1: Úvodní stránka aplikace

A.2 Založení procesu

Druhou ukázkou je samotné založení procesu. Na hlavní obrazovce (A.1) uživatel klikne na tlačítko *Create new*, jež ho přesune na obrazovku A.2. Zde může vyplnit jméno projektu, jeho popis a přidat k němu dohlížející z řad klientů a zaměstnanců. Jakmile přidá nějaký soubor, aplikace mu automaticky nastaví práva, jež umožní komukoliv s odkazem si tento soubor zobrazit (tento odkaz je poslán klientovi a tak je zaručeno, že klient nemusí vlastnit žádný účet, podle požadavku F.8 v sekci 2.3).

A.3. Práce s procesem z pohledu dohlížejícího

The screenshot shows a web interface for creating a process. At the top right, the user's profile is visible: Zdenek Haták, myemail@company.com. The main form includes a text input for the project name, currently containing "Test project to show this application". Below this is a blue button labeled "+ Add a file". A message for the recipient is shown in a text area: "Hi, can you take a look on this project, please? It deals with the new campaign we were discussing last week on the meeting." Below the message, there are two columns: "Clients" and "Company". The "Clients" column lists "Patrik Eliáš" and "David Kočí", each with "Send" and "Delete" buttons. The "Company" column lists "Lukáš" and "František", also with "Send" and "Delete" buttons. At the bottom of the form, there are two sections for adding new participants: "Add a new client" with fields for Name and E-mail, and "Add a colleague" with an E-mail field and radio buttons for "Attending" (selected) and "Owner". Both sections have a blue "+ Add" button. At the very bottom of the form is a blue "Create new" button.

Obrázek A.2: Stránka založení procesu

Všem přidáním osobám je následně doručen email s informací, že byly přidány do schvalovacího procesu společně s odkazem do aplikace. Zároveň se zapojené soubory na *Drive* označily, jak je vidět na obrázku A.3

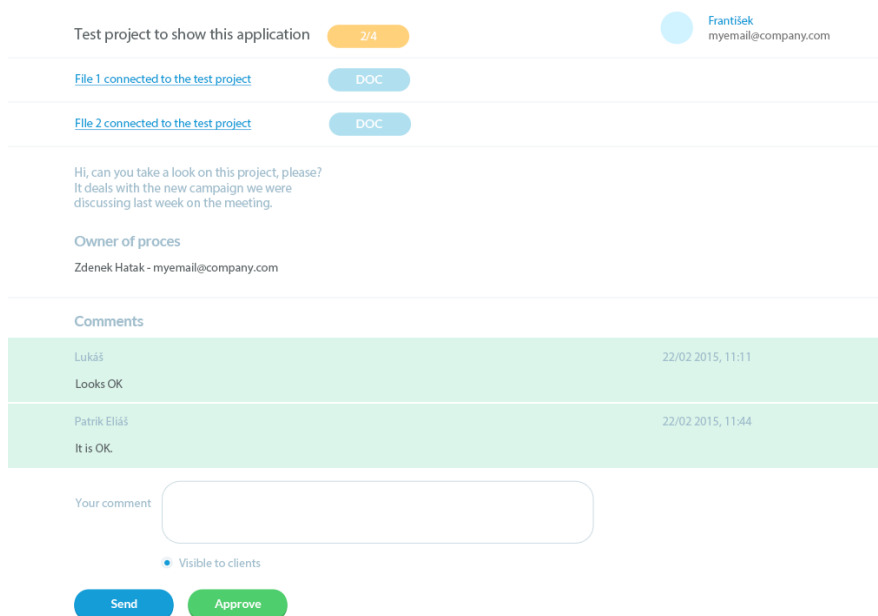
The screenshot shows a notification in the Drive interface. It consists of a blue square icon with a white list symbol on the left, followed by the text "▷ File 1 connected to the test project".

Obrázek A.3: Označení souboru na Drive

A.3 Práce s procesem z pohledu dohlížejícího

Jak jsem již zmínil v kapitole 3, dohlížejících existují dva typy. První je zaměstnanec firmy a druhý je klient. Na obrazovce A.4 má možnost zaměstnanec zobrazit soubory připojené k procesu v prostředí *Drive* pomocí zobrazených odkazů v horní části obrazovky. Dále může celý projekt okomentovat a v případě, že souhlasí s obsahem souborů ho může zeleným tlačítkem *Approve* schválit.

A. PRŮCHOD APLIKACÍ



Obrázek A.4: Detail procesu z pohledu dohlížejícího

Po úspěšném schválení všech zainteresovaných, se proces na hlavní stránce *vlastníka* obarví a na firemním *Drive* se souborům přidá symbol značící, že je vše hotovo A.5.



Obrázek A.5: Označení souboru na Drive - hotovo

Seznam použitých zkratk

- 3D** Trojdimenzionální zobrazení
- AJAX** Asynchronous JavaScript and XML
- API** Application Programming Interface
- CRUD** Create, Retrieve, Update, Delete
- CSS** Cascading Style Sheets
- DOM** Document Object Model
- DSN** Data Source Name
- HTML** HyperText Markup Language
- IDE** Integrated Development Environment
- NDA** Non-disclosure agreement – dohoda o mlčenlivosti
- OSVČ** Osoba samostatně výdělečně činná
- PHP** Hypertext Preprocessor
- SDK** Software Developer Kit
- UI** User Interface – uživatelské prostředí
- XML** Extensible Markup Language
- XSS** Cross-site scripting

Obsah přiloženého CD

readme.txt.....	stručný popis obsahu CD
src	
_ impl.....	zdrojové kódy implementace
_ app.....	backendová část
_ app-frontend.....	frontend část
_ bin	
_ vendor	
_ www	
_ Gruntfile.js	
_ thesis.....	zdrojová forma práce ve formátu \LaTeX
text.....	text práce
_ thesis.pdf.....	text práce ve formátu PDF