

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MIKROBLOG PRO TÝMY

DIPLOMOVÁ PRÁCE

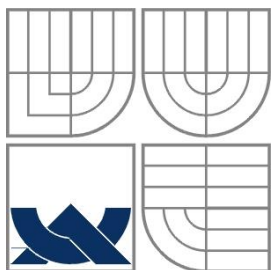
MASTER'S THESIS

AUTOR PRÁCE

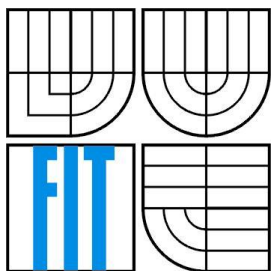
AUTHOR

Bc. Tomáš Krutiš

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

MIKROBLOG PRO TÝMY

TEAM-BASED MICROBLOGGING SERVICE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. Tomáš Krutiš

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. Rudolf Kajan

BRNO 2013

Abstrakt

Tato diplomová práce se zaměřuje na vytvoření mikrobloginovací služby pro týmy. Práce popisuje průzkum a analýzu současných mikrobloginovacích služeb a nalezení jejich silných a slabých stránek. Na základě analýzy je navržena nová služba, která se soustředí na komunikaci mezi členy týmu. Dále je rozebrána implementace navržené služby a na závěr jsou shrnuta budoucí rozšíření pro vylepšení služby.

Abstract

This master's thesis focuses on the creation of microblogging services for teams. The thesis describes the research and analysis of current microblogging services and finding their strength and weaknesses. Based on the analysis is designed new service which focuses on communication between team members. It is also discussed implementation of the proposed service and on the conclusion are summarized future extensions to improve this service.

Klíčová slova

Mikroblog, tým, spolupráce, sdílení informací, hashtag, Titter, Swabr, Twitter

Keywords

Microblogging service, team, collaboration, share information, hashtag, Titter, Swabr, Twitter

Citace

Krutiš Tomáš: Mikroblog pro týmy, diplomová práce, Brno, FIT VUT v Brně, 2013

Mikroblog pro týmy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Rudolfa Kajana. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Krutiš

22. 5. 2013

Poděkování

Chci poděkovat vedoucímu diplomové práce Ing. Rudolfovi Kajanovi za odbornou pomoc a rady.

© Tomáš Krutiš, 2013

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Mikroblog pro týmy	4
2.1 Co je to mikroblog	4
2.1.1 Zprávy.....	5
2.1.2 Hashtag	5
2.1.3 Označení uživatele ve zprávě	6
2.1.4 Diskuze mezi uživateli.....	7
2.1.5 Jednoduchost používání.....	7
2.1.6 Jednoúčelové vs. komplexní služby.....	8
2.1.7 Uložení souborů.....	8
2.1.8 Zkracovací služby	9
2.1.9 API.....	11
2.1.10 Mobilní a desktopový klienti	11
2.1.11 Bezpečnost.....	13
2.1.12 Export dat z mikroblogu	13
2.1.13 Vyhledávání	13
2.2 Existující mikroblogovací služby	14
2.2.1 Twitter.....	14
2.2.2 Swabr.....	17
2.2.3 Identi.ca	19
2.2.4 Yammer	20
3 Návrh aplikace	22
3.1 Uživatelé služby.....	22
3.2 Registrace nového člena týmu	23
3.3 Sítě na mikroblogu.....	24
3.4 Příspěvek na mikroblogu	25
3.5 Událost na mikroblogu	26
3.6 Průzkum na mikroblogu	29
3.7 Doplnování při psaní hashtagů a uživatelů	29
3.8 Odpovědi na příspěvky	31
3.9 Profil uživatele.....	32
3.10 Vyhledávání.....	32
3.11 Filtrování příspěvků.....	33

3.12	Logika aplikace a její části.....	34
3.12.1	Stránky aplikace.....	35
3.13	Schéma dat aplikace	36
4	Implementace.....	38
4.1	Použité technologie.....	38
4.1.1	ASP.NET	38
4.1.2	MS SQL Server.....	39
4.1.3	jQuery	39
4.2	Členství uživatele v mikrobloggeru.....	39
4.3	Schéma uložení dat a manipulace s daty	40
4.4	Aplikační logika služby	43
4.4.1	Příspěvky na mikrobloggeru	44
4.4.2	Odpovědi na příspěvky	47
4.4.3	Vyhledávání v příspěvcích.....	47
4.4.4	Filtry a filtrování příspěvků	48
5	Závěr a další pokračování projektu.....	50
5.1	Budoucí rozšíření projektu	50

1 Úvod

Cílem této práce je zmapovat současné služby, jež jsou zaměřeny na mikroblovování a daly by se využít pro komunikaci skupiny lidí se společným zájmem, tedy řekněme týmem. Může se jednat o spolupracovníky v jedné firmě nebo lidi pracující na společném projektu. Problémem bývá, že každý člen týmu pracuje na své části projektu a nemusí mít ponětí, co řeší ostatní. Právě mikroblov je ideální platforma, jak jim sdělit co právě řešíte, nad čím přemýšlíte a s čím máte třeba aktuálně problém. Často se totiž stává, že i lidé, kteří sedí ve stejné budově nebo kanceláři netuší, co ostatní řeší. Mohou tedy mít stejný problém nebo uvažovat o podobném řešení, ale nijak se spolu nedomluví.

Na mikroblov může každý napsat, co právě dělá nebo jaký má problém a ostatní členové týmu mu mohou poradit nebo o daném problému diskutovat. To je výhoda oproti emailům nebo přímé komunikaci přes různé komunikační nástroje jako je Skype nebo Jabber, protože u těchto médií se sdělení nedozví celý tým. Přitom by informace obsažené v těchto komunikacích mohly být pro zbytek týmu důležité a užitečné.

Podstatná je i rychlost komunikace. Vzhledem k tomu, že u mikroblovů je omezen počet znaků na zprávu, je nutné, aby autor shrnul svoji myšlenku na tuto omezenou délku. Takže ve zprávě není vata, která by ji zbytečně zvětšovala. I rychlost vytvoření takové zprávy je kratší než třeba psaní mailu nebo článku na blogu.

Cílem této práce je projít existující mikroblogy. Zjistit jak by jejich prostřednictvím mohla probíhat komunikace a sdílení dat v týmu. Analyzovat jejich přednosti a také jejich slabé stránky. Z této analýzy vybrat ty nejlepší vlastnosti těchto služeb a zahrnout je do nově vytvořené služby pro komunikaci členů v jednom týmu. Další část práce se zabývá návrhem této nové služby. Při návrhu se zaměříme na pokročilejší vyhledávání mezi příspěvky a možnost tyto příspěvky filtrovat podle různých pravidel. Po návrhu se budeme podrobněji věnovat implementaci navržené služby. Nakonec bude zhodnocení celé práce a definice dalšího pokračování vývoje služby a navržení potřebných rozšíření, které by uživatelé mohli využít v rámci komunikace přes mikroblov.

V této práci budu vycházet z projektu mého vedoucího Ing. Rudolfa Kajana. Projekt se jmenuje Titter a v současnosti funguje v rámci fakulty informačních technologií VUT v Brně. Jeho hlavní význam je v komunikaci pro některé zaměstnance fakulty. Titter je postaven na technologiích od firmy Microsoft. Jako webová platforma je použito ASP.NET a vše běží na databázi MSSQL. U těchto technologií zůstanu i při vývoji tohoto projektu. Navíc bude použit ještě Javascript a framework jQuery pro zlepšení ovládání aplikace na straně klienta.

2 Mikroblog pro týmy

V kapitole 2 se důkladně zaměříme na popis mikroblogu a jak se uplatňuje v dnešní komunikaci. Vysvětlíme si co to vlastně mikroblog je, na jakém principu funguje a co nám přináší. Podíváme se, jaké může být jeho uplatnění při komunikaci lidí, jež pracují ve stejné společnosti nebo se společným zájmem na projektech. Tato týmová komunikace nás bude zajímat nejvíce a uvidíme, jaké může přinést výhody nebo naopak nevýhody a dokonce hrozby. Dále se zaměříme na analýzu již existujících projektů, přes které je možné mikroblogovat v rámci týmu nebo při práci na společném díle. Popíšeme si, jak dané služby fungují a jaké jsou jejich výhody a nevýhody.

2.1 Co je to mikroblog

Mikroblog (webová definice) - zmenšená obdoba webového deníku (blogu), slouží k publikování textů omezené délky (např. do 160 znaků).

Mikroblog je nová forma komunikace, kdy uživatel může pomocí krátké zprávy vyjádřit svoji myšlenku. Dnes zažívá tento koncept komunikace velký nárůst zájmu. Pomocí krátké zprávy může pisatel sdělit zprávu okolí mnohem rychleji než přes dříve existující způsoby prezentace informace. Tato výhoda spočívá v rychlosti napsání zprávy o omezeném množství znaků, jež zabere kratší čas oproti článku, který má větší rozsah, třeba 2000 slov, ale obsahuje spoustu ne tolik důležitých informací. Velmi podstatné je to, že autor musí správně umět shrnout myšlenku zprávy do určitého prostoru např. 140 znaků.

Inspirací pro krátké zprávy byly zprávy SMS, posílané na mobilních telefonech. SMS má omezenou délku zprávy na 160 znaků. Toto omezení navrhl Friedham Hillebrand, který provedl zkoumání a analýzu zasílaných zpráv a došel k závěru, že 160 znaků stačí pro naprostou většinu posílaných SMS zpráv.

Jde především o to, aby se informace dostala k ostatním lidem co nejrychleji. To je ovšem například u klasického blogu nebo redakčního systému složitější. Autor musí nejprve napsat článek, který poté publikuje veřejnosti. Pokud máme zprávu, která by se měla dostat k ostatním co nejrychleji, pak je mikroblog ideální možností. Pro příklad uvedu několika hodinový výpadek vyhledávače Google v roce 2010. Než o něm informovala největší světová média (např. BBC), tak už bylo po výpadku a vyhledávač zase fungoval. Informace se však velmi rychle a téměř okamžitě rozšířila pomocí mikroblogu Twitter. Autoři také publikují zprávy daleko častěji, než kdyby psaly klasické příspěvky na blog.

Mikroblog funguje v podstatě tak, že uživatel přenesle svoje myšlenky, poznatky a aktuální problémy přes zprávu napsanou na mikroblog. Jde o to, že zpráva je určena pro nikoho a vlastně pro všechny. Co to vlastně pro autora znamená? Jeho zprávu si může na mikroblogu přečíst kdokoliv koho

zpráva nebo daný autor zajímá. Pokud máte na mikrobloku účet, tak si váš stream zpráv tvoříte přidáváním odběrů od jiných autorů na mikrobloku. Nebo si můžete vyhledávat zprávy s konkrétním obsahem zveřejněné jinými autory k tématu, které vás zajímá.

Autoři také publikují zprávy daleko častěji, než kdyby psaly klasické příspěvky na blog nebo na vlastní webovou stránku. Pak ale vzrůstá počet příspěvků, kterými se čtenáři musí prokousat. Proto je dobré, když mikrobloginovací služba obsahuje vhodný nástroj pro vyhledávání, filtrování a zobrazování příspěvků. Na tuhle část se chci především zaměřit při návrhu a implementaci nové mikrobloginovací služby. Při komunikaci v týmu je hodně důležité, aby si člen týmu mohl zobrazit jenom příspěvky, které ho aktuálně zajímají. Pokud například spolupracuje na určité části projektu s některými kolegy, tak by měl mít možnost si vybrat pouze tyto členy a následně si vypsát pouze jejich příspěvky. Tato funkce pomůže udržet v příspěvcích určitý pořádek. Další třídění příspěvků může probíhat na základě vyhledávání ve zprávách anebo podle hashtagů, které se mohou v příspěvku vyskytovat.

2.1.1 Zprávy

Uživatel mikrobloginu vkládají na mikroblogin zprávy (příspěvky), které mohou ostatní číst, reagovat na ně a sdílet je s dalšími uživateli. Zprávám se říká různě podle služby, kterou používáme. Na Twitteru se jim říká tweety na identi.ca jsou to denty, jinde to jsou posty, příspěvky nebo prostě zprávy. Zpráva může obsahovat text, odkazy na webové stránky, odkazy na uživatele a hashtagy.

Zprávy lze na mikroblogin umísťovat různými cestami. Záleží na konkrétní službě, které kanály pro čtení a přidávání poskytuje. Těmito kanály jsou webové rozhraní, které mají všechny služby, se kterými jsem se doposud setkal. Dále to může být email, SMS zpráva, XMPP protokol pro zasílání zpráv a zjištění stavu nebo API poskytované autorem dané aplikace.

2.1.2 Hashtag

Hashtagy byly poprvé použity při komunikaci přes IRC síť, kde označovaly skupiny a konkrétní témata na těchto sítích. Na mikrobloginích se s nimi také setkáváme. Hashtagy jsou klíčová slova nebo řetězce slov, která stručně popisují, o čem je příspěvek autora pojednává, jakým problémem se zabývá nebo o jakou informaci se chce autor podělit, dělí tedy zprávy do témat. Začíná znakem „#“ a potom následuje klíčové slovo nebo více slov, které však nesmí obsahovat mezeru, aby tento řetězec tvořil jedno slovo. Hashtag je například „#modulFinance“, to znamená, že příspěvek obsahuje informace o finančním modulu, na kterém autor zrovna pracuje. Jestli mě zajímají zprávy o finančním modulu, tak si je vyhledám právě podle tohoto hashtagu. Vzhledem k tomu, že zprávy na mikrobloginu mají omezenou délku, tak se do nich nevejdou všechny informace, které by ale autor chtěl v příspěvku mít. U zpráv také nejsou nadpisy, podle nichž se na webu vyhledává. K vyhledávání na mikrobloginu tedy pomáhají hashtagy. Pokud někdo na mikroblogin napíše zprávu k tématu, ke kterému se chcete vyjádřit a nechcete mu přímo odpovídat, tak stačí napsat stejný hashtag jaký použil autor. Pak pokud

by někoho zajímaly příspěvky na dané téma, tak bude stačit vyhledat zprávy s daným hashtagem a zobrazí se všechny příspěvky, jejichž autoři je označili stejným hashtagem.

Největší rozmach hashtagů nastal s příchodem sítě Twitter a jejího masového rozšíření. Prvním hashtagem na mikrobloginovací síti byl #barcamp a napsal ho Chris Messina, 23. srpna 2007 [1].



Obrázek 1 - První tweet, který obsahoval hashtag na Twitteru (#barcamp)

2.1.3 Označení uživatele ve zprávě

Ve zprávě, kterou autor umístí na mikrobloginovací síť může označit další uživatele, jimž je zpráva určena anebo se jich jiným způsobem týká. Takovéto označení se provádí pomocí znaku „zavináč - @“. Za zavináčem následuje nick, který patří uživateli, jehož chceme ve zprávě zmínit. Funguje to tak, že autor zprávy začne psát text zprávy a když chce vložit odkaz na jiného uživatele mikrobloginovací sítě, tak napíše zavináč a začne psát jméno. Pod textem zprávy se objeví nabídka, jež se mění podle toho, co autor píše postupně do jména, jak je vidět na obrázku níže. Po výběru z nabídky se jméno uživatele změní na nick, který má v rámci mikrobloginovací sítě. Uživatel si pak může vypsat všechny zprávy, v nichž byl označen a sledovat, tak příspěvky, které se ho nějakým způsobem týkají.



Obrázek 2 - vložení odkazu na uživatele do zprávy



Obrázek 3 - vytvořená zpráva s odkazem na uživatele (@Druidik) a hashtagem #Brno

2.1.4 Diskuze mezi uživateli

Pokud se zaměříme na komunikaci v týmu, tak je důležité, aby ostatní členové týmu mohli reagovat na zprávu od autora. Buď může být odpověď provedena tak, že se autor příspěvku, na který chceme odpovědět, označí v příspěvku přes odkaz na autora, jak je popsáno v předešlé kapitole. Nebo je k dispozici přímo pole pro odpověď. Tato odpověď se zařadí přímo k příspěvku, na který reaguje, takže bude celá konverzace k danému tématu udržena pohromadě. Když se bude někdo další zajímat o dané téma, najde vše na jednom místě a nemusí hledat odpovědi, které se nachází jinde.



Obrázek 4 - ukázka reakce uživatelů na zprávu (swabr.com)

2.1.5 Jednoduchost používání

U mikroblovacích služeb je hodně důležitá i jednoduchost používání. Uživatel musí být na daném mikroblovu registrován. Při registraci by po něm neměli být požadovány zbytečné informace, které pro kvalitní používání služby nemají žádný efekt. Uživatel by však měl mít možnost editace již registrovaného účtu, aby ho například lépe poznali jiní uživatelé, kteří chtějí sledovat příspěvky, jež jsou od tohoto autora umístěny na mikroblovu. Je dobré, když si uživatel může doplnit nebo změnit své údaje, jako jsou email, místo pobytu, ICQ, Skype a další údaje. Tyto údaje sice nejsou nezbytně nutné,

ale mohou posloužit pro zkvalitnění služby. Třeba při vyhledávání příspěvků můžeme zohlednit zeměpisnou polohu uživatele a poskytnout mu co nejpřesnější výsledky vyhledávání.

Je dobré uživateli vytvoření profilu, co nejvíce zjednodušit a zpříjemnit. Chtít po něm pouze email, přes nějž se bude přihlašovat, uživatelské jméno, pro zobrazení u vytvořených příspěvků a heslo pro přihlašování do mikrobloginovací služby. Například vložení avatara (obrázek identifikující uživatele) do služby, kterým budou označeny příspěvky, si může služba opatřit z jiného zdroje, kde už uživatel má svého avatara uloženého. Tímto zdrojem může být třeba <http://www.gravatar.com>. Je to i usnadnění pro uživatele, když totiž využije takovéto služby, nemusí si všude měnit svého avatara, ale stačí ho vyměnit na jediném místě a změna se projeví na všech webech, kde se avataři uživatelů načítají z daného zdroje. Gravatar používá třeba web StackOverflow.com, který je programátorům jistě dobře známý. Takovéto strategie se budeme držet i u tohoto projektu, kde se uživatelské avatary budou načítat právě z Gravatar.

2.1.6 Jednouúčelové vs. komplexní služby

Mikrobloginovací služby by se daly rozdělit na dvě skupiny. První skupinu tvoří služby, které plní pouze základní funkci mikrobloginu. Poskytují pouze základ, tedy sdílení informací pomocí krátkých příspěvků a neposkytují další podpůrné funkce. Pokud na takovou službu chceme například přidat video nebo obrázek, tak to nejde udělat přímo. Autor, jenž chce něco takového vložit na mikroblogin, musí tyto multimediální data umístit na jinou službu v rámci webu (YouTube, Flickr, Picasa, ...) a do příspěvku jenom vložit odkaz na tyto data. Se vkládáním odkazů je však menší problém. Délka zprávy je omezená a daný odkaz může zabrat celý příspěvek nebo jej dokonce může přesáhnout. Tento problém vyřešíme využitím tzv. zkracovací služby. Použití zkracování odkazů je popsáno níže.

Druhou skupinu tvoří služby, které se také řadí mezi mikrobloginovací služby. Uživatel na nich má ale k dispozici i další části. Zástupci komplexních služeb jsou například Facebook nebo Google+. Nachází se tady třeba chat pro komunikaci mezi konkrétními dvěma účastníky, možnost nahrát video nebo obrázek přímo na službu, vytvoření události, které se můžou zúčastnit další uživatelé, videokonference přímo přes web, vytvoření skupiny a mnoho dalšího.

V tomto projektu budeme vytvářet jednouúčelovou mikrobloginovací službu, ale v dalších budoucích rozšířeních mikrobloginu by se mohli objevit i některé části z komplexních služeb. Zde to spíše závisí na aktuálních trendech a tom, co si žádají uživatelé takovýchto webů.

2.1.7 Uložení souborů

Při práci v týmu budou uživatelé chtít čas od času vložit na mikroblogin datový soubor. Může se jednat o zdrojový kód, obrázky s vývojovými diagramy pro vytvářený produkt, instruktážní videa, PDF soubory s analýzami, textové soubory a další. Některé služby ukládají soubory přímo u sebe na svých

serverech, jiné zase odkazují uživatele na jiné služby, které slouží pro ukládání takovýchto souborů na internetu a jejich pozdější šíření po webu přes odkaz na tento soubor.

V této práci bych chtěl zvážit, jestli by nebylo možné některé soubory ukládat přímo na server, kde mikroblog poběží. Každý uživatel by měl přidělenou datovou kvótu, do které by se musela data zde uložená vejít. Pak by také byla omezena velikost vkládaného souboru, aby uživatelé nevkládali příliš objemné soubory a nedošlo tak k zaplnění paměti serveru. Časem by možná byla potřeba zvětšit datové úložiště. To hodně závisí na počtu uživatelů mikrobloggerů, kteří budou ukládat soubory přímo do příspěvků. Při zaplnění přidělené kvóty nebo při pokusu nahrát větší soubor než je limit, by byl uživatel odkázán na službu, kam je možné tento soubor bez problémů nahrát a do příspěvku by vložil pouze odkaz na web, kde je nahraný soubor k dispozici pro čtenáře příspěvků s přiloženým souborem.

Další možnost by bylo využití zatím neurčité služby pro ukládání souborů, která poskytuje API, přes který by šel soubor uložit a vrátil by odkaz na tento soubor. Vracený odkaz by se vložil do zprávy. Služba by se musela zvolit podle typu souboru vkládaného do zprávy. Pro vložení videa by se dalo použít API YouTube, pro obrázky Flickr nebo ImageShack.

2.1.8 Zkracovací služby

K jejich většímu rozšíření došlo právě z rozmachem mikrobloggerových služeb. Kvůli omezené délce zprávy je výhodné, někdy i nutné odkaz vkládaný do zprávy zkrátit. Původně se takovéto služby používaly na zkracování odkazů vkládaných do emailů, aby se jeden odkaz vešel na řádek monitoru, takže nesměl překročit 80 znaků, aby nedošlo k zalomení odkazu. Zkracovací služba (nebo také zkracovač URL) poskytuje nástroje, jak docílit zkrácení původního odkazu. Funguje to tak, že zkracovací služba vezme odkaz od uživatele, uloží si jej do databáze a vygeneruje k němu krátký jedinečný kód v rámci aplikace a připojí ho k URL domény určené pro zkrácené odkazy. Pro příklad uvedeme například zkrácení odkazu

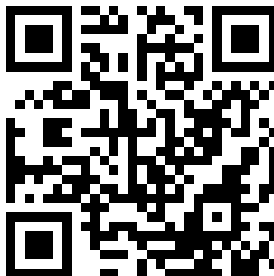
`http://msdn.microsoft.com/en-us/library/system.threading.barrier.aspx`.

Po zkrácení bude odkaz vypadat takto `http://goo.gl/aGG08`.

Pokud na zkracovací aplikaci přijde dotaz na tuto URL (`http://goo.gl/aGG08`), tak se podle záznamu v databázi vyhledá původní URL, na kterou se odkazuje zkrácená URL. Aplikace provede přesměrování na uloženou URL adresu, pomocí http zprávy s číslem 301. Dojde tak přesměrování na původní stránku. Uživatel tedy pracuje s odkazem stejně, jako by pracoval s původním a rozdíl nemusí v prohlížeči ani poznat, protože k přesměrování dochází okamžitě.

Aplikace pro zkracování URL poskytují také další doplňkové služby, které zpřístupní autorovi různé statistické informace o odkazu. Může poskytnout informace o počtu kliknutí na odkaz, kolikrát byla zkrácená URL adresa odkazována, z jakých operačních systémů a webových prohlížečů čtenáři přistupují k odkazu anebo z jakých zemí se čtenáři připojují. Další užitečnou funkcí je vytvoření QR kódu, v němž je zakódovaná zkrácená URL adresa. Tento kód lze pak použít i v reálném světě mimo

internet. Uživatel přes čtečku QR kódů v mobilním telefonu naskenuje fotoaparátem kód a aplikace jej převede na zkrácený odkaz. Některým autorům se tato funkce může hodit. U pár zkracovacích služeb se dá odkaz zabezpečit heslem a tak zamezit neoprávněnému přístupu k původnímu odkazu.



Obrázek 5- QR kód zkrácené URL (<http://goo.gl/gFtky>)

Zkracování odkazů má ovšem i své nevýhody. Pokud chceme přejít na konkrétní odkaz, tak nemáme jistotu, kam se skutečně dostaneme. Takže se může stát, že se dostane na stránku, na které se útočník snaží získat od nás citlivé informace (phishing). Vkládají do procesu přejítí na odkaz další články, který může být potenciálně nedostupný nebo nebezpečný. Je potřeba také vybrat službu, která je dostatečně spolehlivá a její dostupnost se blíží 100%. Problém je totiž v tom, že pokud je služba na zkracování odkazů nedostupná, tak se neprovede přesměrování a my se nedostaneme na stránku, na kterou nás chtěl autor odkázat. Tabulku s dostupnostmi jednotlivých služeb můžeme vidět na obrázku 6. [5] Dalším faktorem pro posouzení je také rychlost najetí odkazu v databázi a přesměrování. V současnosti mezi nejrychlejší služby patří `goo.gl` a `youtu.be`. [5] Zkracování může taktéž způsobit komplikace při SEO optimalizaci, která hodnotí web podle zpětných odkazů a zkrácená URL ji zastiňuje.[6]

K dispozici jsou uzavřené aplikace, tak i open source projekty. V současnosti mezi nejlepší aplikace jak podle rychlosti, tak podle dostupnosti patří `goo.gl`, `youtu.be` a `bit.ly`. Mezi open source projekty je dobrá služba `url.ca`. Důležitým faktorem je také API jednotlivých služeb, které se bude dát využít při vytváření tohoto projektu, pro automatické zkracování odkazů vkládaných do zpráv. Google poskytuje ke své službě slušné API [7]. Nebo je možnost místo celého odkazu vložit použít zástupné slovo za vložený odkaz, tak jak je to na Titteru implementováno teď.

#	Website	Uptime percentage	Downtime per month
1	BudURL	100.00	0 minutes
1	fb.me	100.00	0 minutes
1	goo.gl	100.00	0 minutes
4	amzn.to	99.99	4 minutes
4	binged.it (bitly.Pro)	99.99	4 minutes
4	bit.ly	99.99	4 minutes
4	is.gd	99.99	4 minutes
4	lnkd.in	99.99	4 minutes
4	ow.ly	99.99	4 minutes
10	j.mp	99.98	9 minutes
11	mcaf.ee	99.97	13 minutes
11	lnk.ms	99.97	13 minutes
11	youtu.be	99.97	13 minutes
14	y.ahoo.it	99.95	22 minutes
15	wp.me	99.94	27 minutes
16	fwd4.me	99.93	31 minutes
17	tinyurl.com	99.90	45 minutes
18	tcrn.ch (bitly.Pro)	99.88	54 minutes
19	su.pr	99.76	1 hour, 47 minutes
20	t.co	99.71	2 hours, 9 minutes
21	snurl.com	98.11	14 hours, 4 minutes
22	digg.com	37.71	19 days, 7 hours, 26 minutes
23	to.	0.00	31 days, 0 hours, 0 minutes
23	tr.im	0.00	31 days, 0 hours, 0 minutes
23	twurl.cc	0.00	31 days, 0 hours, 0 minutes

NOTE: The Twitter URL shortener t.co is blocked from China, which caused some downtime as measured from our monitoring stations (two consecutive errors from two different locations in China is regarded as downtime). If the errors from the China locations are excluded the uptime for t.co is 100%.

Obrázek 6 -statistika dostupnosti zkracovacích služeb (watchmouse.com)

2.1.9 API

Každá lepší mikrobloginovací služba by měla poskytovat programátorské API. Je to výhodné hlavně proto, že mohou vznikat produkty třetích stran právě pro konkrétní mikroblogin. API by mělo poskytovat možnosti pro manipulaci se zprávami, jako jsou přidání, odebrání, editace. Dále možnost přihlášení uživatele a způsob, jak upravit údaje v profilu přihlášeného uživatele. Vytvoření nového uživatele se určitě také může hodit. Velmi důležitou funkcionalitou je vyhledávání v příspěvcích uživatelů, takže API by mělo mít část právě pro toto vyhledávání. To jak podle textu obsaženého ve zprávě, podle zadaného hashtagu, tak i podle autora a uživatele zmíněného ve zprávě. Vyhledávání by mohlo umožnit kombinaci těchto variant vyhledávání.

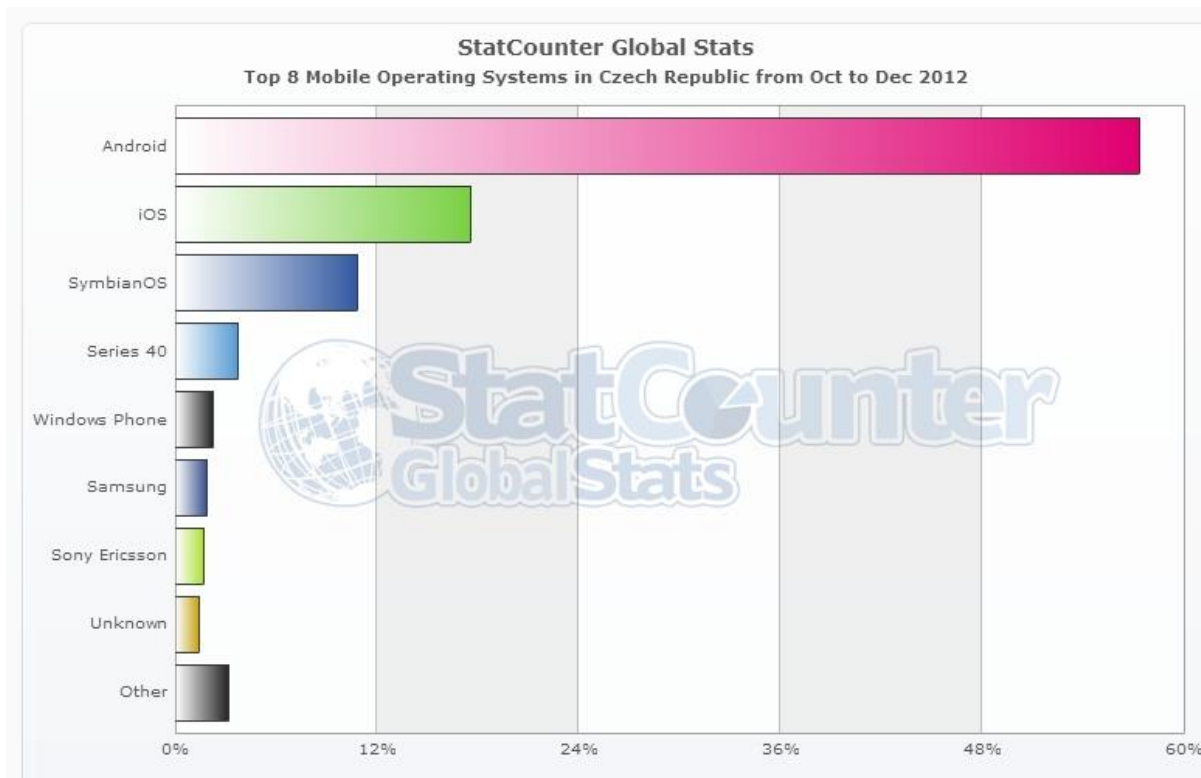
2.1.10 Mobilní a desktopový klienti

Pro úspěšný mikroblogin je podstatné, aby jej používalo co nejvíce lidí. Což je v dnešní době důležité vlastně pro jakýkoliv projekt na internetu. Více než polovina přístupů na mikrobloginovací

služby a sociální sítě se odehrává přes mobilní aplikace. [8] At' už se jedná o mobilní telefony, tablety a jiná zařízení. Proto je velmi důležité zohlednit ve vývoji mikrobloggerů vytvoření klienta pro tyto zařízení.

To je ovšem poněkud náročnější kvůli docela velké roztržitosti operačních systémů na takovýchto zařízeních. Systémy s nejvyšším zastoupením na současném trhu jsou Android od firmy Google, iOS od firmy Apple, Symbian od firmy Nokia (tento systém je na ústupu, Nokia jej nasazuje na mobilní telefony nejnižší třídy) a Windows Phone od firmy Microsoft. Problémem je také spousta verzí jednotlivých operačních systémů, zejména systému Android. [9] Vývoj klientů pro tolik operačních systémů, může být náročné jak na finance, tak na čas. Je důležité se zaměřit na nejrozšířenější platformy a pro ně vyvíjet. Pokud je dostupné API je možné v budoucnu klienty vytvořit bez zásahu do aplikace. API však musí poskytovat dostatečnou funkcionalitu. Samozřejmě, že se lze na mikroblogger dostat i přes webový prohlížeč v mobilu. Mobilní klient má však poskytovat jistou přidanou hodnotu a snadnější používání na menších displejích.

Někteří uživatelé používají desktopové klienty, kteří je informují o dění na mikrobloggeru. Nemusí se tedy neustále dívat přes webový prohlížeč, co se změnilo a kdo něco nového řeší anebo vyřešil. Přes klienta lze mikroblogger obsluhovat stejně jako přes webové rozhraní. Pokud je desktopový klient vytvořen v programovacím prostředí běžícím na více operačních systémech, například Java, nemusíme ani řešit různé systémy na počítačích uživatelů.



Obrázek 7 - používané mobilní operační systémy v ČR (StatCounter)

2.1.11 Bezpečnost

Pokud máme tým komunikující a sdílející informace přes mikroblog, musíme zvažovat i problém, že některé informace by se neměly dostat mimo tým. Bylo by tedy výhodné mít určitý systém, jak určit, kdo bude moci získat a přidávat informace a kdo taková práva mít nebude. Řešením by mohlo být vytvoření skupin, do kterých by se mohli členové dostávat například na pozvánky, a tak by se dalo zajistit, aby v dané skupině byli pouze lidé, jenž k tomu mají oprávnění. Každá skupina by měla správce nebo moderátora, který by mohl udělovat pozvánky do skupiny.

Pokud by nebyly bezpečnostní podmínky moc vysoké, mohli by pozvánky udělovat všichni uživatelé, kteří už ve skupině jsou. Pokud by informace mohly být veřejné, pak by se do skupiny mohli přidat sami všichni uživatelé, kteří mají o informace ve skupině zájem a chtějí zde také publikovat. Řekněme, že takovou veřejnou skupinou by mohla být například podpora pro vyvíjený produkt, kde by odpovídali vývojáři na dotazy ohledně produktu nebo v případě potíží.

2.1.12 Export dat z mikroblogu

Užitečná je možnost exportu informací zveřejněných na mikroblogu do souboru na lokálním disku a jeho procházení a archivace off-line. Můžeme si tak procházet informace z mikroblogu, i když nemáme přístup k internetu nebo je mikroblog momentálně nedostupný. Export může mít více forem. Data jsou exportována do XML souboru, kde jsou náležitě strukturována. Dále mohou být ve formě HTML dokumentu i se správnými CSS styly, takže po zobrazení budou zprávy vypadat stejně jako na webu.

Důležitý je také výběr zpráv, které export bude obsahovat. Zde by měl být určitý výběr pro uživatele, co by v exportu chtěl mít. Například všechny konverzace, do kterých se jakkoliv zapojil, všechny jím vytvořené zprávy nebo všechny zprávy v určité skupině.

2.1.13 Vyhledávání

V tomto projektu se máme především zaměřit na efektivní vyhledávání v příspěvcích. Vyhledávání může mít různé podoby. Základem je vyhledávání přímo v textu zprávy, tak že se najdou všechny zprávy, které obsahují zadaný řetězec. Můžeme také vyhledávat podle hashtagu v příspěvcích. Také můžeme vyhledávat podle autora zprávy nebo uživatele uvedeného v příspěvku. Všechny tyto možnosti vyhledávání se dají kombinovat, takže můžeme vyhledat všechny příspěvky obsahující určitý řetězec, od konkrétního autora a obsahující zadaný hashtag.

Dalšími vlastnostmi vyhledávání může být filtrování. Filtrovat příspěvky lze podle určitého časového období. Zadá se přímo interval (datum a čas) od kdy a do kdy se mají příspěvky vybírat. Pokud není vybrán interval, lze vybrat časový úsek od aktuálního času. Jako zobraz příspěvky za poslední hodinu, den, týden, měsíc, rok atd. Vyhledávané příspěvky je možné seřadit vzestupně nebo

sestupně podle času vložení na mikroblog. Možná by šlo filtrovat zprávy i podle toho jestli obsahují odkaz na web nebo ne.

Vyhledávání je důležitá část mikroblogu, protože pokud uživatel nenajde, co hledal, pak takovou službu nadále nebude používat. Uživatel by měl mít k dispozici vyhledávání a také pokročilé filtrování, aby vždy našel co nejkvalitnější výsledky na jeho vyhledávací dotaz.

2.2 Existující mikroblogovací služby

V této kapitole se podíváme na již existující mikroblogy. Popíšeme si, jakým způsobem fungují, jak v nich lze komunikovat v rámci týmu nebo skupiny. Jestli mají nějakou výhodu oproti ostatním službám. Zda poskytují vlastní mobilní nebo desktopové aplikace anebo programátorské API, pro využití mikroblogu i v jiných aplikacích. Z těchto služeb budeme brát inspiraci do této práce. Pokusíme se vybrat z každé služby ty lepší vlastnosti.

2.2.1 Twitter

Twitter je dnes nejpoužívanější a nejrozšířenější mikroblogovací služba. Byla založena v roce 2006 Jackem Dorsey a dnes sídlí v San Francisku v Kalifornii. Má kolem 150 zaměstnanců, zajišťujících jeho chod. Znakem mikroblogu je modrý ptáček z profilu. Tato služba umožňuje uživatelům psát a číst příspěvky, které jsou na Twitteru označovány jako „tweety“. Tweety jsou příspěvky, jejichž délka je omezena na 140 znaků. Ty se pak zobrazují na profilové hlavní stránce uživatele anebo na stránkách s příspěvky odběratelů (followers).



Obrázek 8 - logo Twitter

Každý uživatel je zároveň autor příspěvků a odběratel. Odběr příspěvků se provádí tak, že si vybereme osoby, které na Twitter umísťují zajímavé a přínosné informace. Po vybrání takového uživatele stačí kliknout na tlačítko „Sledovat / Follow“ a pak už se všechny příspěvky začnou objevovat v příspěvcích na naší hlavní stránce. Pokud nás už daný uživatel nezajímá, stačí přejít na jeho profil a zvolit „Přestat sledovat / Unfollow“. To je hlavní podstata fungování Twitteru. Ve streamu příspěvků se však můžeme začít ztrácet, pokud začneme sledovat příliš mnoho uživatelů.

Příspěvatel může omezit, komu se budou jeho příspěvky zobrazovat. A to buď určitému kruhu jeho odběratelů anebo jsou jeho příspěvky dostupné komukoliv, což je výchozí nastavení při založení účtu. Pokud bychom tedy chtěli sdílet informace v určitém týmu, museli bychom mít založený účet pouze k takovému účelu a do okruhu svých odběratelů pustit jenom členy týmu. Pokud tedy nechceme,

aby informace sdílené v týmu neviděli uživatelé, kterým to nechceme dovolit z libovolného důvodu. V nastavení si zvolíme „Protected tweets“ a po uložení se budou moci k odběru přihlásit pouze uživatelé, které potvrdí k odběru sám vlastník účtu. Pokud si takto nastavíme svůj účet, tak naše příspěvky uvidí pouze naši odběratelé, příspěvky se nebudou veřejně zobrazovat mezi tweety ostatních uživatelů, nepůjdou vyhledat přes webový vyhledávač (google.com, seznam.cz, ...) a nepůjdou retweetovat uživatelům, kteří nejsou naši odběratelé. Stejně jako i označení uživatelé v našich zprávách (@Pavel) je neuvidí, pokud nebudou našimi odběrateli [12] Jako nevýhodu zde vidím právě nemožnost vytvořit si skupiny, do kterých by se soustředili členové týmu a mohli mezi sebou sdílet data a nastavit si, jak chtějí mít skupinu chráněnou proti čtení neoprávněnými osobami.

Příspěvky lze přidávat na Twitter přes webové rozhraní, mobilní a desktopové klienty a také v některých zemích přes SMS zprávy. Příspěvky přes SMS zprávy jsou zpoplatněny standartní částkou za odeslání SMS zprávy u daného operátora.

Výhodou Twitteru je masové rozšíření po celém světě. V současnosti má Twitter 200 milionů aktivních uživatelů měsíčně. To jsou uživatelé používající tento mikroblog alespoň jednou měsíčně. Zhruba polovina z tohoto počtu, tedy asi 100 milionů uživatelů, jej používá každý den. Všech registrovaných uživatelů je pak přes 500 milionů. Denně přibývá dalších 222 tisíc nových účtů. Produkuje 340 milionů tweetů denně. Českých a slovenských uživatelů je zhruba kolem 150 tisíc. Od 6. srpna 2012 má Twitter i českou jazykovou mutaci, i když zatím v beta verzi. [11]

Na Twitteru má účet spousta celebrit (@BarackObama, @JimCarey), pro nás by však mohli být zajímavý spíše osobnosti a firmy z oblasti IT. Mezi zajímavé firmy na Twitteru bych zařadil @Dropbox, @KenticoCMS, @google ... Účty přímo pro konkrétní produkty @Windows, @IE, @firefox ... a dále významní lidé z oblasti IT @BillGates, @jansedlak, @ID_AA_Carmack, @timheuer (skvělý vývojář), ... Zajímavé je také sledovat vesmírné vozítko Curiosity (@MarsCuriosity), které samo posílá z Marsu tweety.

Vyhledávání na Twitteru je hodně důležité, pokud se chceme dozvědět něco o určitém tématu, musíme to na Twitteru vyhledat. Tweety přibývají velkou rychlostí, takže se jedná o velké objemy dat. Je potřeba mít rozumný vyhledávací nástroj pro hledání informací. Když chceme od někoho odebírat příspěvky, je dobré mít možnost ho vyhledat. Vyhledávat uživatele jde z vyhledávacího pole v horní části stránky. Jedná se o univerzální vyhledávací pole, jak pro vyhledávání uživatelů, tak i pro vyhledávání příspěvků. Uživatele můžeme najít podle adresáře emailů v mailové službě, kterou používáme (Gmail, Yahoo, Hotmail). Pokud uživatel, kterého hledáme, ještě nemá na Twitteru účet, můžeme mu poslat na email pozvánku, aby si jej vytvořil.

Při vyhledávání příspěvků vyhledáváme podle slova nebo slov obsažených přímo v textu zprávy, kterou si přejeme vyhledat. Do vyhledávacího pole se napíše pouze slovo a nechá se vyhledat. Dále se může vyhledat podle hashtagu v příspěvku. Při vyhledávání hashtagu je nutné zadat před slovo označující hashtag znak „#“. Pokud nám takovéto základní vyhledávání nestačí, dají se použít vyhledávací operátory. Jsou uvedeny níže v tabulce. S těmito operátory jsme schopni přesněji

definovat, co chceme vyhledat. Můžeme zde zohlednit i jiné atributy než jenom obsah příspěvku. Může to být čas vložení, od koho je příspěvek, pro koho je určen příspěvek, kde byl příspěvek pořízen (zeměpisná poloha), zda obsahuje webový odkaz anebo podle klienta ze kterého byl příspěvek poslán (webové prostředí, desktopový klient, mobilní klient). [14]

Operator	Finds tweets...
twitter search	containing both "twitter" and "search". This is the default operator.
"happy hour"	containing the exact phrase "happy hour".
love OR hate	containing either "love" or "hate" (or both).
beer -root	containing "beer" but not "root".
#haiku	containing the hashtag "haiku".
from:alexiskold	sent from person "alexiskold".
to:techcrunch	sent to person "techcrunch".
@mashable	referencing person "mashable".
"happy hour" near:"san francisco"	containing the exact phrase "happy hour" and sent near "san francisco".
near:NYC within:15mi	sent within 15 miles of "NYC".
superhero since:2010-12-27	containing "superhero" and sent since date "2010-12-27" (year-month-day).
ftw until:2010-12-27	containing "ftw" and sent up to date "2010-12-27".
movie -scary :)	containing "movie", but not "scary", and with a positive attitude.
flight :(	containing "flight" and with a negative attitude.
traffic ?	containing "traffic" and asking a question.
hilarious filter:links	containing "hilarious" and linking to URLs.
news source:twitterfeed	containing "news" and entered via TwitterFeed

Obrázek 9 - tabulka vyhledávacích operátorů na Twitter s popisem (twitter.com)

Twitter zpřístupňuje uživatelům i tzv. trendy. Je to jakýsi seznam slov a slovních spojení, které se aktuálně nejvíce na Twitteru používají. Jde z nich vyzorovat, o čem se uživatelé nejvíce baví a co za téma řeší. Trendy se určují celosvětově nebo podle lokality, kde se nachází uživatel, pro něhož jsou Trendy určovány. Záleží, zda uživatel povolí zjišťování polohy nebo má svoji pozici uloženou na profilu.

K příspěvkům na Twitteru se připojují atributy, které ho dále definují a umožňují lepší možnosti vyhledávání. Atributy jsou čas vložení příspěvku, může být uchována informace o poloze autora, jako

ve které části světa se nachází. Zda používá webové prostředí, některé z mobilních nebo desktopových klientů.

Pro Twitter je velká spousta různých klientů. Seznam některých klientů můžeme najít zde [15]. Právě velké množství klientů je výhodou Twitteru. Poskytuje skvělé vývojářské API a vývojáři tak mohou tvořit vlastní klienty. Dnes jsou dostupní klienti, jak pro všechny běžně používané desktopové operační systémy, tak pro všechny mobilní operační systémy.



Obrázek 10 - oficiální mobilní aplikace pro telefony s OS Android

2.2.2 Swabr

Swabr je mikrobloginovací služba pro podniky, organizace a vzdělávací instituce. Založena byla v Berlíně a spuštěna v prosinci 2010 a v roce 2011 byla uvedena veřejná betaverze. Swabr je stále v betaverzi. Aktuálně Swabr používá více než 1500 společností v 80 zemích světa.



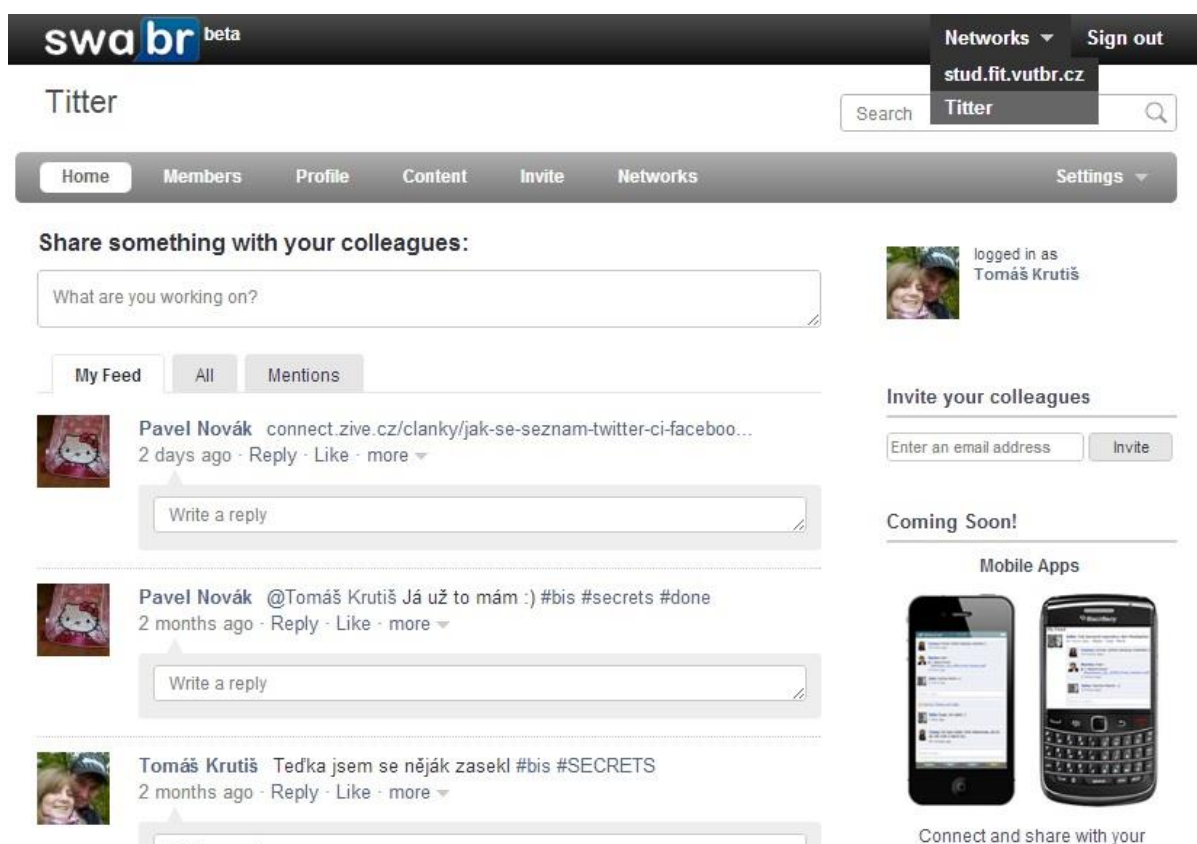
Obrázek 11 - logo německé mikrobloginovací služby

Centrálním prvkem Swabru jsou krátké zprávy, i když jsou delší než u Twitteru (3000 znaků). Uživatelé odpovídají na otázku, která se zobrazuje v poli pro vložení zprávy. Tato otázka je „Na čem pracuješ“ a uživatelé napíší co aktuálně v rámci projektu nebo týmu řeší. Takto zůstávají spolupracovníci a kolegové v kontaktu s minimálním úsilím. [17]

Na Swabru jsou uživatelé součástí sítě „Network“. Uživatel může být současně ve více sítích. Do skupiny se může uživatel dostat na pozvánku někoho, kdo už v dané síti je. Nebo na základě firemního, školního nebo jiného společného emailu, který má stejnou doménu emailu. Pokud tedy

v rámci určitého týmu mají všichni členové společné emaily, tak stačí při vytváření sítě zadat část emailu za zavináčem a Swabr pustí do sítě pouze členy s korektním emailem. Do sítě pak podle nastavení zve členy buď pouze administrátor sítě anebo všichni členové sítě. Koncept sítě se mi hodně líbí pro spolupráci členů týmu, něco na tomhle principu bych chtěl implementovat i do mikrobloggeru Titter.

Ke zprávě na Swabru lze připojit i obsah ve formě souboru, který je uložen přímo na serverech Swabru. Není tedy potřeba řešit ukládání souborů jinde. To je však hodně náročné na diskovou kapacitu serveru, kde služba běží. Vytvářené zprávy můžeme řadit do témat pomocí hashtagů, podobně jako tomu je u Twitteru. [16]



Obrázek 12 - rozložení stránky mikrobloggeru Swabr

Na obrázku výše vidíme webové rozhraní Swabru. V horní černé liště vpravo je nabídka pro přepínání mezi sítěmi, ve kterých je uživatel zaregistrován. Takže se může mezi těmito sítěmi přepínat. To je hodně dobrá funkce, pokud pracujeme na více projektech nebo ve více firmách. Prostě pokud jsme členy více sítí, tak se jednoduše mezi nimi přepínáme a můžeme zjistit, co se kde změnilo. V šedé liště níže vidíme na prvním místě tlačítko „Home“, tady se zobrazují všechny zprávy zveřejněné v dané síti. Na zprávy může uživatel reagovat pod nimi v políčku „Write a reply“. Druhé tlačítko je „Members“. Obsahuje všechny členy sítě. Můžeme si vybrat konkrétního člena sítě a zobrazit si jeho zprávy, které umístil na mikroblogger. Třetí tlačítko je „Profile“. Tady si uživatel může prohlédnout své

příspěvky, které na mikroblog napsal. Je to jeho kompletní historie zpráv. Také zde může upravit informace o svém profilu. Čtvrtou položkou menu je „Content“. Po přejítí na tuto nabídku se zobrazí veškeré dokumenty a soubory, které daný uživatel nahrál ke svým zprávám. Může je zde procházet, případně odstraňovat a nemusí je složitě hledat ve všech svých příspěvcích, kde daný soubor vložil. Předposlední tlačítko „Invite“ je nabídka pro zobrazení členů, které uživatel pozval do sítě, ale dané pozvání ještě nepřijali a nabízí i formulář pro vyplnění emailů pro pozvání nových členů. Nově pozvaným přijde email, kde bude odkaz, přes něhož se budou moci přidat do sítě, kam byli pozváni. Nakonec posledním tlačítkem „Networks“ si můžeme zobrazit všechny sítě, ve kterých jsme členy. Vidíme, kolik má daná síť členů a kolik v ní bylo publikováno zpráv. Pokud jsme administrátorem některé sítě, máme možnost upravit vlastnosti sítě.

Vyhledávání na Swabru je pouze základní. Vyhledání podle slov ve zprávě a podle hashtagu. Po vyhledání zprávy je u každé užitečný odkaz „View thread“. Tím se dá zobrazit celé vlákno konverzace, i když jsme z ní našli pouze jednu zprávu.

Swabr neposkytuje žádné API pro vývojáře. Nepodařilo se mi žádné najít. To považuji za nevýhodu. Swabr také připravuje mobilního klienta, který by měl být brzy k dispozici, jak píšou na svých stránkách. Ale dnes (listopad 2012), není žádný klient vydaný a funkční. Swabr oproti Twitteru neposkytuje češtinu v nastavení prostředí mikroblogu.

2.2.3 Identi.ca

Identi.ca je sociální síť a mikroblogovací služba postavená na otevřeném a svobodném softwaru StatusNet implementující specifikaci OStatus v PHP [18]. Do provozu byla uvedena v roce 2008. Uživatelé mohou publikovat zprávy dlouhé 140 znaků. Oproti ostatním službám nabízí navíc integraci protokolu XMPP pro zasílání soukromých zpráv a sdílení stavu. Identi.ca provozuje společnost E14N. Pro neplaticí uživatele je jedinou podmínkou, že musí zveřejňovat své příspěvky na mikroblog pod licencí Creative Commons 3.0. Tato licence umožňuje šíření a modifikování takového díla s uvedením jména původního autora. Kopie lze prodávat. Pokud by některá organizace nechtěla své příspěvky takto publikovat, může využít placenou verzi s vlastní licencí.



Obrázek 13 – logo mikroblogu Identi.ca

Identi.ca nemá rozdělení na sítě jako Yammer a Swabr, ale rozděluje členy do skupin. Funguje to tedy zhruba stejně jako se sítěmi. Můžeme posílat i soukromé zprávy mezi uživateli formou

soukromých zpráv. Jednotlivé příspěvky si můžeme přidávat do oblíbených a později si je procházet v sekci „Oblíbené“. Kromě psaní zpráv můžeme klást otázky, vytvářet události a dělat ankety. Dále se dají vytvářet „Záložky“ což jsou odkazy na web, k nim připojit popis a tagy. Jestli je veřejný nebo soukromý pro určitou skupinu.

Pokud vložíme soubor na Identi.ca, tak se uloží přímo na serveru služby. Nevkládá se tak nespolehlivý článek do ukládání souboru. Pokud totiž nefunguje služba, na kterou ukládáme soubory, pak je soubor nedostupný. Dají se také zobrazit celé konverzace, i když máme z této konverzace pouze jeden příspěvek. Automaticky zkracuje vložené odkazy. Zkracovací službu si můžeme sami nastavit.

Přestože má Identi.ca i českou jazykovou mutaci není překlad příliš dobrý. Některé výrazy nejsou přeloženy vůbec a některé překlady jsou nesrozumitelné.

2.2.4 Yammer

Yammer je podniková sociální služba spuštěna v roce 2008 a v roce 2012 byla prodána firmě Microsoft. Její tvůrci jsou David O. Sacks a Adam Pisoni. Yammer, kteří tuto službu používají pro privátní komunikaci a sdílení informací v organizaci nebo vytvořených skupinách. Přístup do služby Yammer je určený doménou emailu uživatele. Takže pouze uživatelé s náležitým emailem se mohou připojit k příslušné síti.



Obrázek 14 – logo služby Yammer

Yammer mi v podstatě připomíná mikroblog Swabr. Je však více propracovaný ve směru sdílení informací mezi členy týmu. Tuto službu dnes používají přední světové firmy jako Shell, UNICEF, Ford, DHL, ... [19]

Neposkytuje pouze psaní zpráv jako ostatní mikroblogy, ale také jiné formy sdílení dat. Přidávat soubory a obrázky, vytvářet ankety, ocenit zde některého člena sítě, zeptat se na otázku a adresovat ji konkrétním členům sítě a přidat událost, ke které přidáme, o jakou událost se jedná, kde se koná, datum a čas konání a trvání této události. Můžeme vytvořit poznámku, do které můžou ostatní členové připisovat a modifikovat informace. Můžeme ji také odeslat emailem. Tyto další možnosti sdílení informací se mi líbí a mohly by se do mikrobloggeru Titter dostat jako budoucí rozšíření.

Uživatelé mají k dispozici chat s ostatními členy týmu, kteří jsou právě k Yammeru připojeni. Mohou si tedy psát důvěrně pouze mezi sebou, aniž by konverzaci viděli ostatní členové, kteří v konverzaci nejsou. Můžeme si nastavit také zasílání upozornění na aktivitu uživatelů v dané síti nebo skupině. Upozornění chodí z emailu, který je automaticky zřízen pro potřeby sítě. Email má tvar `network@yammer.com`, kde „network“ před zavináčem je jméno dané sítě.

V rámci sítě můžeme vytvářet skupiny, které jsou určitým směrem více zaměřeny. Můžeme mít skupiny pro různé projekty, témata nebo menší týmy v dané síti. Pak přidáváme informace pouze do konkrétní skupiny, takže tím nezatěžujeme členy, kteří s danou oblastí nechtějí nic mít. Skupina může být buď otevřená, takže se do ní může přidat kdokoli z dané sítě anebo uzavřená, což znamená, že je nutná pozvánka pro vstup do skupiny. Podobně to funguje i při přidávání členů do sítí. Jsou taky potřeba pozvánky, aby se do sítě nedostali neoprávnění uživatelé.

Yammer poskytuje vývojářské API pro vytváření aplikací používající tuto službu. Yammer však už přímo na svých stránkách poskytuje velké množství mobilních i desktopových aplikací, které jsou dostupné pro nejrozšířenější mobilní a desktopové operační systémy. Některé aplikace jsou k dispozici zdarma. Aplikace pro používání Yammeru a vývojářské API jsou velkou výhodou.

Nevýhoda je, že se nedá nastavit jiný jazyk prostředí než angličtina, i když v podnikové sféře by to tolik vadit nemuselo.

3 Návrh aplikace

V této části práce se budeme zabývat návrhem aplikace mikrobloggeru pro týmy. Budu vycházet z již vytvořeného projektu Titter mého vedoucího diplomové práce Ing. Rudolfa Kajana. Pokusíme si zde ujasnit, co by mělo být součástí implementace mikrobloggeru a které nové prvky a funkce by se daly zahrnout do služby, aby se stala příjemnější, pohodlnější a užitečnější pro používání členy týmu. Zkusíme v této části navrhnout, jak by principiálně mělo vypadat uživatelské rozhraní pro jednotlivé části webové aplikace a jaké třídy budou tyto části obsluhovat z pohledu aplikační logiky. Nastíníme si také schéma databáze pro ukládání dat aplikace, jelikož se jedná o jednu z nejpodstatnějších částí projektu. Pokud by byla databáze nevhodně navržena, mohlo by to značně znepríjemnit jak přidávání další budoucí funkcionality do projektu, tak i implementaci navržených částí.

3.1 Uživatelé služby

Pokud budeme řešit používání mikrobloggeru v rámci týmu, můžeme si vypůjčit systém sítí tak, jak je ve službě Swabr nebo Yammer. Registrovaný uživatel, pak má ve službě více rolí. Může být vlastník sítě, což mu umožňuje vytvářet nové sítě a dané sítě také rušit. Dále je autorem příspěvků na mikrobloggeru. Může vytvářet všechny typy příspěvků, které jsou zatím na mikrobloggeru plánovány. Jedná se o příspěvky (post), odpovědi na příspěvky (post answer), události (event) a průzkumy (poll). Jednotlivé prvky, které může uživatel vytvořit, budou popsány dále v této práci. Samozřejmostí je možnost mazání a u vlastních událostí i možnost částečné editace. Dále v těchto příspěvcích můžeme označovat uživatele, kteří jsou s nimi nějak spjatí („+user“).

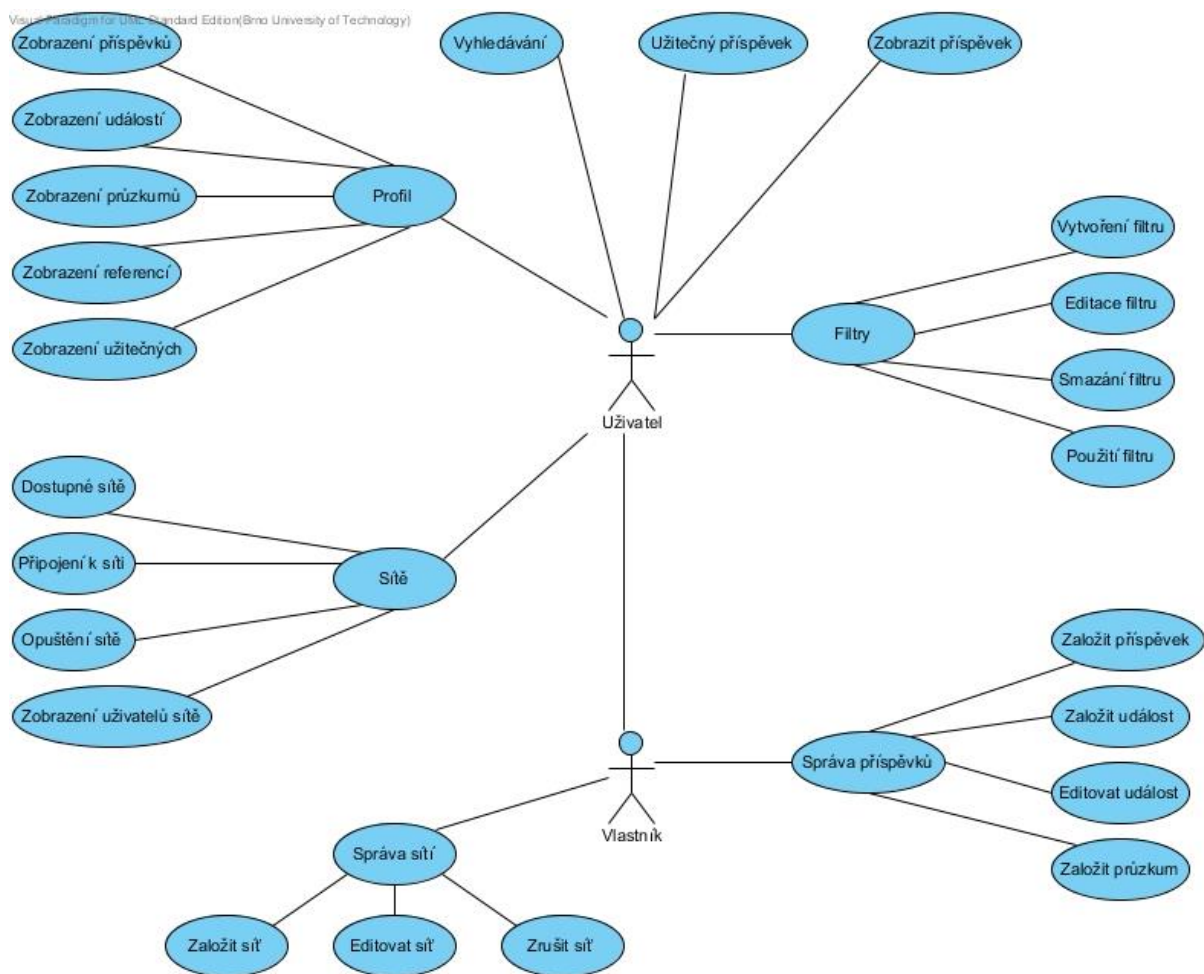
Uživatel vystupuje také v roli čtenáře příspěvků na mikrobloggeru. K dispozici mu bude čtení všech příspěvků ze sítě, ve kterých je členem. Tyto příspěvky si může prohlížet, reagovat na ně odpovědí, která se k danému příspěvku zaznamená a také si může příspěvek, který ho zaujal označit jako „Useful post“, tedy jako užitečný příspěvek. Označené příspěvky si lze později zobrazit, což mu usnadní hledání zajímavých příspěvků po delší době. Může také posloužit autorovi příspěvku, jako zpětná vazba od ostatních členů týmu, že se jim příspěvek zamlouvá a líbí.

U událostí na mikrobloggeru je možné se označit, zda se události zúčastníme či nikoli. U průzkumu pak vybírá jednu z odpovědí, která mu je autorem nabízena a potom si zobrazí výsledek hlasování ostatními členy, kteří již hlasovali. Důležitou funkcí je vyhledávání v příspěvcích podle obsahu definovaného uživatelem, podle toho co konkrétně hledá a dále vytváření filtrů, které mu umožní na základě specifikace filtru zobrazit pouze ty příspěvky, které dané specifikaci odpovídají. Vytváření specifikace filtrů bude popsáno dále v této kapitole.

3.2 Registrace nového člena týmu

Každý nový uživatel se zaregistruje a po splnění všech náležitostí pro registraci, bude mít vytvořený profil, ale bude členem jenom jedné hlavní veřejné skupiny, v níž jsou všichni uživatelé mikroblogu. Potom si může vybrat, jestli vytvoří novou síť anebo se přidá jako člen do již existující sítě. Měl by tedy být dostupný seznam sítí, do kterých se může uživatel přidat. Uživatel si vybere síť, jejímž členem by se chtěl stát a do této sítě se přidá. Dále si může prohlížet příspěvky všech členů v této síti a také vyhledávat a filtrovat obsah dané sítě.

Pro registraci je nutné, aby si uživatel zvolil uživatelské jméno, zadal svůj email a určil si heslo, které musí být delší než 6 znaků, aby byla dodržena alespoň základní bezpečnost uživatelského účtu. Toto by stačilo pro vytvoření uživatelského účtu. Profilový obrázek neboli avatar by se načítal z externí služby Gravatar, která poskytuje podle zadaného emailu avatara pro uživatele. Jedná se o mezinárodně uznávanou službu používanou na mnoha webech (StackOverflow, ...). Novému členovi by pak stačilo mít založený účet na Gravatar.com s vloženým obrázkem, jenž chce použít jako profilový a na náš mikroblog by se tento obrázek automaticky nahrál. Pokud by chtěl uživatel změnit profilový obrázek, stačilo by jej změnit centrálně. U obrázků na službě Gravatar je definován parametr, který označuje, zda daný avatar neobsahuje nevhodný obsah a je tak rozdělen do několika skupin. V tomto projektu bude nejlepší použít skupinu G, která značí nezávadné obrázky, takže by s nimi žádný uživatel mikroblogu neměl mít problém.



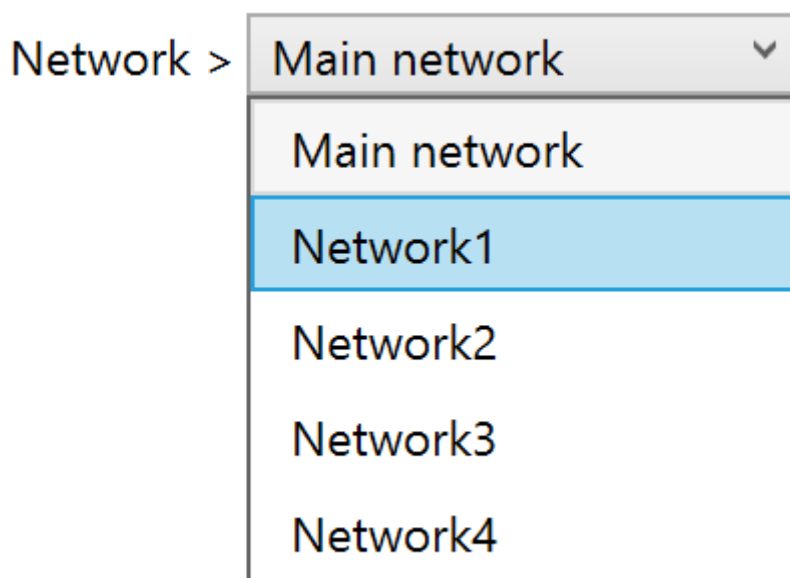
Obrázek 15 - use case diagram uživatele mikrobloggeru

3.3 Síť na mikrobloggeru

Jestliže chceme pracovat v rámci určitého týmu, bude výhodné tuto skupinu členů umístit do skupiny, kde budou všichni pohromadě. Těmto skupinám budeme říkat síť. Uživatel bude standardně po vytvoření účtu členem jedné hlavní sítě (např. Main network), kde by byli všichni uživatelé mikrobloggeru. Pokud by v dané síti nechtěl být, mohl by ji samozřejmě opustit, ale pro začátek a pro správné pochopení fungování mikrobloggeru je lepší ho do této hlavní skupiny zahrnout. Uživatel si bude moci zobrazit seznam sítí, do kterých se na daném mikrobloggeru může přidat. Každá síť by byla identifikována svým názvem, podrobnějším popisem a uživatelským jménem člena, který danou síť vytvořil a vlastní ji.

Každý uživatel má možnost být současně členem více sítí. V menu aplikace by se tedy měla objevit nabídka pro zvolení sítě, do které se může přepnout. Po přepnutí do jiné sítě by se v záhlaví stránky zobrazil název dané sítě, aby uživatel vždy věděl, ve které síti se aktuálně nachází a do které přispívá, aby nedošlo k omylu a nenapsal nesprávné informace do jiné sítě. Na stránce by se pak

zobrazily všechny příspěvky z dané sítě seřazené tak, že jako první bude zobrazen nejnovější příspěvek a postupně by se řadily příspěvky starší. Příspěvky, které by pak uživatel na této stránce napsal, by tak patřily do aktuálně vybrané sítě, do které by se ihned zobrazovaly.



Obrázek 16 - koncept přepínání mezi sítěmi

Uživatel si bude moci vytvořit svou vlastní síť, pokud žádná existující síť nebude splňovat jeho potřeby a požadavky. Při vytváření bude muset zadat jméno sítě a podrobnější popis, k čemu se nová síť váže nebo co se v ní řeší. Aby její noví potenciální členové měli představu, zda právě v této síti najedou informace, které by pro ně byly přínosné a užitečné. Nová síť by se automaticky objevila mezi ostatními dostupnými sítěmi a uživatelé by do ní mohli hned začít vstupovat. Zájemci o členství by stačilo v seznamu u dané sítě stisknout tlačítko pro přidání. Přidaná síť by se objevila v seznamu sítí, kde je členem a ihned by si mohl začít procházet příspěvky z nové sítě.

3.4 Příspěvek na mikroblogu

Přidávání příspěvků na mikroblog bych řešil stejně, jak je tomu u původní verze projektu Titter. Tedy uživatel má k dispozici pole pro napsání zprávy. Po napsání zprávy se zpráva pod daným polem zobrazí ve streamu příspěvků ostatních uživatelů. Pokud jsme v rámci některé vytvořené sítě, tak se příspěvek objeví pouze ve streamu této sítě. Příspěvek se přiřadí uživateli, který jej vytvořil a u příspěvku se objeví jeho jméno a profilová fotka pokud ji má k profilu přiřazenou (Gravatar). Dále se zaznamená datum přidání zprávy, které se zobrazuje pod textem příspěvku a to ve formě, jak dlouhý

časový interval uběhl od přidání. Tedy například „2 hours ago“. Příspěvek si může každý uživatel označit jako přínosný. Později si bude moci u svého profilu zobrazit seznam jeho přínosných příspěvků. K označení dojde po stisknutí tlačítka „Useful“ na pravé straně příspěvku. Pokud je aktuálně přihlášený uživatel zároveň autorem konkrétního příspěvku, tak se mu navíc zobrazí nabídka pro smazání příspěvku. Smazáním se odstraní i všechny označení „Useful“, všechny hashtagy a odkazy uložené společně s příspěvkem.

Napsané zprávy mohou obsahovat označení jiného uživatele, které je uvozeno znakem plus a hashtag, který je uvozený znakem „#“ a kterým lze příspěvek přidělit k určitým tématům nebo skupinám zájmu. Všechny příspěvky se budou ukládat do databáze s vazbou jak na autora, tak na síť, do které patří. Dále se s příspěvkem ukládají hashtagy, označení ostatních uživatelů v textu příspěvku a odkazy na web. Odkazy se uloží do databáze a místo odkazu se do příspěvku vloží náhradní text „link“, který je odkazem na původně napsanou webovou stránku.

UserName posted minute ago

Useful | [Remove]



Post content

Marked as useful by UserX UserY UserZ

Obrázek 17 - náčrt rozložení prvků u příspěvku

Počet příspěvků na mikrobloku není nijak omezen. Zatím jsem neomezoval ani počet znaků na jeden příspěvek. Příspěvek je obecný prvek, který se dá na mikrobloku publikovat. Dále budeme používat i dva prvky, které jsou specializací příspěvku. Jedná se o prvky událost a průzkum. Jejich popis najdete níže.

3.5 Událost na mikrobloku

Událost je rozšířením příspěvku, který je popsán výše. Má všechny atributy stejné jako příspěvek, tedy autora, datum vytvoření, text příspěvku, odkaz na síť, do níž je zařazen a kolekci uživatelů, kteří si jej označili jako „useful“. Navíc se přidávají vlastnosti zaznamenávající datum a čas začátku a konce události. Tyto vlastnosti musí vytvářející uživatel povinně zadat, bez nich by měla událost stejný informační přínos, jako příspěvek. Také se musí kontrolovat, jestli datum a čas začátku události předchází data a času konce události. Pokud je konec před začátkem, uživateli by se mělo zobrazit chybové hlášení, po kterém si zvolí jiné termíny události, aby byla daná pravidla splněna.

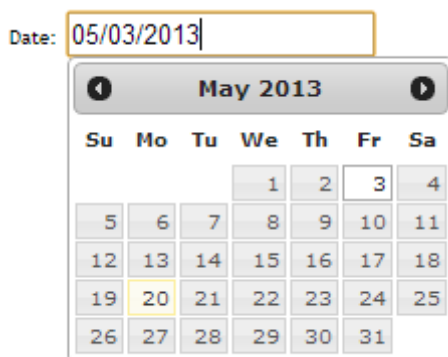
K události navíc přibývá i možnost zaznamenat místo konání události, popis dané události a odkaz na web, kde mohou být uvedeny bližší informace. Tyto tři další vlastnosti jsou pro vytvoření

události nepovinné. Takže pokud uživatel nechce, nemusí je vždy zadávat, na rozdíl od termínů události. Ty musí být správně vyplněné vždy.

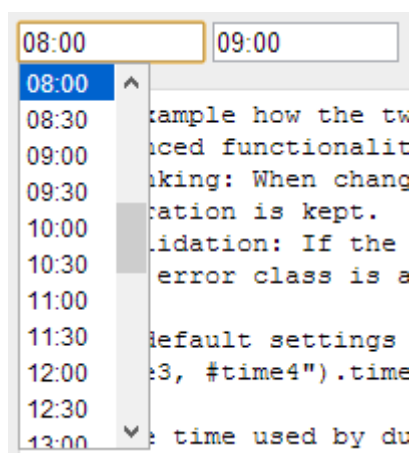
Při návrhu se objevil požadavek na možnost editace termínu u již vytvořené události. Tato funkcionality bude také součástí mikrobloggeru. Vlastník dané události bude mít možnost v případě potřeby upravit termín konání události. Nové termíny však opět musí splňovat pravidlo, že začátek události musí předcházet konci události. Pokud není podmínka splněna, není možné editaci události uložit, a je vysláno hlášení informující uživatele v čem nastal problém. U editované události se upraví datum vytvoření příspěvku na aktuální datum editace. To je proto, aby se editovaná událost dostala opět na přední pozici ve streamu dané sítě a ostatní členové sítě měli možnost zjistit, že se termín události změnil.

Při vytváření události budou vstupní políčka uživateli poskytovat nápovědu pro zadání správného formátu začátku a konce události. Políčka pro datum zobrazí kalendář, ve kterém si uživatel vybere datum pro začátek a pro konec události. V tomto vyskakovacím kalendáři uživatel nebude moci jít zpět za aktuální datum. Asi by to nebylo moc k prospěchu plánovat události do minulosti. Ve vstupních polích pro data bude standardně předvyplněné aktuální dnešní datum. Samozřejmostí je i možnost zapsat požadované datum ručně. Jediný požadavek by byl, aby datum bylo v evropském formátu zápisu, tedy dd.mm.yyyy. První dva znaky označují den v měsíci, další dva znaky identifikují měsíc a poslední čtyři znaky značí rok. Měsíc by se zadával číselně ne slovně. Jako oddělovač se použije znak „tečka“. Aplikace by se pak měla vyrovnat i s jinými oddělovači např. lomítko, ale standardně se použije tečka. Princip výběru data je možné vidět na obrázku níže.

Možnost výběru bude i u zadávání času pro začátek a konec události. Uživateli se po kliknutí na dané políčko rozbalí seznam časů, které obsahují časy během 24 hodin od půlnoci do půlnoci a je mezi nimi rozestup půl hodiny. Čas se může zadat také manuálně. Pro správné fungování je definován čas ve formátu hh:mm. Čas události se rozlišuje na minuty, menší dělení zde není potřeba. Oddělovačem hodin od minut je definována dvojtečka.



Obrázek 18- ukázka principu výběru data pro událost



Obrázek 19 - ukázka principu výběru času pro událost

Vytvořená událost se objeví všem uživatelům dané sítě ve streamu a ti u ní mohou vybrat jednu z možností, která určuje jejich účast na dané události. Nabízí se tři možnosti, pro které se mohou členové sítě rozhodnout. Mohou zvolit „Ano“ pokud se zúčastní, „Ne“ pokud se nezúčastní anebo „Možná“ když ještě nejsou plně rozhodnutí. Každý člen může potvrdit účast pouze jednou a může zvolit pouze jednu z třech odpovědí. Po vyjádření účasti se zobrazí statistika, kolik členů pro kterou možnost hlasovalo a bude zde možnost si tyto členy zobrazit, aby se dalo podívat, kdo se zúčastní, kdo možná a kdo určitě nepřijde.

U události lze samozřejmě stejně jako u příspěvku označit událost jako užitečnou („useful“). Seznam uživatelů, kteří událost označili, jako užitečnou se zobrazí pod obsahem události. Vlastník dané události ji může v případě potřeby editovat (změnit termíny konání). Případně ji celou odstranit. Potom se smažou všechny zaznamenané odpovědi i označení jako užitečný příspěvek.

UserName posted minute ago

Useful | [Edit Event] | [Remove]



Event title

Begin date: 14.04.2013 - 10:00

End date: 15.04.2013 - 12:00

Location: Znojmo

URL: www.znojmo-city.cz

Description: Event description

Answers: *Yes*
Maybe
No

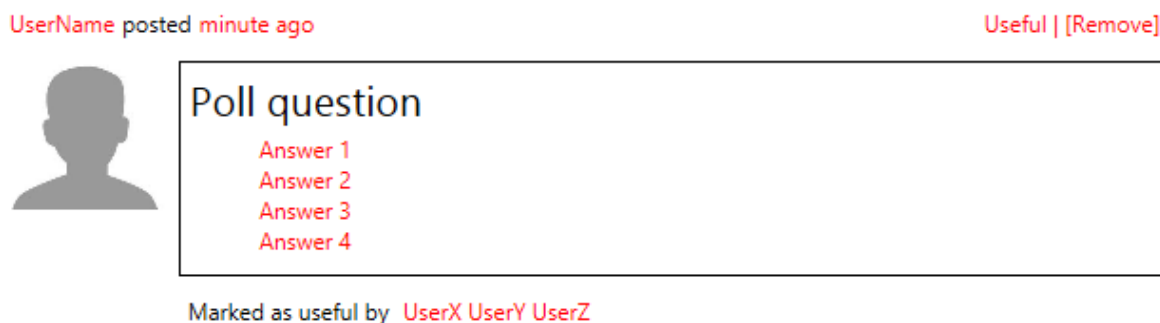
Marked as useful by UserX UserY UserZ

Obrázek 20 - příklad události jak ji vidí uživatel před hlasováním, zda se zúčastní

3.6 Průzkum na mikroblogu

Průzkum je také vlastně příspěvek, který opět přidává další vlastnosti. Uživatel ho může použít pro zjištění, co si o dané otázce myslí ostatní členové sítě. Může tedy mezi nimi udělat anketu. Ke každé otázce jim pak dá na výběr několik odpovědí. Obsah příspěvku zde funguje jako otázka, na kterou uživatelé v průzkumu odpovídají. Otázka musí být povinně vyplněna, jinak by průzkum postrádal význam. K této otázce může autor vytvořit až 5 různých odpovědí. Musí však vždy vyplnit minimálně dvě odpovědi. Otázka s jednou odpovědí by k ničemu nepřispěla. Při vytváření by bylo k dispozici jedno vstupní pole pro otázku a pět polí pro odpověď. Vyplnit by se daly libovolně z nich. Nemusí se tedy vyplňovat postupně. Aplikace by si pak automaticky vybrala pouze vyplněná pole s odpověďmi. V odpovědích i otázce se mohou vyskytovat jak hashtagy tak i označení uživatelů. Jejich obsah nebude jiným způsobem nijak omezen.

Po vytvoření průzkumu se ostatním uživatelům sítě objeví ve streamu. Každý člen včetně autora má možnost k dané otázce vybrat jednu odpověď. Před kliknutím na odpověď se uživateli nezobrazuje počet hlasování, aby nedocházelo k ovlivňování hlasování dalších členů sítě. Po kliknutí na zvolenou odpověď všechny odpovědi zmizí a zobrazí se graf, na kterém je jasně a přehledně vidět kolik uživatelů pro jakou možnost hlasovalo. I u průzkumu lze použít označení užitečný příspěvek („useful“). U průzkumu nemůže autor po vytvoření nijak zasahovat do zadání průzkumu, nejde tedy nic editovat jako je tomu u události. Jeho události lze tedy jenom prohlížet a případně mazat. Příklad průzkumu před hlasováním uživatele je na obrázku níže.



Obrázek 21 - příklad zobrazení průzkumu ve streamu před hlasováním

3.7 Doplnování při psaní hashtagů a uživatelů

Při psaní hashtagů do textu příspěvku se bude zobrazovat nabídka hashtagů, které již v databázi jsou uloženy. Uživatelům to usnadní přidávání zpráv k tématům. Nebudou muset typovat, jak se daný hashtag přesně píše, kde jsou velká a malá písmena, nebo jak přesně se slovo hláskuje. Zároveň s psaním písmen hashtagu se z databáze vybírají slova z tabulky hashtagů, která obsahují zadaný

podřetězec. Podřetězec se nemusí nacházet pouze na začátku slova, ale může být obsažen i uvnitř slova. Vyhledávání se provádí asynchronními dotazy. Uživatel tak pořád může pracovat s uživatelským rozhraním, zatímco se hledají vhodní adepti na zobrazení v nabídce. Pokud se v nabídce objeví hashtag, který uživatel chce použít, stačí, když jej vybere a slovo se doplní do textu zprávy. Zobrazení nabídky pro výběr hashtagu se zobrazí, pokud uživatel napíše znak „#“ a začne bezprostředně za ním psát slovo, tak jak je to vidět na ilustračním obrázku níže. Automatické vyhledávání se spustí po napsání alespoň dvou znaků po začátku hashtagu, tedy #. Vyhledávání z databáze hashtagů se bude provádět napříč sítěmi, takže pokud uživatel již konkrétní hashtag použil při publikaci příspěvku v jiné síti, tak se mu nabídne i v dalších jeho sítích.



Obrázek 22 - nápověda pro psaní hashtagů

Podobně je to i s označováním uživatelů v příspěvcích. V příspěvku se dají označit členové sítě, jichž se daný příspěvek týká. Jako úvodní znak pro zobrazení nabídky jmen je plus, před kterým je mezera. Na obrázcích níže je zobrazen princip přidávání odkazů na členy v mikrobloku Swabr, kde pro identifikaci uživatelé používají znak zavináč. Vyhledává se podle uživatelského jména, které má každý člen týmu přiřazen od registrace svého profilu, kde si ho sám zvolil. Po vybrání ze seznamu se za plus do textu vloží nick přidělený danému uživateli. Po zveřejnění ve streamu se za zavináč doplní celé jméno uživatele. Vložené uživatelské jméno se zobrazí v příspěvku jako odkaz, který vede na stránku, kde si o konkrétním uživateli můžeme zjistit statistiku jeho aktivit a informace získané v rámci mikrobloku.



Obrázek 23 - výběr odkazu na uživatele z databáze (swabr.com)

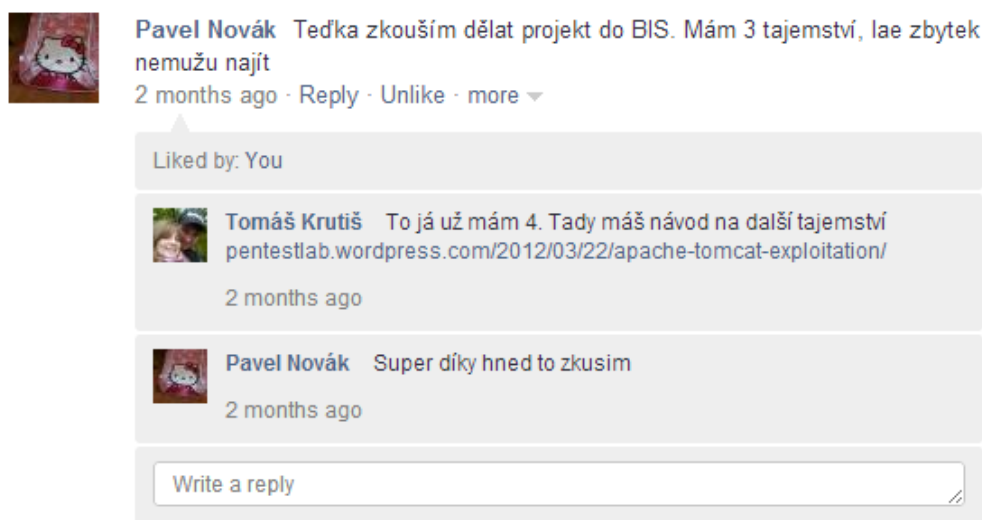


Obrázek 24 – příspěvek, ve kterém je vložen odkaz na uživatele (swabr.com)

3.8 Odpovědi na příspěvky

Dalším rozšířením stávajícího mikrobloggeru, by byla možnost odpovědi přímo u příspěvku. Inspiraci bych bral v systému odpovědí, jak je implementovaný u mikrobloggeru Swabr. Pod příspěvkem je přímo textové políčko pro odpověď. Příspěvek by tak musel udržovat reference na své odpovědi v databázi. Reagovat odpovědi by šlo na všechny typy zpráv na mikrobloggeru. Tedy na příspěvky, události a průzkumy. Pod každým příspěvkem by se pak řadily všechny odpovědi a to tak, že by nejnovější odpovědi byly umístěny nejnižší. Pokud by tedy uživatel psal odpověď, viděl by tu nejnovější nad polem pro zadání odpovědi. Odpověď by mohli používat všichni uživatelé sítě.

Odpověď by tedy byla další speciální varianta příspěvku, na kterou by už nešlo odpovídat, a musela by si držet odkaz na rodičovský příspěvek, k němuž patří. Rodičovským příspěvkem je myšlen příspěvek, který je rodičem pro dané vlákno odpovědí, přičemž všechny odpovědi v tomto vlákne mají stejného rodiče. Autor by měl opět možnost vytvořenou odpověď smazat. Uživatelé by také mohli označovat odpovědi jako užitečné (useful). Ve streamu sítě by se odpovědi musely vždy zobrazovat jenom s daným rodičovským příspěvkem, na který odpovídá. Pokud dojde ke smazání rodičovského příspěvku, budou smazány i ty odpovědi, které souvisely se smazaným příspěvkem. Odpovědi bez rodičovského prvku nemohou samostatně vystupovat, je tedy zbytečné jejich setrvání v databázi.



Obrázek 25 - ukázka odpovědi (swabr.com)

3.9 Profil uživatele

Při registraci se vytvoří uživateli na mikrobloggeru profil. V současné verzi Titteru se dá vložit jméno, nick, email a heslo uživatele. To zůstane u nové verze zachováno, schéma databáze bude navrženo tak, aby se jako budoucí rozšíření daly přidat další vlastnosti uživatele, jako třeba skutečné jméno a příjmení, adresa a další kontakty na daného uživatele (Skype, ICQ, Jabber,...). Avatar, neboli profilový obrázek se získává ze služby Gravatar, kde uživatel musí mít účet a registrovaný stejný email, jako použil při registraci na mikrobloggeru Titter. Avatar se totiž získává pomocí hashe z emailové adresy uživatele. Pokud uživatel nepoužívá službu Gravatar, na jeho profil je umístěn tzv. wavatar. Jde o automaticky generovaný obrázek na základě hashe emailu. To znamená, že i když uživatel nemá vytvořen vlastní avatar, tak je mu vygenerován jedinečný obrázek přímo pro jeho neregistrovaný email. To zvýší přehlednost při čtení příspěvků na mikrobloggeru, protože každý uživatel bude mít jiný avatar. Nebude tak docházet k přehlédnutí autora u příspěvků, jako při použití výchozího avataru, který je pro všechny neregistrované uživatele stejný.

Na stránce profilu si uživatel bude moci zobrazovat aktivitu, kterou generuje na mikrobloggeru. V první řadě si může nechat vypsat všechny jeho příspěvky, události nebo průzkumy, ty se zde seskupí ze všech jeho sítí dohromady a nejnovější jsou ve streamu příspěvků umístěny nahoře. Uživatel si tak může procházet vše, co na mikroblogger umístil a sledovat jak na jeho příspěvky ostatní uživatelé reagují a co na ně odpovídají. Pokud je to třeba, může ihned odpovídat. Může zde i mazat, protože vše co se zde zobrazuje, tak je jeho výtvořem a má tedy právo na smazání. U událostí má možnost editace termínu konání. Může také hlasovat pro všechny své události a průzkumy, pokud to již neučinil dříve.

Za další si zde může zobrazit všechny příspěvky označené jako užitečné. Příspěvky jsou agregovány napříč všemi sítěmi. U každého vybraného příspěvku je pak v horní části i označení sítě, do níž patří. Jelikož se dá označit jako užitečný i odpověď na příspěvek, který se nedá zobrazit samostatně, tak je vždy vybíráno celé vlákno, tedy jak rodičovský příspěvek, tak i všechny odpovědi, které jsou k ní vázány. Na tyto užitečné příspěvky pak může reagovat na jednom místě. Dají se označit, že už pro uživatele nejsou důležité a tím se ze streamu užitečných příspěvků odstraní.

A nakonec si uživatel může procházet příspěvky kde je jeho uživatelské jméno zmíněno v textu příspěvku. Pokud tedy někdo do textu vloží konkrétního uživatele a jeho uživatelskému jménu předchází znak plus „+“, tak je tato vazba zaznamenána do databáze. Uživatel si tedy může takové příspěvky vylistovat a zjistit kdo o něm něco na mikroblogger píše. Může tedy na tyto zprávy reagovat pěkně na jednom místě. Příspěvky jsou opět řazeny od nejnovějších a jsou vybírány napříč sítěmi.

3.10 Vyhledávání

Důležitou částí tohoto projektu je vyhledávání informací na mikrobloggeru. Aby se uživatel v příspěvcích neztrácel a našel, co potřebuje, bude k dispozici vyhledávání. Vyhledávání bude probíhat

na základě textu obsaženého v příspěvcích, událostech, průzkumech a také v odpovědích na všechny typy příspěvků. Množina příspěvků, ve které se bude vyhledávat, se stanoví na všechny příspěvky, které se nacházejí v sítích, kde je uživatel, jenž vyhledává, členem.

K dispozici bude jedno vstupní pole, do něhož se zadá požadovaný text, který se má v databázi vyhledat. Vstupní pole bude mít nápovědu pro hashtagy a uživatele. Bude to fungovat stejně jako při vytváření příspěvků. Stačí, když uživatel napíše znak plus nebo mřížka následované dvěma znaky a pokud je daný uživatel nebo hashtag začínající zadanými znaky v databázi, tak se zobrazí v nabídce pod vyhledávacím polem. Jinak se do pole mohou psát libovolné řetězce, které se následně vyhledávají v obsazích příspěvků. Jsou vybrány příspěvky, u kterých byla nalezena shoda s vyhledávaným textem. Pokud se hledaný obsah nachází v odpovědi na příspěvek, tak se vypíše celé vlákno, tedy rodičovský příspěvek i se všemi odpověďmi včetně vyhledané odpovědi. U všech zobrazených vláken příspěvků se dají přidávat odpovědi a označovat jednotlivé příspěvky jako užitečné.

V současném návrhu bude zahrnuta výše popsaná funkcionalita. Jako další vylepšení by se dala přidat nabídka, ve které by se vyhledávání dalo více specifikovat. Tuto možnost bych realizoval až na základě zpětné vazby uživatelů, pokud by o to byl zájem.

3.11 Filtrování příspěvků

Další možností jak se dostat k informacím na mikroblogu, kromě vyhledávání, je filtrování příspěvků. Uživatel si může vytvořit libovolné množství filtrů a ukládat si je. Filtry je pak možné používat i později a nemusí se znovu nastavovat.

U každého filtru je potřeba zvolit název, bez něj filtr nepůjde vytvořit. Filtr má další vlastnosti, které určují, které příspěvky filtrem projdou a které ne. Při vytváření filtru má uživatel k dispozici vstupní pole, do kterého uživatel zapisuje uživatelská jména členů mikroblogu a hashtagy. Toto pole poskytuje nápovědu při zadávání uživatelských jmen i hashtagů. Při psaní se automaticky nabízí hashtagy a jména uživatelů uložených v databázi. Obsah tohoto pole se rozdělí na jednotlivá slova. Pokud slovo nezačíná znakem plus „+“ nebo mřížka „#“, je vyřazeno, protože se nedá určit, jestli se jedná o uživatelské jméno nebo hashtag. Ostatní slova se rozdělí na uživatelská jména a hashtagy. Hashtagy jsou všechny uloženy do databáze, ať už v databázi dříve byly anebo ne. Takže pokud se hashtag časem vyskytne v některém příspěvku, tak jej filtr zachytí. Uživatelská jména jsou jednotlivě vyhledána v databázi v seznamu uživatelů. Pokud je uživatel s tímto jménem nalezen, je zaznamenán odkaz na tohoto uživatele. Když se žádný uživatel neshoduje, dané jméno se vynechá, protože se nedá zaznamenat reference na uživatele.

Pro uložení uživatele a hashtagy si tvůrce může zvolit, jestli chce, aby je filtrované příspěvky obsahovaly anebo ne. Tuto volbu je možné definovat zvlášť pro uživatele i hashtagy. Je tedy možné nastavit, že chceme zahrnout uživatele, ale nechceme zahrnout zadané hashtagy. Při výběru uživatelů

se jedné o příspěvky, které zapsaný uživatel vytvořil. Neberou se tedy v potaz příspěvky, kde je odkaz na uživatele zapsaný v obsahu příspěvku.

Dále si v definici filtru může uživatel vybrat ze seznamu sítí, ve kterých je členem, ty které chce do filtrování zahrnout. Může u nich také zvolit, jestli chce filtrovat příspěvky z vybraných sítí anebo nechce. Uživateli se síť obrazí v seznamu. Může si vybírat pouze ze svých sítí. V seznamu si uživatel vybere síť a tlačítkem je přenesení do druhého seznamu, kde jsou síť filtru. Síť může uživatel libovolně přesouvat do filtru a zpět podle libosti. Po uložení vytvořeného filtru se objeví v seznamu dostupných filtrů a uživatel ho ihned může začít používat. Po vybrání filtru a spuštění filtru se zobrazí příspěvky odpovídající specifikaci.

Filtrování probíhá tak, že na začátku se vybere množina všech příspěvků, ze všech sítí, v nichž je uživatel členem. Z této množiny se postupně vybírá podmnožina, která odpovídá specifikaci filtru. Po projití všech vlastností je na výstupu podmnožina, která odpovídá konkrétnímu filtru, a příspěvky z této podmnožiny jsou zobrazeny ve streamu. Pokud filtr vybere odpověď na příspěvek, zobrazí se celé vlákno i s rodičovským příspěvkem a všemi odpověďmi. I ve streamu filtru se dá odpovídat na příspěvky a označovat je jako užitečné.

Uživatel všechny své filtry může spravovat. Filtry může editovat a přidávat nebo odebírat uživatele, hashtagy a síť. Může je také mazat.

3.12 Logika aplikace a její části

Mikroblog je koncipován jako webová aplikace. Stěžejní částí aplikace bude databáze pro ukládání dat uživatelů mikroblogu. Bez potřebných dat by aplikace celkem postrádala význam. Aplikace na webu pak slouží jako pohled na data a pro vytváření nového obsahu. Na webu běží aplikační logika, která řídí požadavky uživatele a poskytuje mu potřebné informace. Základem budou třídy, které obstarávají přístup k datům v databázi. Tyto třídy spravují uživatele, příspěvky a veškeré modifikace příspěvku. Další třídy pak obstarávají správu filtrů a vyhledávání.

Při každém zobrazení kteréhokoli streamu příspěvků se z databáze získají potřebné identifikátory. Dále se použije definovaná šablona vlákna příspěvků a podle identifikátoru se z databáze načtou všechna potřebná data, přičemž se naplněná šablona zobrazí ve streamu. Toto se opakuje pro všechny vybrané příspěvky. Pokud je připravena šablona, tak zbytek zobrazování požadovaných příspěvků je pouze o správném výběru příspěvků.

Šablona vlákna příspěvku bude vytvořena tak, že bude zvládat zobrazení normálního příspěvku, události i průzkumu a to včetně všech jejich vlastností, jako graf průzkumu nebo termíny události. Zobrazí se i odpovědi na příspěvek, kterých může být libovolné množství. Pro odpovědi bude vytvořena zvlášť šablona, pro zobrazení jedné odpovědi. Šablona pro odpověď je pak vygenerována v cyklu pro každou jednu odpověď na rodičovský příspěvek. Šablony budou automaticky aktualizovat čas

od vytvoření příspěvku, označení uživatelem jako užitečné a nové nebo smazané odpovědi na rodičovský příspěvek.

The image shows a template for a forum post. It is enclosed in a green border. At the top, it says "UserName posted one hour ago" in red, followed by "Useful | [Remove]" in red. Below this is a grey silhouette of a person's head. To the right of the silhouette is a large rectangular box labeled "Post content". Below the content box, it says "Marked as useful by UserX UserY UserZ" in red. Below this is another section enclosed in a blue border. It starts with "UserName2 posted minute ago" in red, followed by "Useful | [Remove]" in red. Below this is another grey silhouette of a person's head. To the right of the silhouette is a rectangular box labeled "Answer content". Below the content box, it says "Marked as useful by User1 User2 User3" in red. Below the blue-bordered section is a text input field with the placeholder "Write a reply ...". To the right of the input field is a button labeled "Send answer".

Obrázek 26 - šablona pro generování příspěvku (zeleně je označena šablona pro celé vlákno příspěvku a modře je šablona pro odpověď, která se generuje v cyklu pro každou odpověď, dole je pole pro vložení nové odpovědi do daného vlákna)

3.12.1 Stránky aplikace

Pro přístup na všechny stránky mikrobloggeru, kromě domovské stránky, je nutné, aby byl uživatel přihlášen přes svůj účet, který si předtím vytvořil. Bez přihlášení by na ostatních stránkách nebylo co zobrazovat, protože by aplikace nevěděla co z databáze vybrat. Aplikace je tedy koncipována tak, že se nepřihlášený uživatel nedostane nikam jinam, než na domovskou stránku.

První ze stránek aplikace je Stream. Na této stránce se v horní části zobrazí dostupné sítě pro přihlášeného uživatele. Pro každou vybranou síť se zobrazí příspěvky patřící do této sítě. Přepínáním sítě se mění obsah streamu. Tato stránka je také jediná, na které se dají přidávat rodičovské příspěvky. Tedy příspěvky, události a průzkumy. Jsou zde formuláře pro vytvoření všech tří typů příspěvků. Nový příspěvek se vloží do sítě, jež je aktuálně vybraná v horním panelu a také se zobrazí jako první příspěvek v této síti. Toto vytvoření by se mělo co nejdříve promítnout i ostatním členům sítě, aby se o tomto příspěvku dověděli. Stránka by si měla pravidelně kontrolovat stav databáze, aby zjistila, co nového bylo přidáno, vytvořeno a odstraněno od jejího načtení. To platí nejen o nově přidaných příspěvcích, ale i odpovědích na příspěvky. Vytváření příspěvků je popsáno výše v této kapitole. Uživatel tak bude

mít aktuální stav mikroblogger bez většího zpoždění nebo nutnosti manuálně obnovovat celou webovou stránku.

Další stránka je Profile, kde se zobrazují příspěvky přihlášeného uživatele, jak je popsáno dříve v podkapitole Profil uživatele. Stránka Networks ukazuje síť, kterými je uživatel členem. U nich si může zobrazit aktuální seznam členů každé sítě. Zde se vytváří, editují a ruší síť na mikroblogger. Stránka Filters je pro vytváření, editování a mazání filtrů a taky pro používání vytvořených filtrů. Zatím poslední stránkou je Search na níž je pouze vyhledávací pole a tlačítko pro vyhledání. Po zadání textu a vyhledání se vygeneruje seznam příspěvků podle zadaného textu.

3.13 Schéma dat aplikace

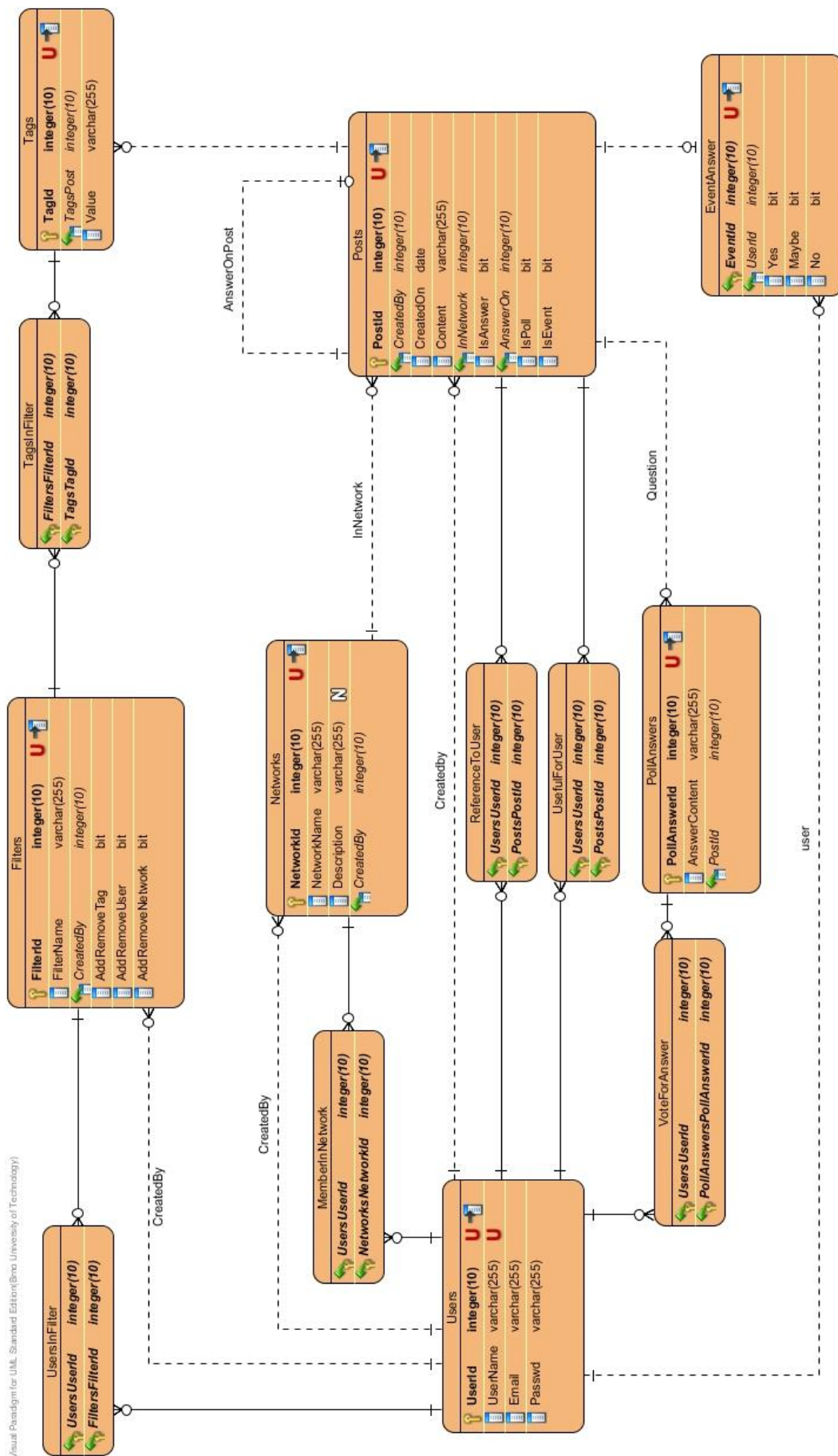
Data, která využívá aplikace mikroblogger pro týmy, budou uchovávána v databázi. Schéma databáze bude realizováno tak, že hlavními tabulkami budou tabulky „Users“, tabulka „Networks“ a tabulka „Posts“.

Tabulka „Users“ obsahuje informace o užívatelích aplikace. Bude použita při všech identifikacích užívatelů v rámci mikroblogger. Tedy při určení autora příspěvku, označení uživatele v příspěvku, užívatelé, kteří označili příspěvek jako užitečný, při správě sítí atd. Každý uživatel bude mít v rámci této tabulky jedinečné identifikační číslo a jedinečné uživatelské jméno.

Tabulka „Networks“ obsahuje ke každé síti jedinečný identifikátor, jméno a popis sítě a odkaz na uživatele, kterému síť patří a vytvořil ji.

Tabulka „Posts“ je hlavní pro všechny typy příspěvků. Ať už se jedná o rodičovský příspěvek, událost, průzkum anebo odpověď na příspěvek, mají všechny společné vlastnosti, které jsou zaznamenány právě v této tabulce. Tyto vlastnosti jsou jedinečný identifikátor příspěvku, odkaz na autora do tabulky „Users“, odkaz na síť do které náleží z tabulky „Networks“, datum a čas vytvoření, obsah daného příspěvku (otázku průzkumu, nadpis události). Dále obsahuje čtyři příznaky nastavené podle druhu příspěvku (rodičovský příspěvek, událost, průzkum, odpověď na příspěvek). V atributu „AnswerOn“ je zaznamenáno identifikační číslo rodičovského příspěvku, když se však nejedná o odpověď na příspěvek je tam nastaveno null. Pokud je daný příspěvek událostí, pak se zaznamenávají účasti užívatelů na události. Pro každé odpovědi uživatele jestli se zúčastní, se vytvoří nový záznam do tabulky s jeho odpovědí. Odpovědi na událost jsou s tabulkou „Posts“ spojeny přes tabulku „Events“, která rozšiřuje příspěvek o potřebné atributy události.

Je-li příspěvek specializací průzkum, tak se v tabulce „PollAnswers“ vytvoří záznamy s konkrétními odpověďmi a přes tabulku „VoteForPollAnswer“ se zaznamenávají odpovědi jednotlivých užívatelů.



Obrázek 27 - navržené schéma databáze

4 Implementace

V této části práce se zaměříme na implementaci mikrobloggeru pro týmy. Podíváme se, jak jsou implementovány jednotlivé části aplikace, jaké technologie a postupy byly použity a jakým způsobem jsou ukládána data pro fungování služby a popis manipulace s těmito daty. Aplikace je implementována webovou technologií ASP.NET jež je součástí .NET frameworku od firmy Microsoft.

4.1 Použité technologie

V této podkapitole si popíšeme jednotlivé použité technologie a nastíním, proč byly právě tyto technologie při tomto projektu použity. Všechny potřebné nástroje jsem získal zdarma z programu Microsoft DreamSpark for Academic Institutions.

4.1.1 ASP.NET

Aplikace je v základu postavena na technologii ASP.NET. Jedná se o webový framework pro vývoj webových stránek a aplikací za použití HTML, CSS a Javascriptu. Jedná se o součást kompletního vývojářského .NET frameworku, o jehož vývoj se stará firma Microsoft. Vychází z předchozí technologie ASP (Active Server Pages), což je skriptovací platforma také od firmy Microsoft. Obě technologie se však v současnosti od sebe velmi liší. ASP.NET je založen na CLR (Common Language Runtime), na němž fungují všechny části .NET frameworku. Projekty se tak dají programovat ve všech programátorských jazycích, které CLR podporuje. Nejčastěji se vyskytují jazyky Visual Basic a C#. C# jsem si zvolil i pro tento projekt, jelikož s ním mám z dřívějších pozitivních zkušeností. Použil jsem ho již v mé bakalářské práci, kterou jsem dělal pomocí technologie Silverlight, kde se také tento jazyk dá použít. Zdrojové kódy se na serveru překompilují na knihovny DLL, takže jejich zpracování je rychlejší než v případě interpretace skriptovacího jazyka, kterým je např. PHP.

ASP.NET WebForms je součástí ASP.NET frameworku a v tomto projektu bude použita. WebForms poskytuje vývojáři stavové prostředí nad protokolem HTTP, který je bezstavový. K udržení stavu se používá HTML kód a Javascript. Zapamatování stavu pak zajišťují dvě techniky. Zaprvé ViewState, která kontext udržuje ve skrytých polích přímo v HTML kódu na stránce. Není tedy potřeba nic dalšího pro zachování stavu, ale velikost přenášené stránky se tím zvětšuje, což může vést ke zpomalení přenosu stránky. Druhou technikou je SessionState. Ta ukládá všechny informace kontextu na straně serveru a klientovi je předává jako část URL nebo cookies. WebForms je řízený událostmi, takže všechny dostupné komponenty reagují na jakýkoliv vstup uživatele a přes Javascript vyvolají obslužnou rutinu, které se nachází na serveru v přeložené DLL knihovně. Vybraná rutina se

provede a odešle se zpět na stranu uživatele jako HTML kód. Rutiny jsou v tomto případě psány v jazyce C#.

Tento mikroblog je založen na nejnovějším .NET frameworku 4.5, zvolil jsem jej, protože poskytuje přepracovaný model členství uživatelů (Membership), který je použit pro správu uživatelů a jejich uživatelských účtů v rámci aplikace, a to včetně obsluhy přihlášení a odhlášení z aplikace [26][27]. Mikroblog jsem vyvíjel ve Visual Studiu 2012, které poskytuje vývojáři skvělý komfort při psaní aplikací. Hlavně debugger, který je zde integrovaný, hodně pomůže při hledání chyb v logice kódu. Také IntelliSense, tedy automatické doplňování a nápověda při psaní kódu ušetří spoustu času a chyb v překlepech.

4.1.2 MS SQL Server

Pro ukládání dat jsem použil Microsoft SQL Server 2008 R2. Jedná se relační databázi, která ukládá data v relacích (tabulkách). Vybral jsem ji proto, že se snadno integruje do .NET projektů a není k ní potřeba žádný další konektor pro připojení a správu dat v databázi. Pro správu databáze jsem využíval nástroj SQL Server Management Studio, kde se dají přehledně spravovat jednotlivé tabulky a hlavně vytvářet asociace mezi tabulkami. Mohl jsem rychle a přehledně prohlížet obsah tabulek a při vývoji testování je patřičně modifikovat. Verzi 2008 R2 jsem zvolil, protože s ní mám velice dobré zkušenosti z jiných projektů.

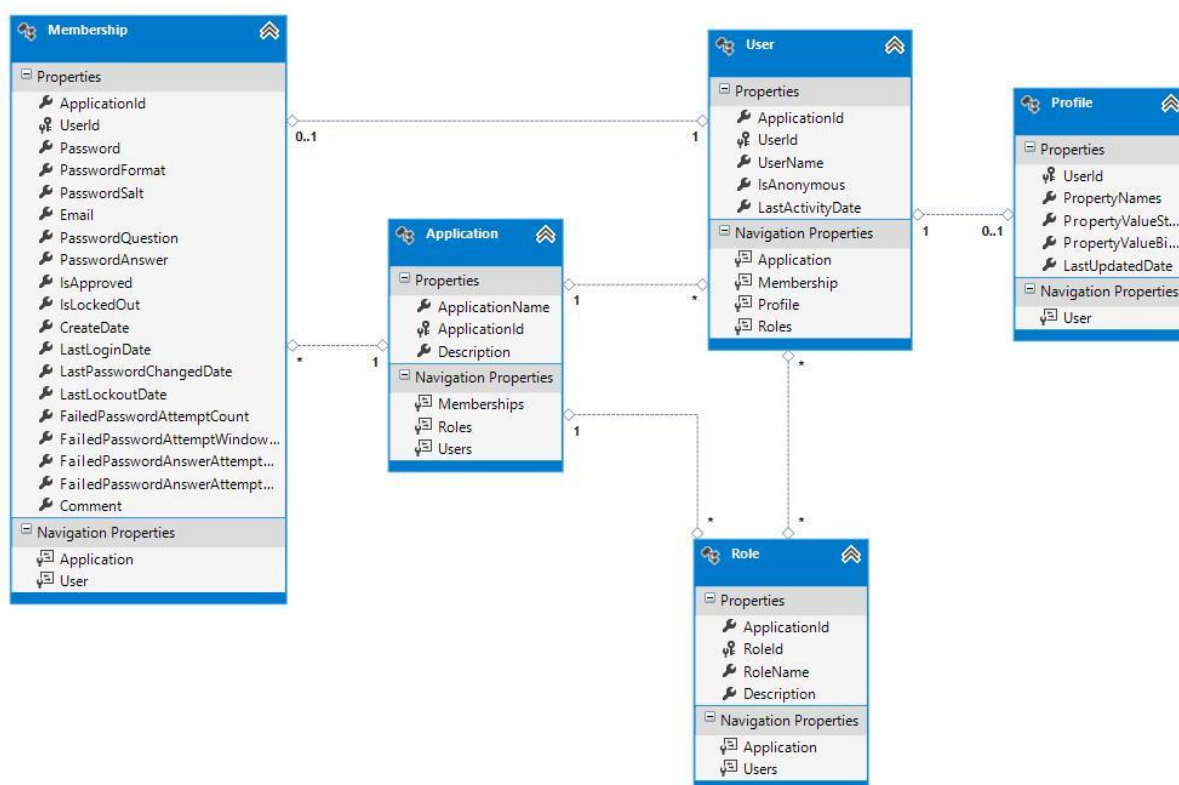
4.1.3 jQuery

Jedná se o javasriptovou knihovnu, fungující napříč všemi webovými prohlížeči. Je navržena pro snadnější scriptování na straně klienta v HTML kódu. Zjednodušuje navigaci v dokumentu. Snadněji se dostaneme k DOM elementům na stránce. Jednodušší vytváření animací a obsluha událostí na stránce a vytvářet asynchronní komunikaci mezi serverem a klientem (AJAX). Microsoft zahrnul jQuery do Visual Studia, takže při vytvoření webové aplikace je jQuery již v aplikaci obsaženo a lze s ním pracovat.

4.2 Členství uživatele v mikroblogu

Pro uživatelské účty jsem použil integrovaný systém pro správu uživatelů, ASP.NET Membership. Systém jsem využil pro registraci uživatelů a určení, kam v mikroblogu mohou přistupovat. Aplikace díky tomu v jakékoliv chvíli ví, který uživatel je přihlášený a do které sekce má přístup. Při prvním spuštění webové aplikace se automaticky vytvoří tabulky, které jsou potřeba pro správné fungování systému členství. Schéma je vidět na obrázku níže. V tomto schématu je definována tabulka pro aplikaci, kde jsou uložena data celé aplikace. Tabulka obsahuje pouze jeden řádek, jelikož

daná databáze obsluhuje pouze jedinou aplikaci. Dále je zde nejpodstatnější tabulka „User“ obsahující informace o registrovaných uživateli. Další doplňující data uživatele jsou v tabulce „Membership“, která určuje hodnoty dalších atributů uživatele. Do budoucna bude potřeba tabulka „Profile“, do které se dají zaznamenat vlastnosti, které o uživateli jinde nejsou, tedy vše, co by se k informacím o uživateli časem přidávalo. Ukládá se sem název nové vlastnosti a také její hodnota. Lze do ní uložit prakticky jakékoliv textové informace.



Obrázek 28 - schéma pro uložení dat pro ASP.NET Membership

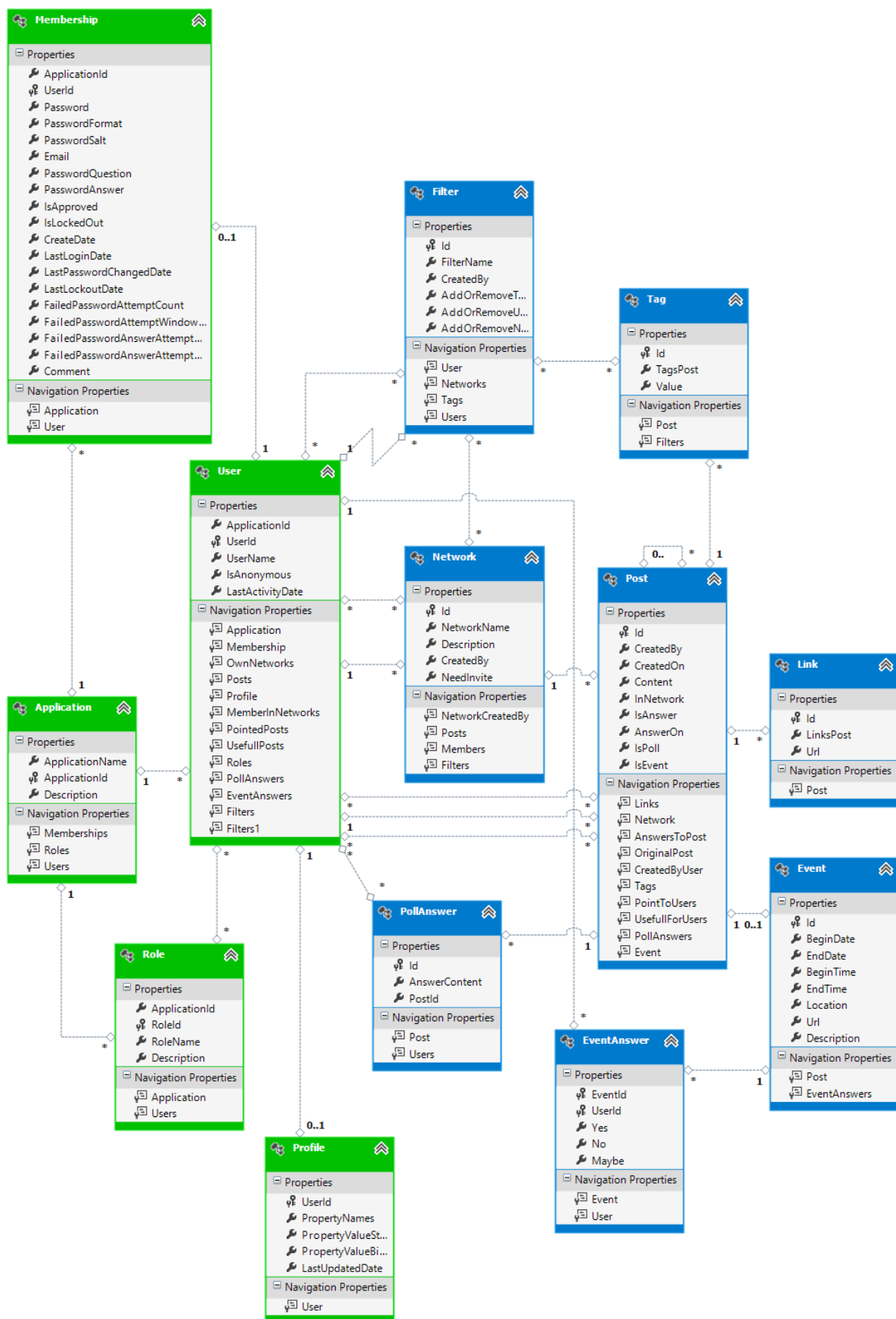
V návrhu byla tato datová část zastoupena pouze jedinou tabulkou „Users“, při implementaci jsem se ale rozhodl použít systém členství, takže se schéma rozrostlo o popsané tabulky. Tento systém však ušetřil spoustu času a kódu při implementaci a je také robustní, takže například hesla nejsou ukládána v podobě prostého textu, ale jako hash hesla, z čehož plyne, že útočník i při prozrazení dat uživatele není schopen dostat se k původnímu heslu, jelikož pro hashovací funkci nejde najít opačná, která by vrátila heslo. Vždy se tedy proti databázi ověřují pouze hashe hesel.

4.3 Schéma uložení dat a manipulace s daty

Data jsou uložena v relační databázi podle schématu vytvořeného při návrhu aplikace. Návrh se trochu změnil kvůli použití ASP.NET Membership pro správu uživatelů. K přístupu a manipulaci

s daty se použije pomůcka od Microsoftu pro manipulaci s daty v databázi Entity Framework. V tomto projektu je použit EF verze 5. EF se připojí k definované databázi, kde se nacházejí tabulky, které byly popsány v návrhu. EF tyto tabulky vezme a pro každou vytvoří třídu, přes kterou se pak dá manipulovat s daty v databázi. Každá třída má atributy mapované na jednotlivé sloupce v odpovídající tabulce. Jeden objekt dané třídy pak odpovídá jednomu řádku v tabulce. Třídy obsahují metody pro vytváření, úpravu a smazání řádku.

Vztahy jsou pak definovány jako reference na objekty anebo kolekce objektů. Vztah 1:N mezi tabulkami je definován jako reference na objekt, stačí tedy do reference přiřadit příslušný objekt a databáze se podle toho změní. Vztah M:N je definován jako kolekce objektů třídy tabulky na druhé straně asociace. Přidávání, úpravou a mazání v kolekci se mění i asociace mezi konkrétními řádky tabulek v databázi. Kolekce asociace je dostupná z obou tříd, vždy se však jedná o kolekci objektů třídy na druhé straně asociace. Nižší je na obrázku zobrazeno schéma tříd vytvořené podle databáze. Ve spodní části každé třídy jsou uvedeny „Navigation Properties“. Stačí tedy vytvořit instanci na EF, která zpřístupní namapované tabulky v databázi, provést jejich úpravu a říct EF, aby dané změny promítl do databáze. Takže vývojář pracuje pouze s třídami a objekty a nemusí vymýšlet SQL dotazy, kterými by se dotazoval přímo do tabulek. Je tak odstíněn od datové vrstvy. EF si z databáze do tříd zaznamená i integritní omezení stanovená schématem databáze. Vlastnosti daných tříd tedy mají nastaveny datové typy podle odpovídajících sloupců v tabulce, která k třídě patří. Stejně tak se natáhnou i primární klíče, které se mají samy inkrementovat, tudíž není potřeba se starat o unikátnost primárního klíče u nově přidávaného záznamu. I všechny kolekce a reference na objekty kontrolují, jestli jsou tam vkládány objekty příslušné třídy. Pokud se třídy neshodují, není dovoleno takovou operaci provést. To zaručí, že nedojde k nekonzistenci v databázi vložením nesprávného objektu.



Obrázek 29 - schéma tříd mikroblogu Twitter

4.4 Aplikační logika služby

Aplikační logika je určena především třídami, které vytvoří EF a jsou popsány výše. Všechny třídy EF jsou implementovány s příznakem „partial“, což znamená, že lze vytvořit stejnou třídu také s příznakem „partial“ a tak se k již vytvořené třídě EF přidávají další vlastnosti a metody pro obsluhu dané třídy. K většině tříd jsem tedy takto přidával metody, které mi pomáhaly a zlepšovaly práci s objekty dané třídy.

Zpočátku jsem také chtěl používat hotové komponenty pro práci se stránkami ASPX, které definují uživatelské rozhraní u uživatelem mikrobloggeru. Aby se při nějakém požadavku nemusela obnovovat celá stránka, kvůli jednomu prvku na stránce jsem používal komponentu UpdatePanel, která zajistí asynchronní provedení požadované operace na straně serveru a vrátí pouze aktualizovaný obsah UpdatePanelu. Požadované příspěvky byly zase umístěny v kolekci, takže jsem chtěl využít komponentu Repeater, kterou poskytuje ASP.NET framework. Tato komponenta pro každý záznam zdrojové kolekce vytvoří stejný výstup na stránku, pouze na definovaná místa zobrazí data z konkrétního prvku kolekce. Když jsem chtěl však tyto dvě komponenty používat současně v sobě zanořené, tak se vyskytl problém. Pokud jsem totiž v Repeateru chtěl použít komponentu UpdatePanel, aby se obnovovala pouze malá část vygenerované stránky, tak to nefungovalo. Jediná věc, co s tím šla takto udělat, bylo vytvořit jeden hlavní UpdatePanel a v něm mít Repeater, který pak generoval příspěvky podle zdrojové kolekce. To však vedlo k tomu, že při jakékoliv události, kterou vyvolal uživatel, se musely opět znovu přenášet všechny příspěvky, pouze malinko pozměněny.

Proto jsem se rozhodl aplikaci předělat a zvolil jsem knihovnu jQuery a webové služby, které poskytuje ASP.NET framework. Z knihovny jQuery si vypůjčíme funkci jQuery.ajax(), přes niž se dá zavolat libovolná webová služba běžící na serveru. Funkce .ajax() zavolá na serveru požadovanou webovou službu a ve formátu dat JSON, jí předá vstupní parametry. Po úspěšném zpracování dat webovou službou jsou data odeslána asynchronně zpět klientovi. Pro plynulejší načítání streamu příspěvků je tedy každý příspěvek zpracováván zvlášť v asynchronním dotazu. Webová služba pak podle šablony vygeneruje kód pro daný příspěvek podle jeho identifikátoru a přes JSON se tento kód odešle zpět klientovi, kde se zobrazí na stránce. Jsou tak řešeny i veškeré aktualizace a změny u příspěvků. Pro každý takový požadavek je samostatně volaná webová služba.

Touto úpravou se stránka načítá plynuleji a při nějaké změně se obnoví vždy jenom malá část stránky a nepřenáší se zbytečně i příspěvky, u kterých k žádné změně nedošlo. Automaticky se v určitých časových intervalech kontrolují nové a smazané příspěvky a odpoví na příspěvek. Kontrola je prováděna jednou za 60 vteřin. Největší zpoždění, které může tedy mezi publikací příspěvku a zobrazením ostatním členům sítě vzniknout, je jedna minuta.

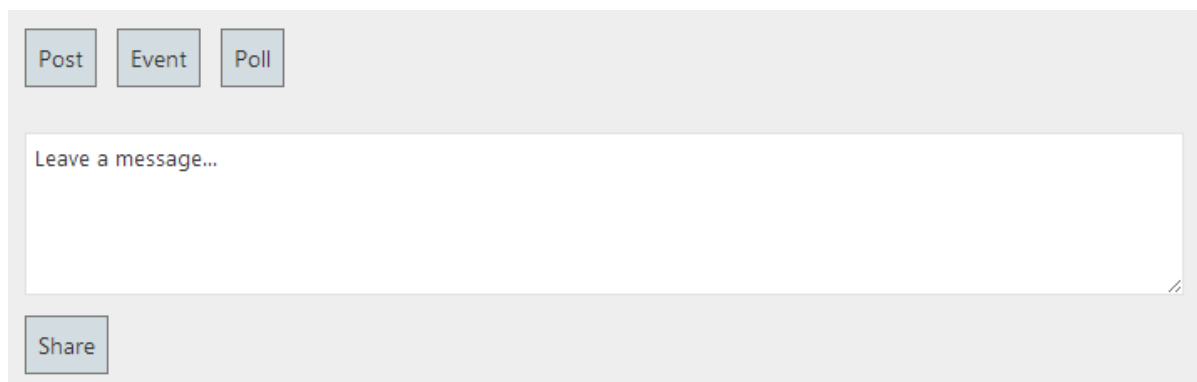
4.4.1 Příspěvky na mikroblogu

Nejdůležitější částí mikroblogu jsou příspěvky a to ve všech podobách. Tedy rodičovský příspěvek a jeho specializace událost a průzkum, a také odpověď na tyto příspěvky. V této části se podíváme, jak se tyto příspěvky vytváří a zobrazují ve streamu příspěvků. Rodičovské příspěvky je možné vytvářet pouze na stránce „Stream“, na ostatních stránkách se příspěvky vybírají podle různých pravidel a nově vytvořený příspěvek by se nemusel na dané stránce vůbec zobrazit.

Při zobrazování všech příspěvků na stránce se používají šablony kódu, kde se podle id příspěvku načtou všechna potřebná data z databáze. Každá tato šablona obsahuje skryté pole, do kterého se po vygenerování doplní id daného příspěvku, aby se po spuštění události k danému příspěvku dalo zjistit, o který konkrétní příspěvek se jedná a se kterým v události pracovat.

4.4.1.1 Příspěvek (Post)

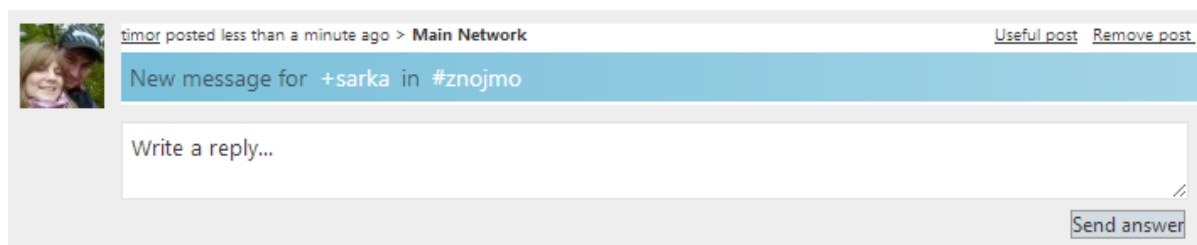
Zde se myslí rodičovský příspěvek. Ten obsahuje pouze text. Při psaní do textového pole pro obsah příspěvku se aktivuje nápověda pro vložení hashtagů a uživatelů. Tuto funkcionalitu zabezpečuje webová služba, která po každém vloženém znaku kontaktuje asynchronně server a pokusí se v databázi vyhledat odpovídající vzorky na zadaný text. Pokud něco v databázi najde, zobrazí se to v nabídce pod vstupním polem. Pokud uživatel některý vzorek vybere, tak se celý vypíše do vstupního pole.



Obrázek 30 - formulář pro vytvoření příspěvku

Po stisknutí tlačítka „Share“ se obsah asynchronně odešle na server i s vybranou sítí v záhlaví stránky, kde webová služba vytvoří nový objekt pro třídu příspěvku. Tento objekt uloží do databáze a zpět klientské straně vrátí identifikační číslo vytvořeného příspěvku. Klient toto id vezme a opět jej odešle zpět jiné webové službě, která na základě přijatého id příspěvku vyhledá potřebná data pro příspěvek v databázi a vloží je do šablony příspěvku. Z té následně vygeneruje HTML kód příspěvku, jenž odešle zpět klientovi, který tento kód vloží před aktuálně první příspěvek ve streamu dané sítě. Nový příspěvek je tak ihned autorovi k dispozici. Ostatním členům sítě, kteří mají stejnou síť

také zobrazenou, se zobrazí v krátké chvíli, záleží, v jaké části intervalu se nachází časovač pro aktualizaci příspěvků. Při pravidelné kontrole, zda se v databázi neobjevily nové příspěvky, se webové službě, která to kontroluje, posílá id nejnovějšího příspěvku ve streamu sítě. Pokud se v databázi takové příspěvky nachází, pošlou se zpět jejich id a klient si pak pro ně nechá vygenerovat HTML kód, jež opět vloží jako nejnovější příspěvky.



Obrázek 31 - nově vložený rodičovský příspěvek s polem pro napsání odpovědi

4.4.1.2 Událost (Event)

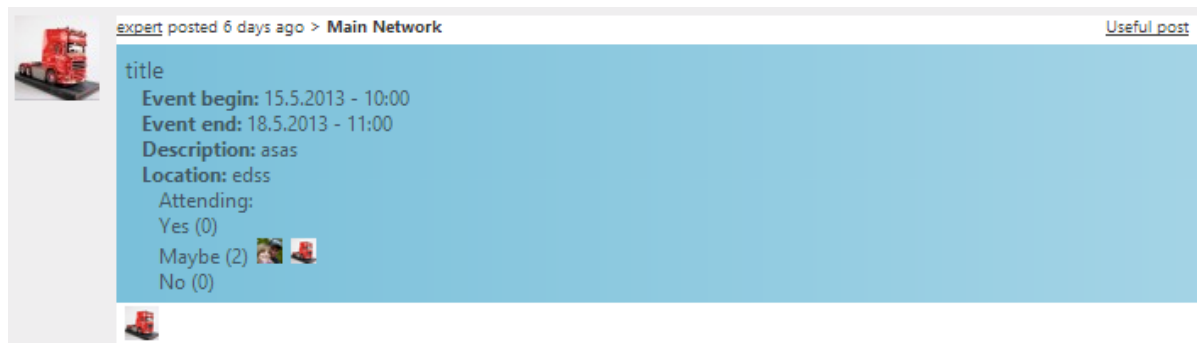
Událost je speciální variantou příspěvku. Má tedy některé vlastnosti navíc jinak vše funguje stejně, jak bylo popsáno výše u příspěvku. Navíc se u události před uložením do databáze kontrolují termíny události. Tedy jestli jsou ve správném formátu a jestli začátek předchází konci události. Pokud některá podmínka není splněna, je vypsáno chybové hlášení.

A form for creating an event. It has several input fields: 'Event title...' at the top, followed by 'Event begin' with sub-fields for date (21.5.2013) and time (10:00), then 'Event end' with sub-fields for date (21.5.2013) and time (11:00). Below these are 'Location...', 'Web link...', and 'Description...' fields. At the bottom left is a 'Share' button.

Obrázek 32 - formulář pro vytvoření události

Po uložení už se provádí stejné asynchronní operace jako při ukládání normálního příspěvku, včetně principu automatické aktualizace. Při prvním zobrazení se zobrazí odkazy, jako možnost

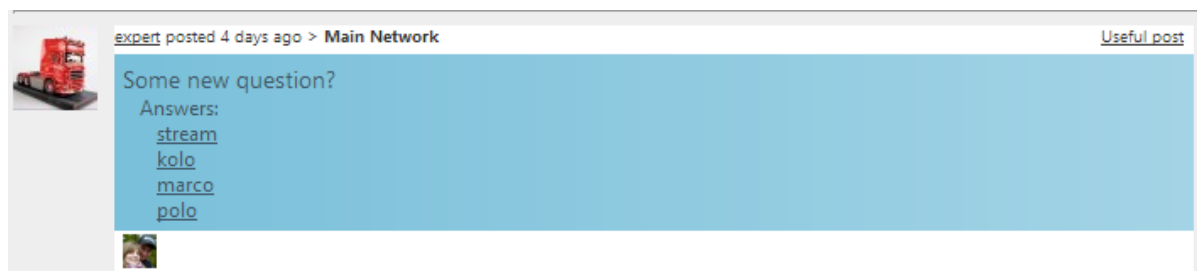
odpovědět na účast na události. Zobrazí se tedy „Yes“, „Maybe“ a „No“. Uživatel jednu z možností označí a ta se asynchronně uloží do databáze a zpět klientovi se přes JSON vrátí statistika hlasování i s uživateli, kteří hlasovali (viz. obrázek níže).



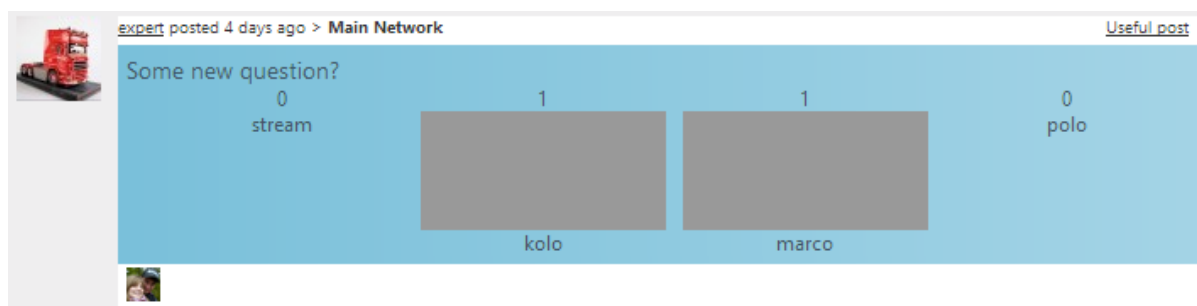
Obrázek 33 - příklad události, kde dva uživatelé hlasovali, že se možná zúčastní

4.4.1.3 Průzkum (Poll)

Průzkum je druhou specializací rodičovského příspěvku. Uživatel musí zadat otázku průzkumu a alespoň dvě z pěti odpovědí. Při stisknutí tlačítka „Share“ se odešlou všechny textová pole webové služby, která zkontroluje, jestli jsou všechny podmínky splněny. Pokud nejsou, vrátí se chybové hlášení, jinak jsou vybrány všechny vyplněné odpovědi a průzkum je uložen do databáze a vráceno je id příspěvku. Podle něho se pak vygeneruje kód pro zobrazení příspěvku.



Obrázek 34 - průzkum před hlasováním uživatele

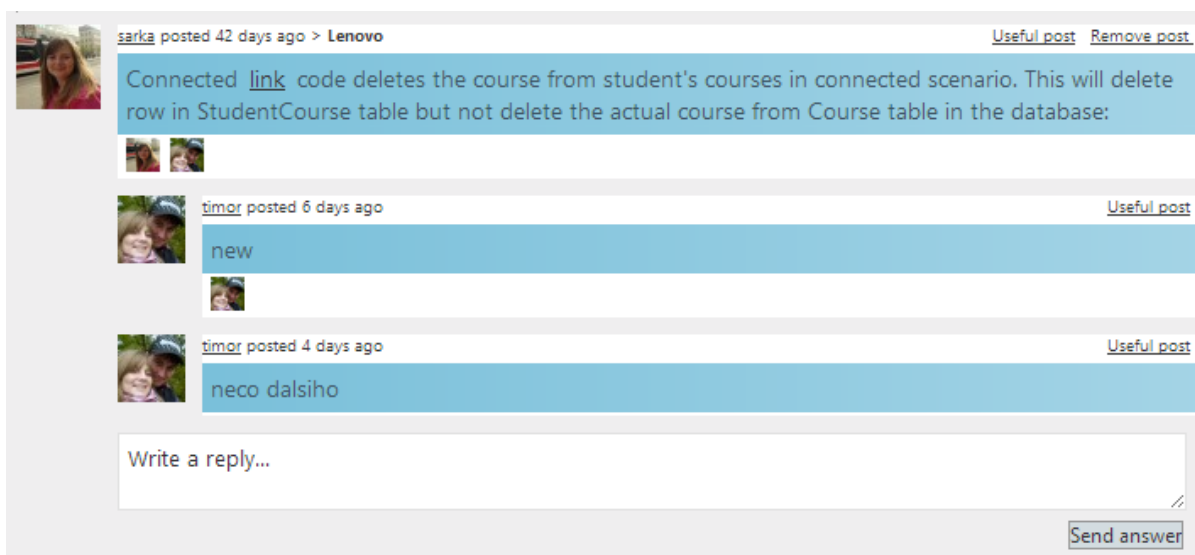


Obrázek 35 - průzkum po zvolení jedné z odpovědí

Na obrázcích výše je vidět, jakým způsobem má uživatel dostupné odpovědi na průzkum a jak se zobrazí statistika hlasování všech uživatelů. Na vykreslení grafů jsem použil zdarma dostupný plugin pro jQuery s názvem jqGraph.

4.4.2 Odpovědi na příspěvky

U každého rodičovského příspěvku se dá napsat libovolné množství příspěvků, proto je k dispozici pole pro odpověď. Ukládání probíhá přes stejnou třídu jako u příspěvku, akorát se nastavuje příznak, že se jedná o odpověď a odkaz na objekt rodičovského příspěvku. Nová odpověď se poté zobrazí ve vlákne pod příspěvkem, který je jeho rodičem. Samozřejmostí je automatická periodická aktualizace nových a smazaných příspěvků jiných uživatelů. Opět zde dochází k menšímu zpoždění mezi publikací a zobrazením dalším uživatelům, maximální prodleva je však 40 vteřin.



Obrázek 36 - příspěvek a dvě jeho odpovědi

4.4.3 Vyhledávání v příspěvcích

Vyhledávání se nachází na stránce „Search“. Hledají se příspěvky, které v textu obsahují zadané řetězce. Požadavek se asynchronně odešle na server webové službě, která má vyhledávání na starost. Ta rozdělí řetězec na jednotlivá slova a ta se pokouší vyhledat v obsahu příspěvků. Po nalezení id všech příspěvků, které hledání odpovídají, se ještě musí postarat o odpovědi na příspěvky. Místo nich do kolekce id příspěvků vloží id jejich rodičovského příspěvku. Kolekce s id je typu „HashSet“, to zaručuje, že se každé id bude v kolekci vyskytovat pouze jednou. Nebude tedy docházet k duplikaci vygenerovaných příspěvků. Kolekce se následně odešle zpět klientovi, ještě před odesláním se seřadí podle id od největšího po nejmenší. Takže nejnovější příspěvky budou nahoře ve streamu. Klient po přijetí vezme každé id a nechá si pro něj vygenerovat podle šablony HTML kód, který postupně zobrazí

na stránce vyhledávání. Pořadí příspěvků zůstane zachováno i při asynchronním zpracování, protože prázdné elementy div jsou do stránky vloženy ještě před generováním HTML kódů.

4.4.4 Filtry a filtrování příspěvků

Uživatel si definuje fungování filtru pomocí formuláře na jeho vytvoření. Při vytvoření si uživatel vybere od kterých členů mikrobloggeru chce nebo nechce zobrazit příspěvky, které hashtagy mají nebo nemají příspěvky obsahovat a ve kterých sítích mají nebo nemají být.

Filter name

Filter configuration

Filter select posts made by user and with tags write in field below or invert
Write users (+user) and tags (#tag) ...

Write users (+user) and tags (#tag) ...

☒ include users ☐ exclude users
☐ include write tags ☒ exclude write tags

Networks

- ToastPrinter
- Net2
- Lenovo
- PulpFiction
- Lama

Networks in filter

- Main Network
- Test network 3

☒ include select networks ☐ exclude select networks

Save filter

Obrázek 37 - formulář pro vytvoření filtru

Když pak chce nějaký filtr použít, tak ho v seznamu vybere a stiskne tlačítko použít. Asynchronně se zavolá webová služba, která má na starost filtrování příspěvků. Jako parametr se předá id filtru. Webová služba si vytvoří objekt třídy Filter a ten jí přístupní požadovaný filtr.

Po výběru filtru se načte množina všech příspěvků ze všech sítí, ve kterých je přihlášený uživatel členem. Toto je výchozí množina, ze které se následně vybírá podmnožina, která bude odpovídat specifikaci filtru. Dále se vyberou množiny příspěvků všech uživatelů zapsaných ve filtru, všech kde se v obsahu vyskytují zadané hashtagy a které se nachází v sítích zadaných ve filtru.

Nyní máme tyto čtyři množiny. Vezmeme hlavní množinu všech dostupných příspěvků a postupně bereme zbylé tři množiny. Pokaždé se podíváme, jestli množina (uživatelů, hashtagů, sítí) má nebo nemá být obsažena. Pokud má být obsažena, tak provedeme průnik množin, takže se vyberou všechny společné příspěvky. Když nemá být obsažena, dělá se rozdíl množin, takže se z množiny všech příspěvků ty nežádoucí odstraní. Po projití všech tří množin máme k dispozici seznam příspěvků, které filtrem prošly a mohou se tedy uživateli zobrazit.

Klientovi se posléze pošle seznam id příspěvků, kde id odpovědi na příspěvek je nahrazeno id rodičovského příspěvku, jsou zde odstraněny duplicity a jsou seřazeny od nejvyššího id. Klient po přijetí seznamu vytvoří pro každé id příspěvku nový div do stránky, aby nemohlo dojít k přeskokování příspěvků, a pro všechny id odešle asynchronní dotaz, jenž do každého divu následně nagenereje příspěvky.

5 Závěr a další pokračování projektu

Tato diplomová práce rozpracovává zadání projektu mikroblogger, zaměřeného a spolupráci a komunikaci v týmu uživatelů. Bylo v ní rozebráno co je to mikroblogger, jaké prvky a části obsahuje a zaměřili jsme se na několik již existujících mikrobloggerovacích služeb na webu. Z těchto zkoumaných služeb byly vybrány ty lepší a užitečnější funkce a prvky, které byly použity v mikrobloggerovací službě Titter, jejíž návrh a implementace je cílem této diplomové práce.

Při této práci jsem se naučil a poznal spoustu nových postupů a řešení, které se mi v budoucnu jistě budou hodit. Využil jsem i spoustu znalostí, které jsem získal během celého studia na FIT VUT a to jak z teoretické stránky, tak i z praktické při programování různých projektů do mnoha předmětů. Mikroblogger je v současnosti nasazený v rámci interní sítě firmy Univa Obchod, kde ho využívají vývojáři pro komunikaci na společných projektech. Mikroblogger je v této firmě nasazen od března 2012. Zatím jsem se setkal vesměs s pozitivními reakcemi a podněty. Ve vývoji mikrobloggeru budu do budoucna pokračovat, jelikož je zde stále potenciál pro přidávání další funkcionality a vylepšování té stávající. Práce na tomto projektu mě velice bavila, hlavně po nasazení mikrobloggeru ve firmě, kde jsem mohl sledovat, že tento projekt má opravdu smysl a význam.

Při návrhu a implementaci jsem se inspiroval u již fungujících podobných služeb a vybíral jsem z nich vlastnosti, které se mi zdály pro tento projekt přínosné.

Funkční verze se nachází na adrese `dp.area59.eu`. Aktuální zdrojové kódy jsou ke stažení na adrese `dp-source.area59.eu/src.rar`.

5.1 Budoucí rozšíření projektu

Do budoucna bych chtěl přepracovat grafiku mikrobloggeru, aby se dala přizpůsobit potřebám různých týmů. Další důležitou částí by bylo vytvoření API, přes které by se do mikrobloggeru mohly připojovat externí aplikace. Na to by navazovalo možné vytvoření mobilní aplikace pro chytré telefony. Bylo by potřeba více verzí pro různé mobilní operační systémy. Za zvážení by jistě stála i desktopová aplikace. Dále by se dala přidat možnost nahrávání souboru potažmo obrázků přímo na server, kde mikroblogger běží. Záležet bude také na zpětné vazbě od uživatelů.

Literatura

- [1] O'REILLY, Tim a Sarah MILSTEIN. *The Twitter book*. 2nd ed. Sebastopol, CA: O'Reilly, c2012, ix, 248 p. ISBN 14-493-1420-1.
- [2] ZARRELLA, Dan a Sarah MILSTEIN. *The social media marketing book*. 2nd ed. Sebastopol, CA: O'Reilly, c2010, vi, 232 p. ISBN 05-968-0660-4.
- [3] 7 things you should know about Microblogging. EDUCAUSE. *Educause* [online]. 2009 [cit. 2013-01-01]. Dostupné z: <http://net.educause.edu/ir/library/pdf/ELI7051.pdf>
- [4] Hashtag. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2012 [cit. 2013-01-01]. Dostupné z: <http://en.wikipedia.org/wiki/Hashtag>
- [5] SCHACHTER, Joshua. On URL shorteners. *Joshua schachter's blog* [online]. 2009 [cit. 2013-01-01]. Dostupné z: <http://joshua.schachter.org/2009/04/on-url-shorteners.html>
- [6] What a Difference a Year Makes: URL Shorteners Make the Web Substantially Slower, but Facebook's fb.me, Google's, goo.gl and BudURL Perform Perfectly. *The official WatchMouse blog* [online]. 2011 [cit. 2013-01-01]. Dostupné z: <http://blog.watchmouse.com/2011/04/what-a-difference-a-year-makes-url-shorteners-make-the-web-substantially-slower-but-facebook%E2%80%99s-fb-me-google%E2%80%99s-goo-gl-and-budurl-perform-perfectly/>
- [7] JACEK, Jonas. Why URL shortening services and shortURLs are bad. In: *Rield* [online]. 2011 [cit. 2013-01-01]. Dostupné z: <http://rield.com/faq/why-url-shorteners-are-bad>
- [8] URL Shortener API. *Google Developers* [online]. 2012 [cit. 2013-01-01]. Dostupné z: <https://developers.google.com/url-shortener/>
- [9] SEDLÁK, Jan. Jak se Seznam, Twitter či Facebook přesouvají na mobily. *Connect!* [online]. 2012 [cit. 2013-01-01]. Dostupné z: <http://connect.zive.cz/clanky/jak-se-seznam-twitter-ci-facebook-presouvaji-na-mobily/sc-320-a-164463>
- [10] Top 8 Mobile Operating System. *StatCounter Global Stat* [online]. 2012 [cit. 2013-01-01]. Dostupné z: http://gs.statcounter.com/#mobile_os-CZ-monthly-201210-201212-bar
- [11] Twitter. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2008 - 2012 [cit. 2013-01-01]. Dostupné z: <http://en.wikipedia.org/wiki/Twitter>
- [12] LAUSCHMANN, Jindřich. Twitter oznámil 200 milionů aktivních uživatelů. *Tyinternety.cz* [online]. 2012 [cit. 2013-01-01]. Dostupné z: <http://www.tyinternety.cz/2012/12/19/clanek/twitter-oznamil-200-milionu-aktivnich-uzivatelu/>

- [13] About Public and Protected Tweets. *Twitter* [online]. 2012 [cit. 2013-01-01]. Dostupné z: <https://support.twitter.com/groups/31-twitter-basics/topics/109-tweets-messages/articles/14016-about-public-and-protected-tweets>
- [14] Twitter Statistics. *SocialBakers* [online]. 2012 [cit. 2013-01-01]. Dostupné z: <http://www.socialbakers.com/twitter/>
- [15] How to Use Advanced Twitter Search. *Twitter* [online]. 2012 [cit. 2013-01-01]. Dostupné z: <https://support.twitter.com/groups/31-twitter-basics/topics/110-search/articles/71577-how-to-use-advanced-twitter-search>
- [16] List of Twitter services and applications. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2011 [cit. 2013-01-01]. Dostupné z: http://en.wikipedia.org/wiki/List_of_Twitter_services_and_applications
- [17] Swabr. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2010 [cit. 2013-01-01]. Dostupné z: <http://en.wikipedia.org/wiki/Swabr>
- [18] Swabr-Facebook or Twitter for Companies. *Blograzzi* [online]. 2011 [cit. 2013-01-01]. Dostupné z: <http://www.blograzzi.net/swabr-facebook-or-twitter-for-companies.html>
- [19] *Status.net: Free and Open Source social software* [online]. [cit. 2013-01-01]. Dostupné z: <http://status.net/>
- [20] Yammer: Customer Success. *Yammer* [online]. [cit. 2013-01-01]. Dostupné z: <https://www.yammer.com/customers/>
- [21] BARKDULL, Jacob. 5 Ways Identi.ca is Better Than Twitter. *TildeHash* [online]. 2010 [cit. 2013-01-07]. Dostupné z: <http://tildehash.com/?article=5-ways-identi.ca-is-better-than-twitter>
- [22] *Swabr: Enterprise Microblogging. Made in Germany.* [online]. [cit. 2013-01-01]. Dostupné z: <http://www.swabr.com/>
- [23] E14N. *Identi.ca* [online]. [cit. 2013-01-01]. Dostupné z: <http://identi.ca/>
- [24] YAMMER. *Yammer: The Enterprise Social Network* [online]. [cit. 2013-01-01]. Dostupné z: <https://www.yammer.com/>
- [25] TWITTER. *Twitter* [online]. [cit. 2013-01-01]. Dostupné z: <https://twitter.com/>
- [26] MACDONALD, Matthew, Adam FREEMAN a Mario SZPUSZTA. *ASP.NET 4 a C# 2010: tvorba dynamických stránek profesionálně*. Vyd. 1. Překlad Jan Pokorný. Brno: Zoner Press, 2011, 880 s. Encyklopedie Zoner Press. ISBN 978-80-7413-131-8.
- [27] EVJEN, Bill, Scott HANSELMAN a Devin RADER. *Professional ASP.NET 4 in C# and VB*. Indianapolis, IN: Wiley Pub., c2010, lviii, 1477 p. ISBN 04-705-0220-7.

- [28] WALTERS, R. *Mistrovství v Microsoft SQL Server 2008: [kompletní průvodce databázového experta]*. Vyd. 1. Brno: Computer Press, 2009, 864 s. ISBN 978-80-251-2329-4.

Seznam příloh

Příloha 1. CD (obsahuje zdrojové kódy aplikace, skript pro vytvoření databáze)