

Implementace modulu plánování zdrojů pro systém PERM 3

Jiří Kouřil

Bakalářská práce
2018



Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2017/2018

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: Jiří Kouřil
Osobní číslo: A14645
Studijní program: B3902 Inženýrská informatika
Studijní obor: Informační a řídicí technologie
Forma studia: prezenční

Téma práce: Implementace modulu plánování zdrojů pro systém PERM3
Téma anglicky: The Implementation of the Resource Management Module for the PERM3 System

Zásady pro vypracování:

1. Proveďte analýzu funkčních a nefunkčních požadavků na modul docházky pro systém PERM3.
2. Na základě požadavků specifikujte datové struktury a datový model.
3. Vytvořte návrh uživatelského rozhraní modulu.
4. Implementujte všechny funkce modulu dle požadavků.
5. Vytvořte implementační i uživatelskou dokumentaci a navrhňte metodu testování modulu.

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. CANTÚ, Marco, WOOD, Peter, ed. Object Pascal Handbook. Piacenza (Italy): Marco Cantú, 2014. ISBN-10: 1514349949.
2. GRUBER, Martin. Mistrovství v SQL. SoftPress, 2004. ISBN 8086497623.
3. HODGES, Nick. Coding in Delphi. Nepeta Enterprises, 2014. ISBN-13: 978-1941266106.
4. HODGES, Nick. More Coding in Delphi. Createspace Independent Publishing Platform, 2017. ISBN-13: 9781548879624.
5. MARTIN, Robert C. Čistý kód: [návrhové vzory, refaktorování, testování a další techniky agilního programování]. Brno: Computer Press, 2009. ISBN 978-80-251-2285-3.

Vedoucí bakalářské práce:

Ing. Tomáš Dulík, Ph.D.

Ústav informatiky a umělé inteligence

Datum zadání bakalářské práce:

15. prosince 2017

Termín odevzdání bakalářské práce:

25. května 2018

Ve Zlíně dne 15. prosince 2017



doc. Mgr. Milan Adámek, Ph.D.
děkan



prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s tím, že vyrovnaní případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne



.....
podpis diplomanta

ABSTRAKT

Implementace modulu plánování zdrojů pro systém PERM 3

Klíčová slova: PERM 3, modul, kalendář

ABSTRACT

Implementation of Resource Management Module for PERM 3 System

Keywords: PERM 3, module, calendar

Chtěl bych poděkovat vedoucímu této práce, panu Ing. Tomáši Dulíkovi, Ph.D., za veškerý čas, který investoval do této práce, za všechny rady a za věcný a praktický přístup k řešení všech problémů, jež se vyskytly. Dále bych chtěl poděkovat firmě Kvasar s.r.o. za to, že umožnila zpracování tohoto modulu a za poskytnutí konkrétních aplikačních řešení, jež byly v rámci této práce využity. A na závěr patří velké díky panu RNDr. Vladimíru Anthovi, a to za konstruktivní rady a poznámky, jež poskytoval po celou dobu psaní této práce.

Prohlašuji, že odevzdaná verze bakalářské/diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

OBSAH

ÚVOD	10
I TEORETICKÁ ČÁST	11
1 POŽADAVKY NA PLÁNOVACÍ MODUL.....	12
1.1 PERM 3, PERMWEB A MOTIVACE.....	12
1.2 KONKRÉTNÍ POŽADAVKY NA MODUL KALENDÁŘE	12
2 PŘEDSTAVENÍ AKTUÁLNÍ NABÍDKY PLÁNOVACÍCH KALENDÁŘŮ	14
2.1 ROZDĚLENÍ DLE CENY	14
2.1.1 Aplikace dostupné zdarma	14
2.1.2 Aplikace s poplatkem	15
2.2 ROZDĚLENÍ DLE NÁROČNOSTI ZAVEDENÍ.....	15
2.2.1 Webové aplikace	15
2.2.2 Desktopové aplikace	15
2.2.3 Kombinace webové a desktopové aplikace	15
2.3 VZÁJEMNÁ KOMPATIBILITA	16
2.4 PŘEHLED JEDNOTLIVÝCH APLIKACÍ	16
2.4.1 Desktopové aplikace dostupné zdarma	16
2.4.2 Webové aplikace dostupné zdarma.....	17
2.4.3 Placené desktopové aplikace	18
2.4.4 Placené webové aplikace	19
2.4.5 Kombinace webové a desktopové aplikace	19
2.5 SHRNUÍ	19
3 PROSTŘEDÍ A JAZYK.....	21
3.1 VÝVOJ PASCALU	21
3.1.1 Vznik jazyka	21
3.1.2 Turbo Pascal	21
3.1.3 Delphi	21
3.1.4 Embracadero	22
3.2 JAZYK OBJECT PASCAL	22
3.2.1 Objektový přístup.....	22
3.2.2 Formuláře	23
3.3 FORMÁT XML	24
3.3.1 Cíle formátu XML.....	25
II PRAKTICKÁ ČÁST	26
4 DATOVÉ STRUKTURY	27
5 IMPLEMENTAČNÍ DOKUMENTACE MODULU	29
5.1 VLASTNOSTI HLAVNÍCH TŘÍD.....	29
5.1.1 TkvaWebPraPlan	29
5.1.2 TkvaWebPraZdroj.....	30
5.1.3 TkvaWebPraVztPPV.....	31
5.1.4 TkvaWebPraDenMapa	32

5.1.5	Popis	33
5.2	HLAVNÍ FUNKCE A PROCEDURY MODULU	33
5.2.1	Accessory	33
5.2.2	Mutatory	34
5.2.3	NewPlan	34
5.2.4	NewWebPraVztPPV	34
5.2.5	NewWebPraZdroj	34
5.2.6	GetDobaSmeny	35
5.2.7	FillDenMapa	35
5.2.8	CheckKolize	36
5.2.9	CheckZdrojKolize	36
5.2.10	GetSimplePlanDleMapy	37
5.2.11	GetPlanDleMapy	37
5.3	TŘÍDA PRO SQL.....	37
5.3.1	Read.....	37
5.3.2	ReadByDenMapa.....	38
5.3.3	Write.....	38
6	IMPLEMENTACE PREZENTAČNÍ VRSTVY	40
6.1	FORMULÁŘ Q_PLANKALENDAR.....	40
6.1.1	Inicializace formuláře	40
6.1.2	Přepřínování formuláře	40
6.1.3	Volání podřazených formulářů	40
6.1.4	Uložení dat	41
6.2	FORMULÁŘ L_PLANKALENDAR.....	41
6.2.1	Vkládání nového plánu.....	41
6.2.2	Editace plánu	41
6.2.3	Mazání plánu	41
6.3	FORMULÁŘ E_PLANKALENDAR	41
6.3.1	Hlavní události formuláře.....	42
6.3.2	Přidávání a mazání položek.....	43
7	NÁVRH ROZHRAŇÍ A UŽIVATELSKÁ DOKUMENTACE MODULU	45
7.1	PROHLÍŽECÍ FORMULÁŘ.....	45
7.2	SEZNAM UDÁLOSTÍ.....	45
7.3	EDITAČNÍ FORMULÁŘ.....	46
7.3.1	Popis události	46
7.3.2	Vztažené PPV.....	48
7.3.3	Zdroje	49
8	KNIOVNY PRO KOMUNIKACI S PROSTŘEDÍM PERMWEB	51
8.1	XML SOUBORY	51
8.1.1	Vstupní XML	51
8.1.2	Výstupní XML.....	52
8.2	IMPLEMENTACE KNIHOVEN	52
9	TESTOVÁNÍ MODULU	54

ZÁVĚR.....	55
SEZNAM POUŽITÉ LITERATURY	56
SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	59
SEZNAM OBRÁZKŮ.....	60
SEZNAM PŘÍLOH	61

ÚVOD

Lidstvo skrz celou historii provází potřeba si svůj čas určitým způsobem organizovat, ať už se jedná o rozvržení osobních záležitostí jednotlivců nebo o plánování celých válečných tažení. K tomuto účelu byly používány různé formy kalendářů, které umožňovala lidem nejen si vytvořit náčrt svých budoucích cílů a plánů, ale také zaznamenávání a přesné datování historických událostí.

V dnešní době roste poptávka po efektivním plánování, což se zvláště týká firem s větším množstvím zaměstnanců, kde již vedoucí pracovníci mají akutní potřebu si rozvrhnout přehledně nejen svůj čas, ale hlavně čas podřízených zaměstnanců a dostupných zdrojů. Z tohoto důvodu přestaly být obyčejné kalendáře v papírové podobě dostačující a rozšířily se kalendáře v elektronické podobě, které poskytují výhody nejen ve snadnější a přehlednější editaci, ale i v široké škále možností následně s daty pracovat.

Tato bakalářská práce se zabývá zpracováním plánovacího modulu pro systém PERM 3 firmy Kvasar s.r.o, tento modul umožňuje uživatelům aplikace snadnou a přehlednou správu časových harmonogramů nejen osob, ale i prostorů a svěřených věcí. Cílem práce tedy není prostý kalendář sloužící k správě vlastního rozvrhu, ale nástroj, který také umožní vedoucím pracovníkům a případně i personalistům správu a plánování vnitřního chodu firmy. Další součástí bakalářské práce jsou knihovny, jež se zabývají vytvářením a zpracováním strukturovaných *XML* souborů, které slouží ke komunikaci modulu s webovým prostředím PermWeb, v rámci kterého budou mít ke svým pracovním kalendářům přístup i jednotliví zaměstnanci.

Práce je rozdělena na dvě hlavní části: teoretické a praktické. V teoretické části je představena aplikace PERM 3 a požadavky na modul plánování zdrojů, dále se teoretická část zabývá aktuální nabídkou různých aplikací pro tvorbu kalendářů a popisem vývojového prostředí Delphi. Praktická část obsahuje popis struktur databáze, implementační a uživatelskou dokumentaci modulu a popis knihoven pro zpracování *XML* souborů.

I. TEORETICKÁ ČÁST

1 POŽADAVKY NA PLÁNOVACÍ MODUL

1.1 PERM 3, PermWeb a motivace

Specializací firmy Kvasar s.r.o. je vývoj programů pro zpracování mzdového účetnictví a personalistiky. Stěžejní aplikací je systém PERM 3, poskytující uživateli možnost správy organizační struktury firmy, mezd, personalistiky, docházky, uchazečů a mnohé další. Nadstavbou tohoto systému je portál PermWeb umožňující správu vybraných dat i dalším uživatelům mimo mzdové četní, ať už dalším vedoucím pracovníkům nebo i všem dalším zaměstnancům. Zásadní funkcionalitou systému PERM 3 a PermWeb je řešení docházky a všech s ní spjatých okruhů jako například žádosti o dovolenou, žádosti o pracovní cesty nebo správa nemocenských. Vzhledem k roztržitosti docházkové agendy vznikla potřeba vývoje modulu pro plánování (nejen) lidských zdrojů, ve kterém by byla umožněna správa všech nepřítomností na jednom místě a zároveň by umožnil plánování pracovních schůzek, porad a dalších aktivit.

1.2 Konkrétní požadavky na modul kalendáře

Tento modul byl zpracováván na základě požadavků firmy Kvasar s.r.o, které byly doplněny o požadavky vedoucího práce:

- Realizace modulu plánovacího kalendáře, který umožní vytváření a editaci jednotlivých plánů zaměstnanců pro systém PERM 3.
- Detailní popis konkrétního plánu a nastavení libovolného kódu docházky, tedy určení zda se jedná o například pracovní cestu, nemocenskou, návštěvu lékaře nebo firemní porad.
- Přehledné grafické zpracování formuláře pro plánovací kalendář včetně záznamů o již proběhlých rozvrhových akcích ve formě zobrazení denní mapy v rámci systému PERM 3.
- Vytvoření formuláře, který bude umožňovat editaci jednotlivých záznamů a formuláře na jejich případný výběr.
- Možnost jedné akci přidat libovolné množství účastníků včetně jednoduchého způsobu editace jejich časového harmonogramu.

- Při více PPV účastníka umožněn výběr konkrétního z nich, kterému bude akce náležet.
- Pro každý plán možnost obsahovat libovolný zdroj v podobě financí, svěřené věci a/nebo případně konkrétních prostor a určit dobu jejich používání.
- Vyhodnocování kolizních stavů jak osob, tak zdrojů, jejich zobrazování a upozornění uživatele.
- Dvě knihovny zpracovávající požadavky portálu PermWeb ve formě strukturovaných XML souborů a vytvářející odpovědi také ve formě souborů XML, první knihovna je určena pro správu pracovního kalendáře běžným zaměstnancem s možnostmi žádat o přiřazení fyzických zdrojů nebo posílání pozvánek k daným událostem ostatním zaměstnancům. Druhá knihovna byla vyžádána zákazníkem firmy Kvasar s.r.o. a jde omezenou verzí předchozí knihovny odlehčenou o zdroje a pozvánky, funguje tedy čistě jako osobní pracovní kalendář.

2 PŘEDSTAVENÍ AKTUÁLNÍ NABÍDKY PLÁNOVACÍCH KALENDÁŘŮ

Protože se tato práce zabývá zpracováním plánovacího modulu, je tato kapitola věnována nabídce různých plánovacích kalendářů a jejich srovnání. Jelikož je těchto aplikací nepřeberné množství, je třeba je nejdříve rozdělit do několika skupin, jako kritérium členění nejlépe slouží cena, dále je možné tyto aplikace rozdělit podle náročnosti jejich zavedení a podle kompatibility s ostatními aplikacemi.

2.1 Rozdělení dle ceny

Jednotlivé aplikace lze podle jejich ceny rozdělit na 3 základní skupiny: dostupné zdarma, placené měsíčním poplatkem a pořízené jednorázovým poplatkem. [1] Často se můžeme setkat s různou kombinací zmíněných možností, kdy může být program poskytován po určitou dobu zdarma, nebo je zdarma s určitým omezením funkcionality a uživatel si může zakoupit jednotlivá rozšíření. [2]

2.1.1 Aplikace dostupné zdarma

Tyto aplikace se často značí relativně omezenou nabídkou funkcí, splňují jednoduché požadavky na funkčnost jako náhrada papírového kalendáře a i přes snahu je doplnit o zajímavé funkce se často příliš nehodí ke zpracovávání složitějších požadavků. Také je zde obvyklý výskyt různých reklam a inzerce.

Dále se dají rozdělit na ty, které spadají pod určitou všeobecnou veřejnou licenci (například GNU General Public License) a jsou tzv. „open source“ a ty, za kterými stojí určitá organizace a její zdrojový kód není volně přístupný. V prvním případě je výhodou možnost si aplikaci pozměnit dle vlastních představ, avšak pro běžného uživatele bez pokročilých znalostí v informačních technologiích velmi obtížná, navíc velkou nevýhodou je, že při výskytu problémů s aplikací se často nedostane uživateli žádná podpora. V případě aplikací, které jsou sice zdarma, ale nemají otevřený kód je situace s podporou uživatelů lepší, avšak stále je poskytována v omezené míře, zároveň jakékoli vlastní modifikace programu nejsou možné.[3][4]

2.1.2 Aplikace s poplatkem

Placené verze jsou většinou bez reklamy, nejsou zde omezení počtu záznamů a případně je podporován import dat z jiných běžně používaných aplikací skrz standard iCalendar, což je u aplikací zdarma spíše výjimkou. Velkou výhodou je také širší podpora uživateli. Cena se často odvíjí od množství funkcí, které jsou uživateli poskytnuty, základní kalendáře pro jednoho uživatele se pohybují v cenovém rozmezí mezi 5 a 10 dolary za pořízení aplikace, případně některé kalendáře vyžadují měsíční respektive roční placení poplatků místo jednorázového nákupu.[4][5]

2.2 Rozdělení dle náročnosti zavedení

Dalším základním kritériem pro výběr plánovacího kalendáře vedle ceny je jeho forma a náročnost jeho zavedení, tedy zda je nutné jej instalovat, případně jakou formou.

2.2.1 Webové aplikace

Velkou výhodou webových aplikací je, že není třeba jejich instalace, stačí často jednoduchá registrace a poté pouze připojení k internetu a internetový prohlížeč, což umožňuje jejich použití na libovolném zařízení s libovolnou platformou, avšak nutný přístup k internetu může být i nevýhodou, ať už z bezpečnostního hlediska nebo z možnosti nedostupného připojení k internetu. [6]

2.2.2 Desktopové aplikace

Desktopové aplikace oproti webovým mají výhodu v rychlejší odezvě, jsou dostupné ihned bez nutnosti připojování k serveru a poskytují plnou funkčnost i bez připojení k internetu. Nevýhodou pak ale je přístup pouze z jednoho zařízení a s každým dalším je nutné software znovu instalovat, dokonce může být používání aplikace na více strojích i zpoplatněno. Další nevýhodou je právě nutnost instalace a možné problémy s kompatibilitou, jelikož tyto programy nejsou vždy multiplatformní.[2][7]

2.2.3 Kombinace webových a desktopových aplikací

Jedná se o programy fungující na bázi klient-server, kdy si uživatel nainstaluje daného klienta ve formě aplikace (nebo případně plug-in do prohlížeče), která mu poskytne nejen výhody desktopové aplikace včetně možnosti využívat kalendář v „offline“ režimu,

ale i poskytuje stejně jako webové řešení možnost přístupu pouze pomocí prohlížeče, tedy je možné si kalendář zobrazit nehladě na platformu nebo zařízení. [8]

2.3 Vzájemná kompatibilita

Díky velkému množství různých aplikací, které se zabývají tématem kalendářů a plánování času a zdrojů vznikla potřeba stanovení určitého standardu, podle něhož bude možné kalendáře sdílet mezi různými programy a platformami.

Z tohoto důvodu byl v roce 1998 poprvé definován standard RFC 2445 s názvem iCalendar společností Internet Engineering Task Force, který určil podobu datového formátu bez požadavků na práci s ním. Tento standard byl v letech 2009 a 2016 redefinován (jako RFC 5545 a RFC 7986) z důvodu jak opravy nedostatků předchozí verze, tak z důvodu přidání dalších údajů. V dnešní době je iCalendar široce rozšířen a je podporován aplikacemi jako například Google Calendar, Apple Calendar nebo Outlook kalendářem v rámci Microsoft Office 365.[9]

2.4 Přehled jednotlivých aplikací

V této části práce jsou shrnuty základní informace o konkrétních aktuálně dostupných kalendářích, rozdělených dle výše uvedených kritérií.

2.4.1 Desktopové aplikace dostupné zdarma

- Osmo

Tato aplikace je poskytována plně zdarma, avšak pouze pro Linuxové platformy, jedná se o kalendář poskytující základní funkcionalitu jako upomínky, poznámky, správu kontaktů, jednoduché textové vyhledávání a nastavitelné GUI. Podpora importu a exportu dat z a do jiných aplikací je poměrně omezena, v aktuální verzi je pouze možná určitá spolupráce s kalendáři využívajícími standard iCalendar. [9]

- Desktop iCalendar Lite

Podobně jako Osmo tento software je poskytován zdarma, nástroj je velmi intuitivní a jeho vzhled je dobře upravitelný. Kalendář obsahuje základní nástroje jako například upomínky a seznamy úkolů, avšak tyto funkce jsou v nástroji pro co největší

jednoduchost odděleny, což ale ubírá na přehlednosti. Program je určen pro platformu Windows a je plně kompatibilní se standardem iCalendar a tím je možný import a export dat z kalendářů jako Google Calendar, Thunderbird nebo Outlook Calendar.[11]

- Apple Calendar

Aplikace Apple Calendar je vestavěnou součástí systémů OS X a iOS, a je plně zdarma a v základu obsahuje dva kalendáře: osobní a pracovní, avšak uživatel si může vytvořit libovolné množství dalších a je možné tyto kalendáře zobrazovat jak samostatně tak hromadně. Všechna data je možné udržovat aktuální díky jejich sdílení přes iCloud, stačí být pouze přihlášen ke svému účtu. Výhodou je možnost přidávání událostí z jiných služeb jako Facebook nebo Yahoo, avšak největším důvodem pro pořízení tohoto kalendáře je velmi propracovaný systém Autocomplete, který podobně jako našeptávače v internetových prohlížečích nabízí údaje, které může uživatel hledat, jako například kontakty nebo adresy restaurací, kalendář také nabízí malou mapku pro zobrazování míst událostí nebo informace o počasí a předpokládané časové náročnosti cest. Dále samozřejmě najdeme základní výbavu jako notifikace, zobrazování narozenin, svátků a událostí a také k zveřejňování svých kalendářů dalším uživatelům.[12]

2.4.2 Webové aplikace dostupné zdarma

- Google Calendar

Jedná se o pravděpodobně nejrozšířenější elektronický kalendář vůbec. Oblíbenost si získal hlavně díky možnosti mít na jednom místě více různých kalendářů a jejich snadné prolínání a skrývání, uživatel si tak může snadno v jednom organizovat jak osobní tak pracovní rozvrhy. Mimo základní funkce poskytuje rozmanitou škálu dalších možností jako zobrazování svátků, vytváření událostí z mailů, vytváření úkolů přes Google Tasks, nahlížení do pracovních kalendářů spolupracovníků nebo posílání pozvánek, avšak při větším množství naplánovaných akcích pokulhává přehlednost zobrazení. Přímou desktopovou aplikaci chybí, avšak každý telefon s operačním systémem Android ji má v továrním nastavení a Google Calendar taky velmi dobře spolupracuje s iCalendar standardem a jinými desktopovými aplikacemi jako například Asana. Navíc fakt, že se jedná o webovou formu kalendáře, zajišťuje přístup z libovolné

platformy. Google Calenar je plně zdarma, k přístupu do něj stačí mít účet Gmail, dokonce je kalendář s emailem uživatele propojen natolik, že může vytvářet události jen na základě příchozích mailů obsahujících například vstupenky. [8][13]

2.4.3 Placené desktopové aplikace

- Fantastical 2

Fantastical poskytuje příjemné prostředí plně nastavitelných barev s kvalitním grafickým zpracováním a zároveň poskytuje i standartní funkcionalitu, tedy psaní poznámek, vytváření upozornění, poznámek a díky možným rozšířením dokáže přebírat události z jiných aplikací a doplňovat je o informace o například adresách nebo dostupnosti osob, se kterými jsou konkrétní události sdíleny. Fantastical 2 je dostupný pouze pro Mac a iOS a poskytuje zkušební dobu tři týdnů zdarma.[8]

- Outlook

Outlook Calendar je jedním z vůbec nejčastěji používaných a nejvíce robustních nástrojů tohoto typu, jeho výhodou je, že je součástí e-mailové aplikace, která dále obsahuje možnost správy kontaktů, upomínek, plánování úkolů a uživatel má vše dostupné v rámci jednoho programu a vše lze přehledně zobrazit v jednom náhledu. Outlook obsahuje funkci sdílení jednotlivých kalendářů mezi uživateli, a to nejen nahlížení do ostatních sdílených kalendářů ale i jejich správu či vytváření poznámek a upomínek k nejen svým ale i sdíleným akcím. Tuto funkcionalitu týmového plánování ocení zvláště větší organizace a společnosti. Další výhodou je nastavitelné zobrazení více kalendářů, případně i jejich vzájemné prolínání, celý vzhled je přitom přizpůsobitelný uživatelem včetně barev nebo nastavení písma. Zjednodušená verze kalendáře je také dostupná ve webové podobě a v aplikacích pro iOS a Android, avšak obsahují pouze základní funkce. Aplikace pro desktop je součástí balíčku Microsoft Office 365, který je možné platit jak měsíčně tak ročně. Verze Outlooku pro iOS, Android a web jsou však zdarma.[8][14]

2.4.4 Placené webové aplikace

- Cozi

Cílová skupina uživatelů Cozi jsou domácnosti a proto je více odlišná od ostatních kalendářů. Poskytuje velmi jednoduché prostředí s množstvím zajímavých funkcí jako například nákupní lístky, recepty nebo možnost vytvářet si elektronický rodinný deník. Což ale neznamená, že je aplikace o to horším kalendářem, poskytuje v této oblasti všechny základní prvky jako upomínky, seznamy aktivit, jejich sdílení s ostatními uživateli, vyhledávání v kalendářích a správu událostí. Výhodou je multiplatformní přístup a verze zdarma s omezenou funkcionalitou a reklamami.[15][16]

2.4.5 Kombinace webové a desktopové aplikace

- Thunderbird

Jedná se o zdarma dostupný plug-in do internetového prohlížeče Firefox sloužící ke správě e-mailů, tvorbu a úpravu kalendářů a správu kontaktů. Kalendář také obsahuje možnost zobrazování svátků, vytváření událostí a pozvánek na ně pro další uživatele, oproti ostatním kalendářům má výhodu v detailně popsanych návodech ke všem funkcím a samotná skutečnost, že se jedná o doplněk prohlížeče, poskytuje uživateli značný komfort v možnosti mít vše na jednom místě a zároveň je možné jej používat bez připojení k internetu. Navíc Thunderbird disponuje otevřeným kódem, takže je možné si jej libovolně upravovat.[17]

2.5 Shrnutí

Nabídka kalendářů je velmi rozmanitá, aplikace poskytují širokou nabídku možností pro správu časového rozvrhu uživatele, některé z těchto funkcí můžeme vidět prakticky u všech kalendářů, jiné jako například sdílení kalendářů mezi více uživateli jsou výsadou pouze některých řešení. Avšak kalendáře jsou v podstatě dimenzovány více jako osobní než pracovní a správa většího množství zaměstnanců z pozice nadřízeného je v nich prakticky nemožná. Zároveň jsou tyto aplikace nevyhovující pro požadované řešení z důvodu komplikovaného zpracování dat pro systém PERM 3, pouze některé z kalendářů podporují standard iCalendar, umožňující určitý import a export událostí, avšak tento standard

nedefinuje velké množství pro společnost Kvasar důležitých položek jako docházkové kódy, zdroje, PPV a rozvrhy směn.

3 PROSTŘEDÍ A JAZYK

Tato kapitola se zabývá stručným představením jazyka a vývojového prostředí, ve kterých je realizována aplikace PERM 3 a spolu s ní i plánovací modul, jež je předmětem této práce. V poslední podkapitole je představen formát *XML*.

3.1 Vývoj Pascalu

3.1.1 Vznik jazyka

Pascal se stal nástupcem vůbec prvního vysokoúrovňového jazyka se strukturovanou a systematicky definovanou syntaxí Algolu. Byl navržen Niklausem Wirthem v roce 1971 a implementován byl v roce 1973. Jazyk byl navržen jako výukový nástroj pro studenty informatiky a oproti Algolu obsahoval navíc dynamické struktury nebo širší škálu datových typů.

3.1.2 Turbo Pascal

Deset let po implementaci jazyka vyšla první verze Turbo Pascalu od společnosti Borderland. Turbo Pascal pokračoval se stejným jádrem jako původní pascal, poskytl ale uživateli integrované vývojové prostředí, ve kterém mohla probíhat editace kódu, kompilování a snadné zobrazování chyb, což vedlo k enormnímu úspěchu a Turbo Pascal se stal jedním z nejprodávanějších kompilátorů.

3.1.3 Delphi

Po předchozím úspěchu vyšla v roce 1995 první verze pojmenovaná Delphi, která rozšířila původní jazyk o objektově orientovaný přístup a formulářovou strukturu, velkou výhodou byl také velmi rychlý kompilátor. Postupem času získalo Delphi celou řadu dalších verzí a rozšíření, ve kterých byly zpřístupněny nové balíčky komponent a řadu nových vlastností jako například dynamická pole, přetěžování metod nebo výběr mezi statickým a dynamickým linkováním knihoven.[18]

Následně avšak přišlo pro společnost Borderland v roce 2007 krizové období a vznikla nová firma CodeGear ve které pokračoval vývoj Delphi, šlo sice z počátku o dceřinou společnost Borderlandu, avšak záhy byla odkoupena společností Embracadero Technologies.

3.1.4 Embracadero

Embracadero dodnes pokračuje ve tvorbě vývojových nástrojů, zásadním zlomem ve vývoji bylo vydání verze XE2, ve které probíhal vývoj tohoto modulu. Tato verze přinesla očekávaný 64bit kompilátor Delphi, kompilátor pro OS X a možnost využití VCL stylů.[19]

3.2 Jazyk Object Pascal

Object Pascal je vysokoúrovňový a objektově orientovaný jazyk, poskytující rozšířenou podporu generického programování. Tento jazyk není tzv. „case sensitive“, tedy nezáleží na tom, zda jsou písmena malá nebo velká. [20]

A ač se nejedná o nejrozšířenější jazyk pro vývoj programů v dnešní době, stále se využívá ke tvorbě uživatelského prostředí v řadě komerčních produktů, zvláště kvůli umožňuje vytvořit profesionálně vypadající „front-end“ podobně jako například Visual C++. Další jeho velkou výhodou je přehlednost kódu, který je velmi dobře strukturován a je snazší jej číst.[21]

3.2.1 Objektový přístup

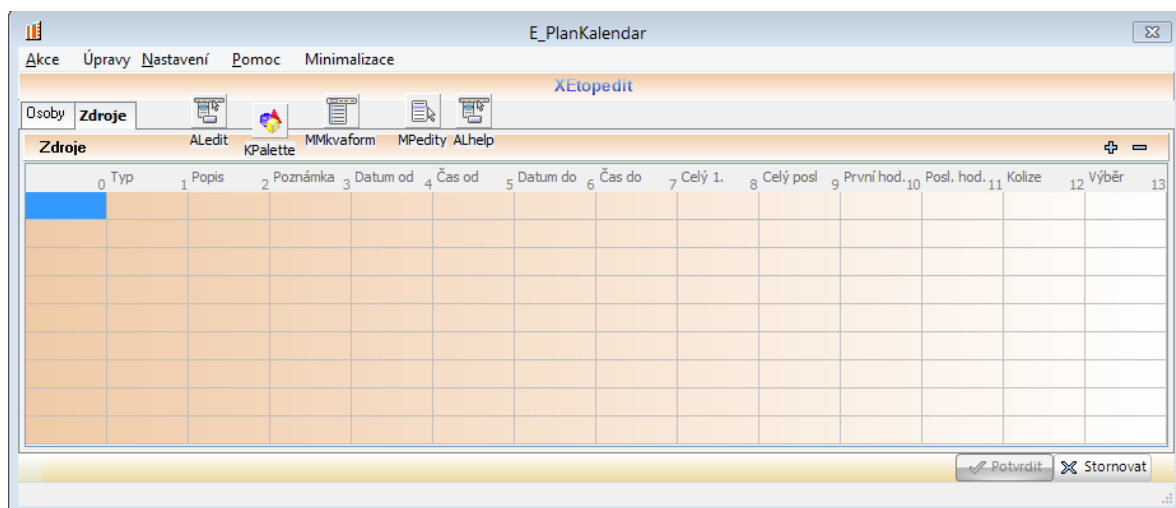
Jak bylo již v této práci uvedeno, Object Pascal je objektově orientovaný jazyk. Toto programovací paradigma má za cíl rozdělit kód na malé, ideálně do určité míry samostatné a opakovatelně použitelné jednotky-tedy objekty. Účelem je zjednodušení kódu samotného, jeho úpravy i jeho lepší přehlednost. Hlavními pojmy jsou v objektově orientovaném programování (OOP) jsou objekty a třídy, kdy objektem se rozumí jedna instance třídy, kdy třída vymezuje určitou skupinu předem definovaných vlastností, jež mohou prvky tohoto typu nabývat. Tedy například bude třída *zaměstnanec* mít vlastnosti *jméno* a *příjmení* a objekt typu *zaměstnanec* bude mít vlastnost *jméno* s hodnotou „Jan“ a vlastnost *příjmení* s hodnotou „Smutný“. Objekty jsou určitým zapouzdřením konkrétních atributů a metod pod jednu samostatnou položku, tento přístup usnadňuje mimo orientace v kódu také jejich sdílení a používání v dalších částech programu.[22]

Základním prvkem OOP je dědičnost, objekty mohou dědit od svých předků jak hodnoty jednotlivé proměnné, tak i metody. S tímto pojmem úzce souvisí také viditelnost, atributy a metody mohou mít v Delphi 3 různé úrovně viditelnosti-*private* je nejpřísnějším omezením viditelnosti, tyto prvky nejsou dostupné kdekoli mimo jednotku, na které jsou

definovány. *Protected* umožňuje viditelnost dané vlastnosti či metody jen pro přímé potomky, poslední úroveň je *public*, jež dovoluje volný přístup. Toto řešení je snahou o jasné oddělení a skrytí implementační části a zpřístupnění pouze rozhraní. [23]

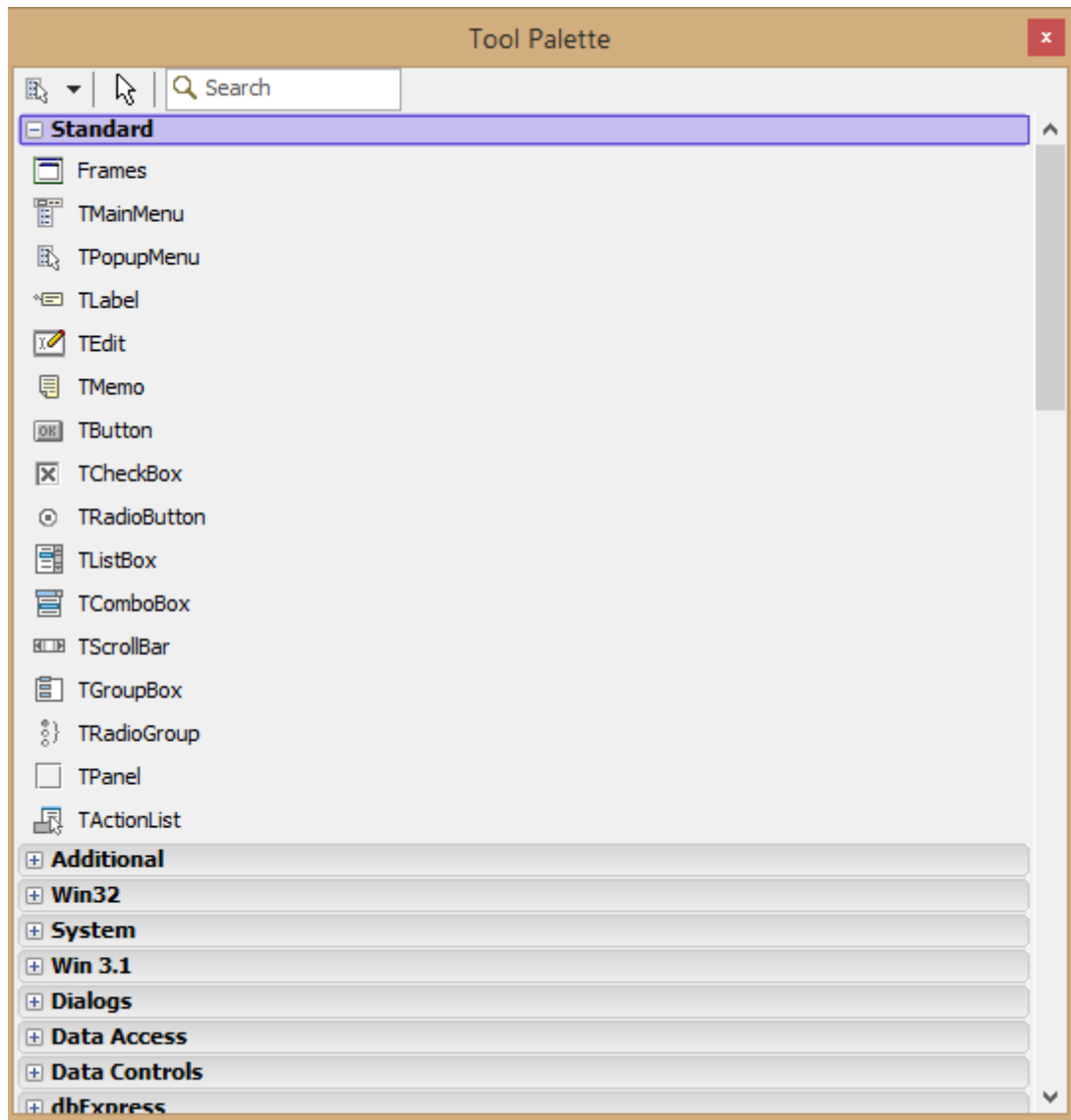
3.2.2 Formuláře

Formuláře jsou v prostředí Delphi základním nástrojem pro vizualizaci a pro zpracování vstupních údajů od uživatele programu. Formuláře mohou být využity nejen pro hlavní okno aplikace, ale i k zobrazování podřízených formulářů nebo dialogových oken. Při vytvoření nového okna v aplikaci je vytvořen i soubor s příponou *.dfm*, skrze který umožňuje v rámci vývojového prostředí přístup do Form Designeru. Form Designer poskytuje základní náhled na formulář a přístup do Object Inspektoru, ve kterém jsou zobrazeny vlastnosti a události formuláře, vlastnosti jsou přímo editovatelné, události se po kliknutí odkazují do kódu.



Obrázek č. 1 - Zobrazení formuláře skrz Form Designer

Formulář je v prostředí Delphi třídou, je tedy umožněna dědičnost mezi a jsou původně odvozeny od předka *TForm*, od kterého dědí množství základních vlastností a událostí, které poskytují vývojáři celou řadu možností, jak zpracovat vizuální stránku aplikace a vstupní data. Nabídka dostupných nástrojů se nachází pod záložkou Tool Palette a je možné zde najít například tlačítka, nadpisy, dialogová okna a pro programátora jednou z nejpodstatnějších položek a to textová pole.[24]



Obrázek č. 2 – Tool Palette

Skrz ně může aplikace zobrazovat jakákoli data, ukládat změny provedené uživatelem a využívat celou řadu dostupných metod nad tímto polem. Pro tyto textová pole je použita VCL komponenta *TAdvStringGrid* společnosti TMS software, tato komponenta poskytuje řadu vlastností a událostí pro práci s mřížkou formuláře a rozšiřuje tedy původní mřížku poskytnutou vývojovým prostředím.[24][25]

3.3 Formát XML

XML je zkratkou pro extensible markup language, tedy rozšiřitelný značkovací jazyk. Jedná se o formát dokumentů, který určuje způsob zápisu dat a jejich významu ve formě, která je čitelná jak strojem, tak člověkem. Tento formát slouží hlavně ke

komunikaci mezi aplikacemi, případně pro prezentaci dokumentů. V rámci této bakalářské práce jsou vytvářeny *XML* soubory právě za účelem komunikace mezi aplikací PERM 3 a prostředím PermWeb. *XML* je svojí strukturou podobný *HTML* kódu, avšak na rozdíl od *HTML* nemá žádné předem definované značky, pouze jsou stanovena určitá pravidla formátování textu, což umožňuje snadnou práci s daty, kdy například v názvu značky bude definována položka a v elementu její hodnota.[26]

3.3.1 Cíle formátu XML

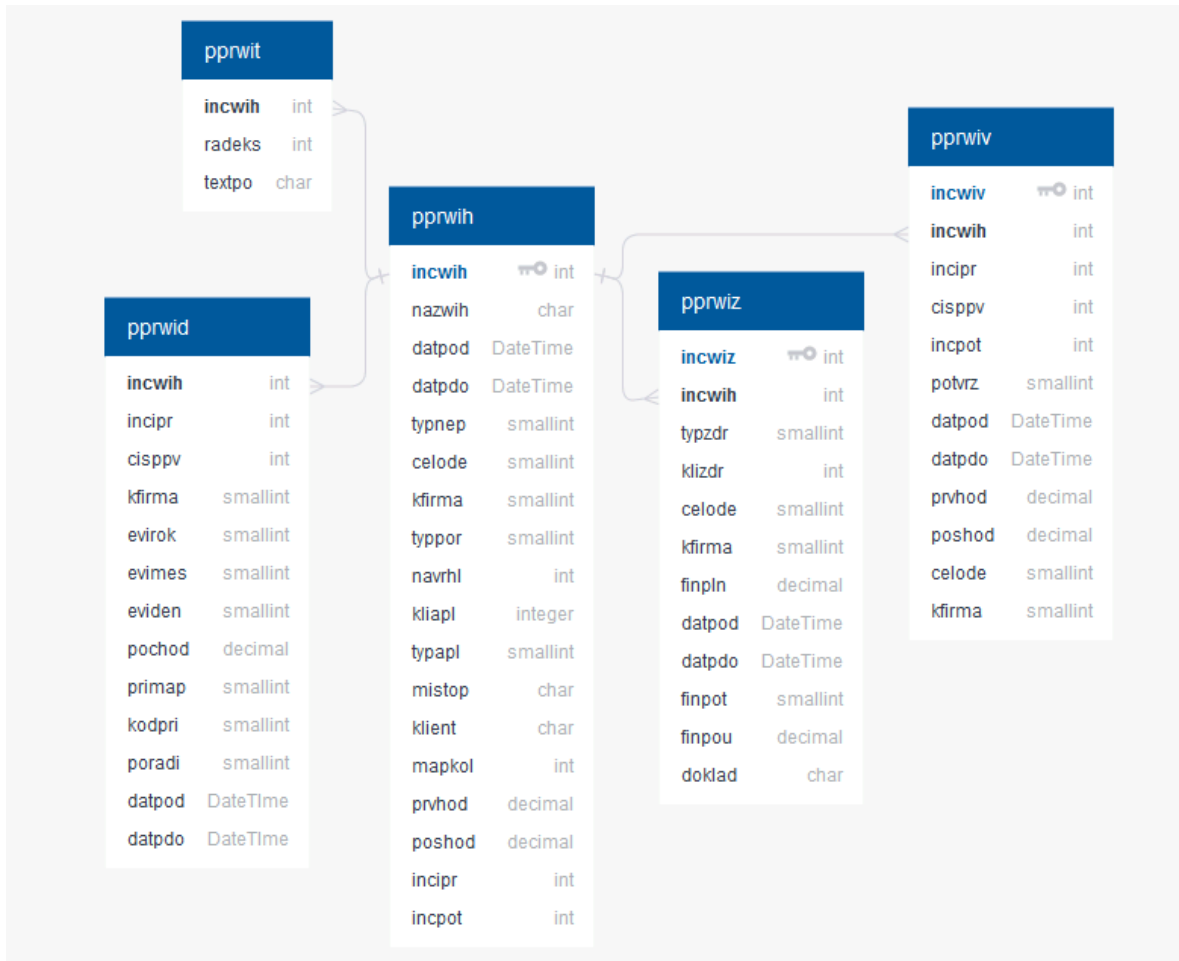
- Široká využitelnost mezi aplikacemi
- Kompatibilita se svým předchůdcem formátem *SGML*
- Jednoduchost programů na vytváření a zpracovávání *XML* souborů
- Minimum volitelných vlastností
- Formální a stručný design[27]

II. PRAKTICKÁ ČÁST

4 DATOVÉ STRUKTURY

V této kapitole se nachází představení datové struktury a datový model tohoto projektu. Plánovací modul využívá celkem 5 následujících tabulek:

- *pprwih* – hlavička naplánované události, primárním klíčem je autoinkrementální položka *incwih*, tato tabulka neobsahuje cizí klíče na tabulky v rámci tohoto modulu, avšak obsahuje řadu cizích klíčů v rámci aplikace
- *pprwiv* – vztažené PPV pro danou událost, primárním klíčem je stejně jako u předcházející tabulky autoinkrementální položka *incwiv*, cizím klíčem v rámci tabulek tohoto modulu je *incwih* ukazující na tabulku *pprwih* vztahem 1:N, dalšími důležitými cizími klíči jsou číslo PPV *cisppv* a interní číslo pracovníka *incipr*
- *pprwiz* – přidělené zdroje k plánu, primárním klíčem je znovu autoinkrementální položka-*incwiz*, stejně jako u minulé tabulky je cizím klíčem *incwih*
- *ppwit* – nejjednodušší tabulka slouží pro textový popis plánu pracovníka, primárním klíčem je kombinace položek *incwih* a *radeks*, cizím klíčem je znovu *incwih*
- *pprwid* – tabulka obsahující mapu dnů sloužící pro urychlení čtení dat, primárním klíčem je kombinace řádků *incipr*, *incwih*, *cisppv*, *kfirma*, *eviden*, *evimes*, *evirok*, *poradi*, cizím klíčem jsou *incwih*, *cisppv*, *incipr* a *kfirma*



Obrázek č. 3 – Databázová struktura modulu

Jednotlivé položky obsažené v těchto tabulkách odpovídají konkrétním vlastnostem objektů, a jsou proto popsány v následující kapitole, jež se zabývá implementační dokumentací modulu.

5 IMPLEMENTAČNÍ DOKUMENTACE MODULU

Tento modul se dá rozdělit do dvou hlavních částí: první je skupina tříd, které zajišťují aplikační logiku kalendáře, komunikaci s databází a veškerou funkcionalitu, tvoří tedy jakousi spodní vrstvu modulu. Druhou, prezentační vrstvou je několik formulářů, které slouží k vytváření, prezentaci a editaci jednotlivých plánů a jsou popsány v následující kapitole. Vrstvy modulu jsou jasně odděleny a to z důvodu možnosti jejího znovupoužití, například pro zpracování dat z *XML* souborů určených ke komunikaci s prostředím PermWeb.

5.1 Vlastnosti hlavních tříd

V této podkapitole jsou představeny jednotlivé vlastnosti tříd a hlavní procedury tohoto modulu jsou pro přehlednost popsány v následující podkapitole. Obě kapitoly popisují kód programu obsažený v první příloze *P3ObjPWI*. Struktura vlastností tříd odpovídá databázové struktuře tohoto projektu, proto je tedy využito několika dvojic tříd, kde první definuje seznam a druhá jeho jednotlivé položky.

5.1.1 TkvaWebPraPlan

Tato třída popisuje jednotlivé položky třídy seznamu *TkvaWebPraPlanList*. V databázi proměnným odpovídá tabulka *pprwh* a jedná se tedy o třídu, jež definuje hlavičku naplánované události. Veřejná část obsahuje 4 následující třídy:

- *WebPraZdrojList*: Jedná se o třídu, která je seznamem všech zdrojů, které jsou konkrétní akci přiděleny.
- *WebPraVztPPVList*: Podobně jako u přechozí třídy se jde o seznam, tentokrát všech vztažených PPV, tedy jinými slovy se jedná o seznam všech účastníků této akce.
- *WebPraDenMapaList*: Tato třída je opět seznamem, v tomto případě položkami jsou informace o jednotlivých dnech všech konkrétních akcí. Účelem této třídy je možnost prezentovat den po dni jednotlivé události a vytvořit tím určitou „mapu“ ve které bude snazší orientace a umožnění rychlejšího čtení databáze.
- *Popis*: Třída obsahující textový popis konkrétního plánu, v podstatě jde o nahrazení proměnné typu *string* a toto řešení bylo zvoleno převážně z důvodu většího rozsahu poznámky a možnosti lepší práce s textem než u obyčejné proměnné.

Dále vedle uvedených tříd se zde nachází také celá řada proměnných, kterým výhradně odpovídají záznamy v tabulce *pprwih*, ty s největším významem jsou popsány v následujícím seznamu:

- *incwih*: ID konkrétního plánu, v databázi se jedná primární klíč a i v tomto modulu slouží k jednoznačné identifikaci plánu.
- *datpod*, *datpdo*: proměnné typu *DateTime*, které obsahují datum i konkrétní čas začátku a ukončení dané akce.
- *typnep*: Docházkový kód určující o jakou akci se jedná. Zda jde například o dovolenou, nemocenskou, pracovní cestu nebo jde o řádně odpracovanou pracovní dobu, což může být například využito při plánování porad apod.
- *prvhod*, *poshod*: údaj o počtu hodin dne prvního a posledního na dané události. Položka slouží k určení přesného počtu hodin v případech, kdy se jedná o vícedenní událost a není tedy možné z intervalu poskytnutého položkami *datpod* a *datpdo* přesně určit délku jejího trvání.
- *celprv*, *celpos*: úzce souvisejí s předchozími proměnnými, jedná se o příznaky, zda je první a/nebo poslední den dané akce celodenní.
- *nazwih*: název události.

5.1.2 TkvaWebPraZdroj

Opět se jedná o popis položek jedné třídy seznamu, tentokrát *TkvaWebPraZdrojList*. V databázi se tato třída odkazuje na tabulku *pprwiz* a jsou zde definovány fyzické zdroje, jež jsou přiřazeny konkrétní události. Každá událost může mít libovolné množství přiřazených zdrojů, jak finančních tak fyzických.

Na rozdíl od předchozí třídy tato neobsahuje žádné další podřízené objekty, takže je možné rovnou přejít k představení nejvýznamnějších vlastností:

- *incwiz*: ID zdroje, v databázi je primárním klíčem.
- *incwih*: ID nadřazeného plánu, ke kterému je zdroj přiřazen, databázově tedy jde o cizí klíč. V rámci jednoho plánu je vždy tento údaj stejný u všech jeho zdrojů a v rámci tohoto modulu je používán výhradně pro komunikaci s databází, jež je popsána v následující kapitole.

- *datpod, datpdo*: Tyto proměnné určují časový interval využívání konkrétního přiděleného zdroje na plánovanou událost. Tento interval nemusí odpovídat intervalu z třídy *TkvaWebPraPlan*, a to z důvodu, že zdroj může být poskytnut pouze po omezenou dobu.
- *prvhod, poshod*: Stejně jako u předchozí třídy se jedná o konkrétní určení počtu hodin prvního a posledního dne.
- *celprv, celpos*: Opět se jedná o stejnou položku jako v třídě hlavičky plánu.
- *typzdr*: Kód určující typ zdroje. Aktuálně tento modul nabízí 3 typy zdrojů: peněžní zdroje, svěřené věci a prostory.
- *klizdr*: Klíč definující konkrétní svěřenou věc nebo prostor. V případě peněžního zdroje se tato položka nevyužívá.
- *Finpln*: Plánovaná finanční částka přiřazená na událost
- *Kolize*: Příznak, že daný zdroj (netýká se financí) je přiřazen na jiný plán ve stejném časovém intervalu.

5.1.3 TkvaWebPraVztPPV

Tato třída popisuje položky seznamu *TkvaWebPraVztPPVList* a odpovídá jí tabulka *pprwiv*. Tato třída popisuje vztažené PPV k události, jde tedy o seznam osob, které se budou účastnit plánované akce s doplněním možnosti, že v případě dvou pracovních poměrů v rámci jedné společnosti je možné vybrat jeden, kterého se týká (např. pracovní cesta) případně doba (například u nemocenských).

Přehled nejvýznamnějších vlastností:

- *incwiv*: ID vztaženého PPV, v databázi položce odpovídá stejnojmenný primární klíč.
- *incwih*: Jako u zdrojů jde o ID nadřazeného plánu.
- *incipr*: Interní číslo pracovníka, každý pracovník má v aplikaci PERM 3 přiřazené jedno toto číslo které slouží k jeho jednoznačné identifikaci. Díky tomuto číslu nejsou v tomto modulu položky jako například jméno a příjmení účastníka události, protože by se jednalo o zbytečnou redundanci.
- *cisppv*: Toto číslo určuje PPV, v rámci kterého se pracovník účastní události. Jak je zmíněno výše, je možné, aby byl jeden pracovník dané události přiřazen

dvakrát, avšak právě toto číslo musí být pro každý záznam se stejnou položkou *incipr* rozdílné.

- *kolize*: Podobně jako u zdrojů tato položka značí, zda má pracovník ve stejný čas dvě překrývající se události. Kolize, respektive její ověřování se nevztahuje k položce *cisppv* ale pouze k položce *incipr*. Výjimkou je případ, kdy oba kolizní záznamy náleží stejnému plánu.
- *datpod*, *datpdo*: Proměnné určující datum a čas, po který se dané události pracovník účastní. Z intervalů všech přiřazených PPV na události se vytváří interval *datpod-datpdo* v nadřazeném objektu konkrétního plánu typu *TkvaWebPraPlan*.
- *celprv*, *celpos*: Příznaky určující zda první a poslední den, ve kterých je účastník přítomen na plánované události celý den.
- *prvhod*, *poshod*: V případě, že jsou *celprv* a/nebo *celpos* na hodnotě *false* tyto proměnné určují, jak dlouho je pracovník přítomen.
- *potvrz*: Příznak, zda je účast na události pracovníkem potvrzena. Této položky se využívá v rámci zpracování XML souborů pro webového rozhraní, kdy je uživatelům umožněno zvát na dané události další pracovníky, u kterých je tedy třeba potvrzení účasti. V rámci aplikace PERM 3 se tato vlastnost nevyužívá.

5.1.4 TkvaWebPraDenMapa

Tato třída je prvkem listu *TkvaWebPraDenMapaList* a v databázi jí odpovídá tabulka *pprwid*. Jedná se třídu, která obsahuje informace o jednom dni v rámci plánované události. V rámci logiky modulu slouží hlavně jako nástroj umožňující rychlé čtení databáze a k prezentaci kalendáře v prostředí aplikace PERM 3. Jednotlivé objekty tohoto typu jsou plněny výhradně procedurou *FillDenMapa* (jež je popsána v této práci níže) a přímo vychází z intervalů jednotlivých PPV vztažených ke konkrétnímu plánu.

Vlastnosti objektu:

- *incipr*, *cisppv*: Podobně jako u předchozí třídy se jedná o přesné určení uživatele a jeho PPV.
- *incwih*: ID plánu, pod který tento objekt spadá.

- *evirok, evimes, eviden*: Určení konkrétního dne. Těchto prvků je využito zvláště kvůli zjednodušení a zrychlení komunikace s databází, kdy je možné hledat a/nebo spravovat prvky spadající do určitého období bez používání nepružného formátu `TDateTime`.
- *pochod*: Číslo určující počet hodin strávených daný den účastníkem na události.

5.1.5 Popis

Popis je v rámci tříd tohoto modulu menší výjimkou, je vlastností třídy *TkvaWebPraPlan* typu *TkvaText*, tento typ byl poskytnutý společností Kvasar a je součástí konvenčního řešení poznámek v aplikaci PERM 3. Tato konkrétní třída *Popis* slouží jako poznámka k plánu a jeho nejdůležitějším atributem je list záznamů typu *string Radek*. Tohoto se využívá pro možnost nahrávat neomezeně dlouhé komentáře a pro jednodušší vyhledávání v nich.

5.2 Hlavní funkce a procedury modulu

Následuje popis hlavních funkcí a procedur tohoto modulu. Zařazení funkcí do samostatné podkapitoly bylo zvoleno zvláště z důvodu přehlednosti této práce, jelikož některé pracují s vlastnostmi ostatních tříd a jejich popis by jinak byl značně komplikovaný.

5.2.1 Accessory

Výše zmíněné třídy obsahují několik privátních funkcí, které jsou takzvanými accessory, neboli gettery a slouží k získání konkrétních hodnot vlastností. Každá třída má několik takových funkcí, jednu pro každý datový typ. Vstupním parametrem těchto funkcí je vždy *Index* typu *integer* určující, o kterou položku se jedná a návratový typ je shodný s datovým typem, pro který je tato funkce určena. Realizace accessorů je poměrně jednoduchá: jeden *case*, jehož výrazem je vstupní parametr *Index* a pro každý vstup zvolí konkrétní prvek, který se zapíše na výstup. Výjimkou jsou metody *GetSmallintBit*, jež nevyužívají *case*, ale čtou z položky typu *Smallint* konkrétní bit, který je určen indexem a návratová hodnota je tedy typu *boolean*.

5.2.2 Mutatory

Mutatory (nebo také settery) plní opačnou funkci než accesory, tedy mají zapisovat do vlastností tříd. Řešení těchto metod je však podobné, opět každá třída má samostatnou proceduru pro každý datový typ a vstupním parametrem je opět *Index* odkazující na konkrétní položku, návratový typ je nahrazen dalším parametrem *Value*, který nese hodnotu, jež má být zapsána do dané vlastnosti.

5.2.3 NewPlan

Tato veřejná funkce třídy *TkvaWebPraPlanList* slouží k vytváření nových plánů, vstupními parametry jsou proměnné typu *integer* *Alncipr* a *ACisppv*, které určují konkrétního pracovníka a jeho PPV a *ADatum* typu *TDateTime* určující pro který den je nová událost naplánována. Funkce vedle nového záznamu v listu *TkvaWebPraPlanList* vytváří na základě vstupních parametrů také nový záznam pro podřazený list vztažených PPV skrz funkci *NewWebPraVztPPV*. Funkce má návratový typ *integer* a vrací index, na kterém je v listu nově vytvořený záznam, pokud dojde k chybě je tato hodnota nastavena na -1.

5.2.4 NewWebPraVztPPV

Veřejná procedura třídy *TkvaWebPraPlan* přidává plánu nový záznam do listu vztažených PPV. Jako předchozí funkce má parametry *Alncipr* a *ACisppv* pro určení pracovníka a jeho PPV a *ADatum* pro určení data, ve kterém se zúčastní události. Dále jsou zde navíc dvě variabilní proměnné typu *integer*. První z nich *ACHyba* slouží pro jako příznak, zda nebyl přiřazen k události účastník, který na ní už přiřazen je, druhá proměnná *Index* definuje, jak její název napovídá, na které pozici v listu se nachází duplicitní položka. Pokud tedy na této události nový účastník není nalezen, je listu nový záznam typu *TkvaWebPraVztPPV* a jsou mu přiřazeny identifikační hodnoty *incipr* a *cisppv* ze vstupních parametrů. Příznaky *celprv* a *celpos* jsou nastaveny na *true* a vlastnosti určující datum a čas jsou získány funkcí *GetDobaSmeny* z rozvrhu směn konkrétního pracovníka. Na závěr je volána procedura *FillDenMapa* pro přeplnění denní mapy.

5.2.5 NewWebPraZdroj

Tato procedura je podobná s procedurou předchozí, slouží k vytvoření nové položky v listu zdrojů. Vstupní parametry obsahují *integer* proměnné *Aklizdr* a *Atypzdr*, které slouží

k jednoznačnému určení konkrétního zdroje a další proměnné *ACHyba* a *Index* mají stejný význam jako u předchozí procedury s výjimkou peněžních zdrojů, kterých může být libovolné množství, a nemají svůj vlastní klíč. Jelikož zdroje nemají vlastní rozvrh směn, tato procedura nemá vstupní parametr, který by určil interval data. Ten je přebrán z hlavičkového objektu typu *TkvaWebPraPlan*, kterému je zdroj přiřazen.

5.2.6 GetDobaSmeny

Jedná se o privátní proceduru listu *TkvaWebPraPlanList*, která slouží k získání plánované doby směny pracovníka pro konkrétní den. První dva vstupní parametry *Aincipr* a *Acisppv* určují pracovníka a jeho pracovní poměr, na kterém tuto směnu funkce hledá, další parametry *ARok*, *AMes* a *Aden* stanoví den směny. Další variabilní parametry *ACasZac*, *ACasKon* a *APochod* jsou využívány jako výstup z této funkce, tedy pro čas začátku a konce směny a počet odpracovaných hodin.

5.2.7 FillDenMapa

Tato metoda třídy *TkvaWebPraPlan* má široký význam. V první řadě slouží k přeplnění položek listu *TkvaWebPraDenMapaList* po změně plánu. Cílem je vytvořit pro každý den události pro každého jejího účastníka jeden záznam v tomto listu. Dalším úkolem procedury je určit počátek a konec dané události a ověřit, zda některý z jejích účastníků nemá jinou naplánovanou kolidující událost. Jedná se o proceduru bez vstupních parametrů, na jejím začátku se ověří status objektu, a pokud došlo k nějaké změně, vymaže se celý list denní mapy skrz funkci *clear*.

Následně postupně prochází v cyklu *for* seznam všech vztažených PPV a pro každou položku je volána funkce *CheckKolize* pro ověření kolizí a také jsou přeplňovány hranice intervalu tohoto plánu. Pro každou položku daného listu se z intervalu jejích proměnných *datpod* a *datpdo* spočítá počet dní a pro každý se vkládá nový záznam do listu denní mapy. Tento nový záznam obsahuje datum konkrétního dne, ke kterému se vztahuje a z objektu vztaženého PPV přebírá proměnné *incipr*, *cisppv* a *incwih*. V závěru cyklu metoda vyplní vlastnosti určující, zda se jedná o krajní dny intervalu, případně jestli jsou celodenní. V tomto případě se ještě vyplní vlastnosti určující časové rozmezí v rámci daného dne

pomocí funkce *GetDobaSmeny*, v opačném případě jsou položky vyplněny také z objektu vztažených PPV.

5.2.8 CheckKolize

Tato procedura třídy *TkvaWebPraPlan* má vstupní parametr interní číslo pracovníka *incipr* a slouží k ověření, zda tato osoba nemá dva nebo více kolizních plánů. Procedura obsahuje několik cyklů, nejdříve postupně prochází všechny existující plány a v nich všechny jejich vztažené PPV. Pro každý prvek typu *TKvaWebPraVztPPV* se stejným interním číslem pracovníka jako na vstupu se odkaz na tento prvek uloží do lokální proměnné *LWebPraVztPPV*, vyplní se mu vlastnost *kolize* na *false* a opět spouští dva cykly na průchod všech plánů a všech jejich podřízených PPV. Bohužel jelikož je možné mít na každém plánu přiřazeno více vztažených PPV se stejnou položkou *incipr* (díky možnosti mít událost, jež se vztahuje ke dvěma a více pracovním poměrům jednoho pracovníka) není možné využít efektivnějších vyhledávacích funkcí, které objekty v prostředí Delphi skrze své předky nabízejí. V průchodu následujících dvou cyklů se ignoruje plán, na němž byl nalezen první pracovník, z důvodu aby nebyla jako kolizní vyhodnocena situace, kdy je daný vztažený PPV porovnáván se sám sebou, případně s PPV vztaženým ke stejnému plánu (například kdy má uživatel dovolenou na obou svých pracovních poměrech). Pokud je tedy nalezen konkrétní *incipr* i na jiném plánu, srovnává se překryv data a v případě překryvu se oběma položkám nastaví *kolize* na *true*.

5.2.9 CheckZdrojKolize

Ověření kolizí pro svěřené zdroje je podstatně jednodušší než ověřování kolizí účastníků. Vstupními parametry jsou klíč zdroje *klizdr* a typ zdroje *typzdr*, pokud se jedná o peněžní zdroj (typ zdroje je roven 3), kolize se neověřují, jelikož peníze nemají konkrétní klíč, pro ostatní vstupy procedura v cyklu projde všechny plány a na jejich seznamech zdrojů pomocí funkce *Find* (jež je zděděna od předka třídy *listu zdrojů*) vyhledá zdroj, odpovídající vstupním parametrům. Pokud je nalezen u některého z plánů, je mu nastaven příznak *kolize* na *false*, odkaz na něj se uloží do pomocné proměnné *OverovanyWebPraZdroj* a stejným postupem, jako byl získán tento záznam, jsou vyhledány další zdroje se stejným klíčem a typem na ostatních událostech, pokud jsou nějaké nalezeny, je porovnán interval dat a v případě překryvů se oběma nastaví vlastnost *kolize* na *true*.

5.2.10 GetSimplePlanDleMapy

Tato funkce třídy *TkvaWebPraPlanList* slouží k získání všech událostí konkrétního pracovníka pro určité časové období ve formě pole objektů typu *TkvaWebPraPlan*. Vstupními parametry jsou interní číslo pracovníka a tři *smallint* čísla určující časové rozmezí, ve kterém se vyhledává *Aden*, *Ames* a *Arok*. Funkce vyhledává všechny plány pro jeden konkrétní den, avšak pokud je vstupní proměnná *Aden* rovna nule, vyhledávají se všechny události konkrétního měsíce, v případě že je rovna nule vstupní proměnná *Ames*, vyhledávají se plány v rámci zadaného roku.

5.2.11 GetPlanDleMapy

Funkce s téměř identickým významem jako ta předchozí, avšak pro jednotlivé položky výsledného pole jsou doplněny o informaci o jejich návaznosti na předcházející a/nebo následující záznam, cílem tohoto řešení je přehlednější grafická prezentace záznamů.

5.3 Třída pro SQL

Poslední třída *TkvaWebPraPlanListSQL* slouží ke správě databáze. Má jednu proměnnou *ReadType* typu *smallint* a 4 procedury pro čtení z databáze a zápis do ní.

5.3.1 Read

Tato procedura bez vstupních parametrů slouží ke čtení záznamů z databáze, k čemuž využívá objektu *Selecter* typu *TKvaSelecter*, třída *TKvaSelecter* byla poskytnuta společností Kvasar a je určena k správě komunikace programu s databází, konkrétně k načítání jednotlivých položek. Jedná se o standardní řešení společnosti a v rámci této třídy jsou jak formovány veškeré *SQL* dotazy tak je i zajištěno připojení k ní, díky čemuž je práce s databází podstatně jednodušší. Objektu *Selecter* jsou nejdříve vyplněny jeho vlastnosti, ze kterých se formují *SQL* dotazy na databázi. Mezi tyto vlastnosti patří konkrétní databáze a tabulka, se kterou se bude pracovat, názvy jednotlivých sloupců, a určení položek, podle kterých bude výsledek seřazen. Podmínka se vyhodnocuje na základě proměnné *ReadType*, která může nabývat čtyř různých hodnot: 0 standardní pro čtení všech záznamů, 1 pro čtení záznamů v aktuálním měsíci a hodnot 2 a 3 pro čtení pouze konkrétních plánů, určených vlastností *StringKey* naplněné dalšími procedurami této třídy, které jsou popsány níže.

Procedurou *OpenCursor* se otevře databázový kurzor a v cyklu *while* se postupně procházejí jednotlivé řádky tabulky *pprwih*, pro každý z nich je vložen do listu plánů *WebPraPlanList* a ve funkci *Transfer* se pak mapují položky tohoto řádku tabulky na vlastnosti objektu *WebPraPlan* dle jejich jména. Poté se pro každou podřazenou tabulku (tedy *pprwiit*, *pprwiz*, *pprwiv*, *pprwid*) stejným způsobem naplní odpovídající objekty s tím, že podmínkou je vždy jen příslušnost k danému plánu (tedy shodná položka *incwih*). V závěru cyklu se vybírá následující řádek tabulky procedurou *fetch*, pokud v tabulce další řádek není, je cyklus ukončen a databázový kurzor uzavřen.

5.3.2 ReadByDenMapa

Tato procedura má vstupní parametry *ARok* a *AMes* určující pro který měsíc má probíhat čtení. *ReadByDenMapa* slouží ke čtení databáze pouze pro konkrétní interval, tím tedy umožní načítat omezené množství dat a urychlit tak práci modulu. Stejně jako v proceduře *read* se využívá objektu typu *TkvaSelector*, ale načítají se pouze položky z tabulky denní mapy *pprwid*, jež odpovídají měsíci ze vstupních parametrů. Pro takový prvek tabulky se do proměnné *StringKey* přidá klíč plánu *incwih* z tohoto řádku, pokud ho *StringKey* již neobsahuje. V závěru metody se nastaví *ReadType* na 2 a zavolá funkce *Read*, která načte tedy pouze omezené množství plánů, jejíž *incwih* obsahuje *StringKey*. Podobný význam má procedura *ReadByDenMapaInciprs*, která omezuje vybírané plány nejen dle měsíce, ale i podle interního klíče účastníků dané události.

5.3.3 Write

Poslední procedura třídy *TkvaWebPraPlanListSQL* provádí zápis z objektů do databáze. V cyklu jsou postupně procházeny všechny prvky listu plánů a pro každý z nich ověřen jeho *Status*, tedy položka určující, zda byl objekt nově vložen, editován, smazán nebo nezměněn. Pro každý z těchto statusů podobně jako u *TKvaSelector* poskytl společnost kvasar samostatnou třídu, spravující konkrétní případ komunikace s databází. Pro každou z těchto možností má procedura samostatnou větev.

Pro *Status rsInsert*, tedy vložení nového záznamu se nejdříve vytvoří objekt *Inserrer* typu *TkvaInserrer*, kterému jsou podobně jako u *Selectoru* při čtení naplněny vlastnosti určující tabulku, do které má být vkládán nový záznam (*pprwih*), sloupce a primární klíč

tabulky (*incwih*). Poté v proceduře *Transfer* proběhne mapování vlastností objektu na prvky řádku databázové tabulky s výjimkou klíče *incwih*. Jelikož se jedná v tabulce o autoinkrementální položku, nesmí její hodnota při insertu nabývat nenulových hodnot. Její reálná hodnota je určena databází samotnou.[28] Procedura *Insert* objektu *Inserter* provede vytvoření příkazu pro zápis do databáze a poté pro každý podřazený list plánu se spustí nový cyklus, ve kterém se opět stejným způsobem vyplní odpovídající objekty typu *TkvaInserter* a jsou na nich zavolány procedury *Inserter*. Avšak samotné vložení prvků do databáze proběhne až v závěru procedury po jejím bezchybném průběhu. Toto ověření je v modulu nastaveno pro zajištění konzistence dat v databázi při případných neočekávaných problémech jako například přerušení spojení s databází za běhu této procedury.

Status rsDelete značí, že se má provést smazání celého jednoho plánu z databáze, tato větev procedury tedy využívá objektů typu *TkvaDeleter*, kdy pro každou tabulku tohoto modulu vytvoří vlastní objekt, vyplní jeho vlastnosti a do podmínky zadá rovnost klíče *incwih* z dané tabulky s vlastností *incwih* objektu ke smazání a jsou zavolány procedury *Delete* podobně jako u ve větvi *rsInsert*.

Předposledním statusem je *rsUpdate*. V této větvi procedury se nejdřív ověří, zda byla změněna některá z vlastností daného plánu, pokud ano tak se objektu typu *TkvaUpdater* vyplní vlastnosti určující tabulku a řádek, na kterém se mají provést změny a je zavolána procedura *Update*. Poté se postupně projdou všechny listy přiřazené objektu plánů a pro každý jejich prvek se na základě jejich položky *Status* ověří, zda má v jim odpovídající tabulce proběhnout editace některého z řádků, případně jeho přidání nebo smazání.

Poslední možností je *Status rsNone*, znamenající že objekt nebyl změněn, proto není třeba žádná úprava databáze.

6 IMPLEMENTACE PREZENTAČNÍ VRSTVY

6.1 Formulář Q_PlanKalendar

První částí prezentační vrstvy modulu je zobrazovací formulář Q_PlanKalendar, jež je k práci připojen v příloze č. 2. Tento formulář slouží k zobrazení denní mapy pro všechny zaměstnance ve vybraném měsíci a pro volání dalších formulářů, určených se správě jednotlivých plánů.

6.1.1 Inicializace formuláře

Inicializace probíhá v několika krocích, při spuštění formuláře je volána procedura *InitForm* zajišťující mimo inicializace i čtení a vykreslení dat. Nejdříve je v rámci této procedury volána metoda *InitFirstGo*, která se stará nejen o základní nastavení formuláře, jako jeho výška, šířka, počet sloupců a jejich nadpisy, ale i o čtení osob a jejich plánů z databáze skrz objekt *FPlanKalendar* a jeho proceduru *InitOsoby*, která ke čtení využívá výše zmíněnou metodu *ReadByDenMapa* se vstupními parametry určujícími měsíc, pro který má být formulář vykreslena. Po *InitFirstGo* se naplní nově vytvořený formulář daty procedurou *RefillGrid*.

6.1.2 Přepřehování formuláře

Přepřehování je realizováno procedurou *RefillGrid*, která je volána při každé změně, která je provedena, tedy například vždy po vložení nového záznamu nebo po změně výchozího měsíce. *RefillGrid* pro každého z pracovníků se pro každý den v měsíci zavolá funkce *GetPlanDleMapy* na objektu listu všech plánů a v případě nalezení záznamu, je vykreslen obrázek značící existenci události pro daného uživatele v daný den.

6.1.3 Volání podřazených formulářů

Skrz tento formulář jsou volány dva jemu podřazené, jedná se o *L_PlanKalendar* a *E_PlanKalendar*, k jejich volání dochází v případě dvojitého kliknutí na libovolné pole denní mapy, konkrétně je volána procedura *ACPlanKalendarExecute*, ve které se zavolá funkce *GetSimplePlanDleMapy* se vstupními parametry, jež jsou získány ze souřadnic pole, na které bylo kliknuto a referenčního měsíce, v případě, že pro tohoto pracovníka pro daný den neexistuje žádná naplánovaná akce, je nová akce vytvořena výše popsanou metodou

NewPlan a je zavolán formulář *E_planKalendar* v rámci kterého je nabídnuta správa tohoto plánu. V opačném případě dojde k zavolání druhého formulář *L_PlanKalendar*, který nabídne správu všech událostí, které se v daném termínu pro zvoleného pracovníka existují.

6.1.4 Uložení dat

Ukládání dat probíhá přes tlačítko „Uložit“, které volá funkci *ACsaveExecute*, až v rámci které jsou data skrz výše zmíněnou funkci *write* na listu plánů uloženy, tedy až po stisku tohoto tlačítka jsou data zapsány do databáze.

6.2 Formulář *L_PlanKalendar*

Příloha č. 3 obsahuje formulář, který slouží k zobrazování již naplánovaných akcí ve vybraný den a umožňuje jejich mazání, přidávání a editaci. Jedná se o podstatně jednodušší formulář než je ten předchozí, avšak principy inicializace a přeplňování dat jsou velmi podobné, proto zde nebudou popsány znovu.

6.2.1 Vkládání nového plánu

Při stisku tlačítka + je stejně jako dvojitě kliknutí na prázdné pole u minulého formuláře vytvářena nová událost pro zvolený den a pracovníka a volá formulář *E_PlanKalendar*.

6.2.2 Editace plánu

Při dvojkliku na některý ze zobrazených plánů se podobně jako u vkládání nového plánu otvírá *E_planKalendar*, avšak nový plán není vytvořen, je pouze na základě souřadnic výběru určen plán, který bude uživatel v podřízeném formuláři spravovat.

6.2.3 Mazání plánu

Pro mazání slouží tlačítko - pro který je zavolána funkce *DeletePlan*, ve které je z listu plánů konkrétní prvek smazán společně se všemi jemu podřízenými objekty.

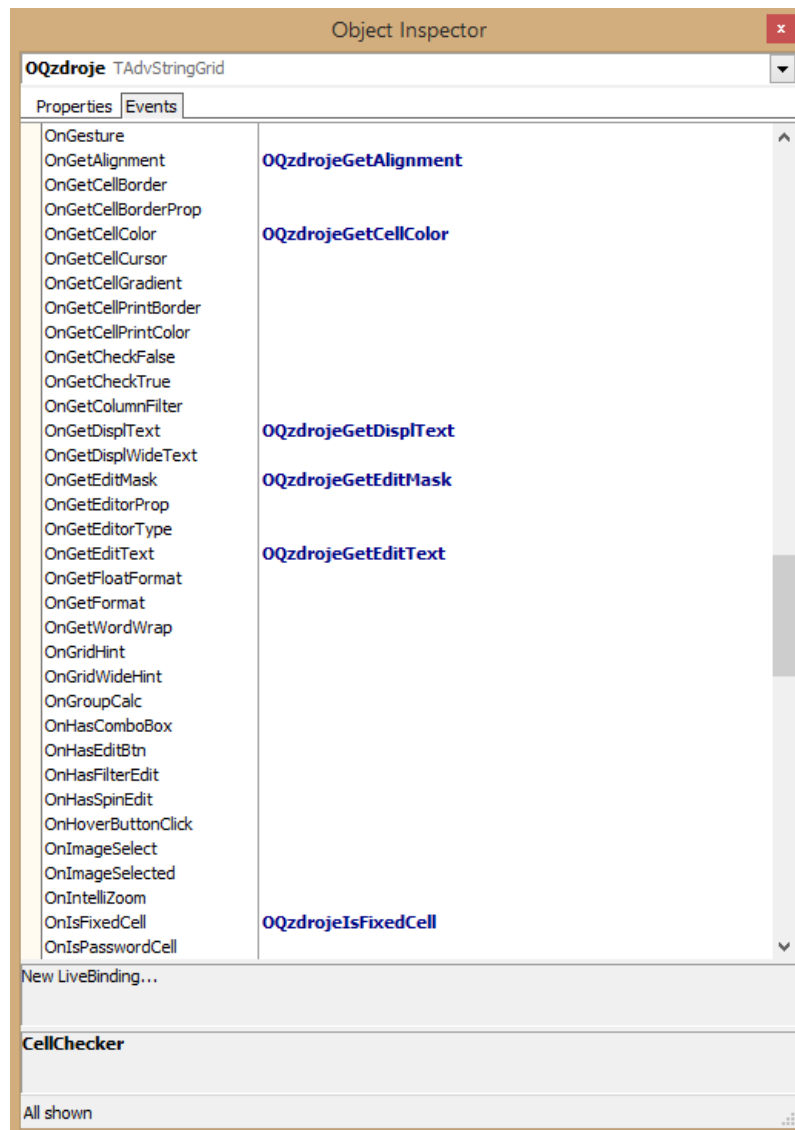
6.3 Formulář *E_PlanKalendar*

Formulář v příloze č. 4 je volán vždy při zadání nového plánu nebo při jeho editaci, obsahuje 2 záložky, první pro správu plánu a přiřazených osob, druhá záložka slouží ke

správě zdrojů. Při spuštění formuláře probíhá spuštění standartní inicializace, v rámci které je rozvržena podoba formuláře a jsou nastaveny hlavičky sloupců (případně názvy řádků) pro obě záložky, první se uživateli zobrazuje záložka osob.

6.3.1 Hlavní události formuláře

Následující funkce a procedury tohoto formuláře byly vytvořeny pomocí Object Inspectoru, který je popsán v teoretické části práce, jedná se konkrétně o události nad mřížkou TAdvStringGrid.



Obrázek č. 4 – Události mřížky TAdvStringGrid v Object Inspectoru

Tyto události mají dva vždy stejné vstupní parametry a to *ACol* a *ARow*, jež vychází ze sloupce a řádku, pro který je metoda volána. Skrz *ARow* události určují se kterým záznamem

listu vztažených PPV případně zdrojů se pracuje, *ACol* udává konkrétní položku daného objektu. Všechny tyto metody jsou na formuláři dvakrát, pro záložku osob mají název doplněn o předponu „*OQedity*“ a pro záložku zdrojů o „*OQzdroje*“.

- *GetDisplText*: Vstupní parametry jsou doplněny o *var* položku *Value*, do které je ukládána ve *String* formátu vybraná hodnota, jež se má zobrazit v mřížce
- *CellValidate*: Ve vstupním parametru *Value* přichází uživatelem změněná položka, před jejím zpracováním probíhají základní ověření, zda v tomto formátu může být vložena a jestli dává smysl (aby například nebylo počáteční datum intervalu větší než datum koncové), v případě jakýchkoli nedostatků je uživatel upozorněn chybovým hlášením. Poté je v případě záložky osob přeplněna denní mapa plánu skrz proceduru *FillDenMapa*.
- *CheckBoxClick*: Umožňuje měnit údaje o tom, jestli jsou první a poslední dny konkrétní akce celodenní, při kliknutí na položku se na základě jejího stavu volá jedna z funkcí pro změnu této položky. Pro osoby je opět na závěr volána procedura *FillDenMapa*.
- *CanEditCell*: Určuje, zda je povolena úprava konkrétní položky, díky této proceduře například není neumožněna editace položky času v případě, že je daný den nastaven na celodenní nebo nelze nastavovat čas a datum peněžním zdrojům.
- *DbClickCell*: Událost, která je aktivní pouze na některých prvcích mřížky formuláře, slouží k volání několika výběrových formulářů při dvojitým kliknutí na dané pole. Pro záložku osob se jedná o výběrový formulář typů nepřítomnosti, tedy v tomto místě se určuje, zda se jedná o například pracovní cestu, nemocenskou nebo mateřskou dovolenou. V záložce zdrojů se jedná o výběr typu zdroje, tedy svěřená věc, prostor nebo finanční částka, pro první dva jmenované zdroje tato událost dále umožňuje vybrat o kterou konkrétní svěřenou místnost, respektive věc se jedná.

6.3.2 Přidávání a mazání položek

Obě záložky mají dvě tlačítka pro přidání a odebrání položek, u zdrojů je nový zdroj automaticky vložen metodou *NewWebPraZdroj* jako peněžní zdroj s množstvím 0 Kč, při mazání zdroje je volána funkce *RemoveZdroj*, v rámci které proběhne také ověření kolizí pro případ, že byl smazán kolizní záznam.

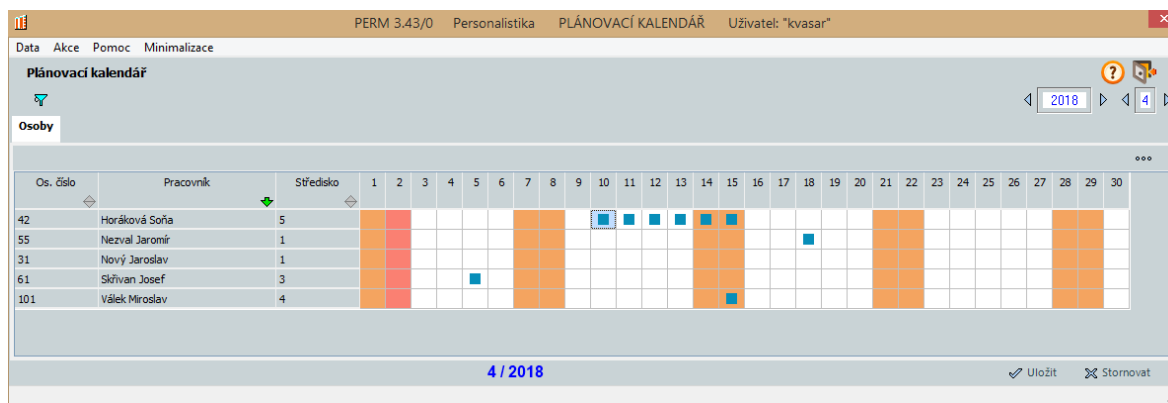
Pro přidání nového účastníka události se nejdříve zobrazí uživateli výběrový formulář, v rámci kterého může vybrat jeden nebo více konkrétních pracovníků a jejich PPV, pro každý tento prvek se zavolá procedura *NewWePraVztPPV*. Pro odebrání vztaženého PPV je volána metoda *RemoveVztPPV*, avšak pokud jsou odebrány všechny položky v tomto seznamu tlačítko „potvrdit“ se stane neaktivním a uživatel nebude moci událost uložit.

7 NÁVRH ROZHRANÍ A UŽIVATELSKÁ DOKUMENTACE MODULU

Tato kapitola obsahuje návrh a prezentaci uživatelského rozhraní modulu. Design prezentační vrstvy byl vytvořen tak, aby odpovídal standardnímu zobrazení formulářů v programu PERM 3

7.1 Prohlížeč formulář

Tento modul lze nalézt v systému PERM 3 pod záložkou Osoby jako plánovací kalendář. První se uživateli zobrazí formulář se seznamem pracovníků a vykreslenou jejich mapou událostí pro aktuální měsíc, kdy jednotlivé akce jsou zobrazeny vyplněním příslušných polí, a při najetí myši na konkrétní položku je zobrazen výpis událostí v daném dni. Měsíc a případně rok, pro který je obrazovka vykreslována lze změnit v jejím pravém horním rohu. Pro vytvoření nové události v prázdném dni stačí dvojitě kliknutí na vybrané pole, po kterém je pro daný den vytvořena událost s odpovídajícím dnem a uživatelem a je zobrazena v editačním formuláři. Pro dny, ve kterých je již nějaká událost naplánována vede dvojklik na formulář, který obsahuje seznam událostí pro tento den.

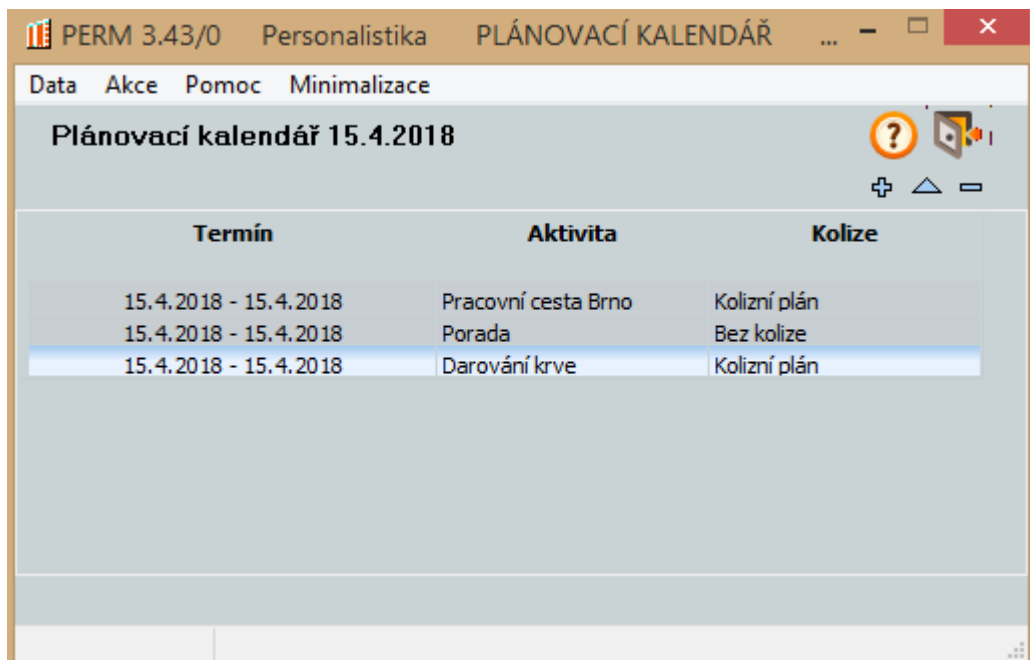


Obrázek č. 5 – Plánovací kalendář

7.2 Seznam událostí

Tento formulář zobrazuje seznam všech naplánovaných akcí, které zasahují do daného dne, tyto události jde spravovat pomocí tlačítek v pravém horním rohu obrazovky, pro tlačítko + je vytvořen nový plán pro daný den a je otevřen editační formulář. Ten je také otevřen v případě editace již existujícího záznamu, po stisku symbolu trojúhelníku nebo pro dvojkliku na daný prvek. Pro smazání události stačí stisknout tlačítko -, důležité je

poznámenat, že toto smaže celou událost, tedy v případě více účastníků události je smazána pro všechny.



Termín	Aktivita	Kolize
15. 4. 2018 - 15. 4. 2018	Pracovní cesta Brno	Kolizní plán
15. 4. 2018 - 15. 4. 2018	Porada	Bez kolize
15. 4. 2018 - 15. 4. 2018	Darování krve	Kolizní plán

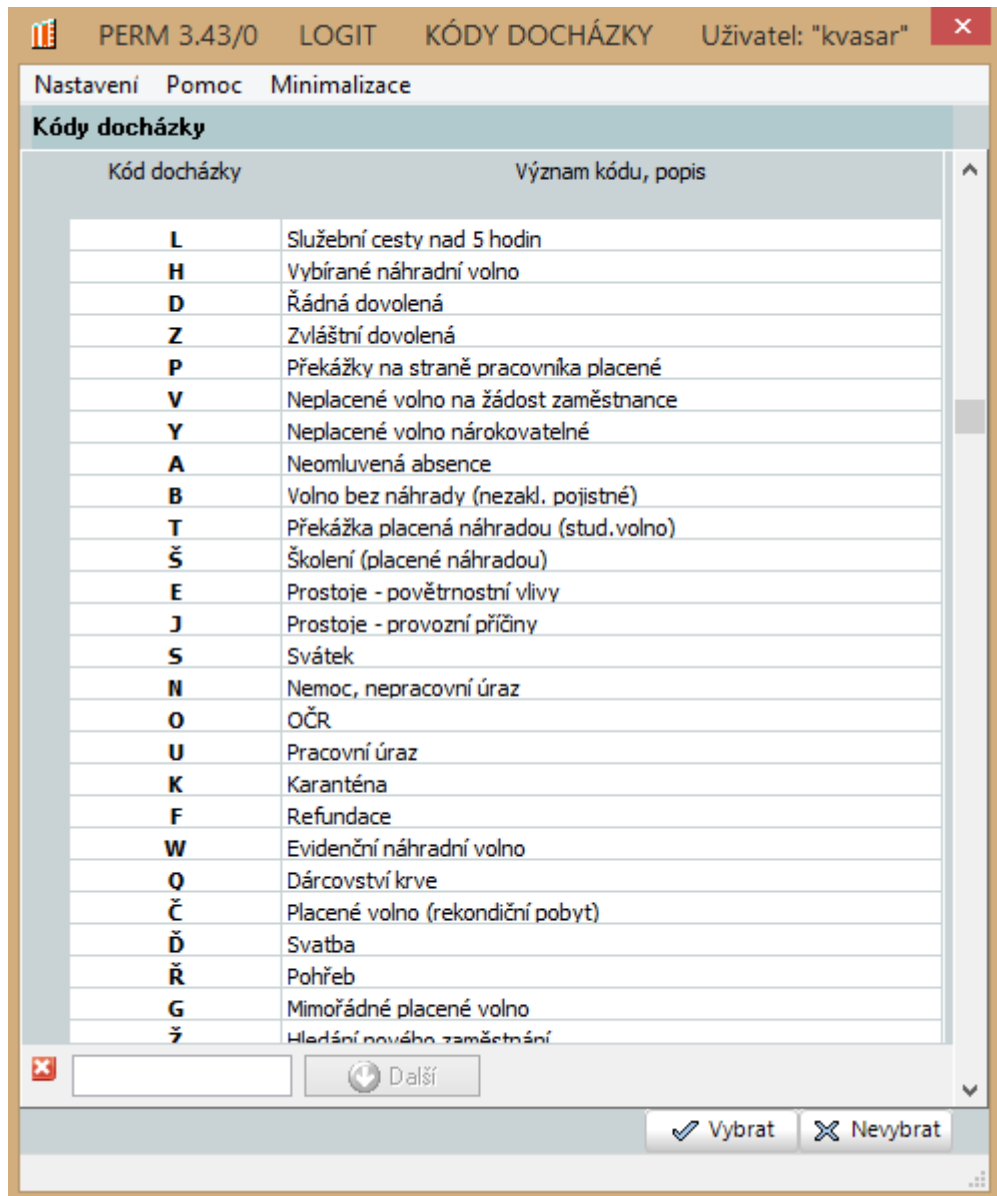
Obrázek č. 6 – Seznam naplánovaných událostí

7.3 Editační formulář

Editační formulář má dvě záložky, Osoby a Zdroje, první z nich je ještě rozdělena na dvě části, první se obsahuje informace o události, druhá seznam osob které se účastní události.

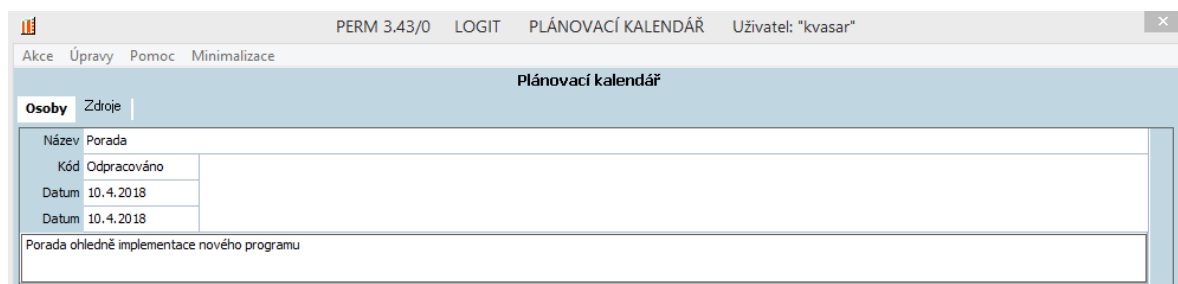
7.3.1 Popis události

Vrchní část záložky Osoby obsahuje popis události, je zde prezentován její název, dalším řádku je zobrazen typ docházky, který je možné změnit dvojitým kliknutím, po kterém se zobrazí formulář nabízející výběr mezi jednotlivými typy.



Obrázek č. 7 – Výběr typu docházky

Data zde plní pouze prezentační roli a určují mezní intervaly všech událostí, jsou určeny z nejnižšího, respektive nejvyššího data mezi vztaženými PPV, poslední pole je určeno pro poznámku.



Obrázek č. 8 – Popis události

7.3.2 Vztažené PPV

V druhé části záložky Osoby se nachází seznam všech vztažených PPV, první 4 položky určují konkrétního pracovníka na události a jeho pracovní poměr a není možné je editovat. Další čtveřice položek určuje interval data a času, po který se daná osoba účastní události. Když jsou krajní dny intervalu určeny jako celodenní, není možné upravovat jejich čas, ten je načten z rozvrhu směn pracovníka. V opačném případě je také možné upravit počet hodin, ve kterých bude v první respektive poslední den pracovník na události. Poslední položkou je zobrazení informace o kolizním stavu s jiným plánem pracovníka, tento prvek je pouze prezentační.

Osobní	Pracovník	Středisko	Prac. místo	Datum od	Čas od	Datum do	Čas do	Celý 1.	Celý pos.	První hod	Poslední hod	Kolize	Výběr
61	Skřivan Josef	3	Řidič Dílna	11.4.2018	08:00	15.4.2018	16:00	<input type="checkbox"/>	<input type="checkbox"/>	8	8	Kolize	<input type="checkbox"/>
42	Horáková Soňa	5	THP	11.4.2018	-	11.4.2018	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-	Bez	<input type="checkbox"/>
55	Nezval Jaromír	1	Dělník Štět	11.4.2018	-	11.4.2018	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	-	-	Bez	<input type="checkbox"/>
101	Válek Miroslav	4	Dělník	11.4.2018	08:00	11.4.2018	16:00	<input type="checkbox"/>	<input type="checkbox"/>	8	-	Bez	<input type="checkbox"/>

✔ Potvrdit ✖ Stornovat

Obrázek č. 9 – Účastníci události

Nové PPV na událost je možné přidat opět pomocí tlačítka + v pravé části obrazovky. Po jehož stisku je podobně jako u typů nepřítomnosti prezentován formulář, ve kterém je možné vybrat libovolné množství pracovníků, kteří mají být k události přiřazeni. Odebrání účastníků je možné tlačítkem -, avšak událost bez aspoň jednoho pracovníka nelze uložit.

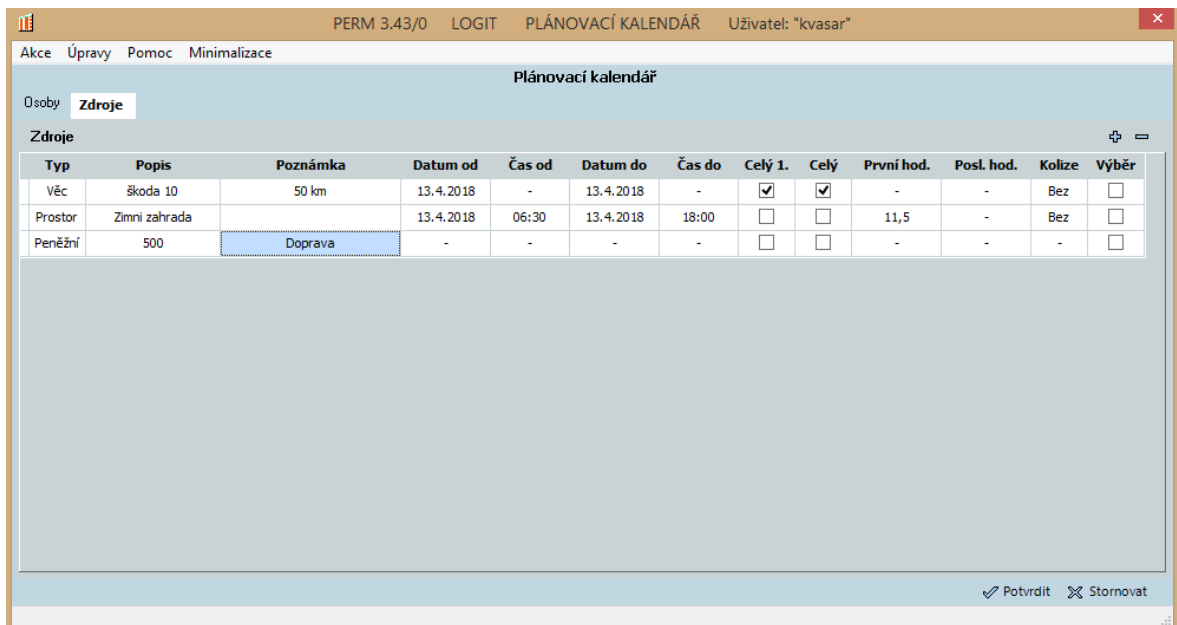
Os. číslo	Jméno	Druh PPV	Středisko	Kategorie	Zakázka	Vznik	Ukončení	MES	Typ mzdy	Změna tar.	Výběr
31	Nový Jaroslav	PP	1	D		2.5.2000			C		<input type="checkbox"/>
42	Horáková Soňa	PP	5	T		1.1.1993			M		<input type="checkbox"/>
55	Nezval Jaromír	PP	1	D		18.4.2001			C		<input type="checkbox"/>
61	Skřivan Josef	PP	3	D		1.10.2001			C		<input type="checkbox"/>
101	Válek Miroslav	PP	4	D		18.11.1996			C		<input checked="" type="checkbox"/>

✔ Vybrat ✖ Nevýbrat

Obrázek č. 10 – Výběr účastníků a jejich PPV

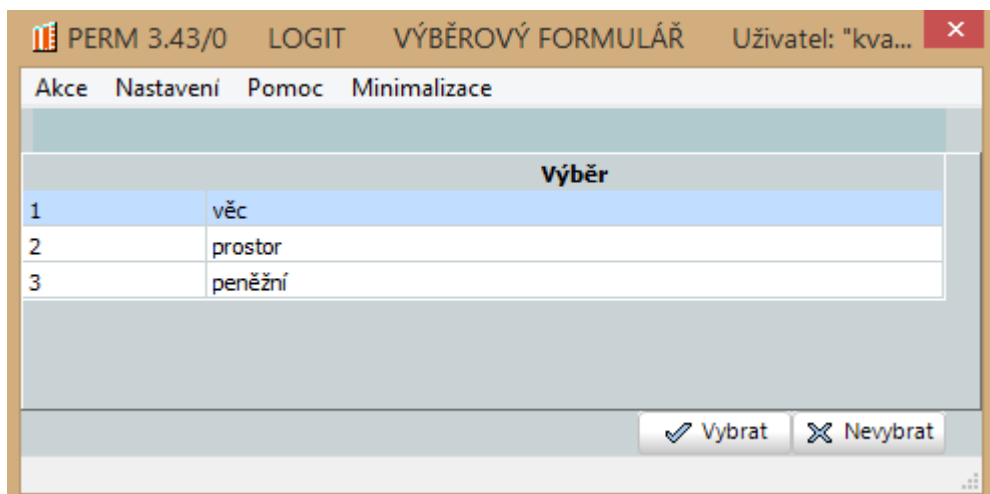
7.3.3 Zdroje

V této záložce je možné přidávat a odebírat zdroje událostí skrz tlačítka + a – v pravém horním rohu formuláře, potvrdit je možné i událost bez jakýchkoli zdrojů.



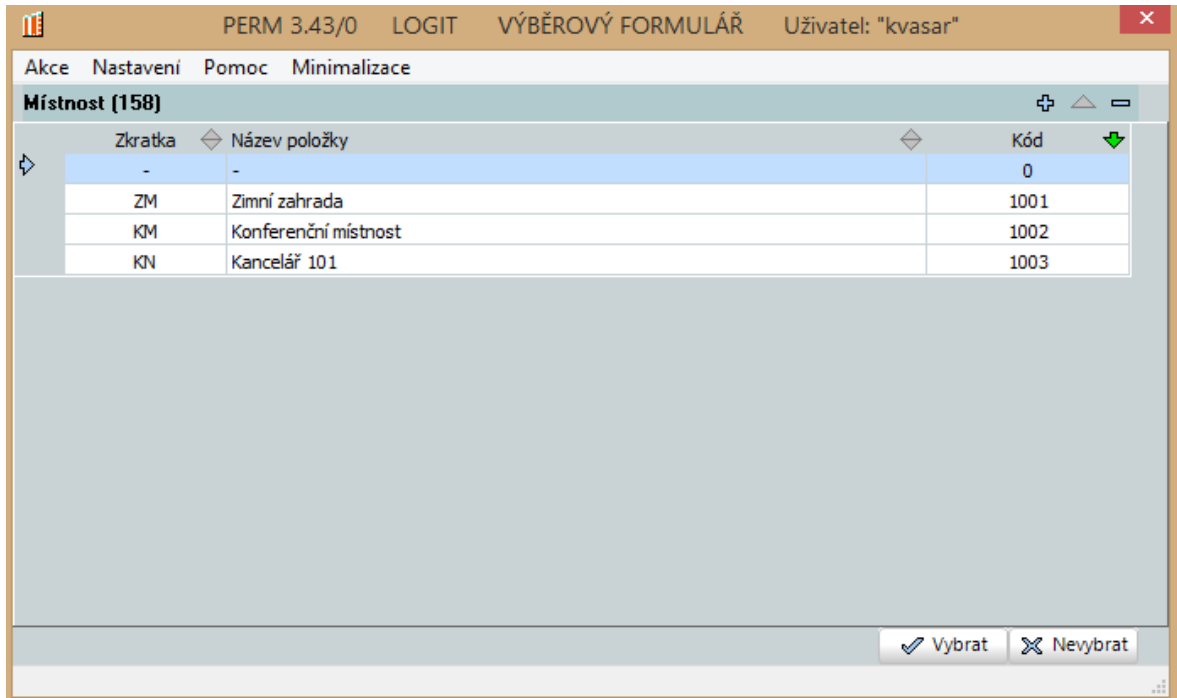
Obrázek č. 11 – Zdroje

Zdroje mohou mít tři různé typy: svěřené věci, prostory a peníze, typ je zobrazen v prvním sloupci a je možné jej měnit dvojklikem, po kterém je zobrazen výběrový formulář se všemi typy.



Obrázek č. 12 – Výběr typu zdroje

Druhý sloupec obsahuje v případě peněžních zdrojů částku a v pro svěřené věci a prostory určuje konkrétní položku, kterou opět lze vybrat skrz formulář zobrazený po dvojkliku.



Obrázek č. 13 – Výběr prostor

Následuje poznámka a určení data a času, jež je identické se vztaženými PPV, pro peníze jsou tyto položky skryty, jelikož jejich editace nemá reálný význam.

8 KNIHOVNY PRO KOMUNIKACI S PROSTŘEDÍM PERMWEB

Pro tento modul byly vytvořeny dvě knihovny na komunikaci s prostředím PermWeb, základem této komunikace je aplikační server P3Server.exe firmy Kvasar, který komunikuje se dvěma frontami Microsoft Message Queue, s jednou vstupní a jednou výstupní. Ze vstupní fronty bere server postupně příchozí (vstupní) XML soubory, dle názvu v jejich hlavičce jim přiřadí konkrétní knihovnu, kterou mají být zpracovány, a tato knihovna poté vytvoří výstupní XML soubor, jež je zařazen do výstupní fronty.

Pro modul plánovacího kalendáře byly vytvořeny dvě knihovny, obě mají za cíl umožnit skrz webové rozhraní přístup k osobnímu kalendáři pro jednoho uživatele a správu jeho událostí. Jelikož jsou si velmi podobné, jejich implementace je v této kapitole popsána společně s vyznačením rozdílů.

První knihovna *P3k_plankal* byla vyžádána zákazníkem firmy Kvasar, konkrétně zajišťuje komunikaci s modulem čistého osobního kalendáře pro zaměstnance. Jedná se tedy o verzi bez možnosti plánování zdrojů nebo přidávání více účastníků k události. Její kód je připojen v příloze č. 5.

Druhá knihovna *P3k_plankalfull* je určitým rozšířením té první, obsahuje tedy náhled na plánované události jedné osoby s možností kompletní správy této události jako přidělování zdrojů a přidávání dalších vztažených PPV. Kód obsahuje příloha č. 6.

8.1 XML soubory

Po každou z knihoven jsou dva hlavní XML, a to vstupní obsahující dotaz a druhý obsahující odpověď na tento dotaz.

8.1.1 Vstupní XML

Tyto soubory v přílohách č. 7 a 9 jsou dotazem, který směřuje z webového rozhraní, jeho cílem je získání odpovědi ve formě dalšího XML souboru, který bude obsahovat vyžádaná data a případná editace těchto dat. Tyto soubory jsou rozděleny na dvě části: **header** a **body**, tedy hlavičku a tělo souboru. V hlavičce přichází informace jako jméno knihovny, ID uživatele, položku *compid* definující konkrétní databázi pro kterou má být

a uzel **filter**, jež obsahuje údaje o tom, která data se mají číst, v případě těchto knihoven se jedná o osobní číslo *oscipr*, číslo PPV a období, dle těchto údajů je generována odpověď.

Uzel *body* slouží k editaci položek, obsahuje seznam všech plánů konkrétního pracovníka v období, jež bylo zadáno ve filtru. Každý dceřiný uzel plánu může obsahovat status, ve kterém je definováno, zda má být daný záznam nově vložen (status je roven **i**), upraven (**u**) nebo smazán (**d**). Následuje seznam položek, které patří ke konkrétnímu plánu, který je identický se seznamem vlastností třídy *TkvaWebPraPlan* doplněný o uzel **popis** obsahující poznámku k danému plánu. Další dceřiný uzel **pracovník** může stejně jako uzel plánu obsahovat status, avšak v první knihovně se tento uzel musí vyskytovat přesně jednou pro každou událost, jelikož se jedná o knihovnu, v rámci které není umožněno přidávání více účastníků k jedné akci, proto i status může nabývat pouze hodnoty **u**. Pro druhou knihovnu může být na události uživatelů libovolné množství, proto položka může mít jakýkoli status, avšak stále platí, že plán musí mít aspoň jeden záznam, tedy smazání posledního záznamu není možné. Dceřiné uzly pracovníka odrážejí vlastnosti třídy *TkvaWebPraVztPPV*. Posledním dceřiným uzlem plánu (pouze pro druhou knihovnu) je **zdroj**, zde opět můžou být tři různé statusy, tentokrát bez žádného omezení a obsah uzlu odpovídá tentokrát vlastnostem třídy *TkvaWebPraZdroj*.

8.1.2 Výstupní XML

Výstupem je také soubor *XML*, který je odpovědí na vstupní *XML*, tyto soubory jsou přílohami č. 8 a 10. V uzlu **header** nese výstup název knihovny a položku *compid*, avšak přihlašovací jméno, heslo a filtr již zde nejsou třeba. Struktura odpovědi je jinak téměř identická se vstupními soubory, rozdílem je absence jakýchkoli statusů, pak navíc na konec uzlu **body** je přidán seznam směn přihlášeného pracovníka a seznam jeho spolupracovníků, jež je obsažen z důvodu možnosti zvaní dalších pracovníků na událost.

8.2 Implementace knihoven

Implementace obou knihoven je velmi podobná, prakticky celou knihovnu tvoří jedna globální funkce *ExecuteTask* ve které obsahuje ještě několik lokálních procedur. Funkce *ExecuteTask* typu *Integer* má 5 parametrů typu *PChar*: *Input*, variabilní proměnou *Output*, *AUserName*, *APassword* a *ALog*, jak již názvy napovídají, jedná se parametry obsahující

vstup a výstup funkce, uživatelské jméno, heslo a log. Funkce na začátku načte vstupní *XML*, a ve funkci *PMCheck* ověří, zda obsahuje všechny náležitosti a funkce *GetPermUserFromUserID* ověří přihlašovací údaje. Z položky **filter** jsou poté získány údaje o zadaném období, osobním čísle a čísle PPV. Pokud vše proběhne bezchybně, pokusí se funkce spojit s databází a načíst data potřebná k vyhodnocení knihovny. Tabulek je čtena celá řada, mezi nejdůležitější patří údaje přihlášeného pracovníka, seznam jeho PPV, rozvrh jeho směn a hlavně jsou načteny všechny plány spadající do zadaného období pro přihlášeného uživatele skrz *ReadDenMapaByInciprs*.

Následně pokud bylo vyplněno **body** se v lokální proceduře *DecodeXmlZaznam* přečte příchozí *XML*, kdy pro každý uzel čte jeho název a ukládá jeho hodnotu do příslušné vlastnosti objektu *XMLWebPraPlanList*. Tímto způsobem může být zpracováno libovolné množství plánů včetně jejich podřízených objektů jako vztažené PPV a zdroje, důležitou položkou je také *WebStatus* všech objektů načtený z položky **Status** u jednotlivých uzlů podle kterého je rozhodnuto, zda se má daný záznam editovat, mazat nebo vložit. Po této lokální proceduře následuje cyklus kde se pro každý načtený plán z *XML* dle položky *WebStatus* změní objekty načtené z databáze. V závěru je proveden zápis dat pomocí procedury *Write*, tyto data jsou pro zajištění jejich správnosti znovu načteny (hlavně z důvodu potřeby prezentace autoinkrementálních klíčů jednotlivých tabulek) opět metodou *ReadByDenMapaInciprs* a poté je vyplněn *Output* hlavní funkce této knihovny lokální procedurou *AddXmlZadost*, která vytváří odpověď pro PermWeb. Tato odpověď je strukturována stejně jako příchozí *XML* soubor, avšak obsahuje všechny plány odpovídající vstupním parametrům, tedy uživateli a období, nehledě na to, zda bylo v databázi při zpracování příchozího souboru něco změněno.

9 TESTOVÁNÍ MODULU

Knihovny popsané v minulé kapitole umožnily efektivní způsob testování tohoto modulu, za účelem testování byla vytvořena řada vstupních *XML* souborů, které následně přes program *DITester* poskytnutý společností Kvasar byly zpracovány těmito knihovnami a výsledek tohoto testování bylo možné zobrazit jak v modulu plánovacího kalendáře systému PERM 3, tak i ve výstupních *XML* souborech. Testování probíhalo po celou dobu vývoje tohoto modulu a chyby byly tedy operativně opravovány a v závěru se již neobjevily žádné vážnější problémy.

ZÁVĚR

Výsledkem této bakalářské práce je modul plánování zdrojů pro aplikaci PERM 3 a to jak zdrojů lidských tak i fyzických a peněžních. Modul byl vytvořen na základě požadavků společnosti Kvasar s.r.o., které byly doplněny o požadavky vedoucího práce, velkou výhodou se stal fakt, že tyto požadavky se spíše doplňovaly, než vzájemně rozporovaly.

V rámci tohoto modulu je uživateli umožněna správa událostí všech pracovníků, jednotlivé události můžou mít přiřazeno neomezené množství PPV a zdrojů. Události mimo údajů o datu konání nesou také příznak, o jaký docházkový typ se jedná a pro osoby i pro zdroje jsou vyhodnocovány jejich kolizní stavy na jednotlivých událostech. Dále byly vytvořeny dvě knihovny zpracovávající XML dokumenty pro komunikaci s prostředím PermWeb, které také za využití programu *DIITester* umožnili testování modulu a ověření jeho funkcionality.

Tato práce dostála všem požadavkům, jež na ni byly kladeny, a splnila tak svůj hlavní cíl, tedy vytvoření modulu, který v systému PERM 3 sjednotí agendu docházky, jež je roztržštěna napříč ostatními moduly. Tento kalendář tuto agendu nejen sjednocuje, ale zároveň ji i rozšiřuje o možnost plánování nových typů událostí jako pracovní schůzky a i o možnost správy fyzických zdrojů. Díky tomu dojde k podstatnému ulehčení práce s plánováním docházky pro vedoucí pracovníky a personalisty, což je zároveň největším přínosem této práce.

Výsledný modul nedisponuje žádnými zásadními nedostatky, i když je samozřejmé, že určité prvky se dají vždy nějak vylepšit, jako například kvalita grafické prezentace modulu. V rámci vytváření práce nedošlo k žádným komplikacím, jež by ohrozily její výsledek, velkou výhodou také byl fakt, že firma Kvasar poskytla k vývoji modulu několik struktur, které částečně usnadnily práci, konkrétně hlavně s tvorbou prezentační vrstvy a se správou tabulek v databázi.

SEZNAM POUŽITÉ LITERATURY

- [1] KHANSE, Anand. Difference between Freeware, Free Software, Open Source, Shareware, Trialware, etc. *The Windows Club* [online]. [cit. 2018-04-09]. Dostupné z: <http://www.thewindowsclub.com/difference-freeware-free-software-open-source>
- [2] WinCalendar: License. *WinCalendar* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.wincalendar.com/license>
- [3] 13 of the Best Calendar Apps. *Digital Trends* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.digitaltrends.com/mobile/best-calendar-apps/>
- [4] Best Calendar Apps 2018. *Toms guide* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.tomsguide.com/us/pictures-story/442-best-calendar-apps.html#s9>
- [5] Kalendáře a aplikace na organizaci času. *SvětAndroida* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.svetandroida.cz/organizaci-casu-svetandroida-201701/>
- [6] Informace o webových aplikacích. *Adobe* [online]. [cit. 2018-04-10]. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>
- [7] WALLEN, Jack. Five apps: Five free but powerful desktop calendars. *Tech Republic* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.techrepublic.com/blog/five-apps/five-free-but-powerful-desktop-calendars/>
- [8] PINOLA, Melanie. Beyond Google Calendar and Apple Calendar: The 18 Best Apps to Manage Your Schedule. *Zapier* [online]. 5.5.2017 [cit. 2018-04-09]. Dostupné z: <https://zapier.com/blog/best-calendar-apps/>
- [9] The iCalendar Standard. *ICalendar* [online]. [cit. 2018-04-09]. Dostupné z: <https://icalendar.org/>
- [10] Osmo. *Clayo* [online]. [cit. 2018-04-09]. Dostupné z: <http://clayo.org/osmo/>
- [11] Desktop iCalendar Lite. *Download.cnet* [online]. [cit. 2018-04-09]. Dostupné z: http://download.cnet.com/Desktop-iCalendar-Lite/3000-2124_4-10802583.html
- [12] Mac Basics: Calendar keeps your appointments. *Apple* [online]. 23.3.2016 [cit. 2018-04-09]. Dostupné z: <https://support.apple.com/en-us/HT202835>
- [13] GUAY, Matthew. Google Calendar - Features, Pricing, Alternatives: Google Calendar Review. *Zapier* [online]. 5.2.2016 [cit. 2018-04-09]. Dostupné z: <https://zapier.com/apps/google-calendar>

- [14] Microsoft Outlook. *Products Office* [online]. [cit. 2018-04-09]. Dostupné z: <https://products.office.com/cs-cz/outlook/email-and-calendar-software-microsoft-outlook?tab=tabs-1>
- [15] Cozi Gold Features. *Cozi: An upgrade for whole family* [online]. [cit. 2018-04-09]. Dostupné z: <http://www.cozi.com/cozi-gold-features/>
- [16] The best online calendars for groups, families, and households: Cozi. *The Next Web* [online]. 2017 [cit. 2018-04-09]. Dostupné z: <https://thenextweb.com/apps/2017/08/28/best-online-calendars-groups-families-households/>
- [17] Thunderbird: Calendar. *Mozilla: Support* [online]. [cit. 2018-04-09]. Dostupné z: <https://support.mozilla.org/en-US/products/thunderbird/calendar>
- [18] GAJIC, Zarko. Delphi History – from Pascal to Embarcadero Delphi XE 2: Delphi history: the Roots. *ThoughtCo* [online]. 17.3.2017 [cit. 2018-04-09]. Dostupné z: <https://www.thoughtco.com/history-of-delphi-1056847>
- [19] Proč je Embarcadero Delphi a ne Borland?. *Delphi.cz* [online]. [cit. 2018-04-09]. Dostupné z: <http://delphi.cz/page/Proc-je-Embarcadero-Delphi-a-ne-Borland.aspx>
- [20] CANTÚ, Marco, WOOD, Peter, ed. *Object Pascal Handbook: The Complete Guide to the Object Pascal programming language*. Piacenza, Italy: Embarcadero Technologies, 2015. ISBN 1514349949. *The Foundations of Digital Signal Processing: Theory, Algorithms and Hardware Design*
- [21] GAYDECKI, Patrick. *The Foundations of Digital Signal Processing: Theory, Algorithms and Hardware Design*. Bibliovault OAI Repository, the University of Chicago Press, 2004. ISBN 3540674012.
- [22] SVOBODA, Josef. Object Pascal: Úvod do OOP. *Programujte* [online]. [cit. 2018-04-09]. Dostupné z: <http://programujte.com/clanek/2008090100-object-pascal-03-uvod-k-oop-datove-typy-2-cast/>
- [23] Delphi Object Pascal. *Delphi.cz* [online]. [cit. 2018-04-09]. Dostupné z: <http://delphi.cz/post/Object-Pascal-zacatecnici.aspx>
- [24] Form Designer. *Embarcadero: RAD Studio* [online]. [cit. 2018-04-09]. Dostupné z: http://docwiki.embarcadero.com/RADStudio/Tokyo/en/Form_Designer

- [25] VLC TMS Grid Pack. *TMS Software* [online]. 5.4.2018 [cit. 2018-04-09]. Dostupné z: <https://www.tmssoftware.com/site/gridpack.asp>
- [26] Co je XML?. *Interval* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.interval.cz/clanky/co-je-xml/>
- [27] Extensible Markup Language (XML). *W3* [online]. [cit. 2018-04-09]. Dostupné z: <https://www.w3.org/TR/REC-xml/#sec-origin-goals>
- [28] SQL AUTO INCREMENT Field. *W3schools* [online]. [cit. 2018-04-09]. Dostupné z: http://w3schools.sinsixx.com/sql/sql_autoincrement.asp.htm

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

PPV	Pracovněprávní vztah
SQL	Structured Query Language, strukturovaný dotazovací jazyk
GNU	GNU's Not Unix, GNU není Unix
RFC	Request for comments, žádosti o komentáře
S.r.o.	Společnost s ručením omezeným
ID	Identification, identifikační položka
XML	Extensible Markup Language, rozšířitelný značkovací jazyk

SEZNAM OBRÁZKŮ

Obrázek č. 1 - Zobrazení formuláře skrz Form Designer.....	23
Obrázek č. 2 – Tool Palette.....	24
Obrázek č. 3 – Databázová struktura modulu.....	28
Obrázek č. 4 – Události mřížky TAdvStringGrid v Object Inspectoru.....	42
Obrázek č. 5 – Plánovací kalendář.....	45
Obrázek č. 6 – Seznam naplánovaných událostí.....	46
Obrázek č. 7 – Výběr typu docházky.....	47
Obrázek č. 8 – Popis události.....	47
Obrázek č. 9 – Účastníci události.....	48
Obrázek č. 10 – Výběr účastníků a jejich PPV.....	48
Obrázek č. 11 –Zdroje	49
Obrázek č. 12 – Výběr typu zdroje	49
Obrázek č. 13 – Výběr prostor.....	50

SEZNAM PŘÍLOH

- [1] P3ObjPWI.pas – hlavní třídy modulu
- [2] P3Q_PlanKalendar.pas – prohlížečí formulář
- [3] P3L_PlanKalendar.pas – formulář zobrazující seznam událostí v jednom dni
- [4] P3E_PlanKalendar.pas – editační formulář
- [5] P3DplanKal.pas – knihovna pro komunikaci s prostředím PermWeb bez zdrojů a dalších účastníků
- [6] P3DplanKalFull.pas – plná verze knihovny pro komunikaci s prostředím PermWeb
- [7] k_plankal_in.xml – vstupní soubor pro zjednodušenou komunikační knihovnu
- [8] k_plankal_out.xml – výstupní soubor pro zjednodušenou komunikační knihovnu
- [9] k_plankalfull_in.xml – vstupní komunikační soubor pro plnou verzi knihovny
- [10] k_plankalfull_out.xml – výstupní komunikační soubor pro plnou verzi knihovny