

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

OSOBNÍ PLÁNOVAČ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN LUTONSKÝ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

OSOBNÍ PLÁNOVAČ

PERSONAL SCHEDULER

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

MARTIN LUTONSKÝ

VEDOUcí PRÁCE

SUPERVISOR

Ing. ŠÁRKA KVĚTOŇOVÁ, Ph.D.

BRNO 2011

Abstrakt

Cílem bakalářské práce je navrhnout a implementovat osobní plánovač pro správu a plánování osobních i pracovních záležitostí a aktivit. Plánovač bude umožňovat evidenci různých typů událostí, správu kontaktů, export a import uživatelských dat. Práce rozebírá a popisuje jednotlivé typy plánování času a základní plánovací techniky a metody. Srovnává a analyzuje funkčnost a použití dostupných plánovacích nástrojů a řešení. Na základě analýzy popisuje specifikaci a návrh nového informačního systému. Nedílnou součástí je také implementace a testování informačního systému s důrazem na jeho funkčnost, použitelnost a optimalizaci. Závěr obsahuje shrnutí a zhodnocení vytvořeného řešení, včetně možností budoucího rozšíření. Výsledkem je funkční a použitelný informační systém, implementovaný jako webová aplikace, umožňující plánování osobních a pracovních aktivit.

Abstract

The aim of the bachelor thesis is to design and implement personal scheduler for administration and planning of personal and business events and activities. The scheduler will provide administration of different types of events, contacts, export and import of user's data. The thesis deals with and describes particular types of time planning and basic planning techniques and methods. The text compares and analyses options and application of available planning tools and solutions. Base on analysis it describes specification and concept of new information system. An integral part is also implementation and testing of information system with stress laid on its utility, usability and optimalization. In conclusion the text summarizes and evaluates achieved solution including possibilities of future extension. The result of the thesis is functional and usable information system, implemented as web application which provides planning and administration of personal and business activities and tasks.

Klíčová slova

osobní plánovač, správa času, PHP, Nette Framework, MVP, JavaScript, optimalizace

Keywords

personal scheduler, time management, PHP, Nette Framework, MVP, JavaScript, optimization

Citace

Martin Lutonský: Osobní plánovač, bakalářská práce, Brno, FIT VUT v Brně, 2011

Osobní plánovač

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením paní Ing. Šárky Květoňové, Ph.D.

.....

Martin Lutonský

15. května 2011

Poděkování

Tímto bych chtěl poděkovat vedoucí práce, paní Ing. Šárce Květoňové Ph.D., za přínosné a velmi užitečné rady a postřehy při řešení bakalářské práce.

© Martin Lutonský, 2011.

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

1	Úvod	3
2	Plánování	5
2.1	Plánování času	5
2.2	Metody plánování času	6
2.2.1	Ganttův diagram	6
2.2.2	Metoda kritické cesty	7
2.2.3	Metoda PERT	7
2.2.4	GTD metoda	8
3	Plánovací nástroje	9
3.1	Ideální plánovač	10
3.2	Srovnání dostupných řešení	10
3.2.1	Osobní diář	10
3.2.2	Mobilní kalendář	11
3.2.3	Microsoft Outlook	11
3.2.4	WinOrganizer	12
3.2.5	Kalendář Google	12
3.3	Zhodnocení dostupných řešení	13
4	Specifikace a návrh plánovače	14
4.1	Požadavky	14
4.1.1	Účel a použití systému	14
4.1.2	Plánování	14
4.1.3	Kalendáře	14
4.1.4	Úkoly	15
4.1.5	Správa kontaktů	15
4.1.6	Bezpečnost	15
4.1.7	Export a import dat	15
4.1.8	Přístupnost	16
4.1.9	Lokalizace	16
4.2	Návrh informačního systému	16
4.2.1	Diagram případů užití	16
4.2.2	ER diagram	16
4.2.3	Databáze	16
4.3	Návrh uživatelského rozhraní	16
4.3.1	Rozvržení kalendáře	18
4.3.2	Rozvržení kontaktů	19

4.3.3	Rozvržení nastavení	20
5	Implementace	21
5.1	Použité technologie	21
5.1.1	HTML 5	21
5.1.2	CSS	21
5.1.3	JavaScript	21
5.1.4	Webový server Apache	22
5.1.5	PHP a Nette Framework	22
5.1.6	MySQL	23
5.2	Opakované události	24
5.3	Lokalizace	24
5.4	Import a export dat	25
5.5	Uživatelské rozhraní	25
5.5.1	Grafické prvky	25
5.5.2	Upozornění na události	25
5.5.3	Klávesové zkratky	26
5.6	Bezpečnost	26
5.6.1	Autentizace	26
5.6.2	Uživatelská data	26
5.7	Optimalizace	27
5.8	Testování	28
5.8.1	Funčnost na různých platformách	28
5.8.2	Uživatelské testování	28
6	Závěr	29
6.1	Možnosti rozšíření	30
	Seznam příloh	34
A	Obsah CD	35
B	Obrazovky systému	36

Kapitola 1

Úvod

V dnešní moderní a dynamické době jsou na každého člověka kladeny poměrně velké nároky na celkovou organizaci osobního a především pracovního života. Kvůli neustálému koloběhu každodenních událostí je nemožné si pamatovat všechny plánované aktivity a činnosti. A proto je efektivní plánování času za pomoci externích nástrojů jednou z nejdůležitějších činností každého pracujícího člověka či organizace.

„Čas jsou peníze“ v dnešním bouřlivě se vyvíjejícím ekonomickém světě platí dvojnásob a především v moderních kulturách je na tomto principu postaven veškerý obchod. Co nejlevněji a nejrychleji vyrobit poptávané zboží, aby byly nasyceny potřeby zákazníků. Neprodáváme tak jen výsledný produkt, ale i energii a čas, který byl do výroby investován. Pro ještě větší efektivitu toho procesu, a tedy i větší výsledné zisky, je proto nutné snižovat náklady spojené s výrobou těchto produktů. A právě efektivní plánování je jednou z možných cest, jak požadovaného snížení nákladů dosáhnout. Pracovní plán, stanovení milníků a termínů hraje důležitou roli v celkovém pracovním procesu, kde správné načasování a rozvržení jednotlivých kroků a postupů může často rozhodnout o úspěšném dokončení projektu.

Historie plánování je úzce spjata s vývojem počítačů. Nicméně již na přelomu 5. a 6. st. př. n. l. čínský vojenský generál, stratég a filozof Sun-c' popsal filozofii války, strategii boje a plánování ve svém rozsáhlém díle Umění války. Stavby pyramid, starých přes 3000 let, transkontinentálních železnic či náboženských chrámů, jako např. indický Tádž Mahal, trvaly desítky let. Žádná z těchto staveb nemohla být úspěšně dokončena bez určité formy plánování. Avšak plánování v moderní podobě, jak jej známe dnes, začalo být studováno a dále rozvíjeno až počátkem 20. století [30].

Na přelomu 19. a 20. století začal americký strojní inženýr Frederick Taylor s podrobnými studiemi práce. Díky svému vědeckému přístupu dokázal, že při rozdělení práce na elementární prvky ji lze lépe analyzovat a zefektivnit. Na Taylorovu práci navázal jeho společník Henry L. Gantt, který podrobně studoval řízení činností a pořadí jednotlivých operací. Jeho studie řízení byla zaměřena na stavbu námořních lodí během první světové války. Výsledkem studie byl návrh pruhového diagramu, skládajícího se z úkolů se značkami milníků, znázorňujících pořadí a dobu trvání všech úkolů a procesů projektu. Nejvíce používané se Ganttovy diagramy staly v 80. letech 20. století, a to především díky velkému rozšíření osobních počítačů [15].

V 50. letech 20. století Morgan R. Walker a James E. Kelley vyvinuli *metodu kritické cesty* (zkráceně CPM), která využívala grafové reprezentace všech činností projektu k nalezení nejdelší plánované doby trvání celého projektu. Předchůdce metody, dnes známé jako metoda kritické cesty, byl vytvořen a aplikován již v 40. letech během prací na projektu

Manhattan. Ačkoliv v původní formě není v současnosti často užívána, stala se metoda CPM základem pro mnohé další, dnes používané metody a techniky určené k analýze a plánování projektů [30, 31].

Roku 1957 byla publikována *metoda vyhodnocení a kontroly programu* (zkráceně PERT). Na rozdíl od metody CPM, PERT využívá tři časové charakteristiky pro vyjádření doby trvání každé z jednotlivých činností projektu. Konkrétně optimistický, nejpravděpodobnější a pesimistický odhad, díky čemuž je možné tuto metodu použít u akcí majících stochastickou povahu [33].

Díky rozmachu osobních počítačů počátkem 80. let 20. století došlo i k rozšíření osobních a podnikových plánovačů a systémů pro podporu řízení projektů. Vůbec prvním osobním plánovačem byl software Micro Planner určený pro počítače Apple II, který je pod názvem X-Pert vyvíjen dodnes. V dalších letech se osobní počítače díky jejich relativně nízké ceně staly hlavní platformou při plánování. Díky neustále rostoucímu výpočetnímu výkonu osobních počítačů vznikly stovky aplikací pro plánování, včetně komplexních systémů pro podporu řízení projektů jako jsou Open Plan, Microsoft Project či Primavera. Mnohé z nich jsou vyvíjeny a používány dodnes. V současné době můžeme najít na trhu široké množství plánovacích nástrojů a systémů určených pro různé oblasti plánování soukromých i firemních aktivit a projektů [30].

Bakalářská práce popisuje analýzu, návrh a implementaci informačního systému pro plánování osobních a pracovních aktivit a jiných záležitostí. Druhá kapitola bakalářské práce rozebírá jednotlivé typy plánování času a základní plánovací metody a techniky. Třetí kapitola popisuje dostupné plánovací nástroje a řešení. Definuje ideální plánovač, jakožto výchozí vzor pro porovnání vlastností jednotlivých řešení, jejich nedostatků a předností. Čtvrtá kapitola se věnuje analýze a návrhu vytvářené aplikace. Definuje požadavky na nový informační systém z pohledu funkčnosti, přístupnosti a zabezpečení. Pátá kapitola popisuje způsob implementace informačního systému, zaměřuje se také na testování funkčnosti a použitelnosti výsledné aplikace. Závěr obsahuje shrnutí a zhodnocení vytvořeného řešení, včetně možností budoucího rozšíření.

Kapitola 2

Plánování

Plánování je proces, během kterého na základě stanoveného cíle projektu¹ určujeme začátek a konec jednotlivých aktivit v závislosti na jejich době trvání, předchozích a následujících událostech a dostupných zdrojích [23].

Plánovací činnost je pro řízení většiny projektů rozhodující. Plány simulují vývoj projektu, jelikož písemnou či grafickou formou popisují hierarchickou strukturu činností nutných k dokončení projektu. Každý z plánů by měl při prvním pohledu zodpovědět co, kdo, kdy a především s jakým cílem má udělat [23]. Samotné vytvoření plánu je pak složeno z několika kroků. V první řadě je potřeba zaznamenat a ujasnit záměry, z nichž vyplývají cíle plánu. Následně je nutno tento plán vhodně časově rozvrhnout. Určení doby začátku a především trvání plánované aktivity může být věc velmi obtížná a vždy je vhodnější počítat s horší variantou, popřípadě si mezi jednotlivými aktivitami ponechat alespoň menší časovou rezervu pro případné další nečekané události. Každý z plánů, a to především v rámci krátkodobého plánování, by měl mít také svou prioritu. Ta může jednoduše rozdělit úkoly na důležité, jejichž včasné provedení je nutné, a na méně důležité, jejichž odklad či nesplnění není kritické. Takto specifikované úkoly poté můžeme zasadit do časového rozvrhu.

S rostoucím počtem plánovaných aktivit je pak kladen ještě větší důraz na přehledné a efektivní plánování. Je potřeba co nejpřesněji odhadnout časový interval trvání jednotlivých aktivit a jejich správné seřazení a zařazení dle aktuálních možností a priorit.

Efektivitu plánování můžeme měřit a porovnávat podle několika kritérií, například poměr vytyčených záměrů a dokončených cílů s ohledem na vynaložené náklady či jako rozdíl plánovaného a skutečného času, který byl potřeba k uskutečnění našich plánů.

2.1 Plánování času

Plánování času lze rozdělit podle několika hledisek, například podle intervalu stanovování plánu či podle počtu subjektů, které budou vytvořený plán vykonávat. V případě stanovování plánu pro jednoho člověka musíme brát ohled na jiné faktory, nežli pro skupinu čítající desítky či stovky osob. Podle intervalu lze plánování rozdělit na krátkodobé, střednědobé a dlouhodobé.

Krátkodobé, denní plánování se zaměřuje na tvorbu rozvrhu do velmi blízké budoucnosti, nejčastěji na následující den. Je potřeba především v případech, kdy je třeba za den

¹Projekt je jedinečný proces sestávající se z řady koordinovaných a řízených činností s daty zahájení a ukončení, prováděný pro dosažení cíle, který vyhovuje specifickým požadavkům, včetně omezení daných časem, náklady a zdroji [23].

stihnout splnit větší množství úkolů.

Střednědobé plánování se vztahuje na delší interval, nejčastěji na jednotlivé týdny nebo měsíce. Zde je zapotřebí především celkový pohled na budoucí naplánované události v širších souvislostech, aby bylo případně možné podle nich přizpůsobit krátkodobé, aktuální plány.

Dlouhodobé plánování v rámci jednoho až několika měsíců je vhodné především u delších projektů a činností, kde potřebujeme obecně stanovit přibližný průběh jejich jednotlivých fází. Jednotlivé fáze pak můžeme detailněji naplánovat v rámci střednědobého plánování. Přináší obecný pohled na delší časový interval se zanedbáním detailů a méně důležitých podrobností.

Lze uvažovat ještě dlouhodobější plánování – v horizontu několika let, například stanovení životních přání a cílů. Dlouhodobější plánování je však natolik nejisté a neurčité, že s konkrétním plánováním časových aktivit nemá mnoho společného a tudíž se jím nebudeme nadále zabývat.

2.2 Metody plánování času

Kromě klasického sestavování seznamů úkolů a milníků² existují také další metody, využívající nejčastěji grafické reprezentace událostí a činností, například formou časového diagramu či síťového grafu. Grafická reprezentace nabízí přehlednější a ucelený pohled na jednotlivé plánované činnosti, jejich trvání a spojitosti. Díky tomu může vedoucí projektu analyzovat stanovené plány a termíny, priorizovat konkrétní aktivity, popřípadě lépe navrhnout či reorganizovat plán vývoje projektu.

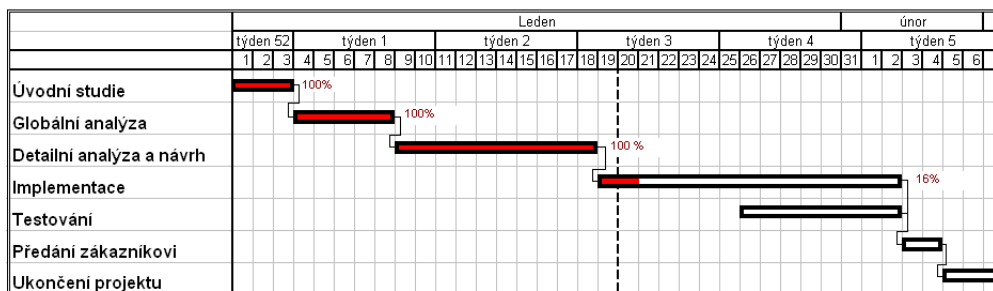
2.2.1 Ganttův diagram

Jedná se o druh pruhového diagramu, jenž byl navržen roku 1896 polským ekonomem Karolem Adamieckim. Zpopularizován byl však až ve 20. letech 20. století americkým strojním inženýrem H. L. Ganttem. Ganttův diagram se využívá pro grafické znázornění naplánování posloupnosti činností v čase. V pozdějších modifikacích obsahuje také vyznačení závislostí mezi činnostmi a míru jejich plnění [30, 31].

Na vertikální osu Ganttova diagramu se zaznamenávají jednotlivé činnosti, ze kterých se projekt skládá, a to vždy jedna činnost na jeden řádek. Horizontální osa tvoří odhadované časové období trvání projektu, rozdělené na stejně dlouhé časové úseky. Na ploše diagramu jsou jednotlivé činnosti označeny pruhy, jejichž levá hrana značí plánovaný začátek a pravá hrana plánovaný konec činnosti. Celková délka pruhu značí předpokládanou délku trvání činnosti. Pomocí postupného vyplňování pruhů je možné zobrazení míry dokončení jednotlivých činností.

Ganttův diagram je díky své jednoduchosti pro mnoho lidí na první pohled snadno srozumitelný. Je vhodný především pro malé projekty, které se vlezou na jednu stránku papíru, případně na jednu obrazovku, jelikož diagram sděluje relativně málo informací na jednotku využitých ploch. Pro projekty s více aktivitami může být nepraktický, jelikož projekty většího rozsahu jsou příliš složité pro efektivní prezentaci pomocí Ganttova diagramu. Dalším problémem je nemožnost reprezentace celkového rozsahu projektu, ani relativní velikosti jednotlivých činností a jejich náročnosti na dokončení. Pro manipulaci se proto používá specializovaný software (např. Microsoft Project), který tyto problémy do určité

²Milník je úkol s nulovou dobou trvání (nula dnů), který slouží k identifikaci významných událostí v plánu, například dokončení určité hlavní fáze nebo započetí další [16].



Obrázek 2.1: Ukázka Ganttova diagramu s vyznačenými závislostmi činností a mírou dokončení [32]

míry kompenzuje formou přidávání dodatečných informací o jednotlivých činnostech, jako jsou např. potřebné zdroje, náklady, atd.

2.2.2 Metoda kritické cesty

Metoda kritické cesty (angl. *critical path method*, zkráceno CPM) je matematický algoritmus plánování průběhu množiny činností projektu. Metoda CPM byla vyvinuta ve 40. letech 20. století a publikována roku 1959, jako výsledek spolupráce Morgana Walkera ze společnosti DuPont a Jamese Kelleyho ze společnosti Remington Rand. Metodu lze obecně aplikovat na plánování jakéhokoliv projektu se vzájemně provázanými a závislými činnostmi [30].

Princip metody spočívá v sestrojení síťového grafu a nalezení kritické cesty – nejdelší cesty z počátečního vrcholu do vrcholu koncového [21]. Uzly grafu představují milníky projektu, hrany činnost nutnou k dosažení dalšího milníku a ohodnocení hrany odhadovanou dobu potřebnou k vykonání činnosti. Součet ohodnocení hran na kritické cestě pak určuje nejdelší potřebnou dobu k dokončení celého projektu.

Metoda umožňuje rozlišit, které aktivity projektu jsou či nejsou kritické. Kritické aktivity jsou takové aktivity, jejichž nesplnění v plánovaném termínu bude mít za následek zpoždění celého projektu [22].

2.2.3 Metoda PERT

Metoda vyhodnocení a kontroly programu (angl. *program evaluation and review technique*, zkráceně PERT) je zobecnění metody kritické cesty. Metoda byla vyvinuta roku 1957 pro zjednodušení plánování a řízení rozsáhlých projektů pro potřeby námořnictva USA. Na rozdíl od metody CPM, PERT umožňuje u jednotlivých dílčích činností určit očekávanou dobu dané činnosti, jejichž trvání má stochastickou povahu. Očekávanou dobu T_e lze vypočítat pomocí empirického vztahu

$$T_e = \frac{O + 4M + P}{6} \quad (2.1)$$

kde O značí optimistický odhad, M nejpravděpodobnější odhad a P pesimistický odhad doby trvání konkrétní činnosti [23]. Jsou-li známy či odhadnuty parametry všech činností projektu, lze zjistit pomocí algoritmu CPM kritickou cestu projektu, stejně tak i nejpravděpodobnější a nejkratší očekávanou dobu trvání projektu. Umožňuje nám také spočítat pravděpodobnost, že projekt bude dokončen v určitém čase.

2.2.4 GTD metoda

Getting Things Done (zkráceně GTD) patří mezi moderní, organizační a plánovací metody. Metoda byla navržena a popsána Davidem Allenem ve stejnojmenné knize. Princip metody spočívá ve vytváření seznamů různých druhů, které poté slouží jako připomenutí plánovaných aktivit. Člověk se tak může přímo zaměřit na plnění úkolů, namísto vzpomínání, co je potřeba udělat. Cílem je tedy vytvoření si důvěryhodného, externího systému, který umožní plně se soustředit na jednu danou aktivitu. Na rozdíl od klasického plánování, GTD metoda klade velký důraz na zpracování, prioritní organizování plánů a jejich následné i průběžné hodnocení.

Metoda definuje pět fází pracovního procesu [1]:

- Shromáždění veškerých soukromých i pracovních informací, úkolů, požadavků, plánů, předsevzetí a záměrů.
- Následuje jejich roztřídění do skupin podle proveditelnosti. Neproveditelné jsou pak dle potřeby odloženy na později či zahozeny.
- Zařazení proveditelných aktivit do seznamů podle jejich typu, např. seznam pro události s přesně stanoveným termínem, seznam pro dlouhodobější plány, jejichž úplné dokončení není jednorázová akce, nebo seznam pro aktivity, u nichž v daném okamžiku nelze rozhodnout, zdali je budeme chtít v budoucnu realizovat.
- Zhodnocení námi sestavených plánů a seznamů.
- Plnění naplánovaných aktivit. Plnění je založeno na prioritním systému uspořádání úkolů podle aktuálního kontextu, ve kterém se člověk nachází, času, který má k dispozici a prioritě úkolu.

Kapitola 3

Plánovací nástroje

Plánovacích nástrojů můžeme na trhu nalézt celou řadu. Liší se především v možnostech a funkcích, které uživatelům nabízejí. Plánovací nástroje můžeme rozdělit do dvou hlavních kategorií – interaktivní a statické. Mezi statické se řadí například osobní diář či veškeré ostatní formy papírových poznámek. Jejich používání je velmi jednoduché a nevyžaduje žádné jiné speciální znalosti a dovednosti. Také proto je tato forma plánování tak oblíbená a často používaná.

Interaktivní plánovací nástroje jsou nejčastěji počítačové a mobilní aplikace. K jejich ovládání jsou již zapotřebí alespoň základní znalosti ovládání dané aplikace a zařízení, na kterém jsou provozovány. Interaktivní plánovací nástroje můžeme dále rozčlenit do skupin podle typu zařízení, pro něhož jsou určeny. Dvěma největšími skupinami v této kategorii jsou plánovače určené pro osobní počítače a plánovače pro mobilní zařízení, jako jsou například mobilní telefony nebo PDA zařízení. Obě skupiny nástrojů se liší množstvím funkcí, které nabízejí a především pak ergonomií a efektivitou jejich používání. Počítačové aplikace obvykle nabízejí vyšší komfort obsluhy a také větší množství funkcí než aplikace pro mobilní zařízení. Je to dáno především rozdílným výpočetním výkonem jednotlivých platforem, který je potřeba pro realizaci složitějších operací nad uživatelskými daty.

Plánovací aplikace určené pro osobní počítače můžeme rozdělit do dalších dvou skupin – webové aplikace, přístupné prostřednictvím internetového prohlížeče a klasické (desktop) aplikace. Webové aplikace mají tu výhodu, že jejich obsah je kdykoliv přístupný z webového prohlížeče, a to třeba i prostřednictvím prohlížeče mobilního telefonu. Klasické počítačové aplikace tuto možnost často suplují některou ze serverových synchronizačních technologií a nástrojů, jmenovitě například Microsoft Exchange Server či Google Sync. U webových aplikací se můžeme také často setkat s veřejným API (*Application Programming Interface*), které umožňuje zpřístupnit svá data dalším aplikacím a službám. Co se použitelnosti a funkcí týče, moderní webové technologie dospěly do takového stádia, že rozdíly mezi webovými a klasickými aplikacemi jsou téměř smazány. S dostatečně rychlým internetovým připojením nabízejí webové aplikace téměř srovnatelný komfort práce jako s klasickou aplikací.

Mezi nepoužívanější webové aplikace patří nástroje od společností Google a Microsoft – Kalendář Google a Microsoft Live Calendar. Obě tyto aplikace nabízejí velmi podobné funkce, způsob ovládání, prezentaci dat i možnosti jejich sdílení a synchronizace s aplikacemi třetích stran.

Nejpoužívanějšími programy z kategorie klasických počítačových plánovačů jsou především Microsoft Outlook a obdobná řešení od konkurenčních firem, jako například IBM Lotus Notes či Mozilla Thunderbird spolu s rozšířením Lightning. Jedná se o komplexní

aplikace, které kromě plánovacího kalendáře obsahují také e-mailového klienta, adresář kontaktů a mnoho dalších funkcí, často dostupných také prostřednictvím rozšíření a doplňků třetích stran. Do stejné kategorie náleží i aplikace, které slouží jen a pouze jako plánovače a správci úkolů, jako například WinOrganizer. Tyto programy většinou nabízejí jednodušší a přehlednější uživatelské rozhraní se základní sadou funkcí a možnostmi plánování a správy kontaktů. Avšak především s nástupem internetu se začaly přesouvat do pozadí zájmu uživatelů, neboť se začaly prosazovat aplikace integrující i e-mailového klienta a některé další pokročilé funkce.

3.1 Ideální plánovač

Na základě předchozí kategorizace a popisu existujících plánovacích nástrojů bude nyní popsán ideální osobní plánovač, jeho vlastnosti a funkce, které by měl uživateli nabízet.

Ideální osobní plánovač by měl být pokud možno co nejjednodušší a uživatelsky přívětivý. Práce s ním by měla být pohodlná, příjemná a intuitivní, s důrazem na jeho bezproblémovou dostupnost na co možná největším počtu zařízení. Aplikace by uživateli umožňovala jednoduše spravovat a plánovat své osobní i pracovní aktivity v rámci jednoho či více kalendářů. Taktéž by umožňovala jednoduchou správu kontaktů formou adresáře, v němž by bylo možné jednotlivé kontakty třídit do libovolného množství skupin.

Pro zajištění co největší funkčnosti a použitelnosti by výsledná aplikace měla brát ohled na zařízení, na kterém je provozována. Měla by být dobře a jednoduše ovladatelná na klasickém osobním počítači s velkým monitorem, stejně jako na malé zařízení s omezeným rozlišením displeje, například malém notebooku či PDA zařízení. Hlavní funkčnost by však i přes diametrálně rozdílná zařízení měla zůstat vždy zachována. Samozřejmostí by také měla být podpora co nejširšího množství formátů souborů pro import a export uživatelských dat, aby byla zajištěna možnost spolupráce s ostatními aplikacemi. Mezi podporované formáty by měl zcela jistě patřit univerzální textový formát pro reprezentaci kalendářních dat iCalendar [4] a také některý z univerzálních textových formátů pro výměnu tabulkových a strukturovaných dat, jako jsou například XML (*Extensible Markup Language*) či CSV (*Comma-separated values*).

3.2 Srovnání dostupných řešení

3.2.1 Osobní diář

Plánování za pomoci osobního diáře je jednou z nejstarších a dodnes nejpoužívanějších forem. Klasický osobní diář obsahuje kromě denního kalendáře pro každý den velmi často také abecedně řazený adresář kontaktů.

Nespornou výhodou osobního diáře je jeho jednoduchost a přenositelnost. Pro použití není nutné si pořizovat mobilní telefon či počítač a učit se dovednostem spojeným s jejich používáním. Avšak jeho největší nevýhoda vychází již z primárního účelu, ke kterému je osobní diář určen – k plánování a správě času jedné osoby. Kromě opisu událostí přímo z diáře zde není žádná jiná možnost, jak uživatelská data automaticky sdílet s jinými osobami či je jakkoliv jinak publikovat.

3.2.2 Mobilní kalendář

Většina dnešních mobilních telefonů obsahuje ve své základní výbavě plánovací kalendář. Typický mobilní kalendář je jednoduchá, relativně přehledná aplikace, nejčastěji určená pro plánování v rámci jednoho osobního kalendáře. Nabízí základní plánování událostí na konkrétní datum a čas, s možností volby priority a intervalu opakování. Většinou umožňuje také alespoň částečně propojení konkrétních událostí či výročí s kontakty v telefonním seznamu.

Výhodou mobilního kalendáře je především jeho okamžitá dostupnost. Mobilní zařízení se v posledních letech staly nedílnou součástí každodenního života, tudíž jejich uživatelé mohou prakticky kdykoliv spravovat své plánované aktivity jejich prostřednictvím. Chytré mobilní telefony a PDA zařízení často také nabízejí synchronizaci kontaktů a kalendáře s některými z počítačových nebo internetových aplikací. Synchronizace se nejčastěji provádí prostřednictvím pevného připojení mobilního telefonu s počítačem, popřípadě za pomoci mobilního internetového připojení a proprietárních synchronizačních technologií, např. Microsoft Exchange ActiveSync či Google Sync.

Jsou však také situace, kdy použití mobilního telefonu není možné. Může to být způsobeno omezenou dobou výdrže baterie mobilního telefonu nebo například i pravidly etikety. Dalším negativem je také nepříliš velký komfort ovládání mobilních aplikací, vzhledem k rozlišení displeje a ergonomii klávesnice, které mobilní telefony mají.

3.2.3 Microsoft Outlook

Microsoft Outlook, aktuálně ve verzi 2010, je e-mailový klientský software od společnosti Microsoft, který je standardně součástí základního balíku Microsoft Office. I přestože se jedná primárně o klienta elektronické pošty, nabízí také řadu dalších pokročilých funkcí, jako například vedení a správu kontaktů, organizaci úkolů, poznámek a také času. Prostřednictvím doplňků třetích stran je možné jej rozšířit o další funkce, například o IP telefonii, propojení se sociálními sítěmi, širší možnosti zabezpečení nebo synchronizace. Jelikož se jedná o jednu z několika aplikací softwarového balíku Microsoft Office, je zajištěna dobrá spolupráce mezi jednotlivými součástmi. Software umožňuje spravovat, prezentovat a prohledávat několik kalendářů zároveň. Podporuje také standardní správu úkolů a kontaktů, které je možné spolu s kalendáři rozesílat a přijímat za pomoci integrovaného e-mailového klienta.

Každý z kalendářů je možné publikovat na internetu prostřednictvím serveru Office.com či jiného serveru využívající technologii *WebDAV*¹. Uživatelská data je také možné synchronizovat prostřednictvím technologie Microsoft Exchange ActiveSync. Pro vzdělávací účely je k dispozici i testovací on-line verze aplikace s názvem Microsoft Outlook Live. Aplikaci je možné provozovat na platformách Windows a Mac OS X.

Aplikace nabízí relativně přehledné uživatelské rozhraní. Pro některé uživatele však může být nezvyklá hlavní kontextová nabídka, tzv. „pás karet“, v originále pojmenovaný jako „Ribbon“. Hlavní nabídka mění svůj obsah v závislosti na aktuálně používané sekci a funkci aplikace, čímž by měla uživateli usnadnit ovládání aplikace. Praxe však může být odlišná a především pro uživatele zvyklé na klasickou nabídku může být kontextově závislá nabídka matoucí.

Další nevýhodou tohoto řešení může být cena, která se pohybuje dle verze software

¹WebDAV je sada rozšíření HTTP protokolu, které umožňují kooperaci více uživatelů při správě dokumentů a souborů uložených na webovém serveru [6].

a typu licence v rozmezí 3–9 tisíc Kč. Dalším nepříjemným jevem je relativně dlouhá doba prvního startu aplikace, která se s větším množstvím uživatelských dat (především uložených e-mailů) ještě prodlužuje [14].

3.2.4 WinOrganizer

WinOrganizer je jednoduchý správce úkolů, poznámek, kalendářů, kontaktů a hesel vyvíjený společností TGS Labs, dostupný ve více než 30 jazycích včetně češtiny. Aplikace nabízí základní sadu nejdůležitějších funkcí pro správu času, složek, dokumentů, kalendářních skupin, událostí do nich spadajících a velmi propracovanou správu kontaktů. U každého z kontaktů je možné uchovávat přes 70 různých přednastavených údajů a nechybí ani možnost přidat položky vlastní. Kalendářní skupiny, kalendáře, úkoly, deníky a dokumenty je možné libovolně řadit do složek, které lze do sebe neomezeně zanořovat. Aplikace ihned po instalaci nabízí několik předem připravených kalendářů, obsahujících velké množství událostí z oblastí historie hudby, počítačů a výročí slavných osobností.

Grafické rozhraní působí jednoduše a přehledně. Aplikace nabízí veškeré základní funkce, které by měl osobní plánovač obsahovat. Bonusem je tvorba a správa dokumentů přímo v aplikaci. Software ale nenabízí žádné možnosti doplnění o další funkce, například prostřednictvím doplňků či rozšíření.

Program je placeným produktem. Jeho cena ovšem není v poměru k nabízeným funkcím příliš výhodná – cena osobní licence se pohybuje kolem 1000 Kč. Další problémem je nemožnost synchronizace dat prostřednictvím internetu či s mobilními zařízeními. Aplikace nabízí pouze ruční import a export uživatelských dat ve formátech TXT, RTF (pouze poznámky), XML, CSV a proprietárního GSO.

3.2.5 Kalendář Google

Jedná se o internetovou, bezplatnou aplikaci vyvíjenou společností Google. Nabízí jednoduché a přehledné uživatelské rozhraní využívající technologii *drag-and-drop* a AJAX pro ovládání a manipulaci s objekty aplikace. Kalendář Google umožňuje v jednom pohledu najednou zobrazit události z vybraných kalendářů. Události jsou pak odlišeny barvou kalendáře, kterému náleží. Součástí jsou i předdefinované kalendáře, např. státní svátky, kalendáře sportovních utkání či fáze Měsíce pro jednotlivé dny, které si uživatel může volitelně zobrazit.

Kalendáře je možné sdílet s ostatními registrovanými uživateli aplikace i je veřejně publikovat ostatním uživatelům internetu. Veřejně publikovaný kalendář je pak dostupný přes adresu, která je vygenerována správcem kalendáře po publikování. Kalendáře je možné exportovat pouze ve formátu iCalendar, pro import je navíc možné použít i formát CSV dle specifikace Microsoft Outlook.

Veškerá uživatelská data jsou ukládána na serverech, díky čemuž jsou uživateli po autentizaci dostupná odkudkoliv, ovšem pouze za předpokladu připojení k internetu. Po instalaci zásuvného modulu Google Gears do webového prohlížeče je možné aplikaci s omezeními provozovat i v režimu bez internetového připojení [9].

Aplikace sama o sobě neobsahuje správu kontaktů. Je však dostupná v rámci uživatelského účtu Google prostřednictvím poštovního klienta GMail. Pokud však uživatel požaduje vyhledat určitý kontakt, musí přejít na stránku aplikace GMail, kde může kontakty spravovat. Pokud v aplikaci Google Kalendář uživatel zadává e-mailovou adresu osoby, se kterou chce vytvářenou událost sdílet, je mu automaticky nabídnut seznam podobných e-mailů osob uložených v jeho adresáři v rámci účtu Google.

Pro načtení kompentí aplikace je při prvotním startu potřeba stáhnout přibližně 1,2 MB dat, což může být při použití pomalého připojení k internetu velmi omezující faktor. Díky technologii AJAX jsou však veškeré další datové přenosy omezeny na minimum, díky čemuž je práce s aplikací již relativně rychlá. Funkčnost aplikace je úzce svázána s využitím technologie JavaScript, jelikož většina vykreslovaných dat je posílána právě ve formátu JavaScript a transformována do HTML kódu až v prohlížeči klienta. Proto není možné aplikaci provozovat v prohlížeči s vypnutým JavaScriptem. Neexistuje ani HTML verze pro zařízení a prohlížeče bez podpory této technologie. Z mobilního telefonu je dostupná funkčně omezená verze aplikace, umožňující plánované události pouze prohlížet, nikoliv spravovat. K aplikaci je možné přistupovat také prostřednictvím rozhraní *Calendar Data API* dostupného v programovacích jazycích .NET, Java a Python [8].

3.3 Zhodnocení dostupných řešení

Z vybraných řešení ani jedno řešení úplně nesplňuje požadavky a vlastnosti ideálního plánovače. Všechny srovnávané počítačové řešení nabízejí přibližně stejné základní plánovací funkce, tj. plánování časových i celodenních událostí na konkrétní datum a čas, s širokými možnostmi opakování, volbou priority a dalších atributů. Obsahují také správu kontaktů a úkolů, přehledné a intuitivní uživatelské rozhraní. Ideálnímu plánovači se nejvíce podobá řešení od společnosti Google, jehož největšími nedostatky jsou náročnost na objem přenášených dat při prvním načtení aplikace a nefunkčnost při vypnutém JavaScriptu. Avšak právě silná vazba na JavaScript dělá tuto aplikaci velmi rychlou a uživatelsky přívětivou. Obsahuje také správce kontaktů, který však není přímo integrovaný do aplikace, nýbrž je součástí služby GMail, na kterou se uživatel musí explicitně přepnout.

Nástroje Kalendář Google i Microsoft Outlook umožňují relativně jednoduchou cestou synchronizovat uživatelská data s jinými zařízeními a především sítí internet. Co se počtu funkcí týče, nabízí aplikace WinOrganizer nejmenší funkční výbavu, avšak především kvůli jednostrannému zaměření aplikace. Velkou výhodou softwaru WinOrganizer oproti ostatním řešením je rychlý start, a to i při velkém množství spravovaných uživatelských dat.

Pokud srovnáme cenu, je nejvýhodnějším řešením webová aplikace od společnosti Google, která je dostupná zcela zdarma. Nejdražším softwarem je pak řešení od firmy Microsoft, avšak jelikož je nabízené řešení zároveň součástí sady Office, obsahuje nejen plánovač či e-mailového klienta, ale také například textový či tabulkový editor. Jeho další výhodou oproti ostatním aplikacím je také možnost rozšíření funkcionality aplikace prostřednictvím doplňků. Díky doplňkům je možné software lépe přizpůsobit potřebám uživatele. Kalendář Google také podporuje rozšíření formou tzv. „widgetů“, což jsou nejčastěji drobné doplňky napsané v jazyce JavaScript a využívající Calendar API [8]. Zdaleka však nenabízejí takové možnosti rozšíření a navýšení funkcionality, jako rozšíření pro aplikaci Microsoft Outlook.

Software WinOrganizer je oproti nástroji Microsoft Outlook mnohem levnější, avšak nemožnost rozšíření či on-line synchronizace cenový rozdíl snižuje na minimum. Silnou stránkou aplikace WinOrganizer však zůstává fakt, že ze všech tří počítačových aplikací nabízí klasické uživatelské rozhraní, bez přebytných grafických efektů. Grafické rozhraní je podobné ostatním aplikacím pro Windows. Dá se tedy předpokládat, že především pro uživatele zvyklé pracovat s operačním systémem Windows bude práce s aplikací snazší, jelikož je zachováno standardní rozložení a vzhled ovládacích prvků.

Kapitola 4

Specifikace a návrh plánovače

4.1 Požadavky

4.1.1 Účel a použití systému

Informační systém umožní prostřednictvím internetového prohlížeče efektivní plánování a správu časových aktivit formou vytváření událostí a úkolů, jejich tříděním a vyhledáváním. Systém uživateli umožní také správu osobních kontaktů, jejich řazení a třídění do skupin. Do informačního systému budou mít možnost vstupovat pouze registrovaní uživatelé po úspěšné autentizaci.

4.1.2 Plánování

Systém nabídne plánování aktivit na konkrétní datum, se specifikací délky jejich trvání a s možností opakování vybrané aktivity, počínaje vybraným datem v různých časových intervalech (např. denně, ve vybraných dnech, týdně, apod.). Systém bude rozlišovat mezi dvěma základními typy aktivit – celodenní a časové. Celodenní aktivity jsou vztaženy na konkrétní časový úsek ohraničený počátečním a koncovým datem. Interval trvání časových aktivit je navíc určen konkrétním časem začátku a konce.

U aktivit bude také možné specifikovat jejich další atributy – místo konání, interval a konec opakování a detailnější popis události, využitelný například pro uživatelské poznámky. U každé aktivity bude uživatel také moci určit prioritu, která bude následně použita při vizualizaci události v kalendáři. Uživatel bude mít možnost naplánované aktivity kdykoliv smazat. Bude-li se jednat o opakovanou aktivitu, bude možnost odstranit veškeré výskyty, všechny budoucí nebo pouze jeden konkrétní výskyt. Aplikace bude umožňovat několik odlišných pohledů na plánované aktivity. Nabídne denní, týdenní, měsíční a roční pohled na vybraný časový úsek a události v něm plánované. Součástí aplikace bude také agenda, neboli výhled na události plánované v nejbližší době.

4.1.3 Kalendáře

Uživatel systému bude mít možnost spravovat jeden až několik kalendářů. Každá z událostí bude náležet právě do jednoho z kalendářů. Kalendář a události do něj zasazené bude možné společně barevně či jinak graficky odlišit od ostatních.

Mezi uživateli informačního systému bude možné kalendáře také sdílet, nebo naopak toto sdílení kdykoliv zrušit. Právo spravovat kalendář a události do něj náležící bude mít však jen jeho vlastník. Ostatní uživatelé do něj budou moci jen nahlížet. Iniciovat sdílení

kalendáře bude také vždy jen jeho vlastník, a to zadáním e-mailu osoby, se kterou bude chtít zvolený kalendář sdílet. Příjemce nabídky bude následně upozorněn a v případě souhlasu bude kalendář zařazen do jeho vlastních kalendářů. Avšak pouze za předpokladu, že bude registrovaným uživatelem systému. Součástí e-mailu s nabídkou sdílení kalendáře tedy musí být i informace o registraci do informačního systému.

4.1.4 Úkoly

Informační systém umožní uživateli spravovat také jednoduché úkoly – události nevázané na konkrétní čas ani datum. Uživatel bude mít možnost úkoly spravovat formou jednoduché nabídky, ve které bude možné úkoly jednoduše přidávat, editovat a označovat za splněné.

4.1.5 Správa kontaktů

Správa kontaktů poskytne jednoduchý nástroj pro ukládání a editaci osobních kontaktů. Uživatel bude moci kontakty jednoduše zařazovat do jedné či více kontaktních skupin. Kromě názvu bude možné graficky odlišit jednotlivé kontaktní skupiny také pomocí barvy. Skupiny bude možné libovolně filtrovat – tj. zobrazovat pouze kontakty spadající do vybraných kontaktních skupin. Filtrovaný seznam kontaktů bude možné také řadit, minimálně podle jména a e-mailu.

U každého z kontaktů bude možné uchovávat základní údaje, jako například jméno, e-mail, adresu, telefonní čísla či volitelnou poznámku.

Seznam kontaktů bude možné využít také v případě vytváření nové události v kalendáři, kdy uživatel systému bude mít možnost volitelně si vybrat kontakty ze svého kontaktního seznamu, jimž chce zaslat pozvánku na nově vytvářenou událost.

4.1.6 Bezpečnost

Pro přístup do informačního systému musí uživatel zadat platnou kombinaci přihlašovacího jména a hesla. Hesla uživatelů musí být uložena v transformované podobě, za použití kryptografické hashovací funkce (např. MD5 či SHA-1). Dále je nutné zajistit validaci uživatelských vstupů, a to jak na straně klienta (webový prohlížeč), například pomocí JavaScriptu, tak na straně serveru. Na straně serveru je také nutné ověřovat a vhodně ošetřovat korektnost URL parametrů a v případě neplatného či podvrženého parametru na tuto skutečnost vhodně reagovat – například přesměrováním na úvodní stránku či upozorněním na neexistenci stránky.

4.1.7 Export a import dat

Pro zajištění co nejlepších možností sdílení uživatelských dat nabídne informační systém uživateli možnost importovat a exportovat vlastní kalendáře, a to za použití textových formátů XML a iCalendar. Umožní mu také importování a exportování kontaktních skupin a kontaktů ve formátu XML. Možnost zpracovat soubory ve formátu XML a iCalendar byla zvolena z důvodu velmi dobré podpory těchto formátů v nejpoužívanějších aplikacích určených pro správu kontaktů a plánování času (např. Microsoft Outlook, Mozilla Thunderbird či Kalendář Google) a také z důvodu jejich případné další transformace do jiných formátů.

4.1.8 Přístupnost

Informační systém bude dodržovat základní pravidla přístupnosti webových stránek [11]. Základní ovládání aplikace bude dostupné i prostřednictvím klávesových zkratk. Aplikace by měla být funkční i při vypnuté či nedostupné technologii JavaScript (viz 5.1.3), která bude v systému sloužit především k validaci uživatelských vstupů, ale také k AJAXovému načítání dat. Aplikace bude funkční ve všech nejpoužívanějších prohlížečích [7] (tj. Internet Explorer 8, Mozilla Firefox 3.6, Google Chrome 10). Musí být také zachována funkčnost aplikace ve starších, avšak stále relativně hojně rozšířených verzích internetových prohlížečů, především Internet Explorer 7.

4.1.9 Lokalizace

Informační systém bude v základní verzi lokalizován v anglickém jazyce. Lokalizace zahrnuje překlad všech textů, popisků, chybových a jiných hlášení. Musí však existovat možnost, jak jednoduchou a jednotnou cestou dodat další překlady. Jako nejvhodnější se jeví textový soubor či skript obsahující překlady jednotlivých řetězců.

4.2 Návrh informačního systému

4.2.1 Diagram případů užití

Na základě požadavků byl vytvořen diagram případů užití, který je znázorněn na obrázku 4.1, identifikující interakce probíhající mezi aktéry a informačním systémem. Z obrázku můžeme identifikovat celkem 2 aktéry – uživatele systému a čas.

4.2.2 ER diagram

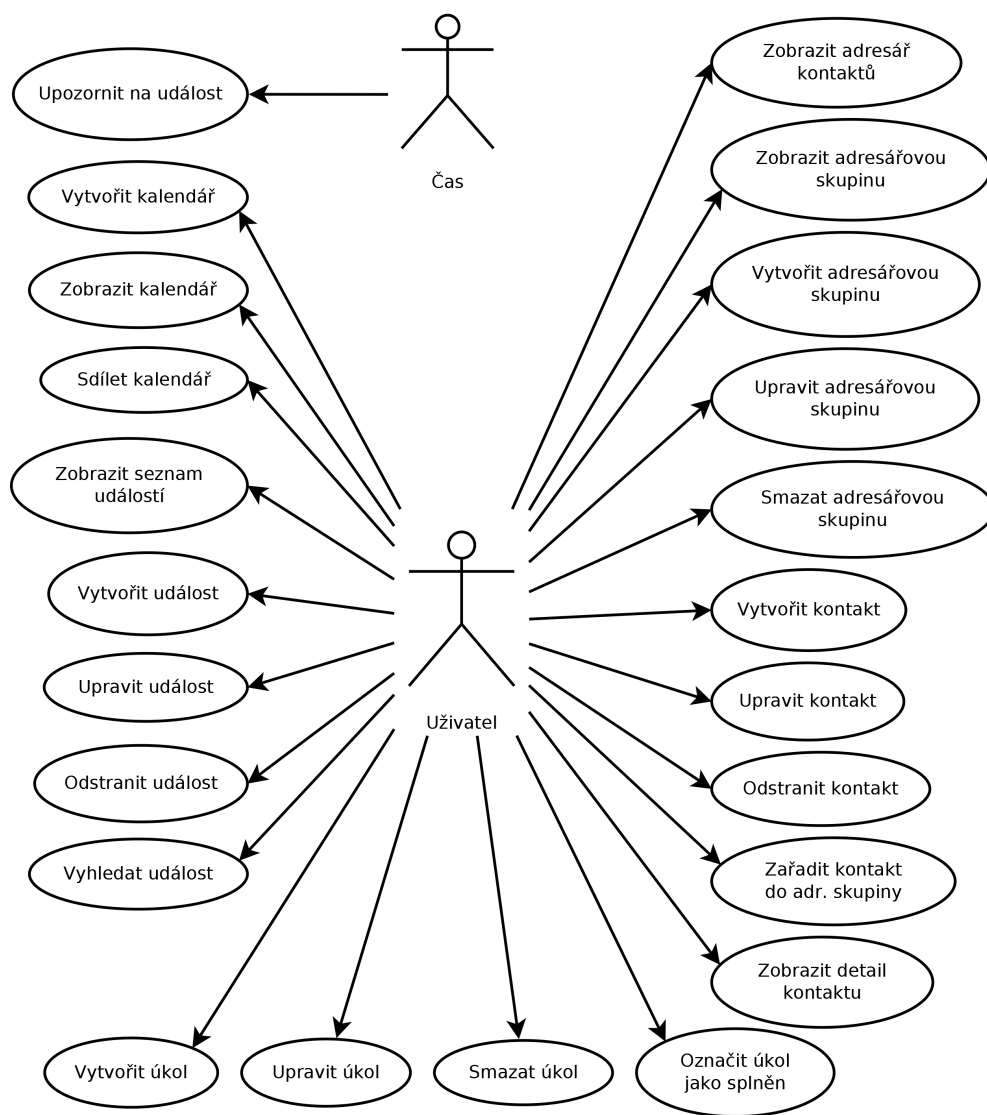
Na základě diagramu případů užití byl vytvořen Entity-Relationship diagram jazyka UML, který je zobrazen na obrázku 4.2. V diagramu můžeme identifikovat celkem 6 entitních množin a jejich atributy. Nejdůležitější částí je uživatel, na kterém závisí další tři entitní množiny (kalendář, adresářová skupina a úkol), které může uživatel spravovat či organizovat, a se kterými je v relaci 1:N. Množina uživatel je navíc v relaci N:M s množinou kalendář, vyjadřující možnost uživatele sdílet několik svých kalendářů s několika dalšími uživateli.

4.2.3 Databáze

Transformací ER diagramu na databázový model vznikla databáze znázorněna na obrázku 4.3. Kromě databázových tabulek byly ještě definovány dva pohledy pro snadnější a přehlednější interpretaci dat, především nad tabulkami v relacích N:M. Databáze se celkově skládá z deseti tabulek a dvou pohledů. Pro účely systému byla navíc oproti ER diagramu vytvořena ještě jedna tabulka pro uchovávání nabídek na sdílení kalendářů, jelikož uživatel má možnost sdílet kalendář i s dosud neregistrovanými uživateli.

4.3 Návrh uživatelského rozhraní

Základní struktura navrhnutého uživatelského grafického rozhraní se skládá ze tří logických částí – hlavní lišta navigace, ovládací prvky vybrané sekce a samotný obsah sekce, který

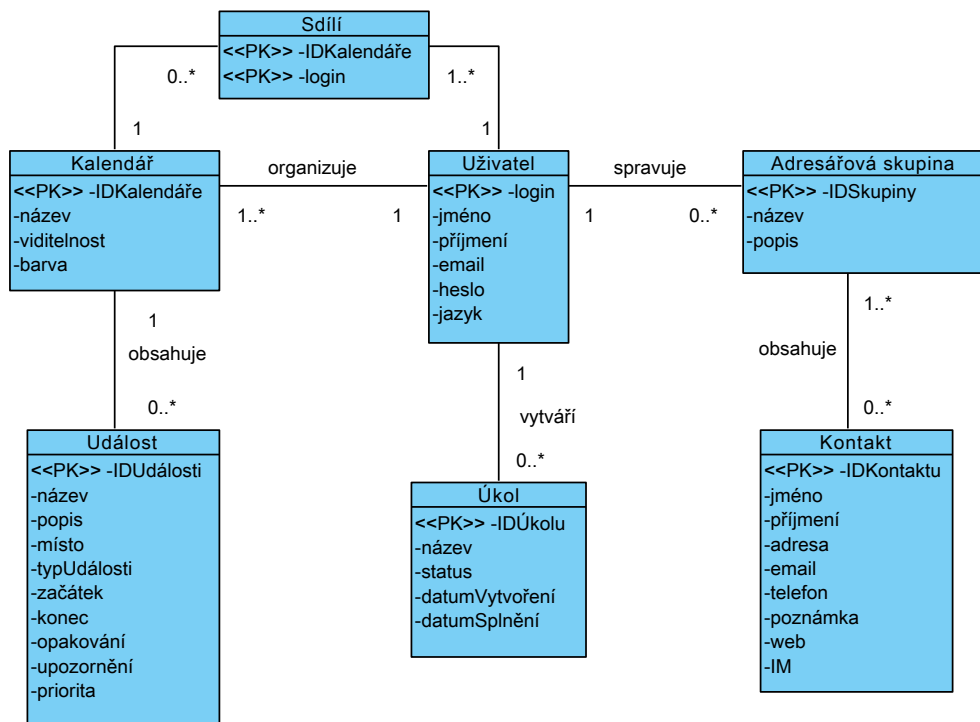


Obrázek 4.1: Diagram případů užití informačního systému

může být ještě dále rozdělen.

Uživatelské rozhraní a ovládání informačního systému bylo navrženo s ohledem na maximální přehlednost a použitelnost. Proto bylo uživatelské rozhraní rozděleno na celkem tři základní sekce - *kalendář*, *kontakty* a *nastavení*.

- Sekce *kalendář* nabízí 4 základní pohledy na uživatelské kalendáře (denní, týdenní, měsíční, roční) a plánované aktivity ve vybraném časovém horizontu formou grafického kalendáře nebo textovým seznamem událostí.
- Sekce *kontakty* umožňuje jednoduchou správu, filtrování a třídění uživatelských kontaktů do skupin.
- V sekci *nastavení* lze spravovat uživatelské údaje, možnosti sdílení kalendářů, importovat a exportovat uživatelská data.



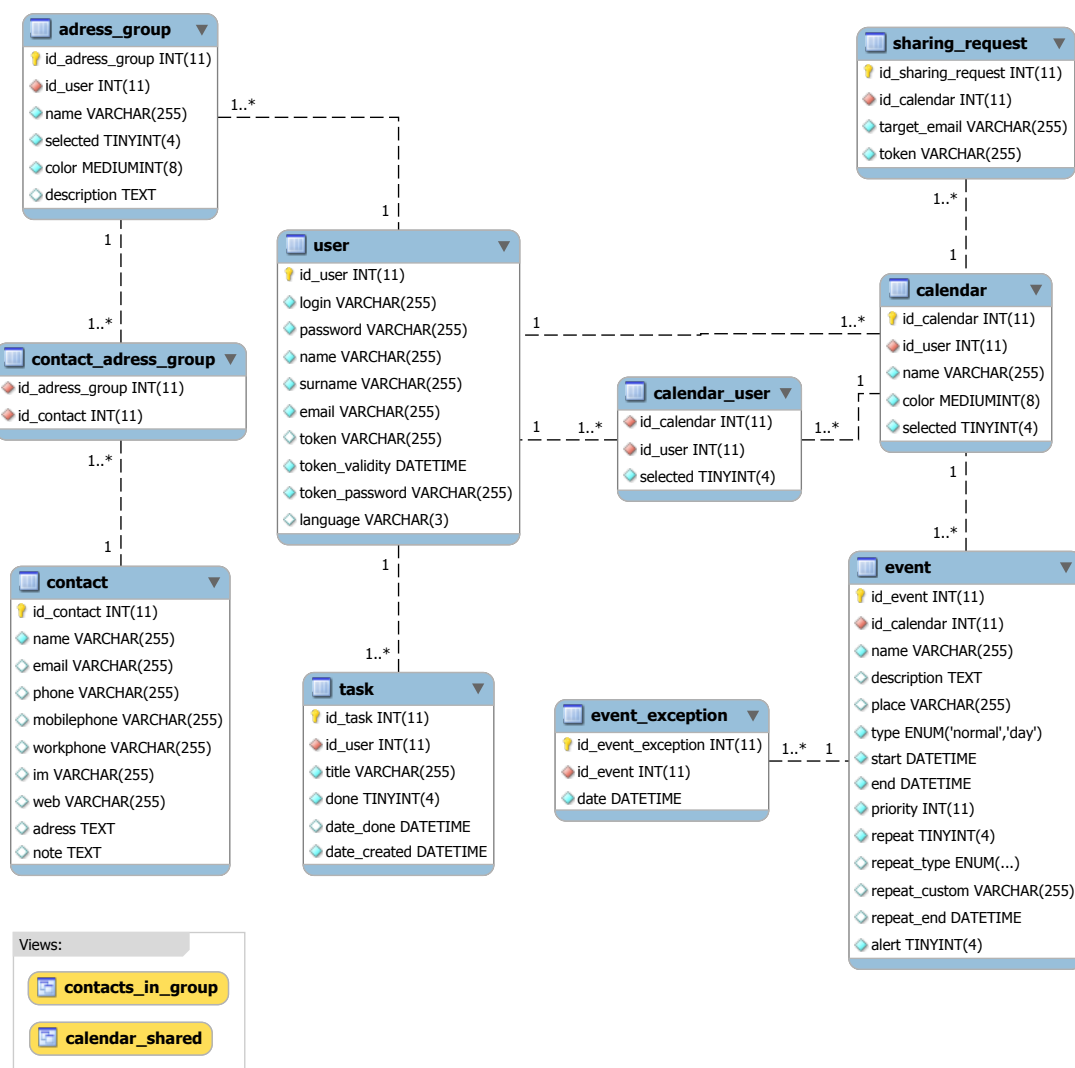
Obrázek 4.2: Entity-Relationship diagram

Jednotlivé sekce jsou barevně odlišeny a veškeré ovládací prvky, formuláře a komponenty uživatelského rozhraní mají jednotný vzhled. Hlavní lišta obsahuje navigaci mezi jednotlivými sekcemi aplikace, volbu jazyka a informaci o přihlášeném uživateli s možností odhlášení. Část s ovládacími prvky může v závislosti na aktuálně vybrané sekci obsahovat například seznam kalendářů s možností jejich zobrazení či skrytí, tlačítka pro přidání nové události a kalendáře nebo seznam kontaktních skupin. Samotný obsah sekce slouží k vizualizaci informací specifikovaných pomocí ovládacích prvků.

4.3.1 Rozvržení kalendáře

V levé části se nachází tlačítka pro přidání nové události, přechod na aktuální datum, dále pak seznam vlastních a sdílených kalendářů s tlačítky pro přidání nového kalendáře. Seznam kalendářů slouží jako filtr výběru – pouze aktivní, vybrané kalendáře (resp. události do něj spadající) budou vykresleny.

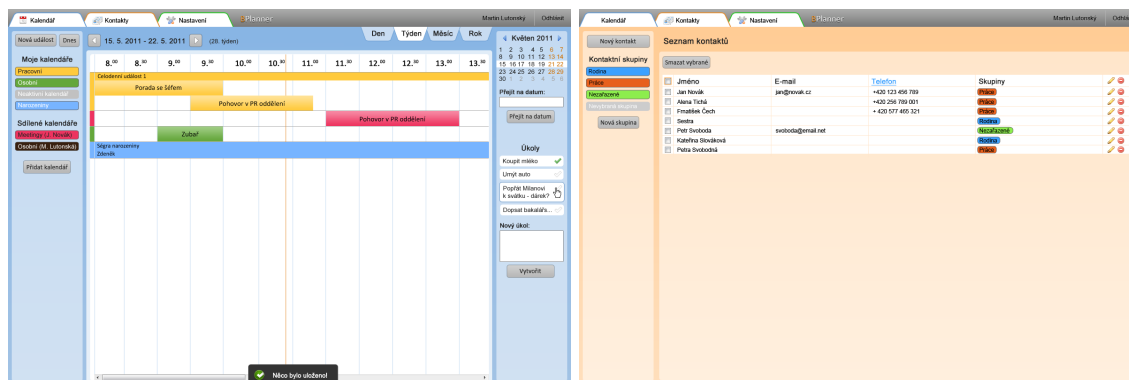
Obsah samotné sekce *kalendář* je rozdělen do tří hlavních logických celků – hlavička kalendáře, samotný kalendář a pravý ovládací panel. Hlavička kalendáře obsahuje aktuálně prezentovaný časový interval, s ovládacími prvky pro posun o jeden interval vpřed, resp. zpět, a dále ovládací prvky pro volbu prezentovaného intervalu. Pravý ovládací panel obsahuje miniaturní měsíční kalendář, který uživateli umožňuje pohodlnější výběr konkrétního data, a dále seznam vytvořených úkolů, spolu s formulářem pro přidávání nových úkolů. Grafická struktura a vzhled sekce *kalendář* je zobrazen na obrázku 4.4.



Obrázek 4.3: Grafická reprezentace návrhu databáze

4.3.2 Rozvržení kontaktů

Sekce *kontakty* vychází z návrhu sekce *kalendář* popsané v 4.3.1. Obsahuje v levé části seznam kontaktních skupin, fungující stejným způsobem jako seznam kalendářů v sekci *kalendář*, tj. jako filtr výběru. V obsahu sekce se nachází seznam kontaktů spadajících do aktuálně vybraných kontaktních skupin. Při vytváření či editaci kontaktu se v obsahu sekce zobrazuje formulář obsahující všechny atributy kontaktu, které lze spravovat.



Obrázek 4.4: Grafický návrh struktury a vzhledu sekcí *kalendář* a *kontakty*

4.3.3 Rozvržení nastavení

V sekci *nastavení* se nachází v levém panelu seznam kategorií, které je možné spravovat (např. údaje o uživateli, kalendáře a kontaktní skupiny), vpravo pak obsah vybrané kategorie a formuláře do ní náležící.

Kapitola 5

Implementace

5.1 Použité technologie

Při implementaci byly použity nástroje a technologie s ohledem na jejich dostupnost a účel použití.

5.1.1 HTML 5

HTML 5 (*Hypertext Markup Language 5*) je nejnovější verzí značkovacího jazyka z rodiny jazyků HTML, určených pro publikaci dokumentů na internetu. Jazyk vznikl aplikací univerzálního značkovacího jazyka SGML (*Standard Generalized Markup Language*).

V současné době (duben 2011) ještě neexistuje oficiální, finální specifikace technologie HTML 5 [28], avšak většina moderních prohlížečů již základní vlastnosti použité při implementaci informačního systému podporuje. V implementaci osobního plánovače jsou navíc použity jen ty elementy a konstrukce jazyka, které jsou zpětně kompatibilní se staršími prohlížeči, podporujícími pouze technologii HTML 4.

5.1.2 CSS

Cascading Style Sheets (zkráceně „CSS“, česky *kaskádové styly*) je technologie vytvořená roku 1996 pracovní skupinou W3C konsorcia, kterým je v současné době také spravována. Zatímco značkovací jazyk udává obsah a význam jednotlivých prvků dokumentu, jazyk CSS definuje vzhled a formu, s jakou bude obsah prezentován uživateli. Pomocí tříd jazyka, identifikátorů a dalších deklarací je možné vybraným skupinám prvků, resp. jednotlivým prvkům, přiřadit konkrétní vlastnost. Výhodou tohoto řešení jsou především mnohem rozsáhlejší možnosti formátování, jednodušší údržba definovaných stylů a také celková přístupnost dokumentu, jelikož je možné například vytvářet styly v závislosti na použitém zařízení (mobilní telefon, počítač, tiskárna) [24, 27].

Při implementaci informačního systému byla použita technologie CSS 3, která nabízí mnohem rozsáhlejší možnosti formátování, nežli předchozí verze. CSS verze 3 je však podporována pouze v novějších verzích prohlížečů, tudíž jsou použity jen ty vlastnosti, které jsou zpětně kompatibilní se staršími verzemi prohlížečů.

5.1.3 JavaScript

JavaScript je multiplatformní, objektově orientovaný skriptovací jazyk, který se v současné době používá především jako interpretovaný programovací jazyk pro WWW stránky,

nejčastěji běžící na straně klienta. Jeho nejsilnější stránkou je podpora pro manipulaci s dokumentem, ke kterému je možné JavaScriptem přistupovat přes DOM (*Document Object Model*) rozhraní. DOM umožňuje přistupovat k objektům dokumentu XML, resp. XHTML či HTML. Těmito objekty mohou být elementy, atributy, text či komentáře.

Jelikož je JavaScript relativně primitivní jazyk, začaly kolem roku 2006 vznikat knihovny a frameworky¹, které vývojářům nabízejí vyšší míru abstrakce a především snadnější manipulaci s modelem DOM. Nejpoužívanějším frameworkem je v současné době jQuery [29], který je také použit při implementaci informačního systému. Společně s rozšířením jQueryUI obsahuje několik vizuálních komponent. Z pohledu implementace je nejzajímavější komponenta *Datepicker*, sloužící pro výběr data.

S pojmem JavaScript souvisí také technologie AJAX (*Asynchronous JavaScript and XML*), což je obecné označení technologií umožňujících dynamicky měnit obsah stránek bez nutnosti jejich opakovaného načtení. Při implementaci byla využita nativní podpora AJAXu v Nette Frameworku (viz 5.1.5), která kombinuje dynamické načítání částí stránek a jejich klasické znovunačtení v případě, že technologie JavaScript není dostupná.

5.1.4 Webový server Apache

Apache HTTP Server je multiplatformní, webový server s otevřeným kódem, vyvíjený organizací Apache Software Foundation [2]. Jeho hlavními přednostmi je velké množství dostupných rozšiřujících modulů, otevřenost, dostupnost a relativně jednoduchá možnost konfigurace, což byly také hlavní důvody pro jeho volbu. V aplikaci jsou aktivně použity rozšiřující moduly *Deflate* a *GZip*, zajišťující kompresi přenášených dat, modul *Rewrite*, starající se o přepis URL a využívaný Nette Frameworkem, dále pak moduly *Header* a *Expires* pro přepis HTTP hlaviček odesílaných souborů (viz 5.7).

5.1.5 PHP a Nette Framework

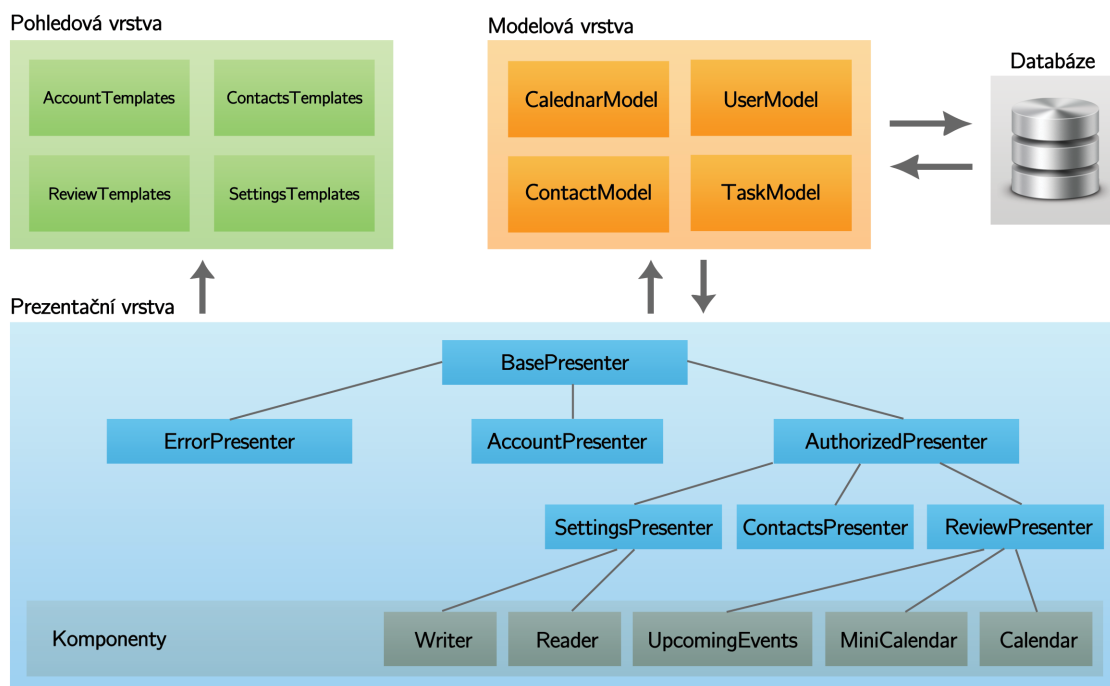
PHP (*PHP: Hypertext Preprocessor*) je skriptovací programovací jazyk původně určený především pro tvorbu dynamických internetových stránek. Skripty PHP jsou narozdíl od skriptů JavaScript prováděny na straně serveru a k uživateli je přenášen až výsledek jejich činnosti – nečastěji v podobě HTML stránky. Od PHP 5 je navíc přítomna i vylepšená podpora pro objektově orientované programování, čehož je při implementaci také využito. Informační systém byl naimplementován, aby byl kompatibilní s PHP 5.2 a novějšími verzemi. Z toho důvodu nejsou při implementaci použity některé vlastnosti dostupné až v PHP 5.3, zejména jmenné prostory a nepojmenované funkce.

Pro lepší dekompozici a efektivnější implementaci informačního systému byl použit Nette Framework. Zvolen byl především z důvodu rozsáhlé a aktivní vývojářské komunity a také proto, že se jedná o jeden z nejvýkonějších PHP frameworků pro tvorbu webových aplikací [5]. Nette Framework umožňuje implementaci webové aplikace s využitím návrhového vzoru MVP (*Model–View–Presenter*), který aplikaci rozděluje na tři nezávislé vrstvy – datový model, uživatelské rozhraní a řídicí logiku, která předchází dvě vrstvy ovládá. MVP architektura vychází z mnohem známějšího návrhového vzoru MVC (*Model–View–Controller*). Rozdíl mezi návrhovými vzory MVC a MVP je především ve vrstvě, která přijímá vstupy od uživatele. Zatímco v architektuře MVC je to řídicí logika (*Controller*),

¹„Framework je softwarová struktura, která slouží jako podpora při programování a vývoji a organizaci jiných softwarových projektů. Může obsahovat podpůrné programy, knihovnu API, návrhové vzory nebo doporučené postupy při vývoji [34].“ Hlavním cílem je převzetí typických problémů dané oblasti, čímž se usnadní celkový vývoj software.

v MVP přijímá vstupy vrstva pohledová (*View*), která je následně deleguje na vstupní body komponent. Prezentační vrstvu tvoří komponenty řídící logiky uspořádané do hierarchické stromové struktury, kde kořen pomyslného stromu tvoří *Presenter* – základní vykreslitelná komponenta návrhového vzoru MVP, na kterou se navazují ostatní komponenty aplikace. Každá z komponent si udržuje svůj vlastní stav, definuje vlastní životní cyklus i způsob vykreslení.

Návrhový vzor MVP je použit i v implementaci informačního systému. Struktura informačního systému je znázorněna na obrázku 5.1.



Obrázek 5.1: MVP architektura informačního systému

Další součástí frameworku je šablonovací systém, jehož nástrojem je Latte filter. Jedná se o sadu metod a maker usnadňujících zápisu šablon. Mimo to Latte filter také umožňuje rozdělit šablonu do souvislých bloků, se kterými je poté možné samostatně pracovat. Jednou z nejsilnějších stránek šablonovacího systému Nette Frameworku je však podpora kontextově sensitivního ošetřování vstupů a výstupů. To je důležité především z pohledu bezpečnosti, neboť v případě jeho správné aplikace jsou znemožněny útoky typu XSS (*Cross-site scripting*) či SQL Injection [18].

5.1.6 MySQL

MySQL je multiplatformní, relační, databázový systém a také jeden z dialektů jazyka SQL, vyvíjen společností Oracle Corporation a šířen pod licencí Open Source. Díky své licenci a rychlosti se jedná o nejpobulárnější databázový systém s otevřeným kódem. Databáze MySQL má však jistá omezení a neobsahuje některé pokročilé funkce, dostupné například v robustních databázových systémech jako Oracle Database či Microsoft SQL Server. V implementaci je použito úložiště typu InnoDB, umožňující spojení databázových tabulek za použití cizích klíčů, díky čemuž je zajištěna lepší integrita dat [17].

5.2 Opakované události

Opakované události, jejich prezentaci a uložení v databázi je možné řešit hned několika způsoby. Jednou z možností je mít všechny výskyty označeny společným identifikátorem a následně je ukládat jako samostatné události. Obzvláště při vysoké frekvenci opakování by však v databázi vznikala velká redundance dat. Na druhou stranu, výhodou tohoto řešení je snadné a rychlejší získávání výskytů opakovaných aktivit, neboť se z databáze vybírají již vygenerované a předpočítané výskyty.

Druhou možností, která je také implementována v informačním systému, je uložení pouze prvního výskytu události, spolu s informacemi potřebnými k výpočtu budoucích. V databázi jsou pak uloženy jen ty nejnutnější hodnoty pro zjištění výskytu opakované události ve zvoleném intervalu, které společně tvoří opakovací pravidlo. Každé opakovací pravidlo se skládá ze začátku opakování, daného počátečním datem události, konce, intervalu opakování a výjimek. Výjimky jsou tvořeny na databázové úrovni samostatnou tabulkou a tvoří je data dnů, ve kterých výskyt události nenastává i když to opakovací pravidlo určuje. Odpadá tak redundance ukládaných dat a jejich náročný přenos na aplikační server.

Z databáze se načítají pouze události, které by potenciálně mohly mít výskyt v požadovaném časovém intervalu. Následně jsou z každé události dle opakovacích pravidel postupně generovány další výskyty události. U každého výskytu se testuje, zdali náleží do zobrazeného časového intervalu. V případě, že do požadovaného intervalu alespoň částečně zasahuje, je přidán do výsledné množiny událostí.

Generování výskytů se provádí již v modelové vrstvě aplikace, tudíž komponenta kalendáře získává finální, připravená data k zobrazení.

5.3 Lokalizace

Překlad textových řetězců v informačním systému je řešen pomocí nástrojů *GNU Gettext*. Varianta překladu řetězců pomocí textového souboru či skriptu zmíněná v 4.1.9 nebyla použita, především z důvodů velikosti, nefektivního ukládání a načítání překladového slovníku.

Gettext je sada nástrojů a knihoven poskytujících framework pro vytváření vícejazyčných aplikací, dostupných i pro PHP. Síla této knihovny spočívá především v podpůrných nástrojích, které umožňují prohledat zdrojové kódy aplikace a vybrat z nich textové řetězce určené k přeložení. Nechybí ani například podpora plurálů či širokého množství kódování vstupních a výstupních souborů [13].

Řetězce určené k překladu jsou označeny v případě PHP voláním funkce `__('řetězec')`, v případě Latte šablon voláním makra `{_ 'řetězec'}`. Vyextrahované řetězce jsou následně uloženy ve formě katalogu, který slouží jako výchozí vzor pro překlad aplikace do dalších jazyků. Katalog je uložen ve formátu *Portable Object*, díky čemuž je zaručena jeho přenositelnost a nezávislost na cílové platformě. Pro katalog řetězců je poté možné jednoduše vytvořit překlad v jiném jazyce, například pomocí programů *POedit* či *Gtranslator*. Součástí sady *Gettext* jsou také nástroje pro vygenerování *Machine Object* souborů, obsahujících překlady zdrojových textů do vybraného jazyka. Překlad je uložen v binární podobě, díky čemuž je vyhledávání přeložených řetězců rychlejší a efektivnější.

Soubor s již vytvořeným katalogem řetězců i PHP skript pro případnou extrakci dalších nebo změněných řetězců z PHP skriptů a Latte šablon jsou součástí zdrojových kódů informačního systému.

5.4 Import a export dat

Informační systém umožňuje import a export kalendářů a událostí ve formátech XML a iCalendar. Dále také nabízí import a export kontaktů a kontaktních skupin ve formátu XML. Pro zpracování XML souborů byly použity rozšiřující PHP moduly *SimpleXML* a *DOM-Document* standardně dodávané se základní distribucí balíku PHP. Tyto moduly nabízejí metody pro jednodušší zpracování a tvorbu validních XML dokumentů [26].

Pro zpracování kalendářních dat byla použita knihovna iCalCreator dostupná z [10]. Knihovna umožňuje generování a zpracování souborů, které vyhovují specifikaci formátu iCalendar dle [4]. Nevalidita dokumentu dle specifikace však může způsobovat problém při importu kalendářních dat z některých, především starších programů, které specifikaci nedodržují.

V případě exportu kalendářů a událostí jsou exportovaná data navíc na aplikačním serveru zkomprimována. Díky tomu je možné zároveň vyexportovat několik kalendářů zároveň a redukovat velikost stahovaného souboru. Komprimace se však provádí pouze za předpokladu, že na aplikačním serveru je nainstalováno PHP rozšíření *ZipArchiv*, starající se o komprimaci souborů do formátu ZIP. V případě, že rozšíření dodstupné není, jsou soubory kalendářů spojeny do jednoho souboru formátu iCalendar a nabídnuty uživateli ke stažení.

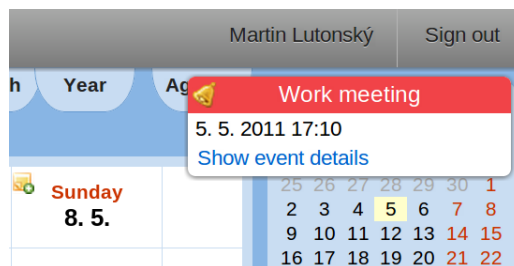
5.5 Uživatelské rozhraní

5.5.1 Grafické prvky

Uživatelské rozhraní bylo implementováno čistě za pomoci technologií CSS a HTML, bez použití obrázků jako grafiky. Jedinými grafickými prvky uloženými jako obrázky jsou ikony. Díky tomu je snížen počet potřebných HTTP požadavků na stáhnutí souborů. Pro složitější grafické prvky, jako jsou zaoblené rohy, barevné přechody či stíny jsou použity pokročilé vlastnosti stylů CSS. Některé z nich však nejsou podporovány staršími prohlížeči (především zaoblené rohy a stíny), proto se vzhled elementů s těmito vlastnostmi může lišit v závislosti na použitém prohlížeči a jeho verzi.

5.5.2 Upozornění na události

V sekci *kalendář* se zobrazují v pravém horním rohu upozornění na případný začátek naplánované události. Při každém obnovení stránky se načítají události, které začnou v nejbližších 12 hodinách. Tato doba odpovídá době vypršení platnosti přihlášení uživatelské identity. Data jsou načtena a vložena na konec stránky jako skript, obsahující kolekci struktur ve formátu JSON. Formát JSON (*JavaScript Object Notation*) je textový formát pro výměnu dat založen na podmožině programovacího jazyka JavaScript [12], což byl hlavní důvod volby tohoto formátu. Zároveň s načtením stránky je nastaven časovač na dobu jedné minuty. Po vypršení časovače je časovač opět resetován a následně je zavolána funkce, která hledá v kolekci struktur události, jenž mají čas začátku stejný jako je aktuální čas, a vkládá je do pole. Po prohledání celé struktury je pro každou událost z pole vygenerováno upozornění v podobě vhodně nastylovaného HTML elementu, který lze vidět na obrázku 5.2. Pokud uživatelský prohlížeč umožňuje přehrání zvuku ve formátu WAV, je navíc uživatel upozorněn i krátkým zvukovým upozorněním.



Obrázek 5.2: Upozornění aplikace na začátek události

5.5.3 Klávesové zkratky

Z důvodů přístupnosti je možné systém částečně ovládat i prostřednictvím klávesových zkratk. Použití klávesové zkratky na aktivaci konkrétního ovládacího prvku bylo voleno s ohledem na jeho důležitost a předpokládanou četnost použití. Klávesová zkratka se vždy skládá z kombinace stisku vybraného znaku a výchozí klávesové kombinace, která se liší dle použitého prohlížeče a operačního systému [35]. Znak klávesové zkratky byl volen především podle anglického názvu akce, jež aktivuje. Například pro aktivaci pole pro vyhledávání událostí bylo vybráno písmeno „S“, vyjadřující počáteční písmeno anglického slova „Search“ – vyhledat. Výjimku tvoří hlavní sekce aplikace, mezi kterými lze přepínat stiskem výchozí kombinace kláves s klávesami 1, 2 nebo 3, což reflektuje pořadí odkazů sekcí v navigačním menu.

5.6 Bezpečnost

5.6.1 Autentizace

Autentizace se provádí klasickým vyplnění přihlašovacích údajů v přihlašovacím formuláři. Jsou-li tyto údaje korektní, je uživateli povolen přístup do systému. Pro lepší zabezpečení a předcházení útoků typu CSFR² je přihlašovací formulář a některé další formuláře opatřeny prvkem s náhodně vygenerovaným řetězcem. Náhodný řetězec generuje serverová část aplikace a s každým vygenerováním formuláře se tato hodnota mění, čímž je útočníkovi velmi ztíženo opakované odesílání podvržených dat [20]. Uživatelská hesla se v databázi ukládají ve formě řetězce o délce 40 znaků, který je zašifrovaný za pomoci jednocestného šifrovacího algoritmu SHA-1. Při případném úniku hesla ze systému je tak zabráněno jeho zneužití. Při autentizaci se odeslané přihlašovací heslo nejprve zašifruje, a až poté porovnává se záznamy uloženými v databázi.

5.6.2 Uživatelská data

Jelikož se identifikátory akcí, jako například ID editovaného kontaktu či události, přenášejí přímo v URL, je důležité kontrolovat, zdali je aktuálně přihlášený uživatel autorizován k vykonání akce, kterou požaduje. Potenciální útočník by mohl URL změnit a údaje tak podvrhnout. Proto se po celou dobu relace uchovávají na serveru informace o přihlášené

²CSFR (*Cross-Site Request Forgery*) je jedna z metod útoku na internetové stránky pracující na principu neočekávaného, resp. nezamýšleného požadavku na vykonání určité akce pocházejícího z nelegitimního zdroje[3].

uživatelské identitně a její unikátní identifikátor na straně klienta ve formě malých textových souborů, tzv. *Cookies*. Při každém požadavku se pak kontroluje, zdali se na serveru nachází daná identita a zdali má právo vykonat požadovanou akci.

5.7 Optimalizace

Informačního systému je hned po několika stránkách optimalizován pro co nejrychlejší běh a odezvu, spojenou s minimalizací přenášených dat. Toho je docíleno hned několika technikami. První optimalizací použitou při generování stránky je souborová dočasná paměť na úrovni šablonovacího systému, která uchovává PHP skripty vygenerované z Latte šablon. Díky tomu se při opakovaném požadavku PHP skripty nemusí znovu generovat. Lokální paměť je použita také na úrovni datového modelu, kde se po dobu existence instance modelu uchovávají výsledky některých SQL dotazů pro případ, že by byl položen ten samý dotaz. Tím je možné redukovat počet dotazů na MySQL server a čas potřebný k jejich zpracování. V případě editačního dotazu se však obsah paměti musí zneplatnit, aby nedošlo k vrácení neaktuálních či nekonzistentních dat.

Další optimalizací je minimalizace přenášených dat. Veškeré CSS styly jsou před odesláním zformátovány, aby neobsahovaly zbytečné mezery a další nepotřebné znaky, čímž se znatelně redukuje jejich velikost. Pro minimalizaci HTTP požadavků jsou navíc ještě spojeny do jednoho souboru. Optimalizace spojením souborů je použita také u načítaných skriptů a knihoven JavaScript. Veškeré optimalizované CSS a JavaScriptové soubory jsou opět ukládány do souborové vyrovnávací paměti. V případě změny některého ze souborů je paměť zneplatněna a soubory jsou znovu vygenerovány. Odstranění přebytných bílých znaků je použito také u některých částí HTML šablon. V konfiguračním souboru serveru Apache jsou navíc vytvořena pravidla prodlužující platnost typů souborů, u kterých se nepředpokládá jejich častá změna, jako jsou například JavaScript soubory, CSS styly nebo ikony. Platnost souborů a také doba, po kterou si prohlížeče mohou udržet stažená data ve svém dočasném uložišti, je u každého stahovaného souboru automaticky nastavena webovým serverem. Nastavení doby vypršení platnosti je prováděno přepisem pole *Expire* v HTTP hlavičce odpovědi. K přepisu hlaviček je použit modul serveru Apache *Expires*, který dle pravidel a typu přenášeného souboru změní hodnotu pole vypršení platnosti souboru. Doba je stanovena dle typu souboru – u obrázků je v řádech týdnů, u JavaScript souborů a CSS stylů v řádech několika dnů, u neměnných HTML stránek (jako je například přihlašovací obrazovka) v řádech hodin. Díky prodloužení platnosti souborů je snížen počet přenášených dat, jelikož prohlížeč nemusí soubory opakovaně stahovat ze serveru, ale může použít již jednou stažené soubory.

Pro minimalizaci přenášených dat a uživatelsky přívětivější chování aplikace jsou jednotlivé funkční bloky stránky načítány asynchronně za pomoci technologie AJAX, díky čemuž se nemusí znovu přenášet celá stránka při každém požadavku, ale pouze její část.

Poslední optimalizací je komprimace přenášených dat na straně webového serveru a jejich dekomprimace na straně klientského prohlížeče. Komprimovány jsou soubory, u kterých je to vhodné, tedy především textové soubory. Ke komprimaci jsou použity metody a rozšiřující moduly serveru Apache, konkrétně *Deflate* a *GZip*. Webový server pak klientskému prohlížeči zasílá data komprimovaná některou z těchto metod, případně nekomprimovaná, v závislosti na podpoře prohlížeče.

Veškerá optimalizační pravidla webového serveru Apache jsou uložena v konfiguračním souboru v kořenovém adresáři aplikace, díky čemuž jsou aplikována i na další souboru a součásti systému. Díky výše zmíněným optimalizacím je možné redukovat velikost přenášených dat o 50–70 % vzhledem k jejich původní, neoptimalizované podobě.

5.8 Testování

5.8.1 Funkčnost na různých platformách

Funkčnost systému byla otestována na dvou platformách – ve vývojovém prostředí na lokálním počítači a v produkčním prostředí na placeném webhostingu. Obě prostředí nabízely odlišné verze webového serveru Apache, PHP a MySQL databáze. Výpočetní výkon obou platform byl diametrálně odlišný. V případě produkčního prostředí se jednalo o výkonný server, na kterém byl pod operačním systémem rodiny Linux spuštěn server Apache 2.2.14, PHP 5.3.6 a databáze MySQL verze 5.0.7. Vývojové prostředí naopak běželo na osobním notebooku, na němž byl nainstalován operační systém Windows 7, webový server Apache ve verzi 2.2.17, PHP 5.2.17 a MySQL 5.5.8. Největší rozdíl mezi těmito platformami tvoří především odlišné verze jazyka PHP. Pro běh informačního systému je doporučeno PHP ve verzi 5.3, především díky výkonostním optimalizacím interpretu jazyka [25].

Mezi oběma systémy nebyl zjištěn žádný funkční rozdíl, informační systém vykazoval totožné chování na obou platformách, i přes odlišné verze nainstalovaného software a konfiguraci webových serverů. Doba generování stránky a celková odezva systému byla dle očekávání kratší na výkonném serveru, a to především díky vyššímu výpočetnímu výkonu a novější verzi PHP.

5.8.2 Uživatelské testování

Informační systém byl podroben uživatelskému testování třemi uživateli s rozdílnými zkušenostmi s prací na počítači a s internetem. Před testováním byli uživatelé seznámeni s účelem a možnostmi použití systému. Poté bylo uživatelům umožněno krátké seznámení se s uživatelským rozhraním a ovládáním aplikace. Po seznámení se systémem byly testovací subjekty vyzvány aby splnily přibližně 15 jednoduchých úkolů, na kterých se ověřila funkčnost a použití navrhnutého řešení. Těmito úkoly byly například vytvoření nového kalendáře, vytvoření události opakující se v určeném intervalu, sdílení kalendáře s jiným uživatelem, odstranění kontaktů spadajících do vybrané kontaktní skupiny a další. Uživatelé byli požádáni, aby v průběhu celého testování popisovali právě prováděné úkoly, pocity, názory na efektivnější možná řešení daných úkolů.

Všichni uživatelé dokázali bez problémů úkoly splnit. Jelikož informační systém nebyl v českém jazyce, jediným problémem při řešení úkolů byla horší znalost anglického jazyka jednoho z uživatelů, avšak i s tímto omezením dokázal splnit všechny požadované úkoly. Je tedy zřejmé, že i když uživatel plně neovládal jazyk aplikace a nerozuměl všem textovým popiskům a dialogům, byl schopen díky intuitivnímu a přehlednému ovládání aplikaci plně používat. Uživateli bylo pozitivně hodnocen především grafické rozhraní a také rychlost odezvy aplikace. Práce s informačním systémem se jim zdála uživatelsky přívětivá a dostatečně komfortní.

Kapitola 6

Závěr

Vytvořený informační systém má uživatelům sloužit jako osobní plánovač a správce aktivit, událostí, úkolů a kontaktů. Na základě popisu plánovacích metod, technik a následné analýzy byla stanovena specifikace a návrh možného řešení. V následující fázi byla vysvětlena samotná implementace. Byly přiblíženy použité technologie a důvod jejich volby, stejně tak i stěžejní problémy a jejich řešení. Výstupem implementace je informační systém pro osobní plánování časových událostí a jiných aktivit.

Formou několika pohledů na různé časové intervaly umožňuje přehledným způsobem prezentovat plánované záležitosti. Aplikace podporuje správu několika typů časových událostí – vázané na konkrétní čas, datum i celodenní události. Všechny typy mohou být i opakované, a to s intervalem jednoho dne, týdne, měsíce či roku. Nechybí ani možnost výběru konkrétních dnů či měsíců, v nichž se má zvolená aktivita opakovat. V systému je implementován také jednoduchý správce úkolů a kontaktů. Kontakty je možné seskupovat, filtrovat dle kontaktních skupin či řadit podle několika atributů.

Jelikož se jedná o webovou aplikaci, byl již při návrhu kladen důraz na co nejrychlejší chod a provoz systému. Kratší odezvy je dosaženo především díky technologii AJAX a také množstvím optimalizací, které jsou prováděny na aplikačním a webovém serveru. Pro lepší dekompozici systému na jednotlivé funkční celky byl systém implementován za použití návrhového vzoru MVP a Nette Frameworku (viz 5.1.5).

Jedním z hlavních cílů bylo efektivní a přehledné uživatelské rozhraní, které má uživateli zpříjemnit a usnadnit používání systému. Proto je grafické rozhraní decentní, ve všech sekcích systému jednotné a bez přebytečných efektů. Díky aplikaci jemných přechodů a stínů však působí moderním dojmem. Uživatelé systému mají možnost exportu a importu plánovaných aktivit, kalendářů a kontaktů ve formátu XML. Aktivita a kalendáře je navíc možné importovat a exportovat i ve formátu iCalendar, který je přímo určen k přenášení kalendářních dat. Díky tomu je možné sdílet uživatelská data i mezi ostatními plánovači či jinými aplikacemi.

Za předpokladu, že bude uživatel vytvořený systém aktivně využívat při plánování, nabízí relativně jednoduchou cestu k udržení přehledu o osobních plánovaných aktivitách a stanovených úkolech. Funkčnost byla otestována na dvou odlišných platformách, i prostřednictvím uživatelského testování. Výsledné řešení tak splňuje požadavky stanovené v zadání ve všech bodech. V porovnání s již existujícími osobními plánovači nabízí přibližně srovnatelnou funkcionalitu a uživatelské rozhraní.

6.1 Možnosti rozšíření

Do budoucna by bylo vhodné aplikaci rozšířit o veřejné API, přes které by uživatelé mohli přistupovat ke svým datům i prostřednictvím aplikací a nástrojů třetích stran. Přístupové rozhraní by se také dalo využít například pro vkládání kalendářů do webových stránek formou plovoucích rámců či obrázků.

Dále by mohlo být výsledné řešení rozšířeno o detekci podpory JavaScriptu na straně klientského prohlížeče. V případě, že by byla podpora dostupná, HTML kód aplikace by byl generován až v internetovém prohlížeči uživatele, pouze pomocí technologie JavaScript, jako je tomu například u aplikace Google Kalendář.

Pro pohodlnější práci s aplikací a především pro větší možnosti přizpůsobení by informační systém mohl také umožňovat přesun objektů uživatelského rozhraní pomocí techniky *drag-and-drop*. Uživatel by si poté mohl uzpůsobit rozmístění jednotlivých komponent dle vlastního uvážení. Technika by se také dala použít pro jednodušší přesun událostí v čase či jejich přímé kopírování.

V rámci kolaborace a spolupráce více členů při plánování by mohl být systém rozšířen o možnost plánování společných pracovních činností, hromadné tvorby a správy sdílených kalendářů a událostí. Uživatelé by pak měli lepší přehled o pracovních činnostech a úkolech jednotlivých členů či týmů. U pracovních událostí by uživatelé systému mohli navíc registrovat a stanovovat například míru splnění stanovených úkolů či dosažení milníků. V současném systému mohou být události seskupovány do jednotlivých kalendářů. V budoucím rozšíření by však mohl systém umožňovat jemnější členění. Události by byly například spravovány v rámci projektů či automaticky řazeny dle nastavených pravidel do vlastních složek apod.

Zajímavá by byla i podpora dalších oblastí plánování, nejen času, ale třeba i zdrojů či financí. Informační systém by se tak mohl stát jednou ze součástí mnohem rozsáhlejšího systému, určeného pro řízení projektů. Pro začátek by bylo možné rozšířit aplikaci například o podporu plánování osobních financí, k čemuž by šla využít výsledná aplikace vytvořená Martinem Odstrčilíkem v rámci bakalářské práce na téma Osobní plánovač financí pro OS Android [19]. Za předpokladu dodateční implementace API, přes které by mobilní aplikace synchronizovala data s informačním systémem, by bylo možné mít neustálý přehled i o plánovaných aktivitách v relaci s jejich finanční stránkou.

V neposlední řadě by bylo vhodné přetvořit informační systém do plně modulární architektury, která by umožňovala použití jednotlivých komponent stávajícího systému jako samostatné celky. Systém by pak nabízel jednotné rozhraní pro přidávání a správu dalších modulů, jako například jednoduchý e-mailový klient napojený na modul kontaktů či správce dokumentů, které by bylo možné připojovat k událostem.

Literatura

- [1] ALLEN, D.: *Mít vše hotovo*. Brno: Computer Press, a.s., listopad 2009, ISBN 978-81-700-8309-2.
- [2] Apache Software Foundation: The Apache HTTP Server Project [online]. <http://httpd.apache.org/>, 2011 [cit. 2011-05-03].
- [3] AUGER, R.: The Cross-Site Request Forgery (CSRF/XSRF) FAQ [online]. <http://www.cgisecurity.com/csrf-faq.html>, 2011 [cit. 2011-05-09].
- [4] B. DESRUISSEAUX, ED.: Internet Calendaring and Scheduling Core Object Specification (iCalendar) [online]. <http://tools.ietf.org/html/rfc5545>, 2009 [cit. 2011-0417].
- [5] DANĚK, P.: Velký test PHP frameworků [online]. <http://www.root.cz/clanky/velky-test-php-frameworku-zend-nette-php-a-ror/>, 2008 [cit. 2011-05-03].
- [6] DUSSEAU, L.: HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV) [online]. <http://tools.ietf.org/html/rfc4918>, 2007 [cit. 2011-03-29].
- [7] GlobalStats StatCounter: Top 12 Browser Versions [online]. http://gs.statcounter.com/#browser_version-ww-monthly-201103-201104-bar, 2011 [cit. 2011-04-11].
- [8] Google Inc.: Google Calendar APIs and Tools [online]. <http://code.google.com/intl/cs-CZ/apis/calendar/>, 2011 [cit. 2011-03-29].
- [9] Google Inc.: Gears API [online]. <http://code.google.com/intl/cs-CZ/apis/gears/>, 2011 [cit. 2011-05-06].
- [10] GUSTAFSSON, K.: iCalcreator [online]. <http://www.kigkonsult.se/iCalcreator/>, 2011 [cit. 2011-05-06].
- [11] Internet Info, s. r. o.: Česká pravidla přístupnosti [online]. <http://www.pristupnost.cz/ceska-pravidla-pristupnosti/>, 2011 [cit. 2011-04-11].
- [12] JSON.org: Úvod do JSON [online]. <http://json.org/json-cz.html>, 2011 [cit. 2011-05-04].
- [13] KAŠPÁREK, T.: GNU - pomoc při tvorbě programů: Gettext [online]. <http://www.root.cz/clanky/gnu-pomoc-pri-tvorbe-programu-gettext/>, 2001 [cit. 2011-05-06].

- [14] Microsoft Corporation: Jak řešit problémy výkonu v aplikaci Outlook 2007 [online]. <http://support.microsoft.com/kb/940226>, 2009 [cit. 2011-05-09].
- [15] Microsoft Corporation: Stručná historie řízení projektů [online]. <http://office.microsoft.com/cs-cz/project-help/HA010351563.aspx>, 2010 [cit. 2011-03-29].
- [16] Microsoft Corporation: Definice fází a úkolů [online]. <http://office.microsoft.com/cs-cz/project-help/HA001078070.aspx>, 2011 [cit. 2011-03-29].
- [17] MySQL.com: MySQL [online]. <http://www.mysql.com/>, 2011 [cit. 2011-05-03].
- [18] Nette Foundation: Nette Framework [online]. <http://nette.org/>, 2008 [cit. 2011-04-17].
- [19] ODSTRČILÍK, M.: *Osobní plánovač financí pro OS Android*. Diplomová práce, Fakulta infomačních technologií Vysokého učení technického v Brně, 2010.
- [20] PEJŠA, J.: Co je Cross-Site Request Forgery a jak se mu bránit [online]. <http://zdrojak.root.cz/clanky/co-je-cross-site-request-forgery-a-jak-se-branit/>, 2008 [cit. 2011-05-03].
- [21] PUMNIA, B. C.; KHANDELWAL, K. K.: *Project Planning and Control with PERT & CPM*. New Delhi: Firewall Media, 2002, ISBN 978-81-700-8309-2.
- [22] ROSENAU, M. D.: *Řízení projektů*. Brno: Computer Press, a.s., únor 2007, ISBN 978-80-251-1506-0.
- [23] SCHWALBE, K.: *Řízení projektů v IT*. Brno: Computer Press, a.s., 2007, ISBN 978-80-251-1526-8.
- [24] STANÍČEK, P.: *CSS Hotová řešení*. Brno: Computer Press, a.s., červenec 2006, ISBN 80-251-1031-1.
- [25] The PHP Group: Performance Considerations [online]. <http://php.net/manual/en/features.gc.performance-considerations>, 2011 [cit. 2011-05-06].
- [26] The PHP Group: PHP: XML Manipulation [online]. <http://www.php.net/manual/en/refs.xml.php>, 2011 [cit. 2011-05-06].
- [27] The World Wide Web Consortium: Cascading Style Sheets, level 1 [online]. <http://www.w3.org/TR/CSS1/>, 1996 [cit. 2011-04-11].
- [28] The World Wide Web Consortium: HTML 5 [online]. <http://www.w3.org/TR/html5/>, 2011 [cit. 2011-04-11].
- [29] Wappalyzer: JavaScript frameworks statistics [online]. <http://wappalyzer.com/stats/cat/JavaScript%20frameworks>, 2011 [cit. 2011-04-17].
- [30] WEAVER, P.: A Brief History of Scheduling - Back to the Future [online]. http://www.mosaicprojects.com.au/Resources_Papers_042.html, 2006 [cit. 2011-03-29].

- [31] WEAVER, P.: The Origins of Modern Project Management [online].
http://www.mosaicprojects.com.au/Resources_Papers_050.html, 2007 [cit. 2011-03-29].
- [32] Wikipedia: Gantt diagram [online].
http://cs.wikipedia.org/wiki/Gantt%C5%AFv_diagram, 2008 [cit. 2011-04-13].
- [33] Wikipedia: Program Evaluation and Review Technique [online].
<http://en.wikipedia.org/wiki/PERT>, 2011 [cit. 2011-03-29].
- [34] Wikipedia: Framework [online]. <http://cs.wikipedia.org/wiki/Framework>, 2011 [cit. 2011-04-17].
- [35] Wikipedia: Access key [online]. http://en.wikipedia.org/wiki/Access_key, 2011 [cit. 2011-05-06].

Seznam příloh

příloha A Obsah CD

příloha B Obrazovky systému

Příloha A

Obsah CD

Příložené CD obsahuje následující materiály.

Technická zpráva

soubor `zprava.pdf`

Instalační a konfigurační manuál

soubor `manual.pdf`

soubor `manual.txt`

Zdrojové kódy aplikace

adresář `./sources`

Zdrojové kódy technické zprávy

soubor `./zprava-latex`

Inicializační skripty databáze

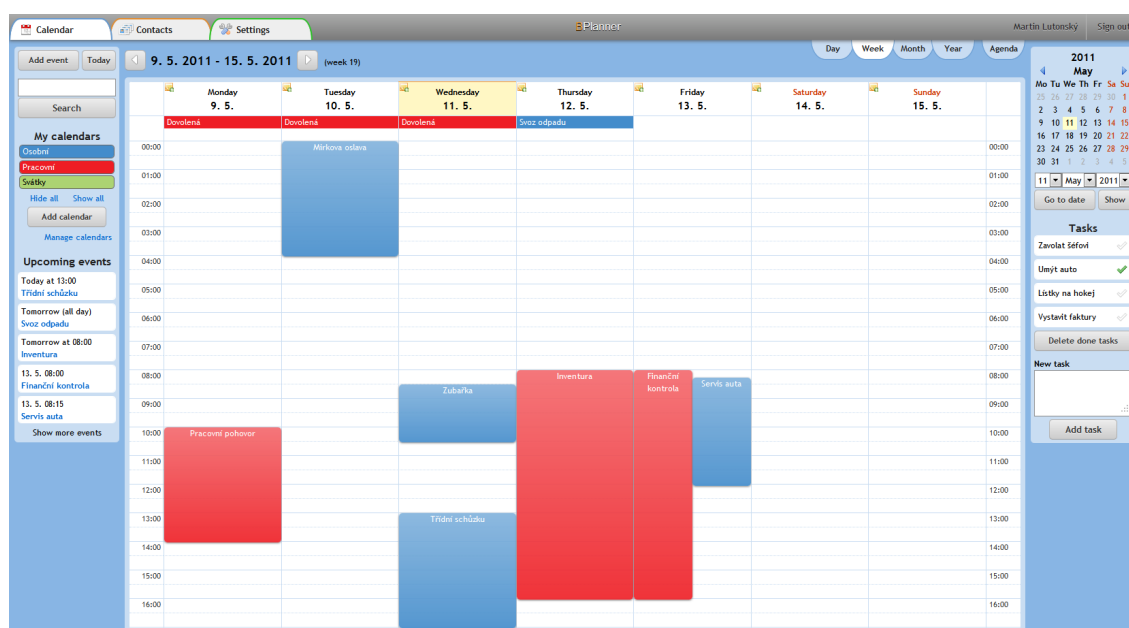
adresář `./database`

Potřebný software a pokyny ke zprovoznění systému

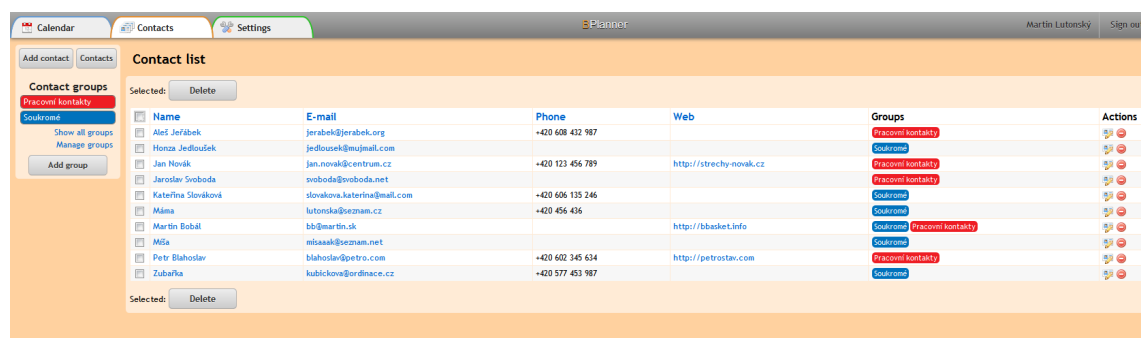
adresář `./prerequisites` a soubor `./prerequisites/README.txt`

Příloha B

Obrazovky systému



Obrázek B.1: Týdenní přehled plánových aktivit



Obrázek B.2: Správce kontaktů