

UNIVERZITA PARDUBICE

Fakulta elektrotechniky a informatiky

System plánování a řízení lidských zdrojů při
organizaci masových akcí

Ondřej Serbousek

Diplomová práce
2013

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Ondřej Serbousek**
Osobní číslo: **I10415**
Studijní program: **N2646 Informační technologie**
Studijní obor: **Informační technologie**
Název tématu: **Systém plánování a řízení lidských zdrojů při organizaci masových akcí**
Zadávací katedra: **Katedra softwarových technologií**

Z á s a d y p r o v y p r a c o v á n í :

Cílem práce je analýza, návrh a implementace aplikace, která bude vhodná pro efektivní plánování a řízení lidských zdrojů pořadatelské a bezpečnostní služby na akcích většího rozsahu. Teoretická část se bude zabývat provedením analýzy a návrhu softwarového systému s využitím CASE nástroje a jazyka UML. Implementační část se bude zabývat tvorbou samotné aplikace podle provedeného návrhu s využitím relační databáze. Vytvořená aplikace bude navržena tak, aby vhodným způsobem doplnila stávající používaný informační systém. V rámci řešení bude realizováno předávání dat z/do existujícího systému. Vytvořená aplikace bude umožňovat zadávání a prezentaci dat v grafickém režimu s napojením na vhodný mapový podklad. Aplikace bude implementována v jazyce C#.

Rozsah grafických prací:

Rozsah pracovní zprávy:

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

Seznam odborné literatury:

ARLOW, Jim; NEUSTADT, Ila. UML 2 a unifikovaný proces vývoje aplikací: Objektově orientovaná analýza a návrh prakticky. 2. aktualizované a doplněné vydání. Praha : Computer Press, 2007. 568 s. ISBN 978-80-251-1503-9. NAGEL, Christian. C# 2005: Programujeme profesionálně. Vyd. 1. Překlad Jakub Mikulaščík, Petr Dokoupil. Brno: Computer Press, 2006, 1398 s. ISBN 80-251-1181-4. RIORDAN, Rebecca M. Vytváříme relační databázové aplikace. 1.vyd. Praha: Computer Press, 2000, 280 s. ISBN 80-722-6360-9.

Vedoucí diplomové práce:

Ing. Petr Veselý

Katedra softwarových technologií

Datum zadání diplomové práce:

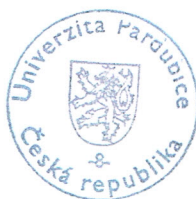
31. října 2012

Termín odevzdání diplomové práce:

17. května 2013



prof. Ing. Simeon Karamazov, Dr.
děkan



L.S.



prof. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 15. listopadu 2012

Prohlášení autora

Prohlašuji, že jsem tuto práci vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 23. 8. 2013

Ondřej Serbousek

Poděkování

Chtěl bych poděkovat panu Ing. Martinu Čechákovi, který je spolumajitelem bezpečnostní agentury ČECHYMEN a zároveň hlavním analytikem informačních systémů firmy. Pan Čechák mi pomáhal při definici business funkcí aplikace a vedl definování nosných požadavků vytvářené aplikace. Dále bych chtěl zmínit pana Romana Zábavského, který mi nejvíce pomáhal při nasazení aplikace pro testování, ale také celý tým vedení bezpečnostní agentury ČECHYMEN, který se aktivně podílel na funkčních definicích a pomáhal mi shromažďovat požadavky systému. Zároveň se v konečné fázi podílel i na testování a ověřování funkčnosti aplikace.

Dále bych chtěl poděkovat svému vedoucímu práce panu Ing. Petru Veselému za jeho vedení práce, konzultace a usměrňování v průběhu vytváření. Děkuji za cenné rady a celkovou záštitu nad prací.

Anotace

Práce je zaměřena na kompletní realizaci projektu vytváření nového pracovního softwaru pro zadavatele, který se příliš v oblasti informačních technologií nepohybuje. Za jedno z krizových míst je považována správná definice funkčních postupů používaných zadavatelem. Práce se detailně zabývá všemi kroky průběhu projektu, od modelování funkčnosti, požadavky, analýzu a design, implementaci a nakonec nasazení a otestování. Zejména v prvních fázích projektu a poté při závěrečném testování bylo využíváno úzké komunikace se zadavatelem, kdy bylo nutné konzultačně řešit definice požadavků a nejasnosti některých funkčních postupů. Vybrané technologie pro realizaci a implementaci projektu byly vybrány pod akademickým vedením s využitím konzultací a odborných rad akademických pracovníků a vedoucího práce. Byly zvoleny moderní technologie včetně některých nezvyklých řešení, které ale přesně vystihovaly požadavky zadavatele. V závěrečné fázi projektu bylo provedeno testování funkčnosti a vyhodnocení úspěšnosti projektu, opět ve spolupráci se zadavatelskou agenturou.

Klíčová slova

analýza, implementace, bezpečnostní agentura, požadavky, praxe, aplikace, systém

Title

System for planning and managing human resources for organization mass event

Annotation

Thesis is based on complete realization project creating new software for client who is not involved in new trends in Information Technology. As one of critical issue is consider correct definition business process used in clients methodic. Thesis is detail describing all steps in project progress, from business modeling, requirements, analysis and design, implementation and for end deployment and testing. Especially in beginning of project and also for finish testing has been use close communication with client. It was necessary by meeting and physics consultation solve it definition of requirements and some not clear points in business processes. Selected technologies for realization and implementation have been chosen under academic leading with communication and advices from academics and thesis leader. It has been chosen modern technologies including some not usual solution which perfectly matched client's requirement. For finally phases of project has been performed testing of functional and final evaluation success rate again with cooperation with client.

Keywords

analysis, implementation, security agency, requirements, practice, application, system

Obsah

| | |
|---|-----------|
| Seznam zkratk | 9 |
| Seznam obrázků | 10 |
| Seznam tabulek | 10 |
| 1 Úvod | 11 |
| 2 Motivace | 12 |
| 2.1 Představení odběratele..... | 12 |
| 2.2 Motivace pro nový systém..... | 12 |
| 2.3 Oblast nasazení nové aplikace..... | 13 |
| 2.4 Rozložení práce | 13 |
| 3 Požadavky | 15 |
| 3.1 Funkční požadavky..... | 15 |
| 3.1.1 Modul 1 | 15 |
| 3.1.2 Modul 2 | 15 |
| 3.1.3 Modul 3 | 16 |
| 3.1.4 Modul 4 | 16 |
| 3.2 Nefunkční požadavky | 16 |
| 4 Modelování případů užití | 17 |
| 4.1 Hranice systému | 17 |
| 4.2 Aktéři | 18 |
| 5 Funkční design | 19 |
| 5.1 Modul 1 | 21 |
| 5.1.1 Business flow..... | 21 |
| 5.1.2 Uživatelské role v Modulu 1 | 22 |
| 5.2 Modul 2 | 24 |
| 5.2.1 Business flow..... | 24 |
| 5.2.2 Uživatelské role v Modulu 2 | 25 |
| 5.2.3 Rozdělení typů pozic | 27 |
| 5.3 Modul 3 | 28 |
| 5.3.1 Business flow..... | 28 |
| 5.3.2 Stavy směny..... | 29 |
| 5.3.3 Uživatelské role v Modulu 3 | 30 |

| | | |
|--|---------------------------------------|-----------|
| 5.4 | Modul 4 | 31 |
| 5.4.1 | Business Flow Modulu 4 | 31 |
| 5.4.2 | Uživatelské role v Modulu 4 | 32 |
| 5.5 | Administrační práce se systémem | 33 |
| 6 | Tvorba scénářů | 34 |
| 7 | Analytické třídy | 36 |
| 8 | Příprava implementace | 38 |
| 8.1 | Vývojové prostředí | 38 |
| 8.2 | Rozvrstvení aplikace..... | 39 |
| 9 | Databázová technologie..... | 42 |
| 9.1 | Databáze Visual Studia 2010..... | 42 |
| 9.2 | SQLite..... | 44 |
| 9.2.1 | Nevýhody SQLite | 44 |
| 9.2.2 | Výhody SQLite..... | 44 |
| 9.2.3 | DB Managery SQLite..... | 45 |
| 10 | Datový model | 47 |
| 10.1 | Popis některých tabulek..... | 48 |
| 11 | Návrhový model..... | 50 |
| 12 | Implementace a GUI | 52 |
| 13 | Testovací strategie | 56 |
| 13.1 | Motivace | 56 |
| 13.2 | Testovací scénáře..... | 56 |
| 13.3 | Testování v reálném provozu | 57 |
| Závěr | 59 | |
| Slovníček pojmů..... | 61 | |
| Literatura | 62 | |
| Příloha A – Kompletní seznam požadavků včetně popisu..... | 63 | |
| Příloha B – Sekvenční diagramy Business flow modulů..... | 69 | |
| Příloha C – Realizace požadavků v rámci Use Case | 73 | |
| Příloha D – Ukázka importovaných a exportovaných dat..... | 76 | |
| Příloha E – Testovací scénáře..... | 77 | |

Seznam zkratk

| | |
|-----|--|
| FS | File system |
| CSV | Coma separated values. Používaný formát pro přenos dat mezi systémy. |
| DB | Databáze |
| IT | Informační technologie |
| UC | Use Case, případ užití |
| PU | Případ užití |
| GUI | Graphic User Interface = Grafické uživatelské rozhraní |
| UP | Unified Process, Unifikovaný proces |
| IDE | Integrated Development Environment = Vývojové prostředí |

Seznam obrázků

| | |
|--|----|
| Obrázek 1 – Fáze UP | 13 |
| Obrázek 2 – Standardní flow práce s modulem 1 | 21 |
| Obrázek 3 – Use case modulu 1 | 22 |
| Obrázek 4 – Standardní flow práce s Modulem 2 | 24 |
| Obrázek 5 – Use case Modul 2 | 25 |
| Obrázek 6 – Standardní flow práce s Modulem 3 | 28 |
| Obrázek 7 – Use case Modulu 3 | 30 |
| Obrázek 8 – Standardní flow práce s modulem | 31 |
| Obrázek 9 – Use case diagram Modulu 4 | 32 |
| Obrázek 10 – Administrace systému | 33 |
| Obrázek 11 – Sekv.Diagram, vytvoření účtu | 33 |
| Obrázek 12 – Scénář k UC editace obrázku pozadí | 34 |
| Obrázek 13 – Ukázka alternativního toku UC změna uživatele | 35 |
| Obrázek 14 – Analytické třídy | 36 |
| Obrázek 15 – Screen vývojového prostředí Microsoft Visual Studio | 39 |
| Obrázek 16 – Interní DB Visual Studio 2010 – Diagram | 42 |
| Obrázek 17 – Interní DB VS 2010 - Zdroje dat | 43 |
| Obrázek 18 – Databázový soubor a dll knihovny | 45 |
| Obrázek 19 – SQLite Manager 8.0 | 45 |
| Obrázek 20 – SQLite Exper Personal GUI | 46 |
| Obrázek 21 – ER Diagram datového modelu | 48 |
| Obrázek 22 – Zjednodušený model tříd | 50 |
| Obrázek 23 – Fyzická struktura tříd aplikace | 52 |
| Obrázek 24 – Přihlašovací formulář aplikace | 53 |
| Obrázek 25 – GUI Úvodní obrazovky | 53 |
| Obrázek 26 – GUI Modulu 1 | 54 |
| Obrázek 27 – GUI Modulu 2 | 54 |
| Obrázek 28 – GUI Modulu 3 | 55 |
| Obrázek 29 – GUI rozhraní Modul 4 | 55 |
| Obrázek 30 – Screen z testovacího scénáře | 57 |

Seznam tabulek

| | |
|---|----|
| Tabulka 1 - Zobrazení časového rozložení práce s aplikací | 20 |
|---|----|

1 Úvod

Impulsem k nápadu tématu této práce byla znalost zavedených postupů bezpečnostní agentury při realizaci zabezpečení velkých kulturních, sportovních a dalších akcí. Pole působnosti firem podobného zaměření není příliš spjata s moderními možnostmi informačních technologií a je vhodné pro zavádění a testování nových postupů.

Po vznesení námětu na vylepšení ve zmíněné firmě byla nabídnuta pomoc při zkoumání funkčních přístupů a business metodik plánovaného systému. Vedení firmy nápad přijalo jako možnou alternativu, i když v počátcích byly vzneseny pochybnosti o realizovatelnosti celého projektu, a celá realizace byla interně předem definovaná jako vytvoření prototypu systému pro ověření funkčnosti. Při prokázání dosažení vytyčených cílů bylo přislíbeno pokračování business podpory a dodání potřebných metodik pro analýzu a funkční vyvíjení systému.

Od počáteční myšlenky byla zřejmá nutnost úzké spolupráce a komunikace s odborníky z firmy, zejména z důvodu správné definice potřebné funkčnosti aplikace. Bylo nutné zajistit i odborné vedení po technologické stránce, kdy bylo vhodné vypracovat projekt jako diplomovou práci s využitím vedení akademických pracovníků a znalostí nabytých v absolvovaných předmětech během studia. Další znalosti potřebné pro realizaci projektu, zejména prohloubení používaných metodik a technologií jsou získány samostudiem a samostatnou prací či konzultací s odborníky. Klíčovou oblastí vyžadující nové znalosti bylo použití některých nezvyklých technologií, které ale přesně odpovídaly požadavkům zadavatele.

Cílem práce je vytvořit ze zmíněného nápadu funkční a použitelnou aplikaci, tedy projít všemi kroky vývoje nového systému. V prvních krocích bylo nutné důkladně analyzovat nutné funkční vlastnosti. Dále převést tyto požadavky na analytické modely, definovat potřebné vztahy a navrhnout chování aplikace. Také bylo nutné s ohledem na technologické možnosti konzultovat požadavky agentury a dospět k funkčnímu a realizovatelnému řešení. Po odsouhlasení návrhu začala fáze implementace dle vytvořené analýzy. Během implementace bylo nutné řešit některé funkční detaily a vzniklo několik změnových požadavků. Některé z nich bylo možné začlenit do aplikace, ale některé, které příliš narušovaly architekturu, bylo vyhodnoceno odložit do případných budoucích verzí. Na závěr bylo plánováno výsledný systém otestovat. Jednak dle připravených testovacích scénářů a jednak v reálném provozu v akci. K tomuto testování přislíbila spolupráci bezpečnostní agentura. Nedílnou součástí bude vyhodnocení úspěšnosti a realizace projektu, s doporučeným rozhodnutím o dalším pokračování prototypu projektu.

Splnění všech těchto podmínek vede k cíli, kterým je vytvoření nástroje pro zjednodušení, ulehčení a systematizaci postupů pracovníků širšího vedení bezpečnostní agentury.

2 Motivace

2.1 Představení odběratele

Bezpečnostní agentura Čechymen byla založena v roce 2004 jako pokračování šestiletého externího působení na trhu bezpečnostních služeb v oblasti festivalové security, ostrahy objektů nebo fyzické ostrahy osob, event security a také zajištění požární ochrany nebo záchranářské služby. Všechny tyto služby poskytuje nejen vlastním zákazníkům, ale také renomovaným společnostem specializujícím se na ostrahu objektů a také pro nadnárodní bezpečnostní agentury působící v rámci celé ČR. V současné době zabezpečuje každý rok průměrně přes 10 velkých akcí s personálním obsazením 100 a více lidí, například velké festivaly, sportovní mistrovství a jiné. K tomu nespočet menších akcí s obsazením do 50 pracovníků.

2.2 Motivace pro nový systém

Aplikace nahradí stávající těžkopádné a pomalé postupy v řízení pracovních směn brigádníků na rozsáhlých akcích. Tabulky v Excelu jsou sice relativně jednoduchým řešením, ale zároveň dost provizorním. Při stávajících hodinových objemech a počtech participujících pracovníků se stávají značně nepřehlednými a vyhledávat či upravovat údaje v nich může pouze několik pracovníků, zpravidla těch, kteří je vytvoří. Nalezení informace v nich je spíše podvědomou orientací, kdy onen pracovník ví, „že to někde napsal“ než že umístěná informace v nich je logická a intuitivní. V praxi se tak běžně stává, že celou organizaci směn na festivale zajišťují jeden, maximálně dva pracovníci, kteří jsou však touto činností plně vytíženi, nestíhají jinou práci a taky jsou vystaveni značné psychické námaze.

Program Excel jako formát uložení dat navíc ani není koncipován pro takto rozsáhlé datové informace, daleko lépe se pro tento typ informací hodí plnohodnotná databáze nebo soubor na FS vytvořených speciálně pro tyto účely. Excel nemá možnost kontroly konzistentnosti nebo integrity dat, neobsahuje žádný kontrolní mechanismus z pokročilejších databází, jako jsou například v dnes nejčastěji používaných relačních databázích pravidla jako normální formy a další. Dále přímé umístění informace v buňce listu takřka znemožňuje jakoukoli automatickou kontrolu, jakou by mohlo být např. překrývání směn, vytěžování pracovníků a jiné.

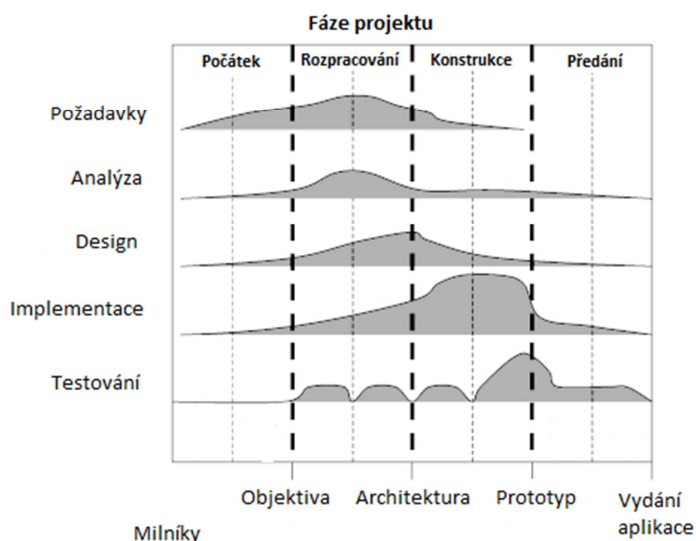
Aplikace bude vhodně doplňovat systém HEAD, který již ve firmě osvědčeně funguje, ale nedokáže pokrýt specifickou oblast použití. Zejména neobsahuje funkce pro plánování zdrojů na akce včetně jejich vizualizace a také nedokáže pružně reagovat na změny v reálném čase přímo v době konání akce. HEAD je spíše evidenční a účetní systém, který pomáhá pracovníkům v kanceláři spravovat objednávky a zakázky ve vyšších úrovních správy. Nová aplikace by měla pomoci vedoucím pracovníkům v terénu, tedy spíše na nižší úrovni managementu.

2.3 Oblast nasazení nové aplikace

Aplikace najde využití pro event akce o obsazení pracovníků 50-ti a více. Pro menší akce potřebné tak důkladné evidování a plánování. Naopak pro velké eventy je důkladné naplánování klíčem k úspěchu, jak zvládnout desetitisíce nebo statisíce návštěvníků. Maximální velikost akce, kterou bude aplikace schopna zpracovat, je téměř neomezená, limity jsou v technických možnostech použitých technologií. Největší eventy, které firma zabezpečuje, se pohybují okolo 250 – 300 pracovníků, což bude aplikace schopná spravovat.

2.4 Rozložení práce

Průběh projektu byl rozložen do několika fází dle metodiky UP. Většina práce prochází přes všechny fáze projektu, ale každá má svou kulminaci jinde, a většinou i v jiné fázi projektu.



Obrázek 1 – Fáze UP

Na počátku projektu je potřeba sesbírat podklady. K tomu slouží zejména tvorba požadavků, co by zamýšlená aplikace měla umět. Pokud nemáme dostatečné funkční povědomí o aplikaci, je nutné si také zjistit, jak procesy probíhají v realitě nyní, za současného stavu.

V další fázi probíhá samotná analýza budoucí aplikace. Tvoří se případy užití, později analytické třídy. Zde se formuje funkčnost budoucí aplikace. Z hlediska milníků se těžiště přesouvá od zjišťování objektiv práce ke konstrukci architektury. Dále nastává tvorba designu aplikace, kdy už je zřejmé, jak bude výsledná aplikace vypadat. V této fázi se formují návrhové třídy, architektura aplikace, použité technologie nebo prototypy uživatelského rozhraní.

Po těchto fázích začíná proces implementace aplikace, kdy již jsou k dispozici první výsledky pro zadavatele. Fáze konstrukce končí předáním prototypu aplikace a během

tohoto období kulminuje fáze testování. Po těchto fázích již nastává nasazení aplikace, předání zadavateli případně předání dokumentace.

Během vytváření této práce byl kladen důraz na dodržování metodiky Unifikovaného Procesu, i když rozsah aplikace nebyl tak velký jako u projektů s vícečlenným týmem. I v tomto případě se ale například na sběru požadavků, funkčnímu designu a nakonec testování podílelo více lidí, takže některé metody týmové práce byly využity.

3 Požadavky

Největší část práce na definici požadavků je vždy v prvních fázích tvorby analýzy, tedy ve fázích zahájení a rozpracování. Toto je logické, protože jinak by nebylo definováno, co vlastně tvořit. Požadavky tedy tvoří základ celé analýzy, a pokud je správně definujeme, máme nakročeno k úspěšnému projektu. Už na začátku musí být jasný přehled o tom, co má budoucí systém dělat, a toto musí být také přesně nadefinováno a zdokumentováno. Navíc, při tvorbě komerčních projektů je toto podstatný krok při definici smlouvy a pozdější akceptaci přejetého projektu. Tvorbu požadavků nelze obecně konstruovat z kanceláře, ale je nutné jít k zadavateli a hovořit s budoucími uživateli softwaru. Budoucí uživatelé sami umí nejlépe popsat, co od systému očekávají a jaké jsou nutné podmínky pro úspěšné začlenění do business procesů firmy.

Požadavky pro zpracovávanou aplikaci byly definovány ve spolupráci právě s odborníky z bezpečnostní agentury. Většina jich tedy vychází ze zkušeností z praxe a potřeb jednotlivých osob, které přijdou s aplikací do styku. Získávání požadavků bylo prováděno formou konzultací s manažery společnosti, a formou jednoduchých dotazníků byli zpovídáni také řadoví zaměstnanci firmy. Tyto základní požadavky byly doplněny a upraveny dle potřeb a možností zamýšleného provedení aplikace. Při definici požadavků byl brán ohled také na použité systémy a plánované HW vybavení společnosti a jejich zaměstnanců. Dalším hlediskem byla nutnost použití otevřených technologií, neboť se nedá počítat s nákupem rozšířených licencí. Výsledná sada požadavků byla konzultována a odsouhlasena business odborníky z praxe. Pro základní definici požadavků použijeme rozdělení na Funkční a Nefunkční požadavky.

3.1 Funkční požadavky

Funkční požadavek je formulace toho, co by měl systém dělat, kdy popisuje požadovanou funkci systému. Funkční požadavky identifikují nutné úkony, aktivity a akce, které musí být vykonány. Analýza funkčních požadavků bude použita jako základ „top-level“ funkce pro funkční analýzu.

3.1.1 Modul 1

Modul 1 bude ze základní funkčnosti obhospodařovat plánování akce. Musí tedy umět vytvořit model akce, musí mít schopnost vložení předem dohodnutých značek, ať už materiálu nebo pozic pro pracovníky. Tyto značky by měly být editovatelné. Musí umět zobrazovat podkladovou grafiku – obrázek nebo mapu. Kompletní soupis požadavků je uveden v příloze A.

3.1.2 Modul 2

Modul 2 bude primárně sloužit pro nadefinování a rozplánování směn pracovníků před začátkem akce. Musí být tedy možné prohlížet seznam naplánovaných pozic v modelu. Dále musí být možné na tyto pozice přidat směny a kompletní management těchto směn, tedy zejména časové úseky a kontrolu stavu směny. Dále je také nutné přidávat pracovníky, kteří budou na daných pozicích a směnách pracovat. Přidání pracovníka musí být jednak dávkové, tedy importem dat v přenositelném formátu ze systému HEAD, ale

také musí existovat možnost zadat pracovníka ručně. Kompletní soupis požadavků je uveden v příloze A.

3.1.3 Modul 3

Práce s Modulem 3 bude probíhat výhradně během průběhu akce. Modul slouží k zaznamenávání směn, odpracovaných hodin a celkové práce pracovníků během akce. V ideálním případě se pouze zaznamenávají změny směny, ale v reálné praxi často nastává i potřeba opětovného zadání nové pozice nebo nové směny dle nových požadavků produkce dané akce. Proto musí být v tomto kroku práce možné i zadání těchto nových dat. Kompletní soupis požadavků je uveden v příloze A.

3.1.4 Modul 4

Ačkoli dle korektního business flow, by se práce s Modulem 4 měla odehrávat až po skončení akce, i zde je velká pravděpodobnost využívání v předchozích krocích. V Modulu 4 je totiž uložena velká část funkčnosti pro export dat, aktuální zobrazení stavu modelu a dalších. Nosné vlastnosti a požadavky čtvrtého modulu jsou tedy tvorba statistik a to i dle zadaného filtru. Exportování dat (statistik) do přenositelného formátu CSV pro využití v systému HEAD. Mezi souhrny patří také finanční souhrn pro jednotlivé pracovníky i jejich hodinové souhrny. Dále zde musí být možnost zobrazení evidence použitého značeného materiálu a jeho export. Zde je také vhodné místo pro definování požadavku na uzavření celé akce (nemožnost editace a přidávání nových dat) po skončení celé akce, a také její opětovné otevření. Kompletní soupis požadavků je uveden v příloze A.

Dodržení těchto požadavků zabezpečí použitelnost a pohodlnou práci s programem. Jakákoli změna základních požadavků musí být konzultována s business analytikem znalým potřeb systému.

3.2 Nefunkční požadavky

Nefunkční požadavek se dá definovat jako omezující podmínka uvalená na daný systém. Kritéria, podle kterých se dá hodnotit funkce systému spíše než jeho specifické funkce. V podstatě tedy definuje kvalitu systému. V našem případě není třeba rozdělovat nefunkční požadavky podle modulů, ale stačí vytvořit jednu společnou sadu nefunkčních požadavků pro celý systém. Kompletní soupis požadavků je uveden v příloze A.

4 Modelování případů užití

Modelování případů užití je následným krokem po specifikaci požadavků. Dá se označit za jednu z forem inženýrství požadavků. Různé zdroje uvádějí různé definice případů užití a jejich použití při analýze systému. Dle oficiální interpretace jazyka UML se dá definice odvodit, není však příliš konkrétní. Pro konkrétnější upřesnění si z pravidla musíme vybrat některou z podrobnějších definic.

Podle definice Jamese Rumbaughu uvedené v knize „The Unified Modeling Language Reference Manual“ (Rumbaugh, 1998) lze tvrdit, že případ užití je specifikace posloupností činností včetně proměnných posloupností a chybových posloupností, které systém, podsystém nebo třída může vykonat prostřednictvím interakce s vnějšími (externími) aktéry.

Případ užití je něco, co aktér od systému očekává. Případy užití jsou vždy iniciovány aktérem a také jsou vždy napsány z pohledu aktéra. Je to funkčnost, kterou systém disponuje a kterou někdo z jeho uživatelů od systému očekává. To samo o sobě souvisí s faktem, že případ užití je tvořen na základě analýzy požadavků. Pokud budoucí uživatel zadá požadavek na systém, tento potom musí systém prostřednictvím případu užití splňovat.

Diagramy případů užití zároveň definují i hranice systému. Případ užití by měl být zpravidla uvnitř systému (jedná se o funkčnost), a měl by být iniciován zpoza hranice systému. V pokročilejší interpretaci se ale lze setkat například s možností specifikace podsystému nebo interakce s jiným systémem nebo okolím.

Případ užití může obsahovat určité prekondice nutné k tomu, aby mohl být proveden. Typicky se jedná například o podmínku být v systému přihlášený. Tyto prekondice se uvádějí v definici jednotlivého případu užití. Označují se jako vstupní podmínky – tedy podmínky, které omezují stav systému před tím, než je možné případ spustit.

Naopak výstupní podmínky omezují stav systému po skončení případu užití. Tedy definují, co musí být vyhodnoceno jako pravda po dokončení PU. Z dalších atributů PU, které je třeba vytvořit, lze jmenovat například jedinečný identifikátor, textový popis, přiřazení k aktérům (viz níže), napojení na požadavky, scénář s jednotlivými kroky (viz kapitola „Tvorba scénářů“) a další.

4.1 Hranice systému

Definice hranice systému nemusí být vždy triviální a v průběhu analýzy se může pohybovat, než je nalezena její ideální poloha. Hranice systému definuje, co je součástí systému a co už leží mimo systém. Určování polohy hranice systému má obrovský vliv na funkční požadavky (a někdy i nefunkční) a jeho správná definice je tedy velmi podstatná. Anglické označení je „System boundary“ a v UML diagramech se značí uzavřeným obdélníkem. Jednotlivé případy užití jsou uvnitř, zatímco aktéři jsou mimo systém.

4.2 Aktéři

Jedná se o role přidělené entitám, které budou v interakci se systémem. Nemusí se jednat vždy o živé (lidské) entity, aktérem může být i předmět nebo jiný systém, pokud není součástí námi modelovaného systému. Je třeba si uvědomit, že za aktéra zpravidla nelze dosazovat konkrétní osobu nebo předmět. Aktér je obalová schránka pro entitu, která komunikuje se systémem. Na konkrétním případě to znamená, že pro roli „Pozorovatele“ můžeme sice dosadit konkrétního člověka XY ale ten samý člověk může být zároveň i správcem systému. Pro správnou interpretaci v diagramu je tak nutné za aktéry považovat vždy „obalové“ entity. Aktérem může být například i čas. Například pokud jsou plánovány procedury spouštěné na základě časové hranice, například každý den v 6:00, potom je aktérem právě čas.

Při hledání aktérů se zpravidla ptáme na otázky typu „Kdo nebo co používá systém?“ či „Kdo nebo co komunikuje se systémem?“. Aktéři jsou vždy externí a tedy mimo naši kontrolu. Chování aktéra nemůžeme naplánovat ani řídit. Lze jen definovat reakce systému po akci aktéra. Tím, že aktér komunikuje bezprostředně se systémem, nám usnadňuje nalezení hranice systému. Ta leží mezi aktérem a případem užití.

Aktéři jsou s případy užití spojeni pomocí relací. To jsou vztahy, které definují propojení mezi aktéry i mezi různými případy užití navzájem. Relací je mnoho druhů podle typu relace, která mezi entitami funguje. Relace musí být vždy smysluplná, pokud není nutná, nemusí vůbec v modelu existovat.

5 Funkční design

V následující kapitole je zaměření na popis funkčnosti modelu. Není kladen důraz na detailnost ani přesnou správnost sémantiky UML diagramů, ale názornost a pochopení procesů které bude později třeba namodelovat. Použité diagramy byly zvoleny jako nejvhodnější pro názornou ukázkou a pochopení business procesů probíhajících v praxi.

System bude funkčně rozdělen na čtyři moduly. Každý z nich bude spravovat rozdílnou část eventu. Nejdříve bude první modul napomáhat naplánování akce, druhý pomůže personální a materiální přípravě, třetí bude využíván v průběhu akce pro fyzické zaznamenávání směn a čtvrtý modul bude pomáhat při vytváření statistik a exportu dat pro stávající systém HEAD.

Přes všechny moduly bude procházet objekt „model“, který znázorňuje virtuální reprezentaci reálné akce. Model se vytvoří v prvním modulu a bude se s ním pracovat jako s ústřední entitou celé aplikace. Přes práci s modelem se ukládají všechna vložená data, charakteristiky a informace, které jsou nutné pro odpovídající reprezentaci zabezpečené akce v realitě.

Model je tedy reflexí reality s omezením na zobrazení nezbytně nutné pro správnou reprezentaci. Detailní zobrazení reality je omezeno z důvodu snížení složitosti a zjednodušení práce s modelem.

V tabulce 1 je graficky znázorněn reálný postup práce s aplikací vzhledem k plynoucímu času. Na nejvyšším řádku je zaznamenáno, kterého modulu se aktuální práce týká. O řádek níže je ilustrativní příklad práce, který se v té době provádí. Na dalším řádku je fáze akce převzatá z reálného světa, tak jak by danou fázi nazvali pořadatelé, tedy zákazník. Níže je pro dokreslení reality zaznačen příklad letošního festivalu Rock for People, kdy se hlavní program konal mezi dny 1.7 – 5. 7. 2013. Jak je z tabulky vidět, příprava celého projektu byla ale mnohem delší.

Pro konkrétní příklad modelu akce hudebního festivalu Rock for People 2013 je tedy možné tvrdit, že životní čára modelu začíná někdy na konci roku 2012 a končí koncem srpna až začátkem září 2013. Co se týče konkrétních dat, které jsou do aplikace při takovéto podobné akci vloženy, se jedná o cca 150 - 170 pracovníků při objemu časové práce 8 000 – 10 000 odpracovaných hodin. Největší zatížení aplikace je v průběhu akce, tedy při práci s třetím modulem. Po celou dobu musí být systém dostupný, musí být schopný dělat zálohy vložených dat a musí umět zálohy obnovovat. System je desktopová aplikace, což zaručuje dostupnost, robustnost, jednoduchost i bezpečnost. Zálohy dat je možné ukládat na pevný disk i výměnná média nebo posílat zabezpečenou cestou přes internet (toto nezabezpečuje vlastní aplikace – je nutné použít nástroje třetích stran).

| | | | | | |
|--------------------------|------------------------------------|-------------------|----------------------|------------------------------|--------------------|
| Modul aplikace: | Modul 1 | | | | |
| Typ práce s modelem: | Vytvoření modelu, Příprava nabídky | Potvrzení nabídky | Areál, mapy, objekty | Pracovní pozice, koridory... | |
| Fáze akce | Oznámení akce | Výběrové řízení | Získání zakázky | Strategické plánování | Detailní plánování |
| Reálný příklad: RfP 2013 | 2012 | Leden 2013 | Březen 2013 | Květen 2013 | 20.5. - 10.6. 2013 |

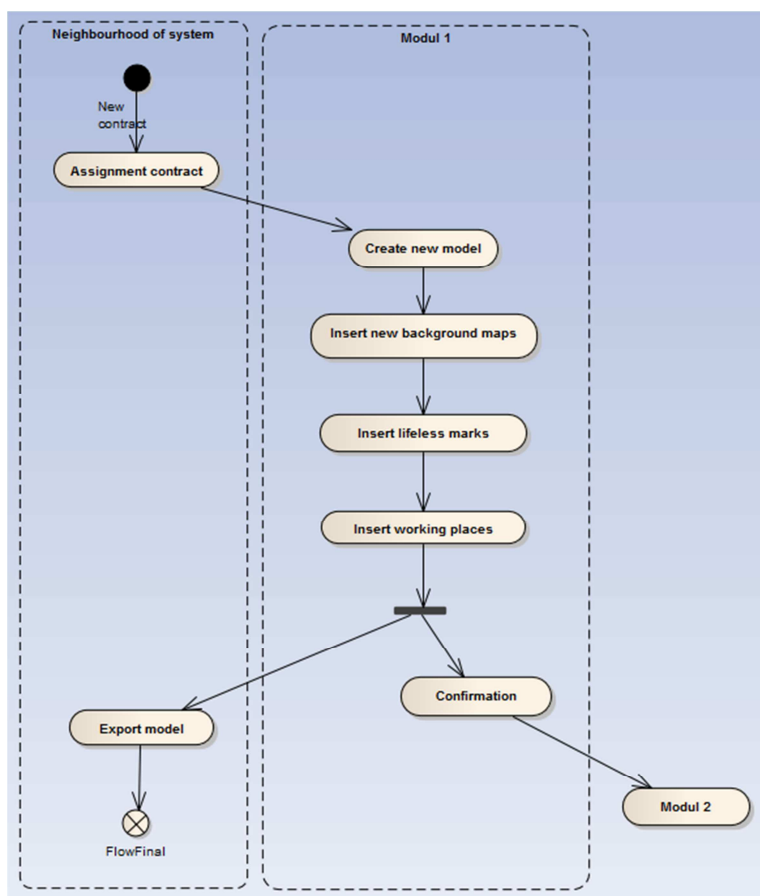
| | | | | | | | |
|--------------------------|--------------------|--------------------|--------------|---------------|--|--------------|-----------------------------|
| Modul 2 | | Modul 3 | | | Modul 4 | | |
| Pracovníci | Skład | Směnový management | | | Vytváření statistik, účtování, export výsledků | | Potvrzení a uzavření modelu |
| Příprava lidských zdrojů | Příprava materiálu | Začátek akce | Průběh | Ukončení akce | Účtování akce | Statistika | Uzavření akce |
| Červen 2013 | 15.6. - 30.6. | 2.7.2013 | 2.-10.7.2013 | 10.7.2013 | Červenec 2013 | 1.-15.8.2013 | Konec srpna 2013 |

Tabulka 1 - Zobrazení časového rozložení práce s aplikací

5.1 Modul 1

První část aplikace je modul pro rozvržení areálu, namodelování zdrojů a naplánování pracovních pozic a materiálu pro zabezpečení celé akce. Tento modul využívají standardně nejprve bezpečnostní manažeři, kteří navrhnu optimální zabezpečení akce a rozplánují zdroje. Tento modul je také vhodný pro prezentaci zákazníkovi, neboť vhodně ilustruje plánovanou situaci. Zákazník může také sám upravit požadavky po seznámení s prognózami bezpečnostních manažerů.

5.1.1 Business flow

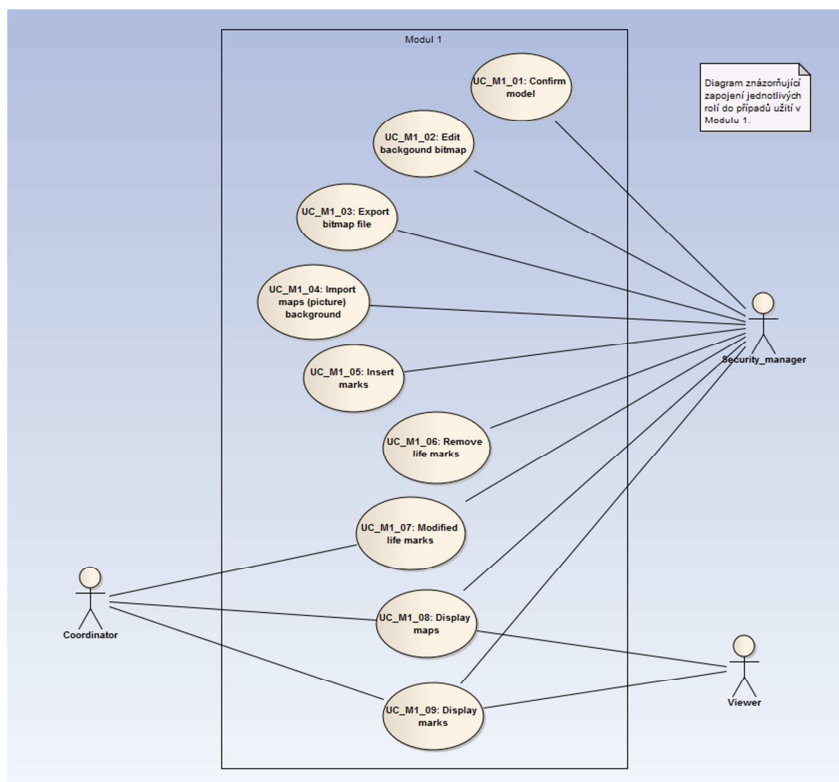


Obrázek 2 – Standardní flow práce s modulem 1

Na obrázku 2 je vidět standardní work flow práce prvního modulu. Work flow není v aplikaci hlídáno žádnými podmíněnými kroky, nicméně doporučená business funkce aplikace vyžaduje naznačené postupy. Flow začíná impulzem z prostředí mimo systém, což je zpravidla zadání kontraktu. Dále následují první kroky práce v systému. Nejprve je nutné inicializovat model a doplnit základní prekvizice (metadata modelu). Poté je možné začít s modelem plnohodnotně pracovat a zkusit jednotlivé návrhy bezpečnostního managementu. Systém zároveň graficky jednotlivé navržené situace zobrazuje a napomáhá tak představě o budoucí podobě eventů. Po dokončení návrhu může být model exportován v grafické podobě pro prezentaci zákazníkovi nebo dalším zasvěceným lidem, a po potvrzení správnosti návrhu by mělo být flow předáno do Modulu 2.

5.1.2 Uživatelské role v Modulu 1

System nabízí několik rolí pro práci s prvním modulem. Jednotlivé role jsou odstupňovány dle jejich přístupových práv. Standardní práva, která role obdrží od systému, jsou zdůvodněna business funkcí dané role a nedoporučuje se práva měnit. Sada práv k jednotlivým funkcím byla odvozena konzultací a požadavky zadavatele. Standardní role, které budou pracovat s modulem 1, jsou *Security manager*, *Coordinátor* a *Viewer*. Systemová role *Administrator* by s modulem neměla přímo pracovat, ale jen přidělovat role ostatním uživatelům. *Administrator* však má plný přístup ke všem funkcím systému.



Obrázek 3 – Use case modulu 1

Security manager

Uživatelská role bezpečnostního manažera je určena pro pracovníky s dostatečným know-how pro řízení zdrojů na masových akcích. Tito manažeři plánují bezpečnostní politiku akce s ohledem na požadavky klienta – pořadatele akce. Pracovníci vykonávající tuto funkci musí mít znalosti finanční náročnosti jednotlivých zdrojů do akce zařazených, znalost požadavků a možností klienta a také znalost dostupnosti zdrojů ze strany vykonavatele – tj. bezpečnostní agentury. Bezpečnostní manažer zpravidla zajišťuje se zákazníkem veškerou komunikaci týkající se věcných požadavků na akci a snaží se nalézt optimální řešení pro obě strany i s ohledem na maximální pohodlí účastníků akce i pracovních sil.

Role bezpečnostního manažera má v aplikaci po administrátorovi největší množství práv. Teoreticky může zajišťovat celou akci včetně všech menších úkolů pro její naplánování potřebných, ale v praxi tyto menší úkoly deleguje na ostatní role a sama obstarává práci

jakéhosi „architekta“ fungování akce. Tato role zpravidla zakládá novou akci a navrhuje rozmístění strategických zdrojů, jako jsou mobilní budovy, bariéry, transportní koridory, únikové východy či rozvržení prostor pro shromažďování lidí, nákupní zóny (prostory pro prodejní stánky) nebo prostory pro dislokaci jednotek záchranných služeb a zázemí akce. U lidských zdrojů (pracovních pozic) zpravidla pouze určí počty, případně rozdělení do zón a přesné umístění již provádějí jiné role.

Coordinator

Koordinátor akce je člověk, který bude řídit jednotlivé (všechny) pracovníky na dané akci. V průběhu konání akce bude dohlížet na korektní průběh provozu jednotlivých pracovních pozic a v době plánování akce (tedy v prvním modulu) se stará zejména o upřesnění umístění přidělených zdrojů pracovních míst. V tomto přesném umístění musí zohlednit rizikovost a důležitost jednotlivých provozů. Dle typu pracovních pozic, tedy možností obsazování jednotlivými pracovníky může určovat například umístění VIP strážců do prostorů backstage, psodů pro ostrahu hranic areálů, specialistů pro vstupní kontroly na vstupy nebo ostrahu stage pro jednotlivé hudební produkce. Koordinátor musí mít znalost celého prostředí a musí umět vytipovat krizové místa, na které pak může posílit pracovní pozice, stejně jako umět rozeznat místa s nízkým rizikem, na které není třeba dohlížet vůbec nebo jen s minimálním zatížením zdrojů.

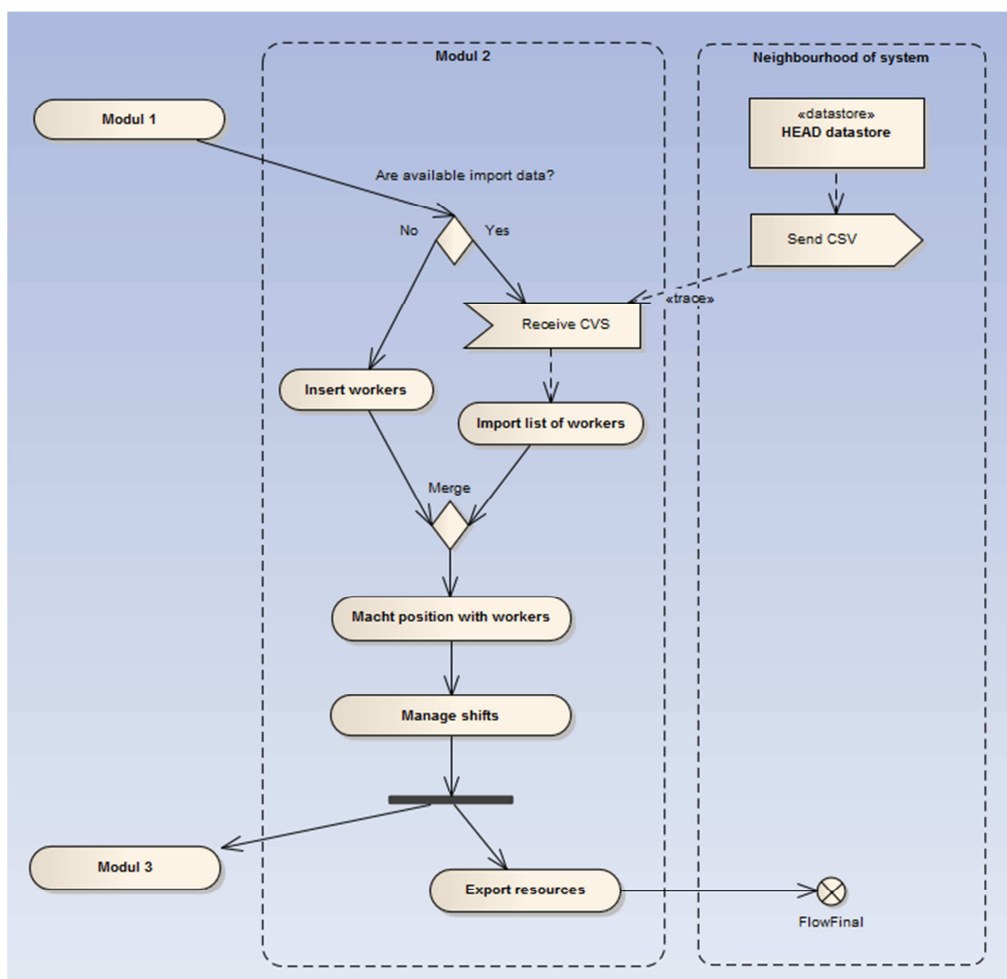
Viewer

Role pozorovatele slouží typicky například pro pracovníky, kteří by měli být seznámeni se strategií zabezpečení akce, ale nemají mít možnost ani práva do strategie jakkoli zasahovat. Roli pozorovatele tak můžou dostat důvěryhodní zaměstnanci, kteří by měli naplánování zdrojů znát (například pro informování jiných zaměstnanců, návštěvníků nebo pracovníků) nebo například typicky zákazníci či produkce dané akce. Zákazník většinou nemá dostatečné znalosti pro určování strategie, ale chce být vždy informován o průběhu plánování celé akce včetně všech detailů a pro tyto účely je vhodná role pozorovatele. Z hlediska aplikace má pozorovatel právo cokoli zobrazit, ale nesmí nic měnit.

5.2 Modul 2

Druhý modul systému bude zajišťovat vhodné obsazení naplánovaných zdrojů modelu. V této části práce se systémem bude možné využít již fungující naplněné databáze stávajícího systému HEAD, který poskytne seznam pracovníků určených pro tuto akci. HEAD jako takový v současném stavu zajišťuje logiku přihlašování pracovníků na jednotlivé akce a také jejich výběr. Konečný seznam lidí vybraných pro danou akci bude možné z HEADu exportovat a importovat do nového systému. V něm potom manažer lidských zdrojů spáruje vhodné adepty na vhodné pracovní pozice. Nový systém navíc strom pracovních pozic i rozšiřuje na atomické rozdělení pozic.

5.2.1 Business flow



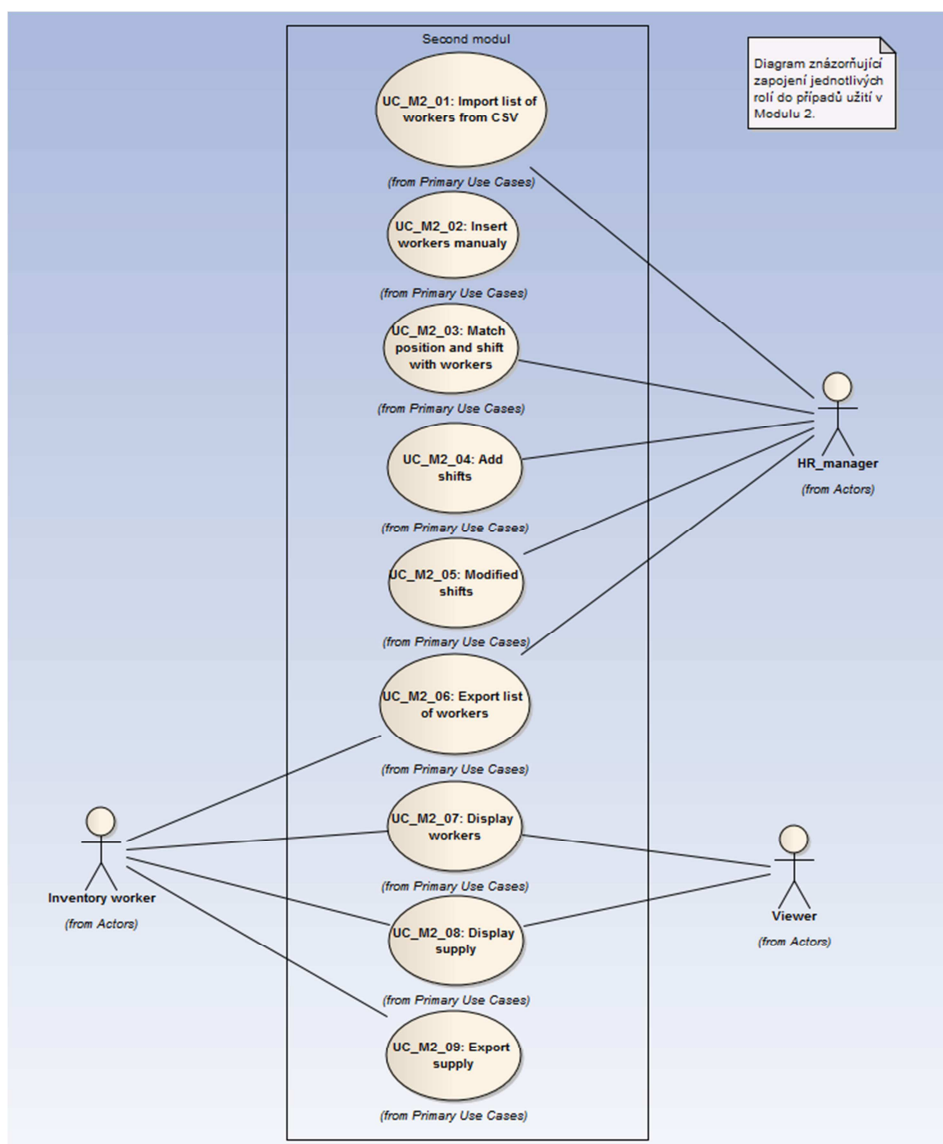
Obrázek 4 – Standardní flow práce s Modulem 2

Po předání flow z Modulu 1 je nutné rozhodnutí, zda je k dispozici import dat z jiného systému, respektive z HEADu. Pokud ano, načteme seznam pracovníků pro danou akci. Pokud ne, systém umožňuje ruční vkládání jednotlivých pracovníků. Obojí lze i kombinovat, pro zjednodušení práce je ale preferován automatický import. Dalším krokem je přiřazení načtených lidí na jednotlivé pracovní pozice, které systém obsahuje již z Modulu 1. Dále provede operátor jednotlivé rozdělení pracovních směn. Nakonec je flow

předáno do třetího modulu, a zároveň jsou připraveny statistiky zdrojů pro skladové oddělení. Sklad podle tohoto seznamu lidských a materiálních zdrojů může připravit materiál pro danou akci. I zde platí, že flow není v aplikaci hlídáno žádnými podmíněnými kroky, nicméně doporučená business funkce aplikace vyžaduje naznačené postupy.

5.2.2 Uživatelské role v Modulu 2

Pro práci s Modulem 2 jsou nejdůležitější role personálního manažera a také pracovníka skladu. Okrajově v druhém modulu stejně jako v prvním působí role pozorovatele, který však zde neslouží jako v prvním modulu pro zákazníka, ale spíše pro ostatní pracovníky firmy, kteří mohou sledovat stav a připravenost akce.



Obrázek 5 – Use case Modul 2

HR manager

Manažer lidských zdrojů je role zejména pro naplánování jednotlivých pracovních pozic pro jednotlivé pracovníky. HR manažer dostane pracovní pozice (z Modulu 1), které naplánoval Bezpečnostní manažer, a obsadí je určenými pracovníky. Výběr pracovníků je

na HR manažerovi, k rozhodování mu ale může pomáhat importovaný seznam určených pracovníků ze systému HEAD.

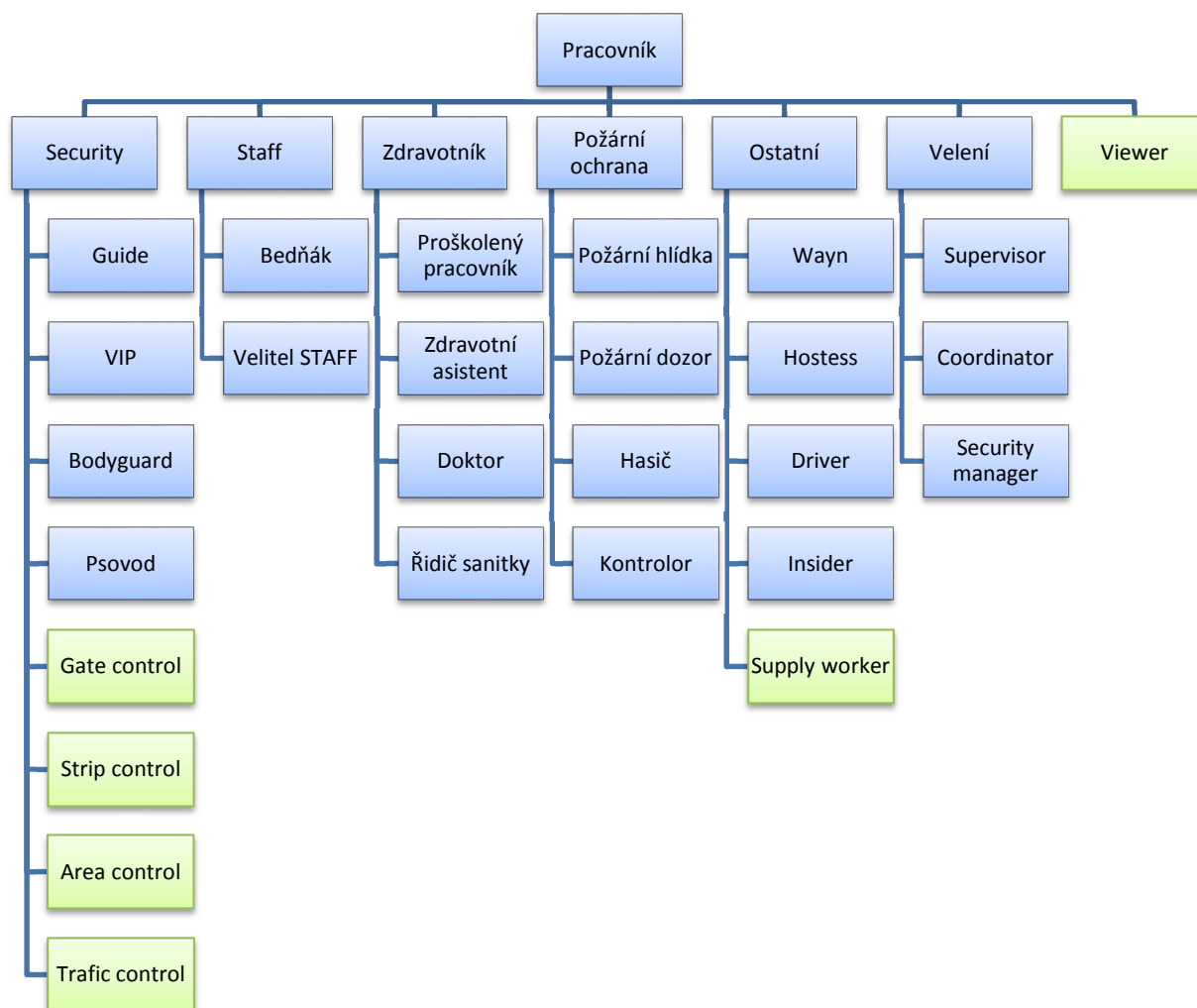
Inventory worker

Pracovník skladu si v aplikaci zobrazí naplánované materiální zdroje a obsazené lidské zdroje. Dle těchto informací může připravit veškerý potřebný materiál pro daný event. Pracovník skladu nemá práva nijak zasahovat do rozložení zdrojů (ani lidských, ani materiálních), ale má právo si veškeré údaje zobrazit a dodat stav připravenosti zdroje, zda je připraven či nepřipraven, není na skladě případně jiný stav. Skladník si také může nechat vyexportovat jednotlivé statistiky v plném i detailním rozsahu.

Viewer

Pozorovatel v tomto modulu může zobrazit pouze aktuální stav modelu. Nemá jakékoli práva na editaci ani na export statistik. Nemůže zasahovat do práce kohokoli. Tato role zde slouží pouze pro informování pracovníků pověřených jinou činností o aktuálním stavu modelu.

5.2.3 Rozdělení typů pozic



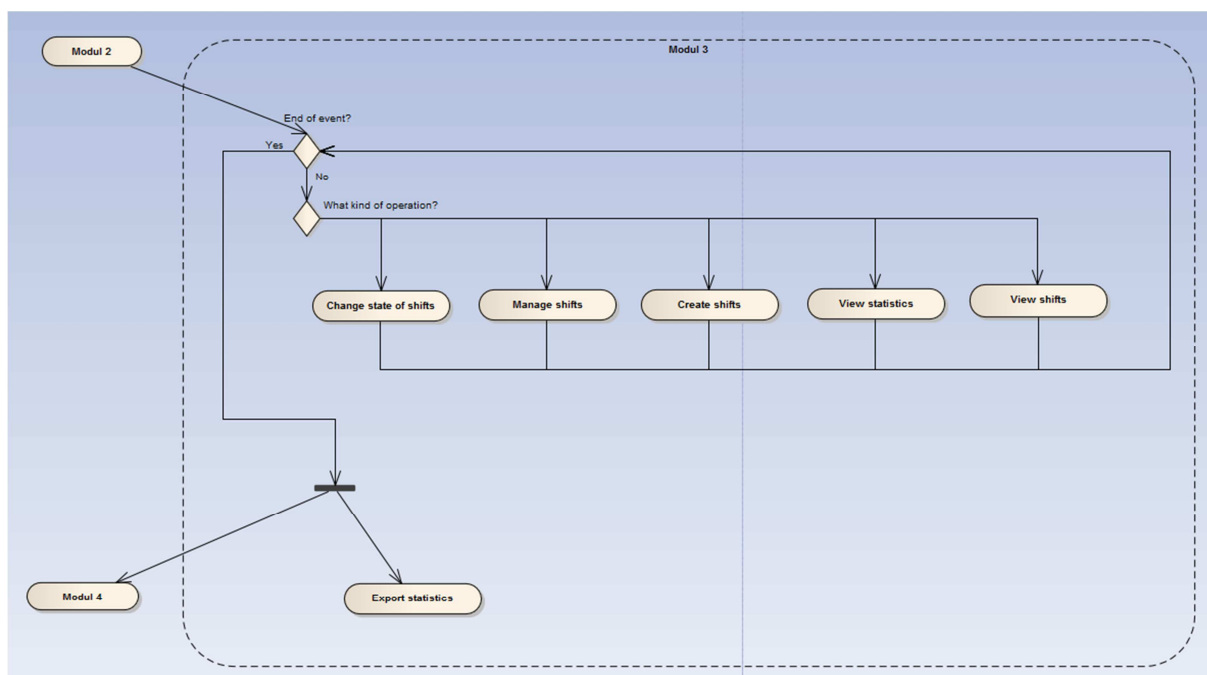
Graf 1 – Rozdělení typů pracovníků

Na Grafu 1 můžeme vidět rozdělení jednotlivých typů pracovních pozic. Světle modré jsou typy pracovních pozic, které obsahuje systém HEAD. Světle zelené jsou pracovní pozice rozšířené pro detailnější atomizaci jednotlivých pracovních úkonů. Při konverzi z HEADu do nového systému nevzniká problém, protože příznaky pozic se mohou (ale nemusí) ručně namapovat na stávající typy. Při konverzi zpět se nové typy (dokud je nebude HEAD podporovat) konvertují na nadřazenou pozici – tedy *Security*, *Ostatní* a pro *Viewer* obecný *Pracovník*.

5.3 Modul 3

Třetí modul slouží k práci s aplikací během samotné akce, přímo v terénu. Supervisor nebo Coordinator můžou pracovat přímo na v místě konání akce se systémem, zadávat do něj data či zjišťovat aktuální stav. Systém dokáže v reálném čase reagovat na požadavky operátorů, a to jak na změny dat (přidání nových směn pracovníků), tak i na požadavky pro zobrazení aktuálního stavu modelu, tedy reflexi reálného stavu akce. Pro řídicí pracovníky je důležité vědět, kolik lidí aktuálně pracuje, kolik volných lidí mají k dispozici a jak dlouhé měli volní lidé přestávky – zda jsou schopní znova nastoupit na směnu.

5.3.1 Business flow

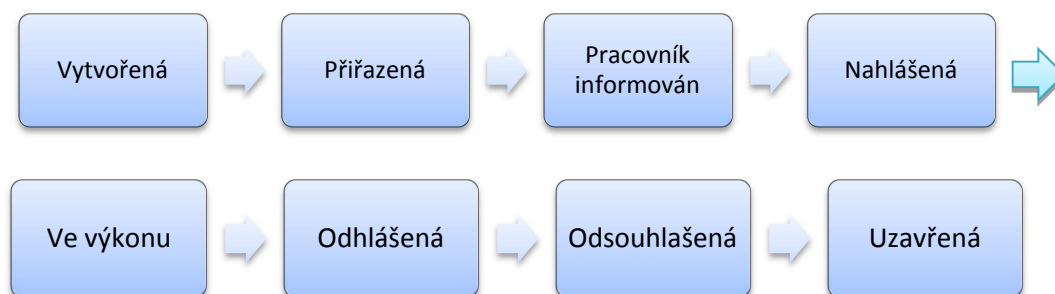


Obrázek 6 – Standardní flow práce s Modulem 3

Po předání toku práce z Modulu 2 se práce dostane do cyklu, který ukončí až fyzický konec akce. V tomto cyklu se opakuje stejná nabídka operací, kdy systém pracuje s daty získanými v předchozích modulech, a doplňuje nové záznamy o pracovních směnách, aktivitách pracovníků nebo zobrazuje statistiky, které pomáhají velitelům v jejich rozhodování o obsazování směn. Také jsou ale účinným nástrojem pro strategická rozhodnutí o přesunech pracovních pozic nebo odhalení slabých míst zabezpečení akce.

Mezi nejčastější operace, které operátoři systému (Coordinator nebo Supervisor) používají, patří vytváření směny, změna stavu směny, přesunování směn, zobrazování statistik nebo zobrazování směn. Jednotlivé stavy směny jsou popsány na grafu 2.

5.3.2 Stavy směny

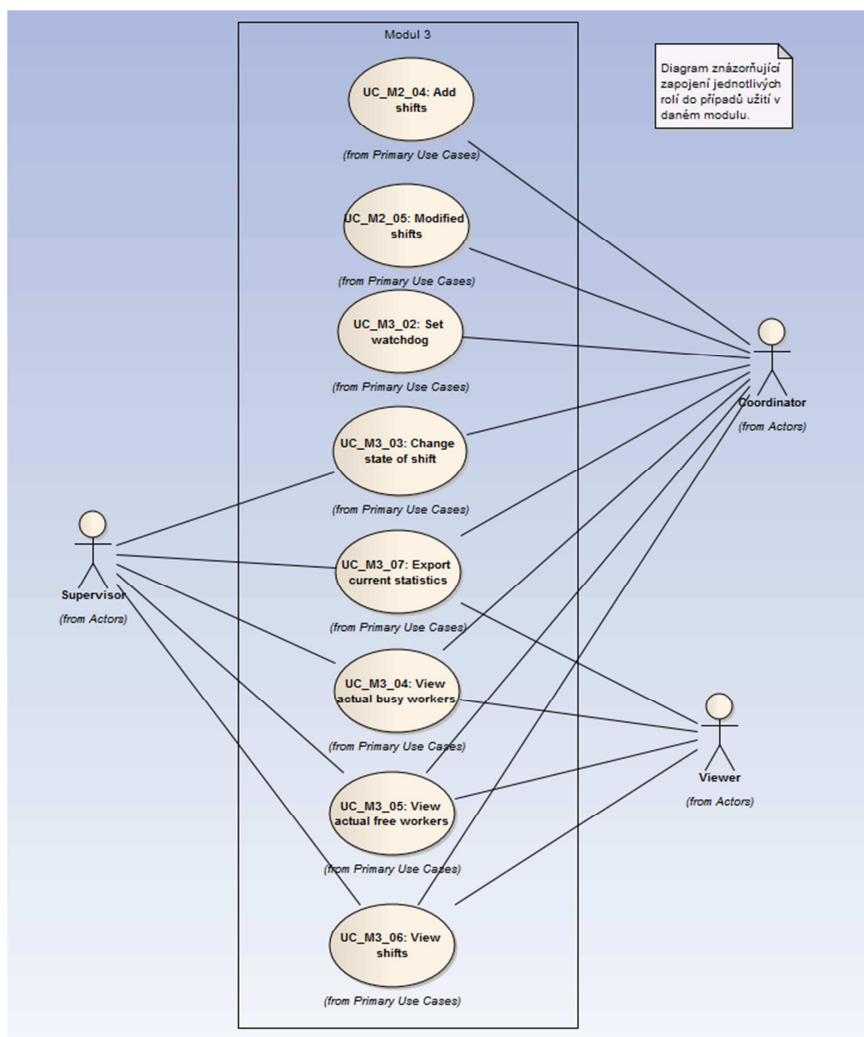


Graf 2 – Stavy směny

Každá směna může nabývat několika stavů, které se mění v postupném procesu. Žádný ze stavů by neměl být přeskočen. Směr procesu se může vrátit, vždy ale postupně krok po kroku. Jednotlivé kroky není možné přeskakovat.

Proces začíná vytvořením směny, která je zanesena do systému. Její naplánování zahrnuje rozvržení jejích časových úseků. Poté je přiřazena jednotlivému pracovníkovi. V dalším kroku je pracovník o své směně informován a ví, v kolik hodin má začít pracovat (podle toho ví, kolik má času na odpočinek). Před začátkem směny se pracovník nahlásí, že nastupuje na svůj pracovní výkon. Pokud je směna ve stavu „nahlášená“, může v daný čas přejít do stavu „ve výkonu“, kdy symbolizuje, že na dané pozici v danou dobu opravdu fyzicky stojí pracovník. Po skončení směny se pracovník přijde odhlásit. Toto zabraňuje „zapomenutí“ některých pracovníků na pracovních pozicích, kdy systém dokáže rozpoznat směnu, která již časově skončila, ale stále není odhlášená. Poté následuje s pracovníkem odsouhlasení jeho hodin, kdy si pracovník s velitelem potvrdí odpracovaný čas. Pokud je vše v pořádku, je možné směnu uzavřít.

5.3.3 Uživatelské role v Modulu 3



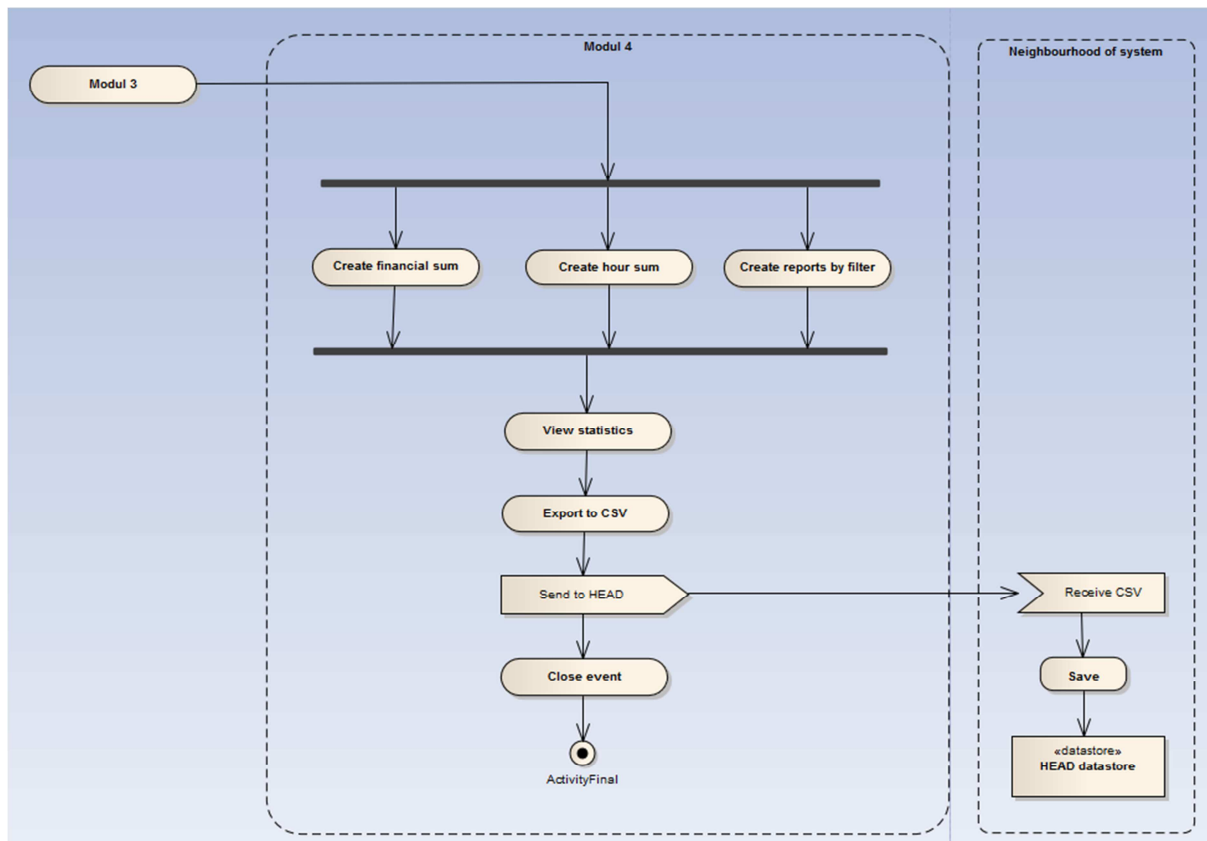
Obrázek 7 – Use case Modulu 3

V třetím modulu aplikace se objevují role známé již z předchozích modulů. Převážnou část práce s modulem provádí Coordinator a Supervisor, kteří zde působí v roli operátora systému. Coordinator má více práv, neboť může pracovní směny vytvářet a přesouvat (měnit jejich časové začátky a konce), ale Supervisor se stará pouze o převádění stavů směn – tedy interakci mezi pracovníky a systémem. Supervisor si také může vytvořit jakékoli statistiky akce v jakékoli úrovni podrobnosti, což mu pomáhá vybírat vhodné pracovníky pro přiřazení do naplánovaných směn. Role pozorovatele (Viewer) je v systému zejména pro kontrolování aktuálního stavu modelu – tedy zobrazení reálného stavu akce. Tato role může být použita pro nezaujatého pozorovatele – experta nebo také zákazníka, klienta – tedy produkci nebo vedení akce.

5.4 Modul 4

Čtvrtý a poslední modul aplikace slouží zejména k závěrečné reflexi a vyúčtování akce. Práce s tímto modulem spadá do post-projektové práce, kdy se využívají informace a data vložená do modelu v předchozích fázích a vytvářejí se závěrečné reporty.

5.4.1 Business Flow Modulu 4

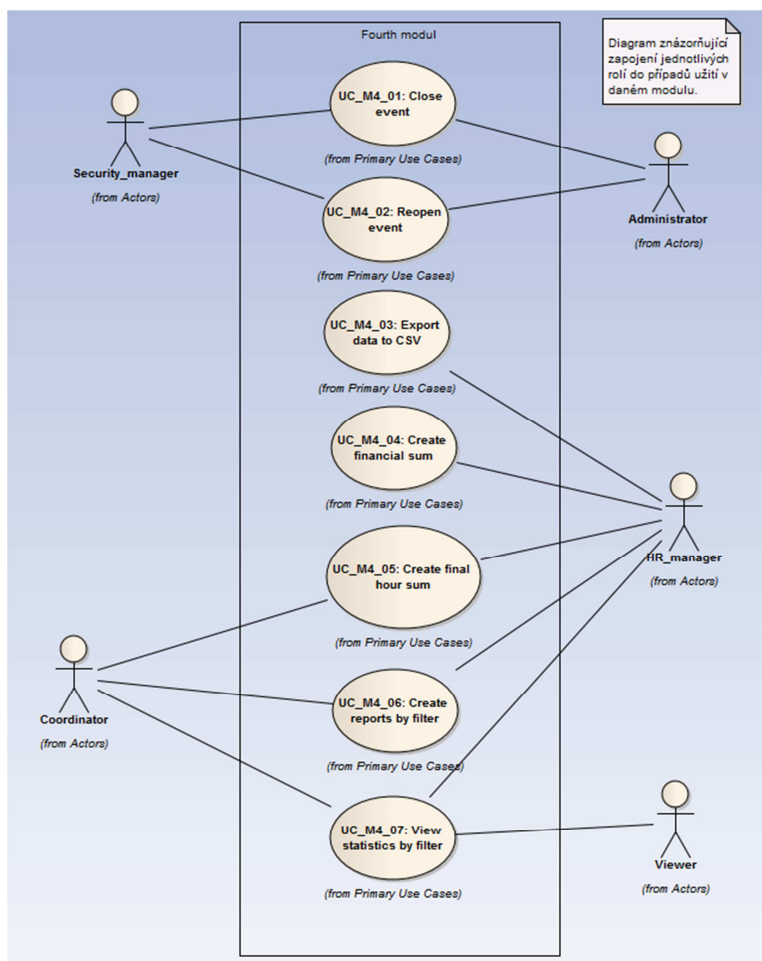


Obrázek 8 – Standardní flow práce s modulem

Po předání řízení z třetího modulu, které nastává po fyzickém ukončení akce, se mohou začít vytvářet finální reporty. Zejména se jedná o spočítání celkových pracovních hodin zaměstnanců, celkové sumy k vyplacení zaměstnanců i pro jednotlivé osoby, vytvoření a případné zobrazení statistik s detekcí nesrovnalostí nebo chyb (neukončené směny, neobvyklé výkyvy v pracovních výkonech atd.). Po vykonání všech těchto kontrolních kroků je možné vyexportovat data o jednotlivých zaměstnancích do přenositelného formátu (CSV), který potom může používat systém HEAD. Předjde se tak ručnímu přenášení dat, které je jednak časově i ekonomicky náročné (trvá dlouhou dobu, kterou musí pracovníkovi někdo zaplatit) a jednak je zde vysoké riziko potenciální chyby. Touto automatizací se částečně předchází chybě lidského faktoru.

Po vyexportování všech dat je možné model akce oficiálně i hodnotově uzavřít. Uzavření nebo zpětné otevření je vratná událost, kterou může provádět pouze Security Manager nebo Administrator. Po uzavření akce nelze již jednotlivé data v modelu měnit.

5.4.2 Uživatelské role v Modulu 4



Obrázek 9 – Use case diagram Modulu 4

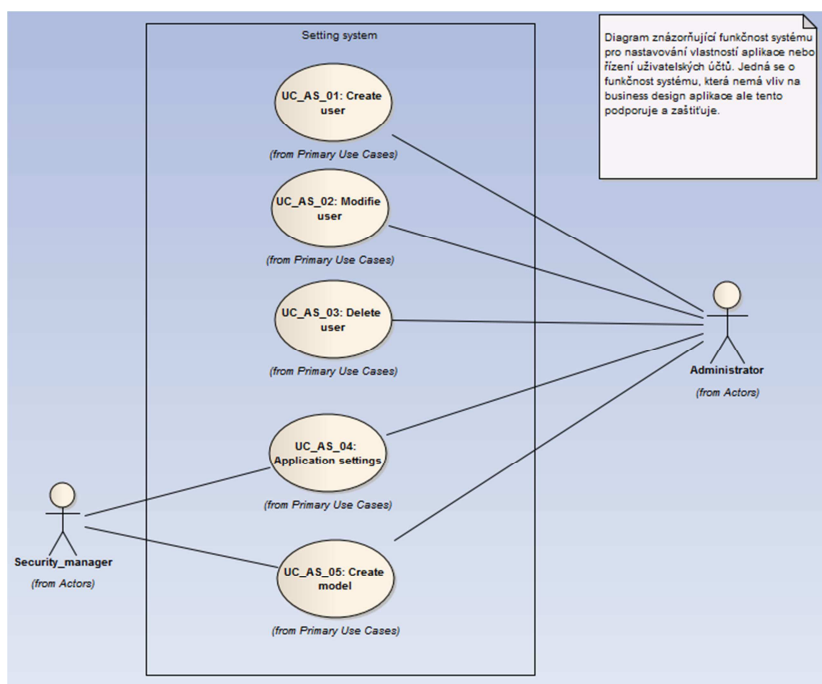
Ve čtvrtém modulu se opět objevují výše popsané role. Coordinator má možnost vytvoření a zobrazení statistik včetně vytváření jeho vlastních filtrů. HR manažer obsluhuje migraci dat do exportních souborů CSV a řídí integraci vytvořených dat ve stávajícím systému HEAD a stará se o finanční reporty.

Pro konečné uzavření akce je potřeba role Bezpečnostního manažera, který tyto úkony po jeho závěrečné kontrole standardně provádí. Bezpečnostní manažer by měl dohlédnout na hladký průběh celé akce a na provedení všech nezbytných úkonů. Potom může model akce uzavřít. Uzavření je vratná operace, která signalizuje, že model je ve finálním stavu a nepotřebuje již žádné změny. V uzavřeném modelu se dají zobrazit data, ale nedá se již s daty jakkoli manipulovat. Operaci uzavření modelu může provést také administrátor systému, jemu však z business logiky tato operace nepřísluší.

Role pozorovatele má i v tomto modelu stejné úkoly a možnosti jako v předchozích modulech. Nemá možnost nic vytvářet, ale může pozorovat dění uvnitř modelu. Tyto práva jí zůstávají i po uzavření modelu.

5.5 Administrační práce se systémem

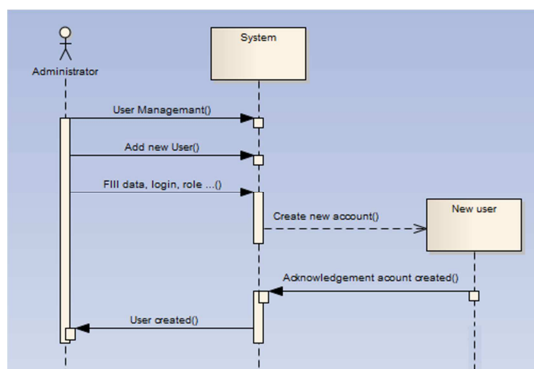
System je třeba také administrovat po architektonické stránce. K tomu slouží role administrátora, který nemusí mít hluboké business znalosti problematiky zabezpečování akcí, ale dohlíží nad fungováním systému. Typická práce administrátora je správa uživatelů a jejich rolí, dohled nad správnou funkcí systému, kontrola systémových zdrojů a nastavení.



Obrázek 10 – Administrace systému

Roli administrátora by měl zastávat technik znalý problematiky informačních technologií. Administraci je možné outsourcovat z vnější firmy. Vzhledem k tomu, že role administrátora má přístup ke všem datům a ovládá celý systém, je nutné pamatovat na bezpečnostní opatření. Citlivá data je vhodné ochránit úmluvou mlčenlivosti nebo podobnými způsoby.

Typický příklad práce administrátora je vytvoření účtu nového uživatele – viz obr. 11.



Obrázek 11 – Sekv.Diagram, vytvoření účtu

6 Tvorba scénářů

Ve scénářích případů užití je nutné zachytit očekávané a navrhované události. Je to podrobný postup, jakým je dosaženo hlavního funkčního záměru případu užití. Scénář popisuje interakci uživatele se systémem. Vždy by se tedy měl střídat vykonavatel, v zásadě „Systém“ a „Uživatel“. Ze základu rozlišujeme tři typy toku událostí.

Základní je ten, který popisuje „happy day“ scénář. Je maximálně stručný, ale dostatečně výstižný. Musí obsahovat kompletní postup při realizaci případu užití. V základním toku se neobjevuje větvení ani podrobné informace, které by nebyly nezbytně nutné. V základním scénáři neřešíme možné alternativy, které mohou nastat, ale definujeme jasný a přímý postup pro dosažení cíle.

Alternativní tok událostí je možná alternativa nebo větvení základního scénáře. Zde se dají zachytit výjimky nebo možnosti pro rozdílné chování. Může se jednat například o chybové výjimky, například nekorektní data. Nejedná se však o popisování chyb aplikace, ale business chyb. Klasický příklad alternativního scénáře je chybný login do aplikace. Zde pak tento alternativní scénář popisuje chování systému po chybném pokusu o přihlášení.

Třetím typem je rozšiřující tok událostí. Ten obsahuje všechny relevantní podrobnosti k bodům postupu v hlavním toku. Může se zaznamenat jako doplňující kroky s odkazem na základní tok. Tvorba scénářů pomáhá objasnit zejména business funkci budoucí aplikace, a dokáže odhalit možné mezery a slabé stránky stejně jako například limity použitých technologií.

Z těchto důvodů byl pro popsání požadované funkce systému ke každému případu užití vytvořen základní scénář, a pokud bylo nutné tak i alternativní kroky. Každý krok je definován jedinečným identifikátorem, číslem, které zároveň označuje pořadí kroku z celého scénáře. Nástroj Enterprise Architekt zároveň umožňuje označení, zda se jedná o základní tok nebo alternativu.

| Name | Type |
|--|------------|
| 1. Uživatel zvolí model | Basic Path |
| 2. Systém načte data modelu | Basic Path |
| 3. Uživatel si zobrazí plánec modelu | Basic Path |
| 4. Systém nabídne uživateli nástroje k editaci plánu | Basic Path |
| 5. Uživatel provede změny v bitmapě modelu | Basic Path |
| 6. Systém zobrazuje změněnou bitmapu. | Basic Path |
| 7. Uživatel uloží bitmapu | Basic Path |

Obrázek 12 – Scénář k UC editace obrázku pozadí

Na obrázku 12 je vidět scénář k základnímu toku pro případ užití editace podkladového obrázku. Je zřejmé střídání interakce uživatele se systémem. Scénář tedy splňuje všechny náležitosti a spadá do kategorie „happy day“ scénářů, tedy pozitivního toku.

| Name | Type |
|---|------------|
| 2. Systém zobrazí seznam existujících uživatelů | Basic Path |
| 3. Admin vybere uživatele, kterého chce změnit | Basic Path |
| 4. Systém zobrazí stávající údaje uživatele | Basic Path |
| 5. Admin změní požadované údaje a uloží změny | Basic Path |
| 6. Systém uloží změny | Basic Path |
| 7A. Systém neuloží změny a ponechá stávající údaje. | Alternate |
| 6A. Admin zruší změny | Alternate |

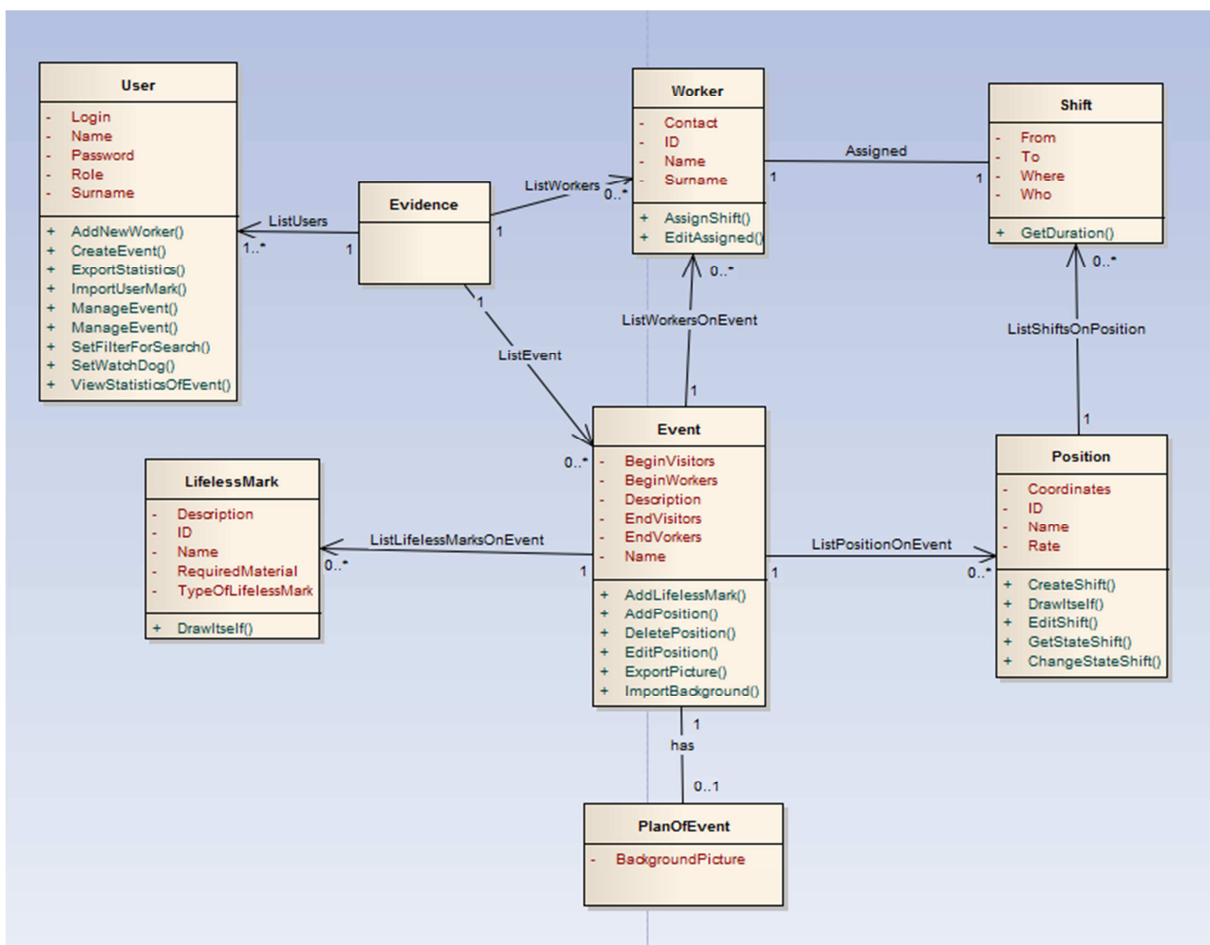
Obrázek 13 – Ukázka alternativního toku UC změna uživatele

Na obrázku 13 je vidět scénář s alternativním tokem. Alternativní kroky jsou označeny prefixem „A“ (Alternate) za identifikátorem kroku. Tento identifikátor početně navazuje na krok, ze kterého bylo provedeno větvení. Pokud by bylo alternativních cest více, bylo by nutné je odlišit, např. XA2 (X-číslo).

7 Analytické třídy

Nalezení analytických tříd je první krok, který později vede k návrhu implementace. Definují základní logické celky aplikace a jejich vztahy. Později jsou základem pro identifikaci návrhových tříd, které již detailně definují strukturu objektů programu se všemi detaily. Diagram analytických tříd definuje logické celky a jejich vztahy. Není třeba uvádět implementační detaily. Z velké části se skládá z objektů doménových tříd. Analytický model musí věrně odrážet entity reálného světa, kdy modeluje reálný svět do objektového přístupu. Doporučených metodik k nalezení analytických tříd je několik, neexistuje však žádná konkrétně určená a jediná správná technologie. Mezi nejznámější metody patří metoda CRC štítků (CRC = Class, Responsibilities & collaboration). Z dalších například metoda analýzy scénářů, typových úloh, metoda hledání sloves atp.

Rozdíly mezi konkrétní interpretací návrhové třídy a analytické třídy jsou například nepodstatnost datových typů u analytických tříd nebo jejich inicializací. Také není nezbytně nutné rozlišovat přístupovou politiku nebo konkrétní definice závislé na programovacích jazycích. Analytické třídy by měly být namapovány na pojmy používané v reálném světě, na problémové domény. Z názvu tříd by měl být jasný jejich účel. Analytické třídy jsou vnitřně soudržné, mají jasně vymezenou množinu dat.



Obrázek 14 – Analytické třídy

Na obrázku 14 je vidět analytický model tříd analyzované aplikace. Dle jedné z definic nalezení analytických tříd bylo vymezeno několik entit reálného světa, které vytvořily základ pro analytické třídy. Jsou to zejména entity uživatele, pracovníka, události (plánované akce), značek materiálu („neživých“ značek), pozic pro pracovníky a směn. Tyto třídy jsou mezi sebou v interakci, navzájem využívají své funkce nebo ukládají svá data. Množina metod pro třídy byla definovaná s pomocí definice funkčních požadavků, ve které byla hledána klíčová slova pro definici operací. Množina atributů byla vyhledávána v datech nutných pro ukládání.

8 Příprava implementace

Pro implementování aplikace byl zvolen jazyk C#. Tento jazyk je vysokoúrovňový objektově orientovaný a vyvíjený firmou Microsoft. Historicky je C# založen na jazycích Java a C++, hlouběji lze říci, že základ čerpá z jazyku C. Z něj také získal syntaxi. Jazyk C# by měl ideově spojovat výhody všech těchto jazyků dohromady a měl by se poučit z chyb, které se za dlouhou dobu používání těchto jazyků projeví. Microsoft jazyk C# stále vyvíjí a v současnosti je nejnovější verzí 5.0 vydaná v roce 2012 spolu s programem Visual Studio 2012, které je vývojové prostředí od firmy Microsoft pro tento jazyk.

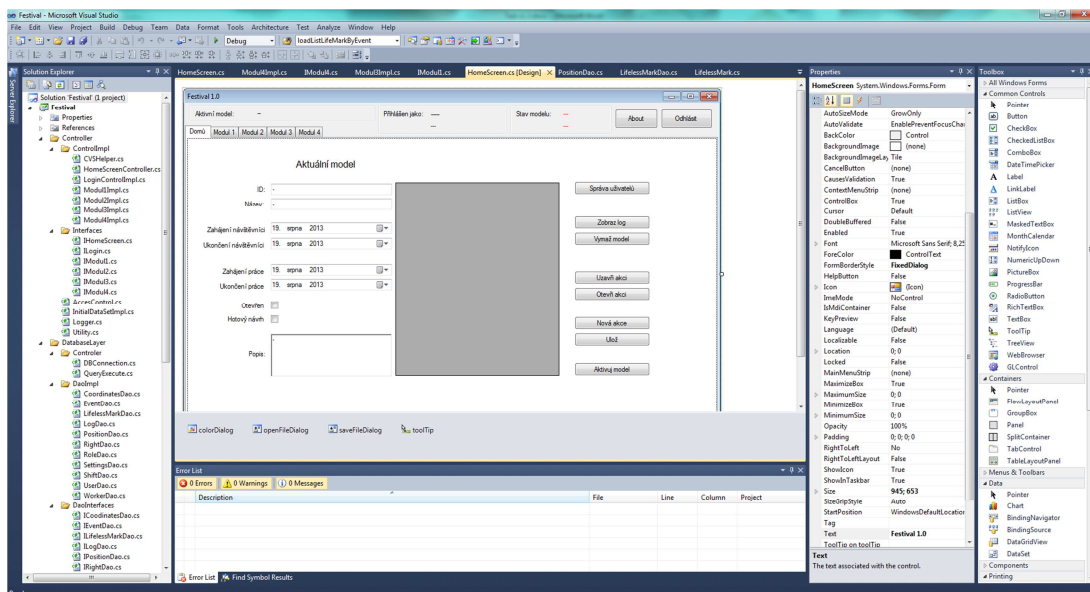
8.1 Vývojové prostředí

Microsoft Visual studio je nejrozšířenější vývojové prostředí (IDE) pro jazyk C#. Je vyvíjeno firmou Microsoft a v současnosti je nejnovější verzí Microsoft Visual Studio 2012. To je spjato s verzí jazyka 5.0. Vývojové prostředí nabízí použití jak pro konzolové aplikace, tak i pro aplikace s grafickým rozhraním. Nabízí technologie Windows Forms, které se dají použít jak pro desktopové aplikace tak i webové aplikace. Technologie zaručuje použití na všech platformách firmy Microsoft, tedy například řadě OS Windows, Windows Mobile, NET Framework a další.

Expresní verze programu je volně k dispozici, vyšší edice jsou však zpoplatněny. Díky programu MSDN Academy mají studenti některých vysokých škol (včetně Univerzity Pardubice) možnost získání legální studijní licence zdarma. Díky této licenci bylo možné uskutečnit vývoj aplikace v tomto vývojovém prostředí v edici Ultimate.

Microsoft Visual Studio obsahuje editor kódu podporující Inteli Sense, automatické refaktorování napsaného kódu, zvýrazňování textu, syntaxe a mnoho dalšího. Integrovaný debugger dokáže pracovat jak na úrovni kódu, tak i stroje. Integrované designery formulářů a standardních Windows objektů značně zjednodušují práci na grafickém rozhraní aplikace. Microsoft Visual Studio navíc nabízí mnoho možností rozšíření podle požadované funkcionality a je tak velmi silným vývojovým nástrojem.

Microsoft Visual Studio podporuje více programovacích jazyků, z těch neznámějších kromě jazyka C# také jazyk C++, VB.NET, Ruby nebo F#. Také podporuje technologie jako XML/XSLT, což je „značkovací“ jazyk, standardizován a schválen konsorciem W3C. Slouží zejména pro přenos či ukládání dat. XSLT technologie slouží pro transformaci dat z/do formátu XML. Z dalších neznámějších podporovaných technologií jsou to například webové technologie HTML/XHTML, CSS pro stylování webových stránek nebo Javascript, což je multiplatformní skriptovací jazyk nejčastěji používaný přímým vkládáním do kódu webových stránek na ovládání různých interaktivních GUI prvků stránek. Jeho použití je však široké.



Obrázek 15 – Screen vývojového prostředí Microsoft Visual Studio

Na obrázku 15 je vidět screen Microsoft Visual Studia, ve vrchní části jsou panely ovládání aplikace a nejčastěji používané příkazy, v levé části „Solution explorer“ stromově zobrazovaná struktura jednotlivých položek otevřeného projektu, nastavení, reference, knihovny, třídy, adresáře nebo formuláře GUI rozhraní. Ve spodní části je aktuálně zobrazen error list, tedy soupis chyb a varování, které IDE dokáže samo vyhledat a identifikovat. V prostřední části je aktuálně zobrazen návrh vytvářeného formuláře, zde se zobrazuje také kód jednotlivých tříd. Vpravo je pak rozpis atributů vybrané položky nebo aktivační události. Zcela vpravo je pak nabídka připravených vytvořených objektů pro GUI rozhraní. Nutno podotknout, že rozmístění prvků je možné kdykoli změnit a pro snadnou orientaci konkrétního uživatele jakkoli přizpůsobit.

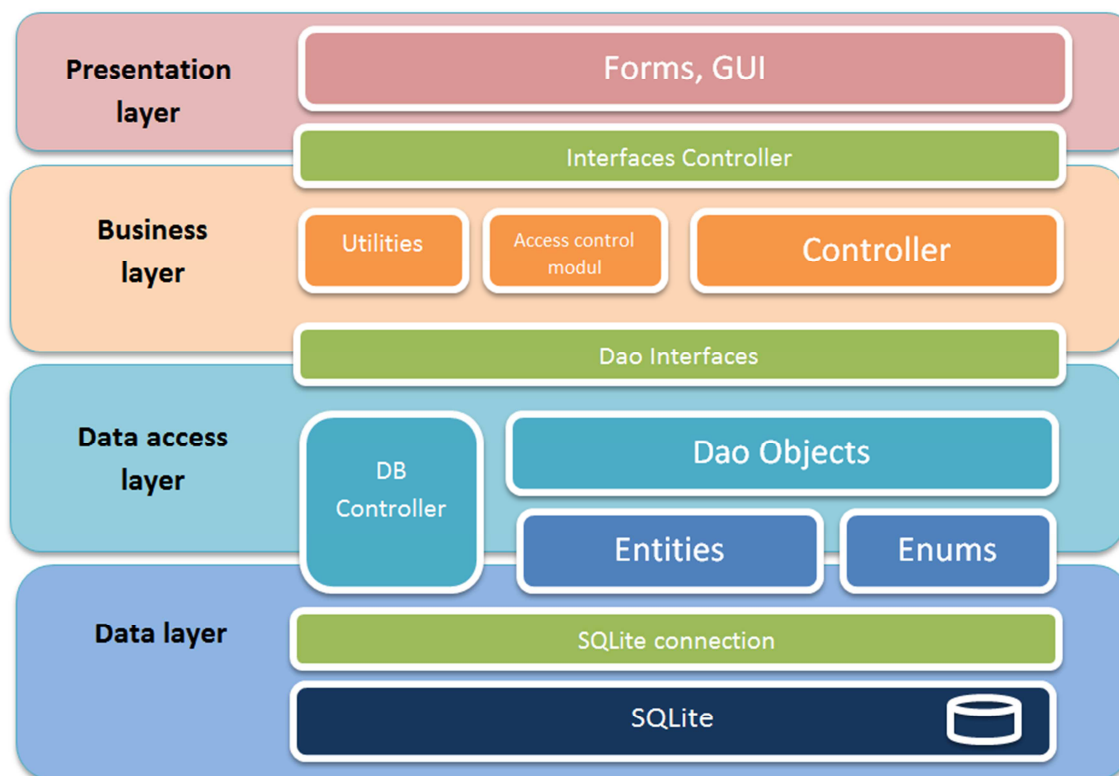
8.2 Rozvrstvení aplikace

Moderní aplikace by měla respektovat vrstevné rozdělení. Každá vrstva by měla být odpovědná za specifickou sadu úkonů, kterou bude vykonávat. Striktní dodržování vrstev je prosazováno zejména u serverových aplikací, kde existuje mnoho abstraktních vzorů, jak aplikace vrstvit a jaké modely používat.

Podobnou logiku lze použít i u desktopové aplikace. Pokud správně navrhne a rozložíme potřebné funkce aplikace do vrstev, bude výsledek přehledný a nebude problém v něm později provádět změny. Pokud dodržíme standardy napojování vrstev na sebe, není také problém vyměnit celé vrstvy nebo jiné mezi ně vkládat. O zabezpečení standardů propojení vrstev se starají rozhraní (interface), kde jsou definovány funkce, které musí rozhraní bezpodmínečně obsahovat. Implementace jednotlivých rozhraní jsou pak umístěny hlouběji v implementační logice vrstvy a při výměně vrstvy (pořažmo celé technologie) nás nemusí konkrétní implementace zajímat. Stačí se pouze napojit na definované rozhraní a máme zajištěno, že aplikace bude funkční. Této výhody se využívá

například při přechodu na jinou databázovou technologii nebo jiný engine. Není potom problém vyměnit použitou SQLite za například MySQL nebo kteroukoli jinou technologii.

Podobný systém funguje i na opačné straně aplikace, na GUI rozhraní. Zde je možné vyměnit například stávající formuláře za nové s novými grafickými prvky nebo jinak definované. Novou technologii pak lze napojit přes rozhraní controlleru na implementaci business vrstvy a systém je opět plně funkční s novým grafickým rozhraním.



Graf 3 – Rozvrstvení aplikace

Na grafu 3 je vidět logické rozvržení aplikace. Ve spodní části se nachází datová vrstva. V té hraje nejdůležitější roli SQLite databáze se svou logikou přístupu, která je definovaná v knihovnách aplikace. Programátor tedy pouze využívá získané instance SQLite connection. Opět, pokud by byl požadavek na změnu, lze zaměnit za knihovny jiného engine a využívat jinou technologii.

Do datové vrstvy částečně zasahují entity a výčtové typy, které obalují vlastní data ukládané v DB. Modul databázového controlleru zajišťuje ukládání entit a enum typů do databáze prostřednictvím connection.

Tento DB controller i s entitami a výčtovými typy patří ale spíše do vrstvy přístupu k datům. Zde hrají nejdůležitější roli DAO (Data Access Object) objekty, které uspořádávají data do správných formátů a zajišťují logiku pro ukládání pomocí DB controlleru.

Vrstva přístupu k datům je s business vrstvou propojena DAO rozhraním. V business vrstvě je ukryta funkční logika aplikace. Zde probíhají výpočty a operace zajišťující správnou business funkčnost. Jsou zde také utility funkce nebo řízení přístupu práv.

Business vrstva je s prezentační propojena opět přes rozhraní. Prezentační vrstva by se měla starat čistě o prezentaci dat. Zde často dochází k chybnému umístění logiky, neboť mnoho technologií včetně .NET a C# nabízí pohodlnou možnost vytváření logiky přímo do kódu formuláře. Tento způsob se může zdát pohodlnější, ale pro pozdější orientaci v programu nebo úpravy tyto chyby způsobují nepřekonatelné problémy. Prezentační vrstva by si od business vrstvy měla pouze předat data, která zobrazí. Měla by je již minimálně upravovat, ideálně vůbec. Úprava dat je správná pouze v případě, že specifická komponenta formuláře vyžaduje specifický formát dat. Právě tyto matcher metody je možné psát přímo do prezentační vrstvy. Rozhraní by vždy měly obsahovat co nejobecnější datové typy, aby bylo případné napojování či výměna co nejjednodušší.

Moderní metody architektury softwaru dokážou jít ještě hlouběji. Kromě výměny celých technologií ukrytých ve vrstvách, resp. vrstev dokáží navrhnout systém tak, aby kterákoli komponenta byla soběstačná a možná výměny. Každá komponenta by měla mít specifickou funkci, kterou bude vykonávat, za kterou bude zodpovědná a kterou bude vykonávat správně. Musí tedy předpokládat a ošetřit všechny varianty vstupů i možných výsledků. Klasickou chybou této koncepce je předpoklad vstupu správných dat, kdy při chybných vstupních datech (klasicky null pointer) nastává chyba a pád komponenty potažmo celé aplikace. Komponenta nemusí vědět nic o okolním prostředí, pouze čeká na zavolání případně dodaná data a provede svou funkčnost. Také se dá kdykoli vyměnit za novější, která ale musí splňovat stejné definiční požadavky.

Tento přístup je nezbytný například při týmovém programování, kdy si členové vývojového týmu můžou mezi sebou jednoduše rozdělit práci. Každý člen týmu má definici co má jeho komponenta umět a je pouze na něm, jak jí implementuje. Ostatní na něho spoléhají, že jeho komponenta bude fungovat správně a mohou jí používat.

Příkladem komponenty použité v implementaci práce je například třída CSVHelper, která nebyla vytvořena mnou, ale byla napsána pod otevřenou licenci s přístupným zdrojovým kódem a nabídnuta k použití. Jiným trochu odlišným příkladem může být použití návrhového vzoru, který je sice vždy upraven pro konkrétní program, ale jeho chování je také předem definované a je třeba jej splnit.

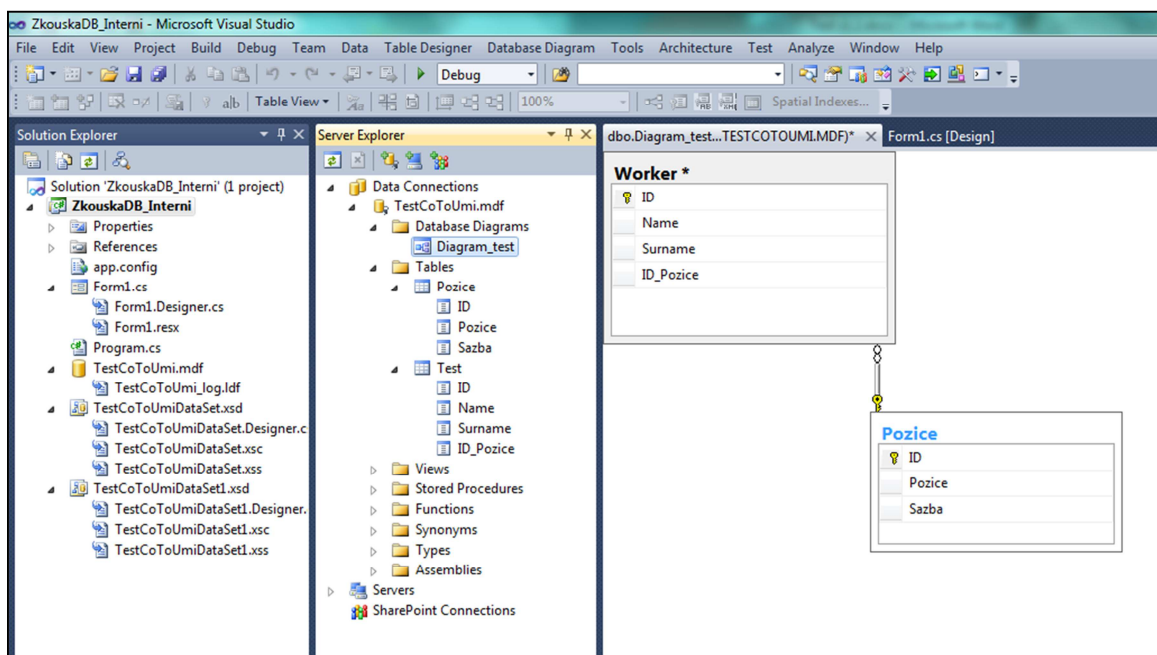
9 Databázová technologie

Vzhledem k požadavkům na bytelnost aplikace a desktopové použití bylo hledáno řešení s minimálními systémovými nároky, ale odpovídajícím výkonem. Nejpoužívanější řešení velkých databází založené na samostatném databázovém serveru, jako jsou Oracle, MySQL, PostgreSQL nebo jiné nejsou příliš vhodné právě z důvodu potřeby vlastního databázového serveru. Je pravda, že toto řešení nese spoustu výhod, spolu s nimi ale narůstá celková složitost řešení a systémové požadavky. Desktopová aplikace sama o sobě svůj vlastní server nepotřebuje, a tak bylo zvoleno embedded řešení formou přilinkované knihovny, což je opak řešení klient-server. Fyzický databázový soubor dat je ve formě samostatného souboru obsluhovaného SQL dotazy.

Další možností bylo zvolit objektovou databázi. To bylo ale zavrhnuto již v počátku, neboť vývoj dnešní doby se stále orientuje více na relační řešení, případně objektově relační. Čistě objektová databáze, i přesto, že vyhovuje požadavkům embedded knihovny – db4o, byla zavrhnuta.

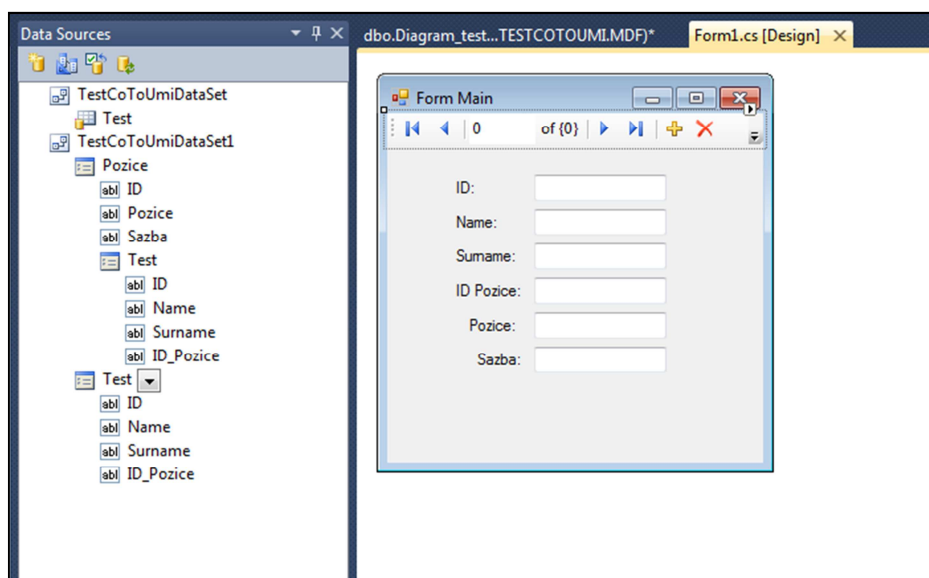
9.1 Databáze Visual Studio 2010

Pro použití se nejvíce hodily dvě technologie. První obsažená přímo ve Visual Studiu 2010, která umožňuje vytvářet jednoduché DB schémata, a ukládat do nich data. Druhá technologie je detailně vysvětlena v kapitole 9.2. Technologie Visual Studio je velice jednoduchá na použití, a databázi lze vytvořit v řádu minut. Nabízí i jednoduché GUI rozhraní pro ovládání, a formulářový (interaktivní) přístup k datům, pro zobrazení i vkládání dat.



Obrázek 16 – Interní DB Visual Studio 2010 – Diagram

Na obrázku 16 je vidět screen z testování funkčnosti tohoto řešení. Vlevo v okně „Solution Explorer“ je kromě běžného stromu projektu vidět i databázové napojení (název „TestCoToUmi.mdf“). V druhém panelu zleva „Server Explorer“ je vidět ovládací strom formulářového přístupu k ovládání databázových objektů. Toto podokno již připomíná podobné přístupy ze známých DB manažerů pro jiné DB. Lze si zde zobrazit diagram DB a přes GUI rozhraní jej ovládat a editovat. Stejně tak lze ovládat jednotlivé DB objekty jako tabulky, pohledy, funkce nebo procedury.



Obrázek 17 – Interní DB VS 2010 - Zdroje dat

Přidání dat do formuláře připomíná hibernate technologii známou z javy EE. Při zobrazení podokna zdroje dat (Data Sources) se zobrazí všechny tabulky (po správném nastavení zdroje dat) a při správně navrženém datovém modelu a dodání omezení cizího klíče se nám jednotlivé tabulky do sebe automaticky spojí. To je názorně vidět na obrázku 17, kdy máme dvě tabulky propojené cizím klíčem a v tabulce „Pozice“ se zobrazí napojená tabulka pracovníků.

Pro zobrazení dat ve formuláři lze v nejjednodušším případě použít drag & drop a přenést atributy na formulář. Automaticky se nám nabídnou data z DB a přenesou na formulář.

Toto řešení je vhodné pro velmi jednoduché DB, řádově v jednotkách tabulek. Je rychlé, jednoduché a intuitivní. Technologie by pravděpodobně obstála i při větším množství dat, ovládání by se ale stalo dost nepřehledné. Navíc samotné umístění přímo ve vývojovém prostředí Visual Studia značně ztěžuje, někdy až znemožňuje ovládání přes DB manažery třetích stran. Vždy jste tak odkázáni pouze na nástroje, které vám dovolí Visual Studio. Pro větší množství dat a tabulek je tedy toto řešení nevhodné. V případě vyvíjené aplikace hledáme řešení pro cca 20 tabulek. Tato technologie by jí tedy pravděpodobně zvládla, přesto bylo rozhodnuto využít jiné řešení.

9.2 SQLite

Relační databázový systém [7] (SQLite) je obsažený v malé knihovně. Knihovna je napsaná v jazyce C a je distribuována pod licencí Public Domain. Tento systém splňuje požadavky, nepotřebuje samostatný server a řešení je jednoduché a stabilní. Je nutné nakopírovat do složky projektu tři .dll soubory a přilinkovat je jako další zdroje projektu. Pak se již dá s DB pracovat podobně jako se serverovými řešeními. Tedy lze vytvářet připojení, načítat a ukládat data. Lze využít příkazů jazyka SQL. Veškerá data jsou uložena v souboru .db který je umístěn přímo ve složce projektu. Tento soubor je přenositelný a lze jej velmi snadno zálohovat nebo přesouvat.

9.2.1 Nevýhody SQLite

Omezení SQLite vycházejí z určení jejího použití. Databáze neobsahuje složitější DB mechanismy, například systém práv na objekty (vůbec nezná GRANT a REVOKE). Konfigurace DB je velmi skromná a celá se řeší přes direktivu PRAGMA, případně atributy připojení. Umístěním celého obsahu do jednoho souboru přináší také nevýhodu fragmentace, při dnešních velikostech a objemech dat, s přihlédnutím k zacílení technologie je ale tato nevýhoda téměř bezvýznamná. Z dalších nejsou implementovány některé operace změny tabulek (ALTER TABLE), některé přidávání omezení (ADD CONSTRAINT) nebo složitější propojení tabulek, např. OUTER RIGHT JOIN. Nutno ale dodat, že například chybějící operaci ALTER TABLE lze obejít složením více operací. Například vytvořením nové dočasné tabulky, zkopírováním obsahu, smazáním a opětovným vytvořením staré tabulky s již požadovanou změnou (například přidáním omezení), zpětným nakopírováním dat a smazáním dočasné tabulky. Podobným skládáním základních operací lze mnoho chybějících příkazů obejít. Obecně lze ale uvést, že SQLite implementuje téměř celý standard SQL-92.

9.2.2 Výhody SQLite

DB SQLite lze použít nezávisle na operačním systému a platformě (Linux, MAC OS, Windows, Android) i v různých programovacích jazycích (C, C++, C#, Java, Delphi, PHP, Perl, Ruby atd.). Samotný soubor s uloženými daty je také mezi těmito platformami a systémy přenosný.

Další výhodou zejména pro vývojáře je fakt, že existuje více možností správy tohoto systému a mnoho variant použití. Je to způsobeno zejména použitím volné licence. Proto lze vybírat z několika SQL manažerů a variant knihovních souborů.

Pro potřeby aplikace bylo zvoleno řešení skupiny vývojářů pod názvem Finisar.SQLite (<http://adodotnetsqlite.sourceforge.net/>). Tento projekt sice ukončil vývoj v roce 2006, knihovna je však stále týmem vyvíjena pod záštitou nového projektu a binární soubory (dll) mají podporu a opravu chyb (podle dostupných pramenů) dodnes. Poslední aktualizace byla nalezena s datem roku 22. 5. 2013.

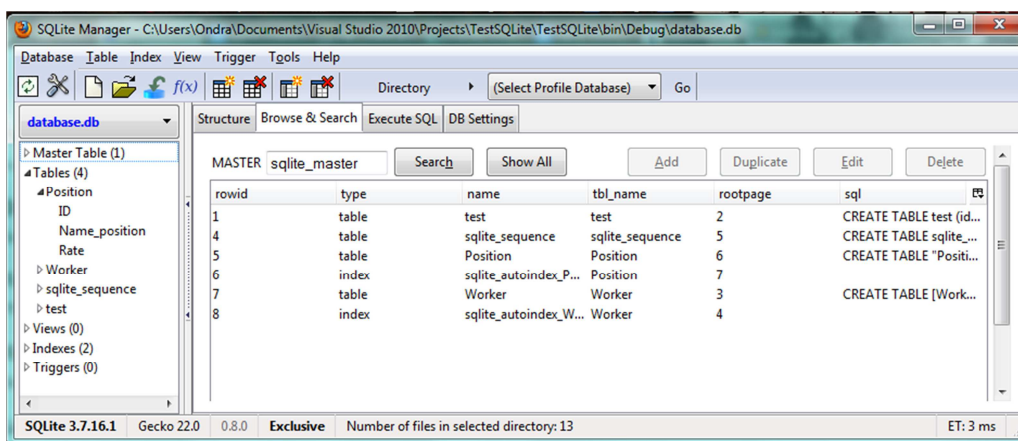
| | | | |
|----------------|----------------|--------------------|--------|
| database.db | 3.8.2013 15:39 | Data Base File | 11 kB |
| SQLite.dll | 12.2.2005 1:06 | Rozšíření aplikace | 252 kB |
| SQLite.NET.dll | 12.2.2005 1:06 | Rozšíření aplikace | 64 kB |
| SQLite3.dll | 12.2.2005 1:06 | Rozšíření aplikace | 264 kB |

Obrázek 18 – Databázový soubor a dll knihovny

Na obrázku 18 je vidět zmiňované tři .dll soubory a soubor database.db, ve kterém jsou obsažena veškerá data. Po přilinkování do projektu přes kontextové menu „Add reference“ již můžeme otevírat databázová připojení.

9.2.3 DB Managery SQLite

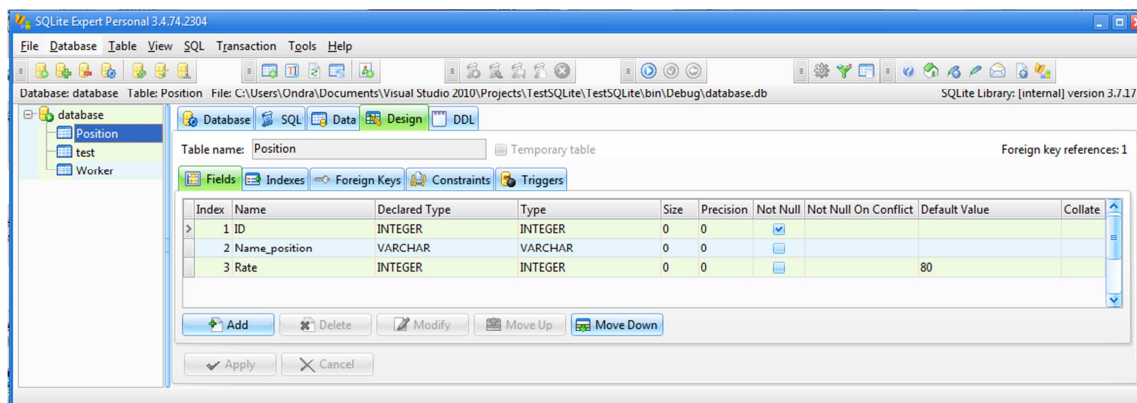
Pro vývojové účely byly testovány různé manažery. Většina jich je volně dostupná. Za zmínku stojí rozšíření do prohlížeče Firefox – SQLite Manager (<https://addons.mozilla.org/cs/firefox/addon/sqlite-manager/>). Tato utilita spouštěná ve Firefoxu je plnohodnotný manažer, který lze spustit jak v tabu prohlížeče, tak jako nové okno. Stačí pouze vybrat databázový soubor a začít pracovat. GUI utility je vidět na obrázku 19.



Obrázek 19 – SQLite Manager 8.0

Dále jsou k dispozici další programy, například SQLite Database Browser (<http://sqlitebrowser.sourceforge.net>) vyvíjený zmiňovanou vývojářskou skupinou, od které jsou použité dll knihovny, nebo SQLite admin (<http://sqliteadmin.orbmu2k.de/>), jednoduchý program, který je dokonce dostupný s českou lokalizací.

Nabízí se i komerční řešení, například SQLite Exper (<http://www.sqliteexpert.com>), které je dostupné buď v 30-ti denní testovací verzi, potom je nutné produkt zakoupit. Firma ale nabízí i řešení pro nekomerční použití SQLite Expert Personal, které postrádá některou funkcionalitu, přesto je ale velmi silným nástrojem. Pro širokou škálu podporovaných operací a nejjednodušší provedení operací ALTER TABLE, což jak již bylo zmíněno, jsou drobné slabiny SQLite, jsem jej vybral spolu s utilitou prohlížeče Firefox jako nejvhodnější program k managování DB.



Obrázek 20 – SQLite Exper Personal GUI

Ovládání je intuitivní a velmi připomíná známé DB manažery větších serverových databází jako například Oracle SQL Developer nebo MySQL Workbench nebo jiné.

10 Datový model

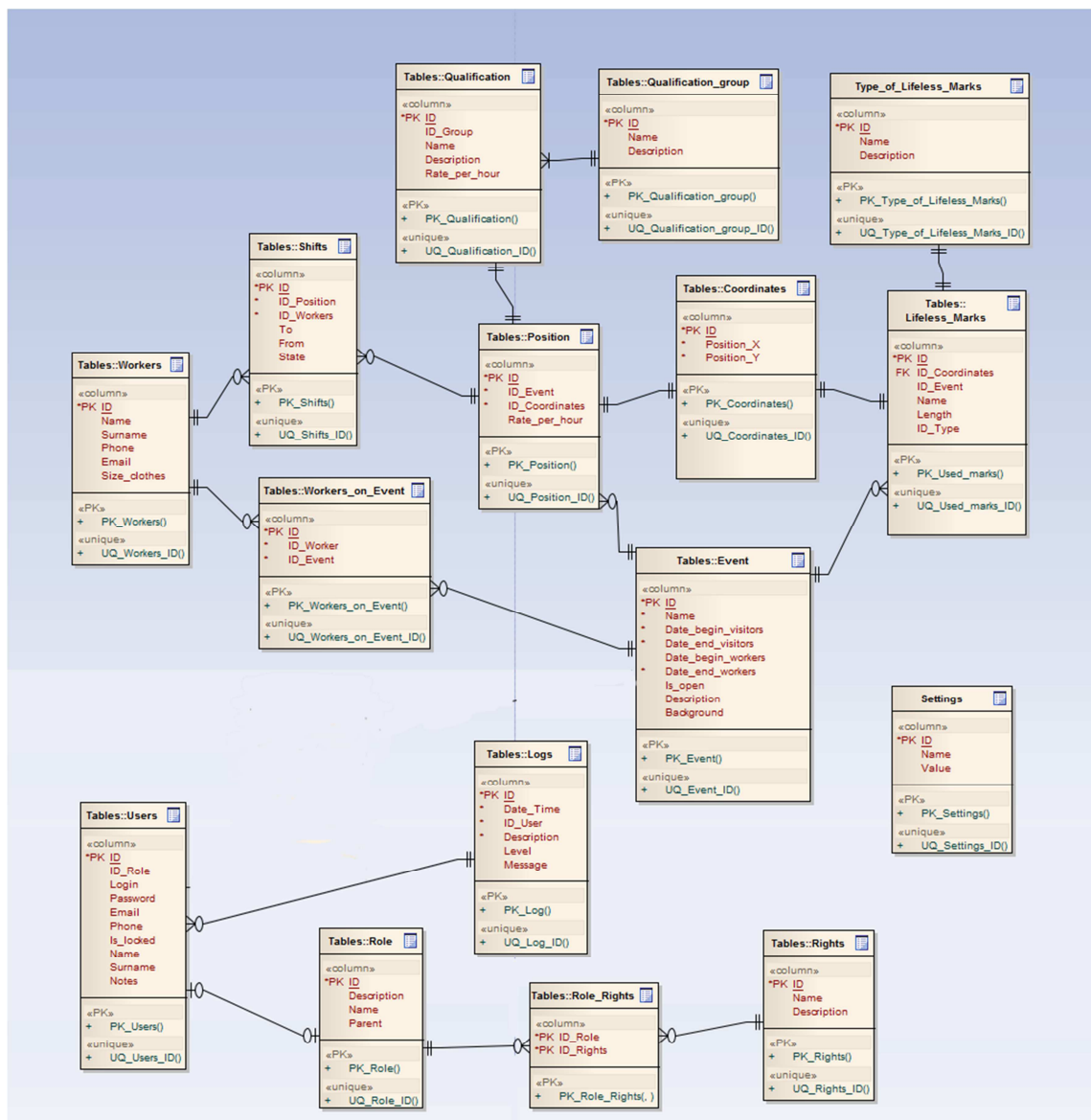
Datový model neboli model databáze je stavebním prvkem pro ilustraci logiky ukládání dat v případě, že aplikace nespolehá na ukládání do běžných souborů na souborovém systému. Pokud používáme relační databázi (případně objektově-relační) je pro zobrazení modelu téměř výhradně používán relační diagram se zobrazením vazeb mezi jednotlivými tabulkami.

Relační model tedy popisuje uložení dat v databázi a při vymodelování ve vhodných nástrojích nabízí i generování zakládacích DDL skriptů (Data Definition Language). Měl by tedy obsahovat i datové typy a i přesto, že by se měl řídit standardem SQL, je vhodné uvádět typ zvolené DB, protože může docházet k rozdílům v generovaných skriptech. Na vizuální zobrazení ale typ DB implementace vliv nemá.

Relační model by měl být minimálně ve třetí normální formě, měl by mít definovány primární a cizí klíče, kardinalitu a parcialitu a vyřešené M:N vazby. Některé tyto podmínky lze vynechat, model pak ale ztrácí na vypovídací hodnotě.

Relační model databáze sestává ze sedmnácti tabulek vzájemně propojených vztahy. Do tohoto datového modelu, zobrazeného na obrázku 21 se ukládají veškerá data sloužící k sestavení modelu – zobrazení zpracovávané akce. Databáze je společná pro všechny modely, tedy akce zpracovávané v aplikaci. Ve většině tabulek je primární klíč ID každého záznamu. Jedinou nepropojenou tabulkou je tabulka „Settings“, která ukládá nastavení aplikace. Odděleným schématem jsou tabulky sloužící k ukládání dat uživatelů. Pro zjednodušení jsou zobrazeny ve společném ER diagramu.

Z důvodu přehlednosti zobrazeného modelu je zde vynecháno automatické zobrazování cizího klíče, jelikož velkým popiskem narůstá velikost zobrazení. Atribut cizího klíče je v každé tabulce zobrazen ve formě „ID_název-navázané-tabulky“ kdy se cizí klíč váže vždy na ID spojené tabulky.



Obrázek 21 – ER Diagram datového modelu

10.1 Popis některých tabulek

Pro lepší pochopení je popsán obsah a funkce některých nejdůležitějších tabulek.

- **Event**
Ústřední tabulka pro uchovávání informací o modelu akce. Obsahuje základní a nejdůležitější informace o dané akci jako je název, termín konání, termín prací ale i metadata potřebná pro správnou funkci modelu, jako například příznak otevření modelu.
- **Position**
Seznam pozic v modelech. Zde jsou uchovávány jednotlivé pracovní pozice a informace o nich. Tabulka využívá cizích klíčů – napojení na Model, kterému pozice náleží a souřadnice umístění v modelu.
- **Coordinates**

Tabulka vzniklá pro normalizaci relačního modelu. Uchovává souřadnice (x, y), které jsou využívány pro umístění značek a pozic do daných modelů. Primárním klíčem musí být jednoznačný identifikátor, neboť dvojice hodnot x a y nemusí být jedinečná.

- **Qualification**

Tabulka kvalifikací je zobrazením rozdělení typů pozic. Záznamy v této tabulce jsou korespondující možné kvalifikace pracovníků s druhy pozic nabízených jednotlivou akcí. Seznam nabízených pozic na dané akci je obsažen v záznamech v tabulce „Position“ a naopak kvalifikace jednotlivých lidí je v tabulce „Qualification of workers“ která je zároveň pomocnou spojovací tabulkou.

- **Workers**

Tabulka pracovníků je druhá ústřední tabulka celého relačního modelu. Uchovává seznam lidí, kteří pracují na akci, tedy ty, kterým budou přiřazovány směny na pozicích, kvalifikace respektive typy pozic a další. Tabulka obsahuje detaily pracovníků, zejména jméno, příjmení, kontakty (telefon, email), velikost oblečení a další.

- **Workers on Event**

Spojovací tabulka, která uchovává seznam lidí pracujících na jednotlivé akci, respektive seznam akcí, na kterých pracovali jednotliví lidé. Tabulky akcí a pracovníků jsou ve vztahu M:N a tato tabulka je spojovací entita.

- **Shifts**

Záznamy o pracovních směnách jednotlivých pracovníků. Zároveň propojovací entita mezi pracovníky a akcí (modelem). Tabulka uchovává také časové záznamy jednotlivé směny a její stav (viz. kapitola 5.3.2 Stavby směn). Záznamy zde jsou nejdůležitější pro pozdější finanční i hodinovou kalkulaci pracovníků i celkové akce.

- **Logs**

Logovací tabulka, doplnění klasického logovacího textového souboru. Logovací tabulka má tu výhodu, že v ní lze jednodušeji vyhledávat a třídit dle stupně událostí, času nebo data a dalších kritérií. Obecně vyhledávání v logovací tabulce je mnohem snazší a rychlejší než v logovacím souboru. Logovací soubor ale dokáže zachytit rozmanitější informace, které sice mohou být uloženy i v logovací tabulce, ale už jim nemusí být přiřazeny specifické atomické stupně nebo příznaky.

- **Users**

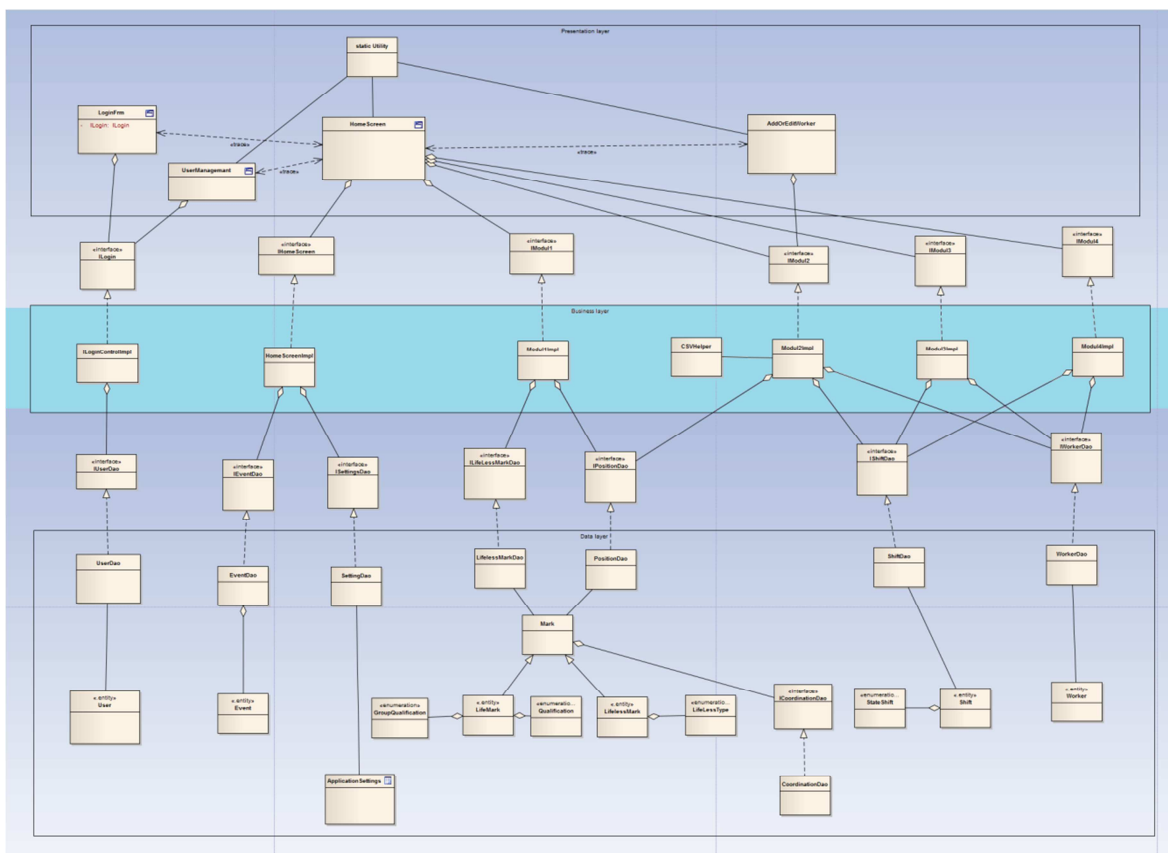
Systémová tabulka programu sloužící k ukládání vytvořených uživatelů, resp. uživatelských účtů. Tabulka obsahuje zejména kontaktní údaje (jméno, email, telefon), login, heslo a přiřazení role uživatele.

11 Návrhový model

Návrhový model tříd rozpracovává analytické třídy do takové podrobnosti, že je již lze implementovat. Na rozdíl od analytického modelu je již implementačně závislý. Je více podrobný a obsahuje již například znovu použitelnost. Pokud je to pro přehlednost vhodné, může obsahovat i systémové třídy, knihovny atd. Podle dostatečně detailního návrhového modelu je možné naprogramovat aplikaci i bez větších architektonických znalostí. Často se využívá přímo generování kódu z modelovacího nástroje.

Návrhový model však nepokryje logiku operací, ty je nutné dopsat ručně. Z definice modelu také vychází nevýhody týkající se dynamiky modulů. Model sám o sobě je statický, a nelze jej jednoduše převést na dynamicky se měnící tak, jak by mohlo být požadováno pro implementaci. Vždy je třeba dobře zvážit míru automatizace a slepého generování či přepisování tříd z analytického návrhu do programovacího jazyku.

Pro vytvořenou aplikaci byl na základě analytických tříd vytvořen návrhový model. Zobrazením kompletního návrhového modelu včetně metod a atributů zde v práci by byla ztracena přehlednost z důvodu velkého rozsahu. Kompletní model včetně popsaného diagramu je k dispozici v elektronické příloze. Diagramy jsou vytvořeny v modelovacím nástroji Enterprise Architect.



Obrázek 22 – Zjednodušený model tříd

Na obrázku 22 je vidět zjednodušený model tříd. Zjednodušený proto, že neobsahuje některé pomocné třídy jako například logiku logování, třídy pro provádění databázových operací a některé statické třídy. Tím, že model neobsahuje tyto položky, neodpovídá definici návrhového modelu, pro přehlednost však byla zvolena tato cesta. Detailní model je k dispozici v elektronické verzi přílohy.

Obrázek znázorňuje rozmístění jednotlivých komponent do příslušných vrstev. Spodní rámeček obsahuje třídy komunikující s databází a zajišťující přístup k datům, včetně samotných entit nebo výčtových typů, ve kterých jsou umístěna data. Nad datovou vrstvou je zavedena sada rozhraní, které zajišťují definici používaných metod.

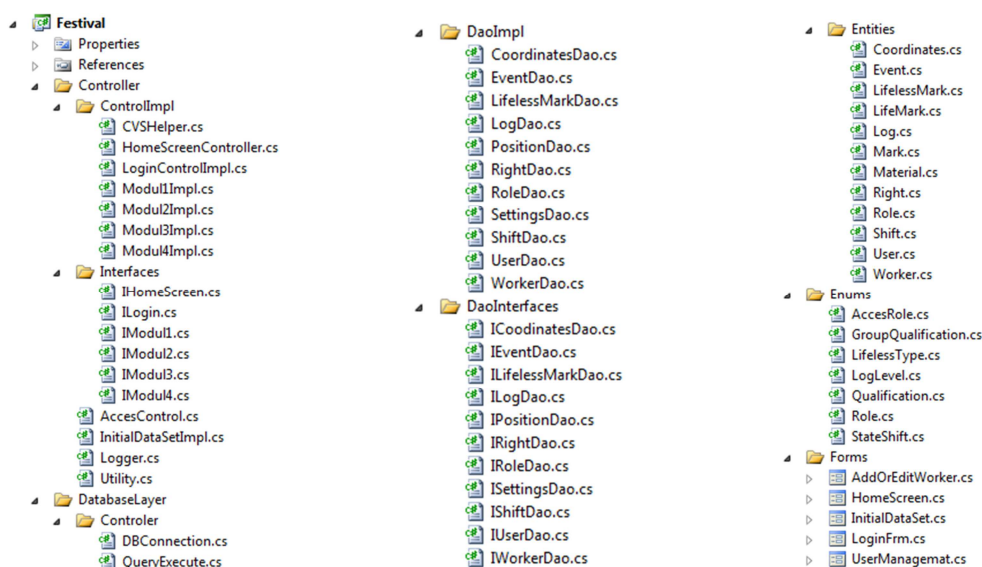
Nad touto vrstvou se nachází business vrstva, která zajišťuje úpravu získaných dat a předává zpracovaná data do prezentační vrstvy, se kterou je opět propojena přes sadu rozhraní. Prezentační vrstva znamená při implementaci v jazyce C# formuláře, tedy definice zobrazení formuláře a jejich pomocný kód pro prezentaci dat. Prezentační vrstva obsahuje také některé pomocné třídy, jako zde třídu „Utility“ pro pomocné operace s daty.

12 Implementace a GUI

Implementace se může zdát jako poslední fáze vývoje aplikace. Při komplexní realizaci IT projektu se však nachází spíše někde uprostřed. Po dokončení fáze implementace nastává ještě testovací fáze, která se překrývá s fází nasazení na prostředí zadavatele a při komerčních projektech ještě závěrečná přejímka projektu, dostání všech smluvních závazků a konečné předání produktu do užívání.

Při implementaci aplikace byl kladen důraz na dodržování principů objektivě orientovaného programování. Vždy byla koncepce implementace vedena snahou dodržet logické rozvrstvení aplikace a mít aplikaci co nejvíce komponentní. Dodržení komponentního přístupu se podařilo pouze z části, protože vytvořit funkční komponenty, které by jednak splňovaly požadované business funkce a zároveň odpovídaly pravidlům komponentového přístupu, bylo někdy velmi těžké. Byl kladen důraz na přehlednost kódu a jeho znovu použitelnost. V rámci možností byly využity některé předem otestované postupy jako například návrhové vzory. Konkrétně byl například pro přístup k databázi použit návrhový vzor singleton. Při některých netriviálních řešeních bylo využito přístupů a metod nalezených s využitím internetu, pokud bylo nalezeno, že nastalý problém již někdo předtím řešil. Třída CSVHelper byla celá převzata při otevřené licenci od neznámého autora, upravena pro požadované vlastnosti, otestována a použita.

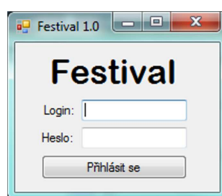
Jak již bylo naznačeno v kapitole „Návrhový model“ a „Rozvrstvení aplikace“, byla snaha aplikaci rozdělit do vrstev oddělených rozhraními z důvodu možné budoucí výměny části aplikace a její modulárnosti. Veškeré rozhraní mají dle zvyklostí jazyka C# počáteční písmeno I (Interface) a za ním následuje název rozhraní. Například IHomeScreen. Vrstva manipulující s daty obsaženými v databázi má všechny třídy s postfixem Dao (Data Access Object). U názvů entit a enum typů byla snaha zachovat konvenci pojmenování tak, aby odpovídala jejich názvům v reálném světě.



Obrázek 23 – Fyzická struktura tříd aplikace

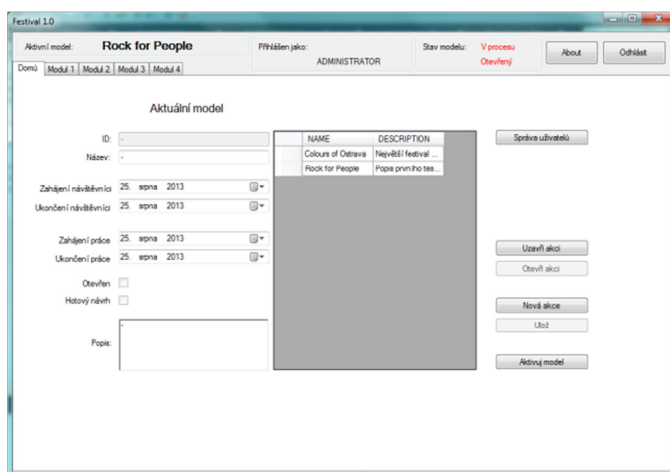
Na obrázku 23 jsou vidět fyzické třídy a jejich rozdělení do „balíčků“ – v terminologii C# jmenných prostorů. Jmenný prostor pomocí tečkové konvence určuje umístění třídy. Například třída „CSVHelper“ leží ve jmenném prostoru „Festival.Controller.ControlImpl“. Veškeré třídy jsou přehledně uspořádány a dají se najít dle logického rozvrstvení a dle účelu, ke kterému třída slouží.

Pokud se zaměříme na vzhled aplikace, tedy rozhraní, se kterým přijde běžný uživatel do styku, nalezneme standardní komponenty známé ze všech programů založených na platformě Windows. Tento vzhled lze přetvořit, pokud bychom nahradili prezentační vrstvu jinými formuláři nebo použili jinou technologii. V této části práce je ovládání aplikace popsáno jen velmi stručně z důvodu poměrně rozsáhlé složitosti a funkčnosti aplikace. Pro zadavatele byla vytvořena uživatelská příručka pro práci se systémem, aby zaměstnanci snadněji pochopili principy použité v aplikaci a přechod z používaných metod byl snadnější.



Obrázek 24 – Přihlašovací formulář aplikace

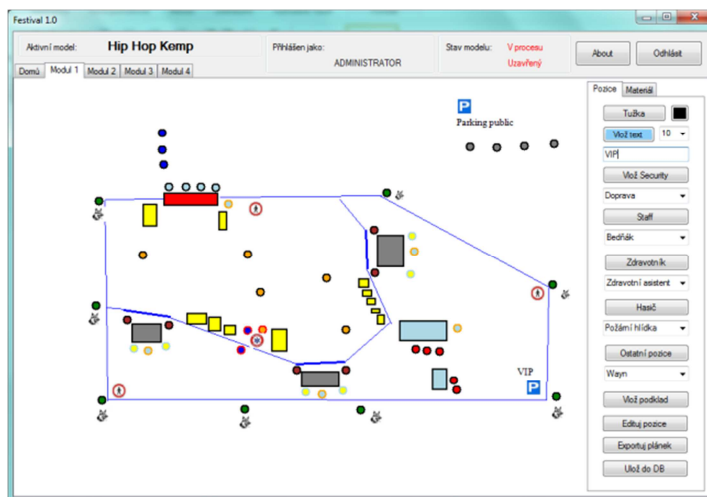
Při spuštění aplikace je třeba se nejprve přihlásit, viz obrázek 24. Každý uživatel má jedinečnou kombinaci uživatelského jména a hesla. Při zadání špatného hesla nebo nesprávných údajů je uživatel upozorněn hláškou „Uživatel neověřen“. Po úspěšném přihlášení se objeví domovská obrazovka – viz Obrázek 25.



Obrázek 25 – GUI Úvodní obrazovky

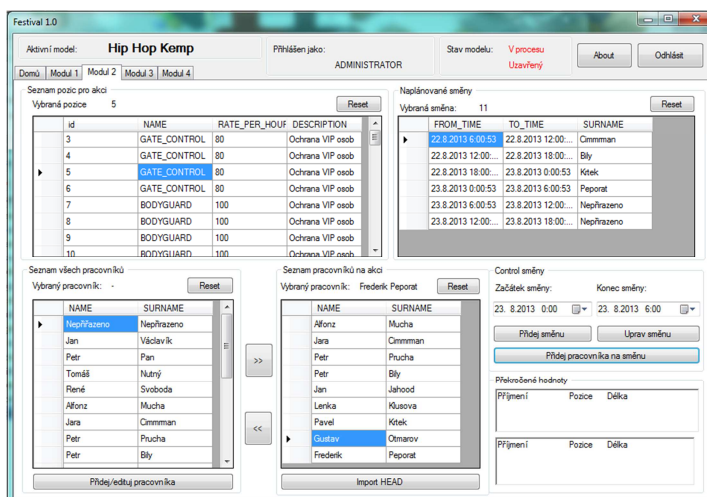
Na obrázcích 25 a 26 je vidět grafické uživatelské rozhraní úvodní obrazovky a Modulu 1. Zcela nahoře je pro všechny moduly stejná část, informace o načteném modelu a přihlášeném uživateli. Dále tlačítko pro odhlášení z aplikace informace o samotné aplikaci.

Na úvodní obrazovce je zejména seznam uložených akcí. Zde jsou pouze dvě testovací akce, v reálném použití bude ale tento seznam zaplněn. V levé části se zobrazují údaje o vybrané akci – pokud je některá z nich vybraná. Také se prostřednictvím těchto input políček dá zadat nová akce. Vpravo nahoře je tlačítko přístupu k ovládání uživatelů. Na pravé straně je ovládání stavu vybrané akce.



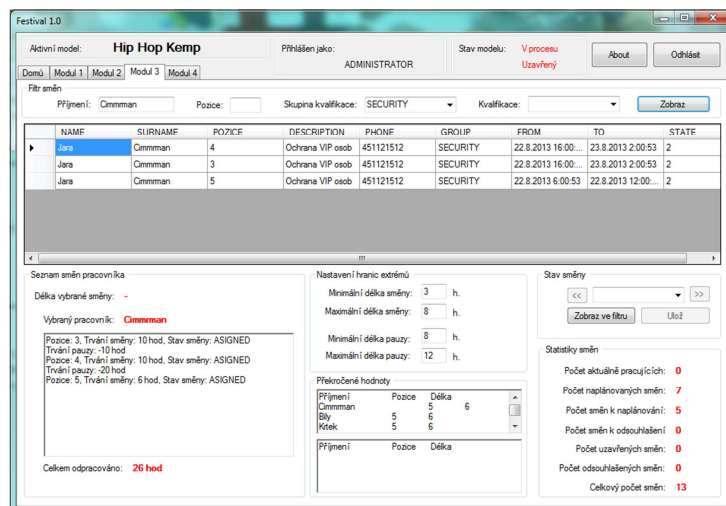
Obrázek 26 – GUI Modulu 1

Na obrázku 26 je Modul 1 s plánkem vybrané akce. Kolečka jsou značky pozic, kdy dle barvy kolečka a jeho výplně lze rozeznat, o jakou pozici se jedná. Barvy byly vybrány na základě již používaného barevného odlišení zadavatele.



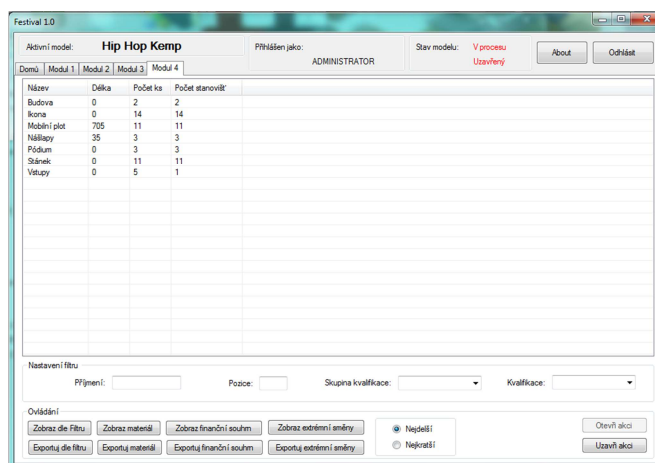
Obrázek 27 – GUI Modulu 2

Na obrázku 27 je vidět grafické rozhraní druhého modulu. V levém horním rohu je seznam pozic na vybrané akci. Vpravo nahoře seznam směn na vybrané pozici. Vlevo dole je seznam pracovníků, kteří kdy byli do systému zavedeni. Toho lze využívat pro často se opakující výběry. Dole uprostřed je seznam pracovníků vybraných na aktivní akci a dole vpravo ovládací prvky a zobrazení směn překračujících navolené extrémní hodnoty.



Obrázek 28 – GUI Modulu 3

Na obrázku 29 je grafické rozhraní Modulu 3. V horní části se nachází input políčka pro vytvoření filtru. Pokud je políčko prázdné, nebere se v potaz. Pod nimi je prostor pro výsledky filtru. V tomto modulu se bude zejména nastavovat stav vybraných směn, toto ovládání je vpravo pod výsledky. V prostřední části je nastavení hodnot pro upozornění na extrémní délky směn a přestávek. V ostatních částech je zobrazení jednotlivých statistik a podpůrných údajů, které pomáhají při obsluze programu. Zejména seznam směn vybraného člověka, zobrazení extrémně dlouhých směn a přestávek nebo jednotlivé souhrny, také aktuální počet pracujících nebo volných lidí a směn.



Obrázek 29 – GUI rozhraní Modulu 4

Na Obrázku 29 je vidět zobrazení Modulu 4. Tento modul je zaměřený na post projektovou práci, tedy na vyhodnocování statistik a předávání dat do systému HEAD. Vizuální schéma je jednoduché, v největší části obrazovky figuruje prostor pro zobrazení statistik před exportem. Statistiky lze zobrazit dle filtru, nebo konkrétní statistiky dle ovládacích prvků ve spodní části. Údaje si je možné vždy zobrazit a ve stejné formě je i exportovat opět do formátu CSV.

13 Testovací strategie

13.1 Motivace

Vytvoření testovacích scénářů je důležitý krok, který slouží jako kontrolní funkční mechanismus pro ověření použitelnosti systému. Dobře napsaný testovací scénář dokáže vyhledat chyby v aplikaci nejen z hlediska výjimek nebo nekorektního chování aplikace, ale zejména architektonické chyby vzniklé chybnou analýzou. Testování je tak závěrečnou fází předávání aplikace před jejím nasazením na produkční prostředí.

Výsledky testů se většinou předávají zákazníkovi jako potvrzení funkčnosti aplikace. Je vhodné, aby se na sestavení testovacích scénářů podílel také zadavatel, pro kterého je aplikace určena. Sám si tak může určit akceptační kritéria, což jsou z jeho strany nutné podmínky, které musí aplikace splňovat, aby byla pro jeho účely použitelná.

13.2 Testovací scénáře

Pro ověření funkčnosti byla vytvořena sada testovacích scénářů, která vychází z vytvořených případů užití a jejich scénářů. Základní sada obsahuje početně méně scénářů než je případů užití. To je způsobeno faktem, že jeden testovací scénář dokáže pokrýt více případů užití najednou.

Sada testovacích scénářů obsahuje název, popis celé sady a výsledky testu. Sumarizaci celého testování a drobné statistiky (například poměr úspěchu nebo neúspěchu k celkovému počtu). Důležité je zaznamenání verze programu, která byla testována, zejména z důvodu opravy případných chyb v následujících verzích. I po opravách chyb je ale vždy nutné otestovat celou kompletní sadu scénářů, aby bylo zaručeno, že opravou chyby nebyla do aplikace zanesena jiná chyba.

Každý jednotlivý scénář obsahuje jedinečné označení (ID), název a popis co konkrétně se v něm testuje. Dále výsledek celého scénáře. Scénář se pak skládá z dílčích kroků, které jsou již atomické. Tyto kroky obsahují opět jedinečný identifikátor, označení o jaký druh kroku se jedná (ověření, vstup dat aj.). Dále číslování kroku, popis vykonávané akce, pokud jsou nutné vstupní data tak popis vstupních dat, očekávaný výsledek – tedy popis co by mělo po provedení kroku být zobrazeno nebo co by systém měl po kroku provést, výsledek jednotlivého kroku a poznámku. Výsledky jak jednotlivých kroků, tak celého scénáře jsou vždy stavy „Netestováno“, „OK“, „Chyba“ a „Varování“.

Vždy platí, že pokud je jeden z kroků scénáře označen chybovou hláškou, pak i celý scénář dopadl podle stavu toho kroku. Pokud je tedy jeden z kroků označen jako „Chyba“ celý scénář je pak označen jako „Chyba“. Poté je třeba chybu v aplikaci najít, odstranit a celou sadu scénářů otestovat znova. Pokud je krok, respektive scénář označen „Varování“ je na dohodě s klientem zda bude aplikaci akceptovat v daném stavu nebo jestli bude očekávat opravu nebo úpravu. Vždy je postup individuální.

Příklad testovacího scénáře je vidět na obrázku 31, zde konkrétně jde o ověření funkčnosti správy uživatelů, tedy vytvoření nového uživatele a modifikaci stávajícího. Kompletní sada testovacích scénářů je k dispozici v příloze této práce a také v elektronické verzi práce.

| Sada testovacích scénářů pro ověření funkčnosti aplikace pro bezpečnostní agenturu | | | | | | | | | | | | | | | | | |
|---|---------|------|--|------|---|----------|----------|-------------|--|--------------|---|--------|---|-----------|---|-----|----|
| Testoval: Ondřej Serbousek Datum: 15.8.2013-20.8.2013 Verze: 1.0 | | | | | | | | | | | | | | | | | |
| Výsledky: <table border="1"> <tr><td>Výsledky</td><td></td></tr> <tr><td>Netestováno:</td><td>0</td></tr> <tr><td>Chyba:</td><td>0</td></tr> <tr><td>Varování:</td><td>0</td></tr> <tr><td>OK:</td><td>2</td></tr> </table> | | | | | | | | Výsledky | | Netestováno: | 0 | Chyba: | 0 | Varování: | 0 | OK: | 2 |
| Výsledky | | | | | | | | | | | | | | | | | |
| Netestováno: | 0 | | | | | | | | | | | | | | | | |
| Chyba: | 0 | | | | | | | | | | | | | | | | |
| Varování: | 0 | | | | | | | | | | | | | | | | |
| OK: | 2 | | | | | | | | | | | | | | | | |
| <table border="1"> <tr><td>Kroky testů</td><td></td></tr> <tr><td>Netestováno:</td><td>0</td></tr> <tr><td>Chyba:</td><td>0</td></tr> <tr><td>Varování:</td><td>0</td></tr> <tr><td>OK:</td><td>15</td></tr> </table> | | | | | | | | Kroky testů | | Netestováno: | 0 | Chyba: | 0 | Varování: | 0 | OK: | 15 |
| Kroky testů | | | | | | | | | | | | | | | | | |
| Netestováno: | 0 | | | | | | | | | | | | | | | | |
| Chyba: | 0 | | | | | | | | | | | | | | | | |
| Varování: | 0 | | | | | | | | | | | | | | | | |
| OK: | 15 | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| AS_01: Vytvoření uživatele Test ID: AS01 Vytvoření nového uživatele Výsledek: Pass | | | | | | | | | | | | | | | | | |
| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka | | | | | | | | | | |
| AS_01_01 | Krok | 1 | Přihlášení do systému jako administrátor | | Úspěšné přihlášení | Pass | | | | | | | | | | | |
| AS_01_02 | Krok | 2 | Zvolení nabídky "Správa uživatelů" | | Zobrazení nabídky správy uživatelů | Pass | | | | | | | | | | | |
| AS_01_03 | Krok | 3 | Zvolení možnosti "Nový uživatel" | | Zobrazení poí pro hodnoty nového uživatele. | Pass | | | | | | | | | | | |
| AS_01_04 | Krok | 4 | Vyplnění údajů pro nového uživatele. | | | Pass | | | | | | | | | | | |
| AS_01_05 | Krok | 5 | Uložení a potvrzení | | | Pass | | | | | | | | | | | |
| AS_01_06 | Krok | 6 | Odhlášení z aplikace | | Odlógování administrátora | Pass | | | | | | | | | | | |
| AS_01_07 | Ověření | 7 | Přihlášení do systému pod novým uživatelem | | Úspěšné přihlášení | Pass | | | | | | | | | | | |
| AS_01_08 | Ověření | 8 | Kontrola funkčnosti a přístupnosti UC v závislosti na roli nového uživatele. | | | Pass | | | | | | | | | | | |
| AS_02: Změna uživatele Test ID: AS02 Změna stávajícího uživatele Výsledek: Pass | | | | | | | | | | | | | | | | | |
| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka | | | | | | | | | | |
| AS_02_01 | Krok | 1 | Přihlášení do systému jako administrátor | | Úspěšné přihlášení | Pass | | | | | | | | | | | |
| AS_02_02 | Krok | 2 | Zvolení nabídky "Správa uživatelů" | | Zobrazení nabídky správy uživatelů | Pass | | | | | | | | | | | |
| AS_02_03 | Krok | 3 | Vybrání existujícího uživatele ze seznamu. | | Zobrazení hodnot vybraného uživatele. | Pass | | | | | | | | | | | |
| AS_02_04 | Krok | 4 | Změna údajů vybraného uživatele | | | Pass | | | | | | | | | | | |
| AS_02_05 | Krok | 5 | Potvrzení a uložení | | Úspěšné uložení hodnot. | Pass | | | | | | | | | | | |
| AS_02_06 | Krok | 6 | Zobrazení uživatele | | Opětovné zobrazení měněného uživatele. | Pass | | | | | | | | | | | |
| AS_02_07 | Ověření | 7 | Kontrola nových hodnot. | | Nové uložení (změněné hodnoty) | Pass | | | | | | | | | | | |

Obrázek 30 – Screen z testovacího scénáře

13.3 Testování v reálném provozu

Po dohodě s firmou ČECHYMEN byla aplikace testována i v reálném prostředí na pořádané akci. Bylo vybráno zajišťování hokejového zápasu, což je pouze jednodenní akce s obsazením pracovníků do 50 lidí. I když na tento typ není aplikace přímo zacílená, pro prvotní test byla vyhovující. Zatím pouze v testovacím režimu, kdy sice obsahovala reálná data, ale byla stále zaštiťovaná stávajícím používaným systémem. Na konci byla porovnáována data zpracovávaná starým systémem a data z aplikace, aby byla ověřena správnost.

Z časových důvodů nebylo možné v programu korektně otestovat reálný návrh akce, jelikož tato akce byla naplánovaná několik měsíců předem. Bylo tedy provedeno pouze přemodelování tak, aby seděla s reálnými daty. Následně byli vloženi pracovníci a rozplánovány směny dle reality. Na konci proběhl export dat a statistik a porovnání s používaným systémem, kdy všechna důležitá data souhlasila s realitou.

Dalším plánovaným velkým zátěžovým testem bylo suplování organizace motocyklových závodů Moto GP na Brněnském Masarykově okruhu. Tato akce je rozsahem a počtem podpůrného personálu největší v České Republice a i počtem návštěvníků kumulovaných v jeden den respektive víkendu na jednom místě patří k největším. V současné době žádná česká firma nedisponuje takovými zdroji, aby celé zabezpečení dokázala pokrýt sama, proto se na akci společně podílí mnoho bezpečnostních a personálních agentur jako

subdodavatelé včetně firmy Čechymen. Přesný počet poskytnutých lidských zdrojů není možné uvést, protože se jedná o citlivá data celé akce. Akce však patřila do kategorie řádově stovek pracovníků v jeden okamžik.

Je nutno podotknout, že se jedná o největší akci v ČR, na které lze tento program otestovat. Proto bylo rozhodnuto pouze o částečném použití. Zde aplikace narážela na některé funkční limity, kdy nedokázala například zobrazit celou rozlohu rozlehlého areálu brněnského Masarykova okruhu. Při tomto testu se dobře projevily limity aplikace a možnosti ke zlepšení do budoucna.

Po skončení těchto akcí byl výsledek konzultován s odborníky z bezpečnostní agentury a společný závěr byl, že aplikace i systém jsou plně funkční, ale mají jisté limity. To, že tyto limity nebyly odhaleny při analýze, je důsledek toho, že nikdy předtím žádná podobná aplikace neexistovala a při modelování funkčnosti byly některé aspekty opomenuty. Tyto detaily bohužel poté způsobily existenci limitů. Zmíněna byla možnost pokračování vývoje, je to však věc dalšího jednání a zatím nebyla iniciovaná ani z jedné strany. Celkově byla konstatována spokojenost a splnění požadavků. O produkčním nasazení aplikace se bude dále jednat, v době uzávěrky práce nebylo vyneseno konečné rozhodnutí.

Závěr

Realizace tohoto projektu je z mé strany hodnocena úspěšně z několika hledisek. Ze strany personálního se jedná zejména o osvojení reálných principů realizací větších projektů, které už bývají v porovnání s touto prací dost podobné. Patrně nejvíce rozvíjeným polem osobních schopností byly „soft-skills“, které byly nutné pro realizaci propojení všech potřebných vstupů projektu. Zejména se jednalo o propojení prostředí reálné praxe s akademickým, kdy často nebylo jednoduché úspěšně nároky obou prostředí jednoduše propojit.

Pro realizaci bylo nutné projít kompletním procesem vývoje nového softwaru pro zadavatele, který nemá jasné představy, co by od produktu očekával. Bylo nutné společně nadefinovat funkční chování procesů, reálně probíhajících u zadavatele v praxi. Toto bylo potřeba nejprve provést teoreticky a pochopené principy ověřit. Z těchto nabytých znalostí bylo nutné opět ve spolupráci s odborníky na business funkčnost vytvořit konkrétní požadavky, které již bylo možné proměnit v prvotní analytické modely. Poté mohla začít důkladná analýza a přeměna analytických modelů na návrhové. Zde probíhaly konzultace s odborníky o technologiích, kdy se často řešila vhodnost použití různých technologií a architektonické metody pozdější implementace. Také bylo často nutné studium konkrétních nových technologií a jejich použití. Zejména se jednalo o některé ne tak často používané, které ale dobře vyhovovaly požadavkům zadavatele. V této fázi bylo využíváno hlavně odborníků z akademického prostředí. Po dokončení architektonického návrhu začala intenzivní fáze implementace, ve které i přes důkladnou analýzu vznikly některé nové poznatky, které předtím nebyly úplně odhaleny. Některé z nich byly dodatečně implementovány, a některé vyhodnoceny jako hlubší zásah do navrhované funkčnosti a odloženy pro možné příští verze.

Po kompletaci začala fáze ověřování dosažených výsledků. Prvotní testování probíhalo podle vytvořených testovacích scénářů. Zde je nutné podotknout, že korektně by kompletní testování měla zajišťovat jiná osoba než analytik či dokonce programátor, z důvodu nutnosti ověření míry pochopení jednotlivých funkcí. Druhou fází testů bylo nasazení v reálném prostředí pro reálné akce. V době dokončení práce pouze jako záložní systém sekundovaný původní metodikou firmy. Tyto testy byly zadavatelem hodnoceny jako úspěšné s výhradami. Po analýze výhrad vyplynulo, že se jedná spíše o opomnění některých principů, což vyústilo v některé nepřesné definice požadavků. Všechny výhrady byly ale označeny jako nekritické, tedy systém byl s konečným stanoviskem prohlášen jako použitelný pro praxi, s doporučením o úpravě pro eventuelní nové verze.

Všechny dílčí cíle práce byly úspěšně splněny. V některých detailech je zpětně vidět částečná nedokonalost a projevily se některé skryté chyby, které ale ve výsledku neměly vliv na celkový splněný cíl – ověření použitelnosti navrženého systému v praxi. Je na další domluvě, zda bude systém v současném stavu využíván, ale i v případě, že by byly nové změnové požadavky na aplikaci tak rozsáhlé, že by bylo nutné zkonstruovat systém znova, tato verze je důkazem proveditelnosti a rentabilnosti vložené práce. Projekt prokázal

použitelnost a dokázal z myšlenky vytvořit hmatatelný výsledek, který dokáže lidem ulehčovat a zjednodušovat práci, vnést do metodik systém a automatizovat postupy. To vše na poli působnosti, kde moderní technologie vytvářené přímo na míru zatím nemají příliš velké zastoupení, čímž se dá tvrdit, že je práce jistým unikátem.

Slovníček pojmů

| | |
|------------------|---|
| HEAD | Stávající informační systém firmy |
| Event | jednorázová akce resp. zakázka. |
| HR manager | personální pracovník resp. manažer |
| Security manager | bezpečnostní manažer |
| Viewer | pozorovatel, divák |
| Supervisor | Supervizor – velitel pracovníků |
| Coordinator | Koordinátor akce, nadřízený supervisora |
| Happy day | označení pozitivního toku událostí. Případ kdy nedojde k chybě a celá prováděná akce proběhne bez problémů. |

Literatura

- [1] Arlow, J., & Narlow, I. (2007). *UML2 a unifikovaný proces vývoje aplikací*. Praha: Computer press.
- [2] Gavrilov, A. (2005, 8 24). *Finisar.SQLite*. Retrieved 7 11, 2013, from sourceforge.net: <http://adodotnetsqlite.sourceforge.net>
- [3] Mozilla Foundation. (2013, 3 16). *Mozilla adons*. Retrieved 7 12, 2013, from Mozilla: <https://addons.mozilla.org/cs/firefox/addon/sqlite-manager>
- [4] Nagel, C. (2006). *C# 2005: Programujeme profesionálně*. Brno: Computer Press.
- [5] Riordian, R. M. (2000). *Vytváříme relační databázové aplikace*. Praha: Computer press.
- [6] Rumbaugh, J. (1998). *The Unified Modeling Language Reference Manual*. Addison-Wesley.
- [7] SQLite. (n.d.). *SQLite*. Retrieved 6 24, 2013, from SQLite: www.sqlite.org

Příloha A – Kompletní seznam požadavků včetně popisu

Modul 1:

- FR_M1_1: System shall create new model
 - Vytvoření nového modelu pro reálnou akci
- FR_M1_2: System shall import new marks
 - Import nových uživatelských značek
- FR_M1_3: System shall insert mark to model
 - Vložení značek do zavedeného modelu
- FR_M1_4: System shall remove life mark from model
 - Odstranění (vymazání) značek pozic z modelu
- FR_M1_5: System shall export picture
 - Export grafického zobrazení modelu do bitmapového formátu – obrázku. Využití pro prezentaci stavu modelu
- FR_M1_6: System shall export list of working positions
 - Export seznamu pracovních pozic v přehledném formátu. Využití pro prezentaci nebo pro předání informace pracovníkům, kteří přímo nepracují s programem.
- FR_M1_7: System shall delete background picture
 - Výmaz obrázkového nebo mapového podkladu.
- FR_M1_8: System shall keep position of marks with background
 - Záruka ukotvení uživatelských značek vzhledem k pracovnímu obrázkovému nebo mapovému podkladu.
- FR_M1_9: System shall load background picture
 - Načtení pozadí (obrázek nebo mapa) uložené v databázi.
- FR_M1_10: System shall import background pictures
 - Načtení pozadí (obrázek nebo mapa) z externího zdroje v bitmapovém formátu
- FR_M1_11: System shall view current state of model
 - Zobrazení aktuálního stavu modelu, včetně zobrazení seznamu pracovních pozic, náčrtu areálu a použitého či potřebného materiálu.
- FR_M1_12: System shall access according user rights
 - Zabezpečení přístupové politiky. Dle typu účtu různé přístupy do systému.
- FR_M1_13: System shall store background picture
 - Ukládání vytvořených nebo importovaných pozadí (obrázkový nebo mapový podklad) do databáze.
- FR_M1_14: System shall export current state of model
 - Export aktuálního stavu modelu pro potřeby druhého modulu nebo prezentaci.
- FR_M1_15: System shall have possibility confirm model
 - Potvrzení správnosti návrhu kvalifikovaným odborníkem (bezpečnostní manažer) a způsobilost předání modelu do druhého modulu.
- FR_M1_16: System shall working on empty background
 - Varianta práce na prázdném pozadí – bez mapového obrázkového nebo jiného podkladu
- FR_M1_17: System shall modified life marks
 - Systém bude umět provádět editaci již zakreslených značek pozic

Modul 2

- FR_M2_1: System shall export list of position for workers
 - Vytvoření seznamu pozic pro pracovníky. Slouží zejména pro párování pracovních pozic s pracovníky.
- FR_M2_2: System shall import workers from other system by CSV files
 - Funkce sloužící k automatizaci práce. Vstupem je CSV soubor s pracovníky. Soubor pochází ze stávajícího systému HEAD který slouží jako personální systém správy lidí pro firmu.
- FR_M2_3: System shall match selected worker with position
 - Klíčová část druhého modulu – spárování pozic získaných z prvního modulu s pracovníky zpravidla importovanými z externího zdroje.
- FR_M2_4: System shall propose most used data
 - Systém si pamatuje používané hodnoty (jména aktivních pracovníků) a umí je vybírat ze seznamu. Pokud tedy importovaný seznam není úplný, je možné data doplnit z těchto již použitých používaných.
- FR_M2_5: System shall control require data
 - Systém bude dohlížet na vyžadované hodnoty. Validace potvrdí nebo vrátí změny.
- FR_M2_6: System shall export list of workers
 - Systém umí exportovat seznam pracovníků. Spárované s pozicemi i samotné.
- FR_M2_7: System shall work with various type of position
 - Pozice mohou být rozlišovány pro obsazení různou kvalifikací. Typy kvalifikací jsou popsány níže v kapitole věnované funkčnímu designu Modulu 2.
- FR_M2_8: System shall add new shifts
 - Systém musí umět přidávat novou směnu na pozici.
- FR_M2_9: System shall view state of model
 - Zobrazení aktuálního stavu modelu. Požadavek je zde pro vyjasnění nutnosti kontrolovat stav kdykoli.
- FR_M2_10: System shall manually insert workers
 - Pro případ nedostupnosti importu pracovníků nebo nedostatečnosti funkce nejčastěji používaných pracovníků je nutná možnost manuálního vkládání pracovníků pro obsazení pozic.
- FR_M2_11: System shall manage shifts
 - Práce s pracovními směnami. Při nutnosti provádění změn musí být toto umožněno.
- FR_M2_12: System shall export list of require supply
 - Export seznamu vybavení. Analogie pro funkci zobrazení seznamu vybavení, s možností exportu do přehledného formátu, zejména pro osoby, které nejsou v přímé interakci se systémem.

Modul 3

- FR_M3_2: System shall currently view of working employers
 - Pohled na aktuální stav pracujících zaměstnanců. Kolik pracovníků je fyzicky na ploše akce a pracuje.
- FR_M3_3: System shall monitor minimal breaks between shifts
 - Automatická kontrola sloužící k zabránění překročení minimální doby odpočinku a přestávky.
- FR_M3_4: System shall monitor extreme shift anomaly
 - Automatická kontrola k zabránění extrémního vytěžování některých pracovníků, nebo naopak minimální odvedené práce jiných.
- FR_M3_5: System shall have extreme border as user settings
 - Nastavení hranic hodnot považovaných za extrémy. Například minimální přestávky, mezery mezi směnami nebo celkové hodinové kvóty jednotlivých pracovníků.
- FR_M3_6: System shall keep all input data
 - Není možný výmaz vložených dat. Požadavek také pokrývá ukládání všech vkládaných dat. Žádná data se nesmí ztratit během práce s aplikací nebo jejím vypnutí.
- FR_M3_7: System shall export filtered data
 - Možnost vytvoření filtru a vyexportování dat podle daného filtru. Platí pro aktuální stav modelu.
- FR_M3_9: System shall view filtered data
 - Vytvoření a zobrazení filtrovaného aktuálního stavu modelu.
- FR_M3_10: System shall searching for data by given filter
 - Vyhledávání dat dle daného filtru. Typický příklad je vyhledávání pracovníka a jeho směn.
- FR_M3_11: System shall change state of shifts
 - Klíčová a nejpoužívanější funkce třetího modulu. Změna stavu pracovní směny pro jednotlivé pracovníky. Stav směn jsou popsány v kapitole Funkčního Designu Modulu 3.
- FR_M3_12: System shall view data according to user rights
 - Požadavek zdůrazňující nutnost omezení přístupu a pravomocí jednotlivých uživatelů dle jejich práv.
- FR_M3_13: System shall create new shifts
 - Vytvoření nové pracovní směny na pracovní pozici.
- FR_M3_14: System shall change shifts
 - Změna pracovní směny. Posunutí časových úseků, nebo její zrušení.
- FR_M3_15: System shall operate with variant state of shifts
 - Nutnost možnost operace s pracovní směnou během všech jejích stavů.

Modul 4

- FR_M4_1: System shall create statistics by filtering condition
 - Obdoba funkce exportu aktuálních statistik v třetím modulu. Vytvoření finálních statistik podle daného filtru.
- FR_M4_2: System shall export data in CSV format
 - Požadavek pro automatizaci práce. Exportní CSV soubory se budou používat pro přenos do systému HEAD případně jiných systémů používaných firmou.
- FR_M4_3: System shall create finally sum working hours
 - Požadavek pro vytváření hodinových statistik jednotlivých pracovníků i všech dohromady.
- FR_M4_4: System shall create final financial sum
 - Peněžní výsledek akce. Dle daných hodinových sazeb lze spočítat jednotlivé odměny pro pracovníky. Podle zadání účtované hodinové sazby lze spočítat i hrubou kalkulaci klientovi.
- FR_M4_5: System shall create sum of resources
 - Výsledná suma použitých zdrojů dle parametrů zadaných při návrhu zdrojů.
- FR_M4_7: System shall close event
 - Konečné uzavření akce. Může provést pouze Bezpečnostní manažer po zkontrolování všech náležitostí, nebo administrátor. Po uzavření akce se nedají měnit jakékoli data daného modelu. Změna je vratná.
- FR_M4_8: System shall view data according user rights
 - Požadavek zdůrazňující nutnost omezení přístupu a pravomocí jednotlivých uživatelů dle jejich práv.
- FR_M4_9: System shall reopen event.
 - Vrácení uzavření akce. Standardně se s tímto krokem nepočítá, pro výjimečné případy je požadavek pro znovu otevření modelu.

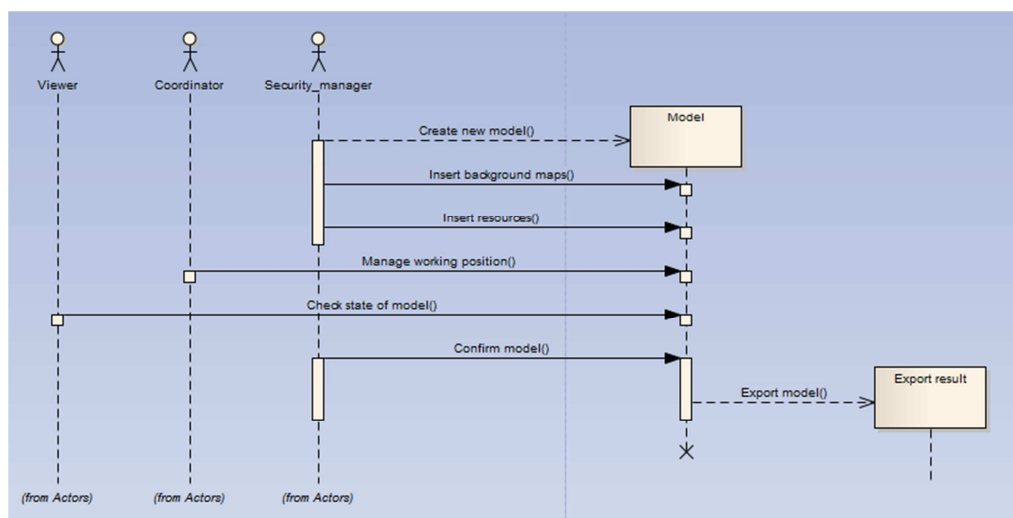
Nefunkční požadavky

- NF_1: System shall be created in C#
 - Pro vytvoření systému byl zvolen jazyk C#
- NF_2: System shall be able to use DB supporting SQL
 - Data aplikace musí být ukládána do databáze. Tato databáze musí podporovat standardy jazyka SQL. Běžné soubory na FS mohou být použity pouze jako doplňkové úložiště dat.
- NF_3: System shall be able to save all input data
 - Veškerá zadaná data do systému musí být možné uložit pro pozdější znovupoužití.
- NF_4: System shall be able to archive data
 - Systém musí obsahovat mechanismy pro archivaci dat.
- NF_5: System shall be able to run without special licence (Except OS Windows)
 - Firma nepředpokládá nákup žádných speciálních licencí pro použití komerčních programů, kromě operačního systému Microsoft Windows. Veškeré použité technologie musí být v licenci Open Source nebo jiné volně dostupné.
- NF_6: System shall be runnable on OS Windows
 - Primární operační systém, pod kterým bude aplikace provozována bude Microsoft Windows. Minimální verze systému je Windows XP s nainstalovaným SP3.
- NF_7: System shall be runnable on Intel platform
 - Aplikace bude primárně provozovaná na platformě Intel.
- NF_8: System shall be available in offline mode
 - Systém nesmí být závislý na internetovém připojení.
- NF_9: System shall be able to check input data types
 - Systém musí sám kontrolovat a validovat vstupní data, minimálně na úrovni datových typů.
- NF_10: System shall be able work with exchange formats
 - Systém musí umět přijímat či odesílat uznávaný formát výměny dat.
- NF_11: System shall be able work with CSV files
 - Systém musí umět přijímat či odesílat minimálně data ve formě CSV souborů.
- NF_12: System shall be oriented on user-rights architecture
 - Systém musí umět pracovat s uživatelskými účty.
- NF_13: System shall propose diferent level of acces for different roles
 - Pro jednotlivé uživatele musí být systém schopen nabídnout různé úrovně přístupu. Použití rolí s různými přístupovými právy.
- NF_14: System shall autentificate user by login and password
 - Systém musí umět ověřit totožnost přihlašujícího se uživatele podle jeho přihlašovacího jména a hesla.
- NF_15: System shall save passwor in hash format
 - Veškerá hesla uložená v aplikaci (do databáze) musí být zakódovaná tak, aby je nebylo možné rozluštit.
- NF_16: System shall be unrestricted with total number of user
 - Systém by neměl být teoreticky omezen v celkovém počtu uživatelských účtů.
- NF_17: System shall be unrestricted with total number of models

- Systém by neměl být teoreticky omezen v počtu uložených modulů – zpracovávaných akcí.
- NF_18: System shall be intuitive for non-informatics users
 - Jelikož hlavní uživatelé systému nejsou odborníci v informačních technologiích, vysoký důraz je kladen na intuitivnost a přehlednost ovládání programu.
- NF_19: System shall be unrestricted with time of using or time zones etc.
 - Systém nesmí mít omezení vyplývající z časových posunů, rozdílů časů nebo době použití.
- NF_20: System shall be administrating by IT workers
 - Administrativní práce v systému bude prováděna osobami se znalostí informačních technologií.
- NF_21: System shall be stable and persistence
 - Systém musí být odolný a stabilní. Aplikace nesmí padat vlastní vinou a při pádu způsobeném třetí stranou nesmí dojít ke ztrátě dat.
- NF_22: System shall be able response in few seconds
 - Doba odpovědi systému musí být v řádech jednotek sekund. Bude se měnit v závislosti na prováděných operacích a obsahu DB, ale opět pouze v řádech jednotek sekund.

Příloha B – Sekvenční diagramy Business flow modulů.

Modul 1

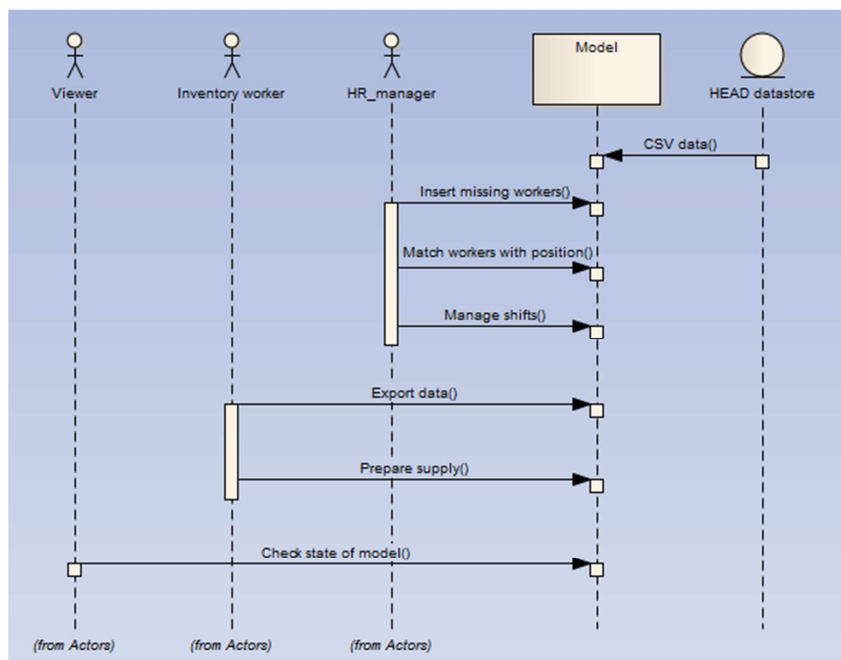


Obrázek 31 – Zobrazení aktivit jednotlivých rolí

Obrázek zobrazuje úkoly jednotlivých rolí při práci s Modulem 1. Nosnou rolí je zde Bezpečnostní manažer, který inicializuje celý model a nakonec jej musí potvrdit. Tato role má také jako jediná právo vkládat nové zdroje (značky) a manipulovat s pozadím / mapovým podkladem.

Role Koordinátora pak upřesňuje rozložení značek, nesmí již však manipulovat s jejich kvantitou. Role pozorovatele pak slouží pouze k zobrazení stavu. Po dokončení práce s Modulem 1 může být model exportován (např. ve vhodném zobrazovacím formátu) nebo předán ke zpracování do modulu 2.

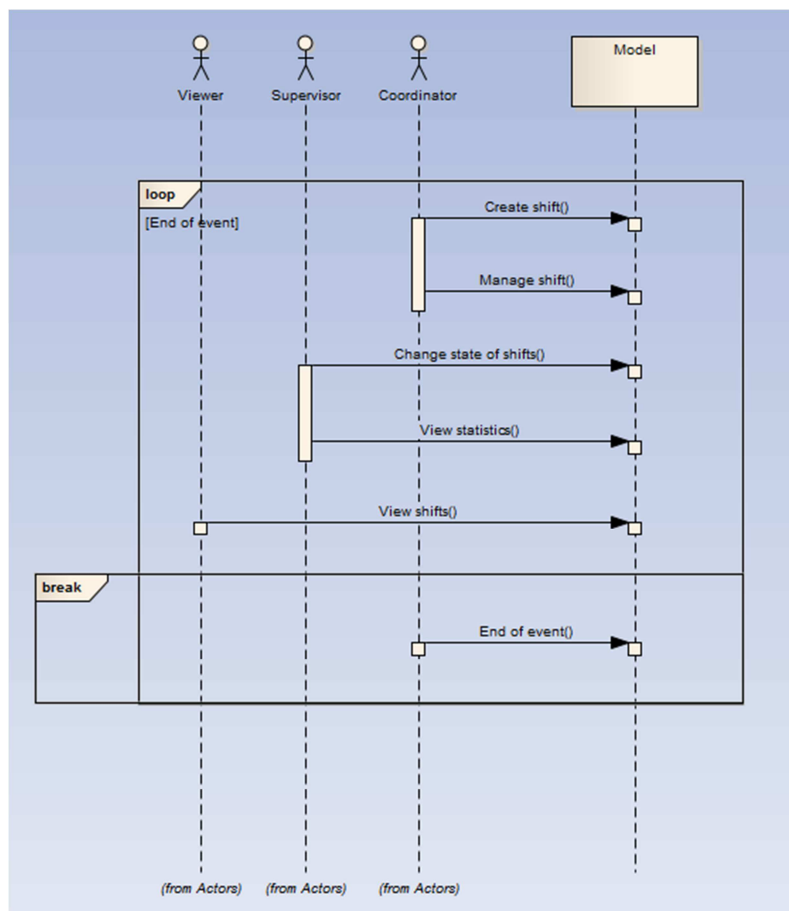
Modul 2



Obrázek 32 – Sekvenční diagram práce Modulu 2

Na sekvenčním diagramu práce druhého modulu aplikace je vidět výše popsané. V modulu jsou zapojeny tři role – Pozorovatel, Skladník a personální pracovník. Z okolních entit do modelu zasahuje systém HEAD respektive CSV soubor vygenerovaný z jeho DB vybraných pracovníků. Personální pracovník doplní a zkompletuje pracovníky na jednotlivé pozice, a dále s modelem pracuje skladník, který podle informací v modelu připraví zásoby pro event. Pozorovatel může zjistit, co se v systému děje, ale nemůže do systému zasáhnout.

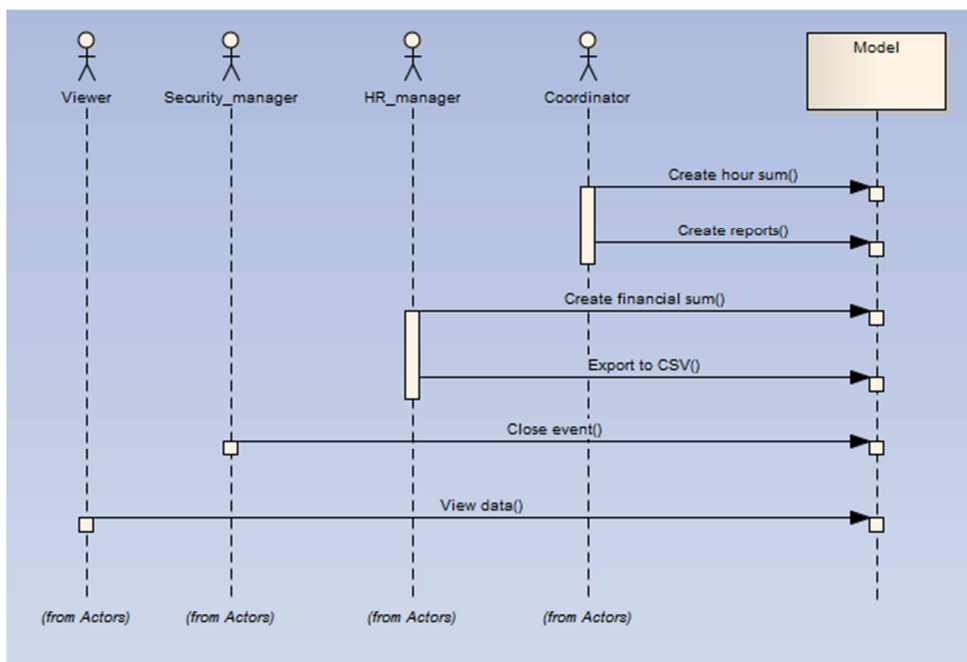
Modul 3



Obrázek 33 – Sekvenční diagram Modul 3

Na sekvenčním diagramu business flow je vidět jednotlivé role operující v systému a jejich možné úkony. Diagram pouze ilustruje možné varianty práce, nedává však výčet všech možných cest práce s modelem. Na diagramu je znázorněn cyklus, který končí impulzem pro fyzické ukončení akce, které dává Coordinator akce. Tento cyklus ve svém těle umožňuje opakovat jednotlivé operace, dokud nenastane podmínka pro jeho ukončení. Z obrázku nemusí být patrné, že v případě zakreslení cyklu v sekvenčním diagramu nemusí být bezpodmínečně nutné v každém průchodu cyklu vykonávat postupně všechny znázorněné operace, ale je možné pouze zvolit ty potřebné.

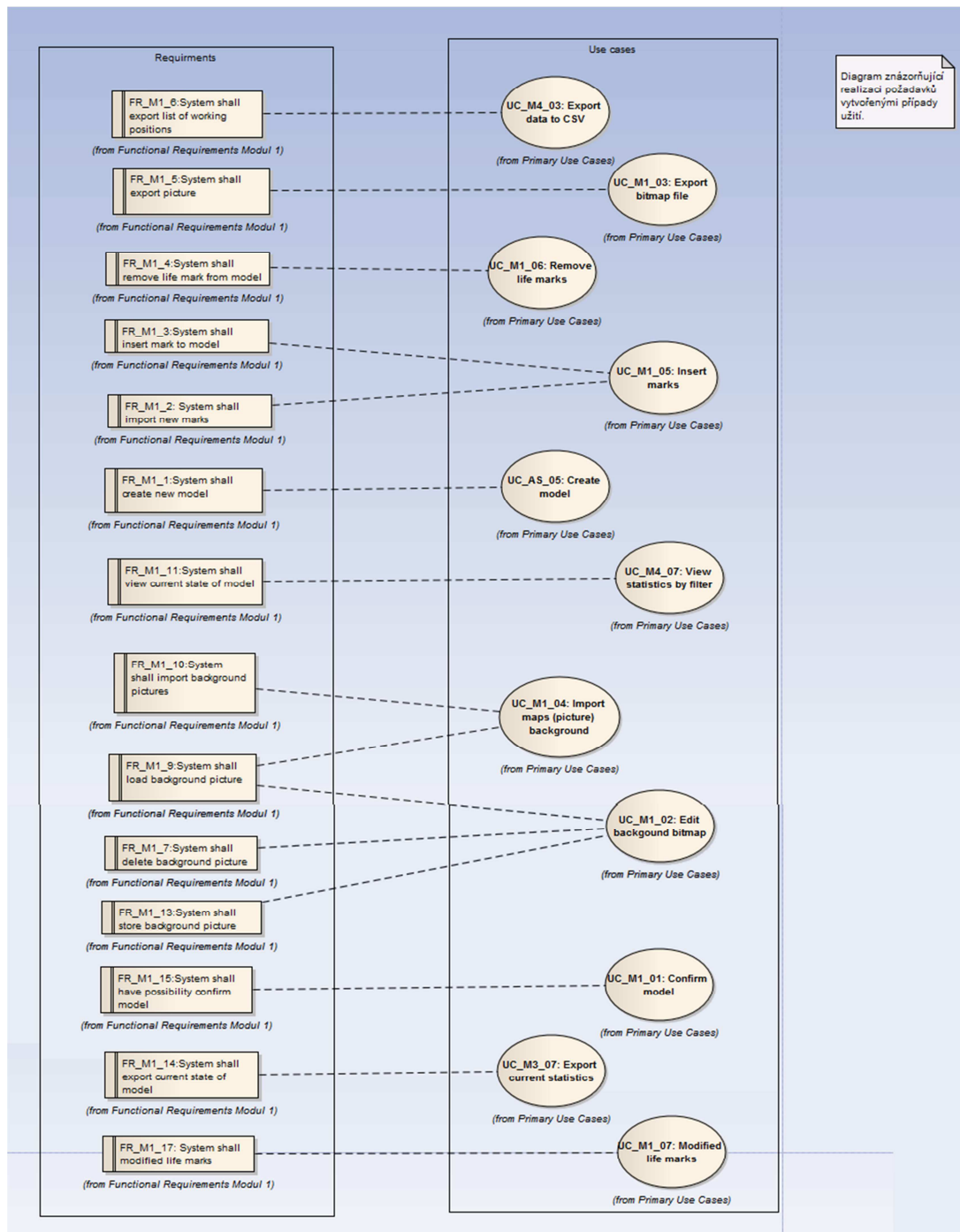
Modul 4



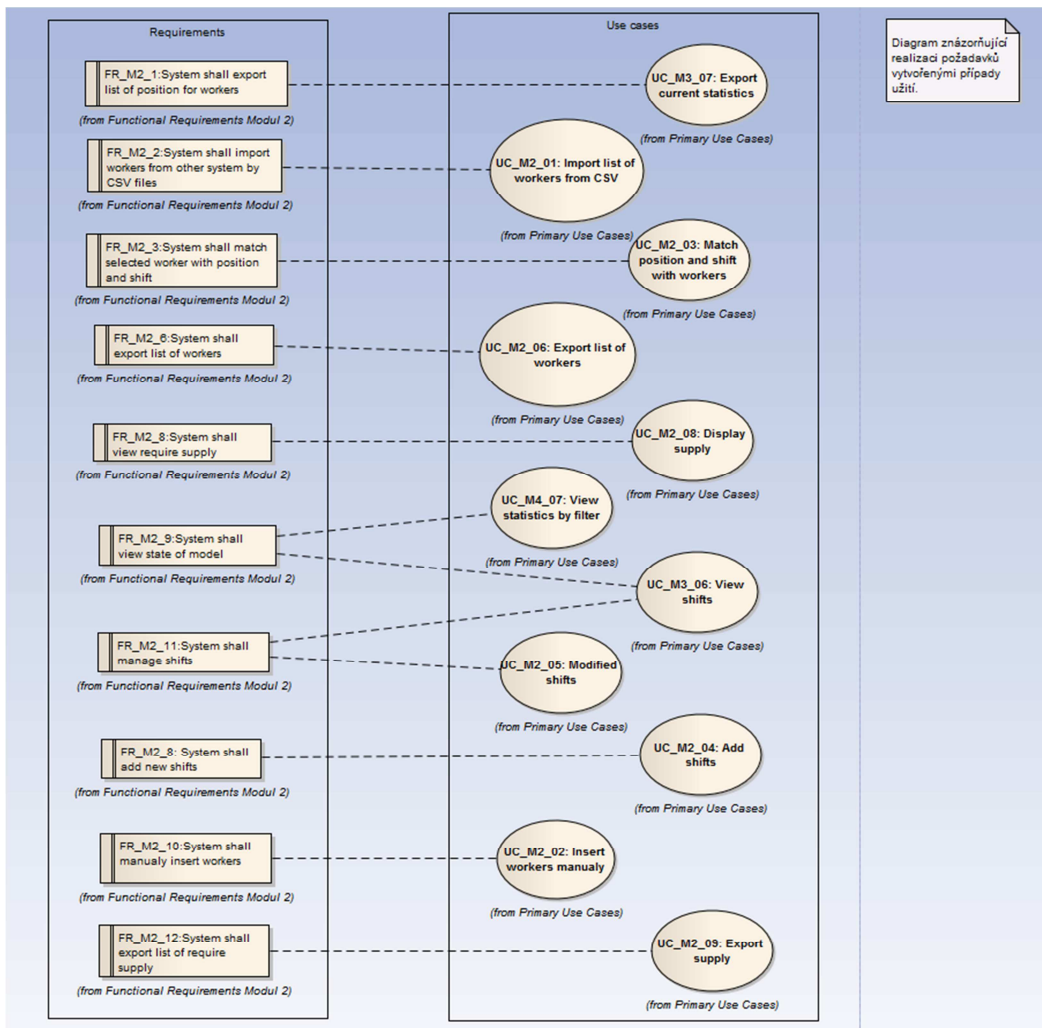
Obrázek 34 – Sekvenční diagram Modul 4

Sekvenční diagram reflektuje práci se čtvrtým modulem podobně jako na předcházejících diagramech. Při standardním scénáři se uvažuje nejprve práce Coordinátora, který předpřipraví statistiky, HR manažer si poté vyexportuje jednotlivá data do CSV formátu a po dokončení práce a kontrole Bezpečnostním manažerem je možné model uzavřít. Zde stejně jako v předchozích případech platí, že flow práce není nijak omežováno programem, a dodržení postupu je na uživatelích. Po celou dobu je možná vnější kontrole pomocí role pozorovatele.

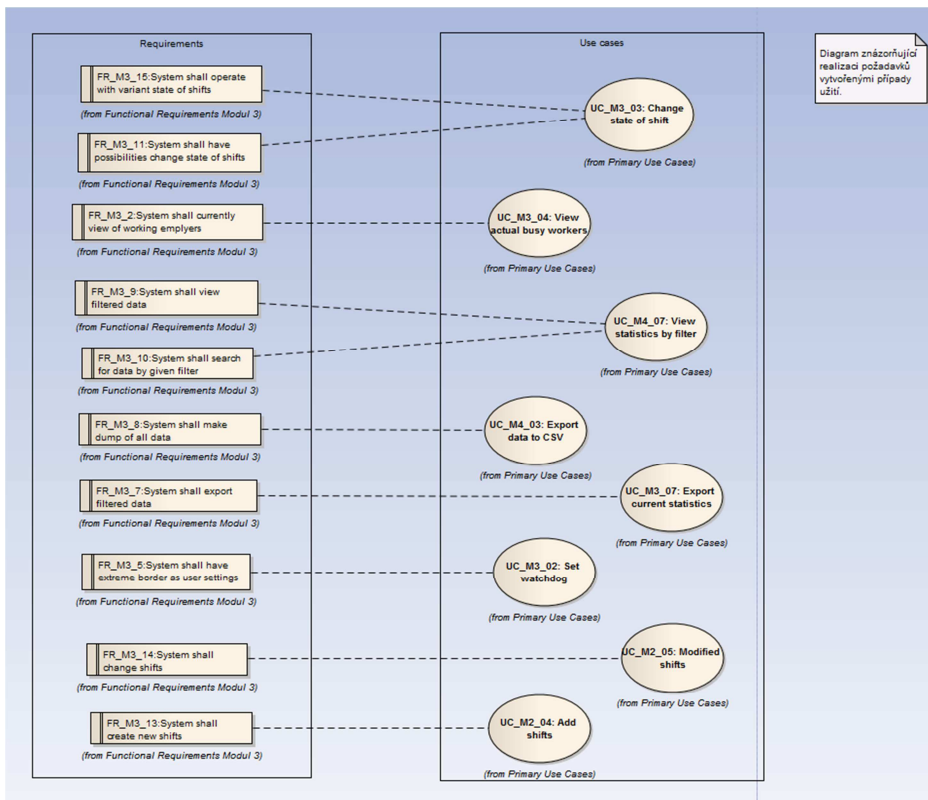
Příloha C – Realizace požadavků v rámci Use Case



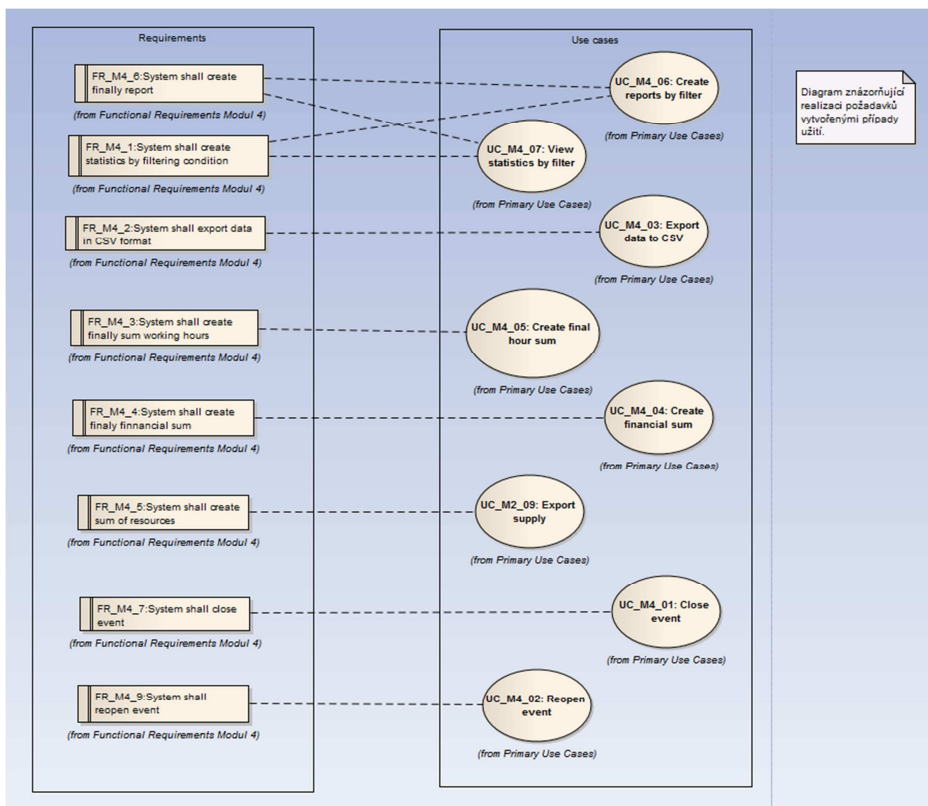
Obrázek 35 - Realizace požadavků na UC Modul 1



Obrázek 36 - Realizace Požadavků na UC Modul 2



Obrázek 37 - Realizace Požadavků na UC Modul 3



Obrázek 38 - Realizace Požadavků na UC Modul 4

Příloha D – Ukázka importovaných a exportovaných dat

Data pro import:

1;Alfonz;Mucha;+451256132;alfonz@mucha.cz;L
2;Jara;Cimrman;451121512;jarda@seznam.cz;XXL
3;Petr;Prucha;;nema;XS
4;Petr;Bily;+451853132;petrbily@seznam.cz;L
5;Jan;Jahod;45854612;jahojan@seznam.cz;XL
6;Lenka;Klusova;+4587413132;lenoolinka@gmail.com;L
7;Jarek;Novák;45854612;jarekjan@seznam.cz;XL
8;Dobroslav;Ičkovský;+4587414132;doboici@gmail.com;L
9;Pavel;Krtek;45854612;krtkuvdort@seznam.cz;XL
10;Hynek;Černý;+4587413132;hyncica@gmail.com;L
11;Ignác;Viktorij;45854612;ingac.viktorij@seznam.cz;XL
12;Bořivoj;Benediktov;+4587414132;borisben@gmail.com;L
13;Gustav;Otmarov;45854476;gustik@seznam.cz;XL
14;Frederik;Peporat;45146312;peper@seznam.cz;XL
15;Karel;Karlovič;+458873132;kayka@gmail.com;L

Příklad exportu informací o použitém materiálu

Název;Délka;Počet ks;Počet stanovišť;
Budova;0;2;2;
Ikona;0;14;14;
Mobilní plot;705;11;11;
Nášlapy;35;3;3;
Pódium;0;3;3;
Stánek;0;11;11;
Vstupy;0;5;1;

Příklad exportu finančního souhrnu pracovníků:

Jméno;Příjmení;Začátek;Konec;Kč / hod;Trvání;Suma za směnu;
Jara;Cimrman;22.8.2013 16:00:53;23.8.2013 2:00:53;80;10;2080;
Petr;Bily;22.8.2013 12:00:53;22.8.2013 18:00:53;80;6;480;
Pavel;Krtek;22.8.2013 18:00:53;23.8.2013 0:00:53;80;6;480;
Frederik;Peporat;23.8.2013 0:00:53;23.8.2013 6:00:53;80;6;480;
Celkem: ;;Kč: ;5760;Hodin: ;72;

Příloha E – Testovací scénáře

Sada testovacích scénářů pro ověření funkčnosti aplikace pro bezpečnostní agenturu

Testoval: Ondřej Serbousek

Datum: 15.8.2013-20.8.2013

Version: 1.0

Výsledky:

| Výsledky | |
|--------------|----|
| Netestováno: | 0 |
| Chyba: | 0 |
| Varování: | 0 |
| OK: | 23 |

| Kroky testů | |
|--------------|-----|
| Netestováno: | 0 |
| Chyba: | 0 |
| Varování: | 1 |
| OK: | 193 |

AS_01: Vytvoření uživatele

Test ID: AS01

Vytvoření nového uživatele

Výsledek: **Pass**

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|--|----------|----------|
| AS_01_01 | Krok | 1 | Přihlášení do systému jako administrátor | | Úspěšné přihlášení | Pass | |
| AS_01_02 | Krok | 2 | Zvolení nabídky "Správa uživatelů" | | Zobrazení nabídky správy uživatelů | Pass | |
| AS_01_03 | Krok | 3 | Zvolení možnosti "Nový uživatel" | | Zobrazení polí pro hodnoty nového uživatele. | Pass | |
| AS_01_04 | Krok | 4 | Vyplnění údajů pro nového uživatele. | | | Pass | |
| AS_01_05 | Krok | 5 | Uložení a potvrzení | | | Pass | |
| AS_01_06 | Krok | 6 | Odhlášení z aplikace | | Odlogování administrátora | Pass | |
| AS_01_07 | Ověření | 7 | Přihlášení do systému pod novým uživatelem | | Úspěšné přihlášení | Pass | |
| AS_01_08 | Ověření | 8 | Kontrola funkčnosti a přístupnosti UC v závislosti na roli nového uživatele. | | | Pass | |

AS_02: Změna uživatele

Test ID: AS02

Změna stávajícího uživatele

Výsledek: **Pass**

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|--|----------|----------|
| AS_02_01 | Krok | 1 | Přihlášení do systému jako administrátor | | Úspěšné přihlášení | Pass | |
| AS_02_02 | Krok | 2 | Zvolení nabídky "Správa uživatelů" | | Zobrazení nabídky správy uživatelů | Pass | |
| AS_02_03 | Krok | 3 | Vybrání existujícího uživatele ze seznamu. | | Zobrazení hodnot vybraného uživatele. | Pass | |
| AS_02_04 | Krok | 4 | Změna údajů vybraného uživatele | | | Pass | |
| AS_02_05 | Krok | 5 | Potvrzení a uložení | | Úspěšné uložení hodnot. | Pass | |
| AS_02_06 | Krok | 6 | Zobrazení uživatele | | Opětovné zobrazení měněného uživatele. | Pass | |
| AS_02_07 | Ověření | 7 | Kontrola nových hodnot. | | Nové uložení (změněné hodnoty) | Pass | |

AS_03: Založení nového modelu

Test ID: AS03

Založení nového modelu a jeho aktivace

Výsledek: **Pass**

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|--|----------|----------|
| AS_03_01 | Krok | 1 | Přihlášení do systému jako administrátor | | Úspěšné přihlášení | Pass | |
| AS_03_02 | Krok | 2 | Zvolení nabídky "Nová akce" | | Vyresetování vstupních políček | Pass | |
| AS_03_03 | Krok | 3 | Vypsání údajů pro novou akci | | | Pass | |
| AS_03_04 | Krok | 4 | Uložení | | Uložení do DB | Pass | |
| AS_03_05 | Ověření | 5 | Kontrola zobrazení nových hodnot | | Zobrazení nových hodnot | Pass | |
| AS_03_06 | Krok | 6 | Zvolení "Aktivuj model" | | Aktivace modelu | Pass | |
| AS_03_07 | Ověření | 7 | Kontrola zobrazení na kontroleru a čistých položek nové akce. | | Nový model označen jako aktivní + příslušné položky z DB | Pass | |

M1_01: Úprava pozadí akce

Test ID: M1_01

Vložení bitmapy pozadí a její editace

Vstupní podmínka: Založený model akce. Založená role "Security managera"

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|---|----------|----------|
| M1_01_01 | Krok | 1 | Přihlášení do systému jako "Security manager" | | Úspěšné přihlášení | Pass | |
| M1_01_02 | Krok | 2 | Aktivace modelu | | Aktivace modelu | Pass | |
| M1_01_03 | Krok | 3 | Zvolení karty "Modul 1" | | Zobrazení grafického editoru | Pass | |
| M1_01_04 | Krok | 4 | Spuštění příkazu "Vlož podklad" | | Spuštění nabídky výběru souboru | Pass | |
| M1_01_05 | Krok | 5 | Vybrání bitmapového obrázku | | | Pass | |
| M1_01_06 | Krok | 6 | Potvrzení | | Načtení bitmapového obrázku | Pass | |
| M1_01_07 | Ověření | 7 | Zobrazení obrázku na pozadí | | Zobrazení obrázku v grafickém editoru. | Pass | |
| M1_01_08 | Krok | 8 | Vložení značky "Stánek" | | Zobrazení značky | Pass | |
| M1_01_09 | Krok | 9 | Vložení značky "Plot" | | Zobrazení značky | Pass | |
| M1_01_10 | Krok | 10 | Vložení značky "Nášlapy" | | Zobrazení značky | Pass | |
| M1_01_11 | Krok | 11 | Vložení značky "Pódium" | | Zobrazení značky | Pass | |
| M1_01_12 | Ověření | 12 | Kontrola zobrazení značek na pozadí | | Korektní zobrazování všech vložených značek | Pass | |

M1_02: Vložení pozic a export bitmapy

Test ID: M1_02

Vložení uživatelských značek pozic a export do bitmapy

Vstupní podmínka: Založený model akce. Založená role "Security managera"

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|---|----------|----------|
| M1_02_01 | Krok | 1 | Přihlášení do systému jako "Security manager" | | Úspěšné přihlášení | Pass | |
| M1_02_02 | Krok | 2 | Aktivace modelu | | Aktivace modelu | Pass | |
| M1_02_03 | Krok | 3 | Zvolení karty "Modul 1" | | Zobrazení grafického editoru | Pass | |
| M1_02_04 | Krok | 4 | Spuštění příkazu "Vlož security" | | Aktivace příkazu "vlož značku" | Pass | |
| M1_02_05 | Krok | 5 | Klik do editoru - vložení značky security. | | Vložení značky - kolečko s výplní | Pass | |
| M1_02_06 | Krok | 6 | Spuštění příkazu "Vlož security", vybrání jiného typu z comboboxu | | Aktivace příkazu "vlož značku" | Pass | |
| M1_02_07 | Krok | 7 | Klik do editoru - vložení značky security. | | Vložení značky - kolečko s výplní jiné barvy | Pass | |
| M1_02_08 | Ověření | 8 | Kontrola rozdílných značek různých druhů security. | | | Pass | |
| M1_02_09 | Krok | 9 | Spuštění příkazu - vložení značky Hasič. | | Aktivace příkazu "vlož značku" | Pass | |
| M1_02_10 | Krok | 10 | Klik do editoru - vložení značky Hasič. | | Vložení značky - kolečko s výplní | Pass | |
| M1_02_11 | Ověření | 11 | Kontrola rozdílných značek různých druhů security a hasiče. | | | Pass | |
| M1_02_12 | Krok | 12 | Spuštění příkazu "Exportuj plánek" | | Zobrazení dialogu ukládání souborů | Pass | |
| M1_02_13 | Krok | 13 | Vybrání názvu souboru a umístění | | | Pass | |
| M1_02_14 | Krok | 14 | Potvrzení a uložení | | | Pass | |
| M1_02_15 | Ověření | 15 | Kontrola uloženého plánu v externím obrázkovém prohlížeči | | Zobrazení bitmapy obsahující všechny vložené značky | Pass | |

M1_03: Editace značek pozic a jejich odebrání

Test ID: M1_03

Editace existujících pozic "živých" značek a jejich odebrání.

Vstupní podmínka: Založený model akce. Založená role "Security managera", existující značky pozic v modelu.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|---|----------|----------|
| M1_03_01 | Krok | 1 | Přihlášení do systému jako "Security manager" | | Úspěšné přihlášení | Pass | |
| M1_03_02 | Krok | 2 | Aktivace modelu | | Aktivace modelu | Pass | |
| M1_03_03 | Krok | 3 | Zvolení karty "Modul 1" | | Zobrazení grafického editoru včetně značek | Pass | |
| M1_03_04 | Krok | 4 | Spuštění příkazu "Edituj pozice" | | Aktivace příkazu "Edituj značku" | Pass | |
| M1_03_05 | Krok | 5 | Přiblížení kurzorem nad existující značkou | | | Pass | |
| M1_03_06 | Krok | 6 | Drag & Drop nad vybranou značkou na novou pozici v plánu | | Přesunutí značky na novou pozici | Pass | |
| M1_03_07 | Krok | 7 | Uložení značek (příkaz Ulož do DB) | | Uložení značek do DB | Pass | |
| M1_03_08 | Ověření | 8 | Kontrola správné nové polohy značky | | | Pass | |
| M1_03_09 | Krok | 9 | Aktivace příkazu pro editaci značek | | Aktivace příkazu "Edituj značku" | Pass | |
| M1_03_10 | Krok | 10 | Přiblížení kurzorem nad existující značkou | | | Pass | |
| M1_03_11 | Krok | 11 | Klik pravým tlačítkem myši nad značkou | | Vyvolání dialogu odstranění značky | Pass | |
| M1_03_12 | Krok | 12 | Potvrzení odstranění značky | | | Pass | |
| M1_03_13 | Krok | 13 | Uložení značek do DB (Příkaz "Ulož do DB") | | Uložení značek do DB | Pass | |
| M1_03_14 | Ověření | 14 | Kontrola správnosti zobrazení | | Chybějící odstraněné značky a nová poloha přesunutých | Pass | |

M1_04: Zobrazení plánu a značek

Test ID: M1_03

Zobrazení plánu a značek, disablování tlačítek pro editace

Vstupní podmínka: Založený model akce. Založená role "Viewer", existující značky pozic v modelu, vložený plánek modelu.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|--|----------|----------|
| M1_04_01 | Krok | 1 | Přihlášení do systému jako "Viewer" | | Úspěšné přihlášení | Pass | |
| M1_04_02 | Krok | 2 | Aktivace modelu | | Aktivace modelu | Pass | |
| M1_04_03 | Krok | 3 | Zvolení karty "Modul 1" | | Zobrazení grafického editoru včetně značek | Pass | |
| M1_04_04 | Ověření | 4 | Kontrola zobrazení plánu | | Editor zobrazuje bitmapu na pozadí | Pass | |
| M1_04_05 | Ověření | 5 | Kontrola zobrazení značek | | Editor zobrazuje uživatelské značky | Pass | |
| M1_04_06 | Ověření | 6 | Kontrola znemožnění spuštění tlačítek pro editaci značek | | Není možné editovat vložené značky | Pass | |

M2_01: Import dat

Test ID: M2_01

Import a načtení dat ve formě CSV souboru

Vstupní podmínka: Založený model akce. Založená role "HR manažera", existující pozice v modelu, vstupní data

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|------|------|---|------------|---------------------------------------|----------|----------|
| M2_01_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M2_01_02 | Krok | 2 | Zvolení karty "Modul 2" | | Zobrazení GUI pro ovládání modulu 2 | Pass | |
| M2_01_03 | Krok | 3 | Zvolení příkazu "Import HEAD" | | Zobrazení dialogu pro výběr souboru | Pass | |
| M2_01_04 | Krok | 4 | Vybrání souboru s daty | import.csv | | Pass | |
| M2_01_05 | Krok | 5 | Potvrzení | | | Pass | |
| M2_01_06 | Krok | 6 | Kontrola zobrazení načtených pracovníků | | Nová data se zobrazují v GUI rozhraní | Pass | |

M2_02: Ruční vložení nových pracovníků

Test ID: M2_02

Ruční vložení dat nových pracovníků

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR manažera", existující pozice v modelu

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|------|------|--|------|--|----------|----------|
| M2_02_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M2_02_02 | Krok | 2 | Zvolení karty "Modul 2" | | Zobrazení GUI pro ovládání modulu 2 | Pass | |
| M2_02_03 | Krok | 3 | Vyresetování výběru pracovníka | | Reset vybraného pracovníka | Pass | |
| M2_02_04 | Krok | 4 | Zvolení příkazu "Přidej / Edituj pracovníka" | | Zobrazení dialogu pro nového pracovníka. | Pass | |
| M2_02_05 | Krok | 5 | Vyplnění dat nového pracovníka | | Vyplnění nových dat | Pass | |
| M2_02_06 | Krok | 6 | Potvrzení nového pracovníka | | | Pass | |
| M2_02_07 | Krok | 7 | Kontrola zobrazení nového pracovníka | | Zobrazení v GUI rozhraní | Pass | |

M2_03: Vložení nové směny a její editace

Test ID: M2_03

Vložení nové směny na pracovní pozici a její editace

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR manažera", existující pozice v modelu

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|--|----------|----------|
| M2_03_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M2_03_02 | Krok | 2 | Zvolení karty "Modul 2" | | Zobrazení GUI pro ovládání modulu 2 | Pass | |
| M2_03_03 | Krok | 3 | Zvolení pozice | | Načtení zvolené pozice | Pass | |
| M2_03_04 | Krok | 4 | Zadání časových úseků a zvolení příkazu "Přidej směnu" | | Reset vybraného pracovníka | Pass | |
| M2_03_05 | Ověření | 5 | Kontrola zobrazení nové směny | | Zobrazení dialogu pro nového pracovníka. | Pass | |
| M2_03_06 | Krok | 6 | Zvolení směny | | Vyplnění nových dat | Pass | |
| M2_03_07 | Krok | 7 | Editace časových úseků | | | Pass | |
| M2_03_08 | Krok | 8 | Zvolení příkazu "Uprav směnu" | | | Pass | |
| M2_03_09 | Ověření | 9 | Kontrola změněných údajů | | Zobrazení v GUI rozhraní | Pass | |

M2_04: Spárování pracovní směny s pracovníkem

Test ID: M2_04

Spárování existující směny na pozici s existujícím pracovníkem

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR manažera", existující pozice v modelu, existující směna, existující pracovník

Výsledek:

Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|-------------------------------------|----------|----------|
| M2_04_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M2_04_02 | Krok | 2 | Zvolení karty "Modul 2" | | Zobrazení GUI pro ovládání modulu 2 | Pass | |
| M2_04_03 | Krok | 3 | Zvolení pozice | | Načtení zvolené pozice | Pass | |
| M2_04_04 | Krok | 4 | Zvolení směny | | Načtení zvolené směny | Pass | |
| M2_04_05 | Krok | 5 | Zvolení pracovníka | | Vybrání pracovníka | Pass | |
| M2_04_06 | Krok | 6 | Vybrání pracovníka na akci | | Vybrání pracovníka na akci | Pass | |
| M2_04_07 | Krok | 7 | Zvolení příkazu "Přidej pracovníka na směnu" | | Přidání pracovníka na směnu | Pass | |
| M2_04_08 | Ověření | 8 | Kontrola vizuálního zobrazení pracovníka na směně. | | Zobrazení v GUI rozhraní | Pass | |

M2_05: Zobrazení pracovníků, pozic a směn

Test ID: M2_05

Zobrazení pracovníků, pozic a směn

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Viewera", existující pozice v modelu, existující směna, existující pracovník vybraný na akci s přidělenou směnou

Výsledek:

Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|-------------------------------------|----------|----------|
| M2_05_01 | Krok | 1 | Přihlášení do systému jako "Viewer" | | Úspěšné přihlášení | Pass | |
| M2_05_02 | Krok | 2 | Zvolení karty "Modul 2" | | Zobrazení GUI pro ovládání modulu 2 | Pass | |
| M2_05_03 | Ověření | 3 | Kontrola zobrazení pracovníků a pozic | | Načtení zvolené pozice | Pass | |
| M2_05_04 | Krok | 4 | Zvolení pozice | | Načtení zvolené směny | Pass | |
| M2_05_05 | Krok | 5 | Zvolení pracovníka | | Vybrání pracovníka | Pass | |
| M2_05_06 | Ověření | 6 | Kontrola zobrazení směn a pracovníků na směnách | | Vybrání pracovníka na akci | Pass | |

M3_01: Vyhledání a zobrazení směny a změna jejího stavu

Test ID: M3_01

Vyhledání a zobrazení směny, změna jejího stavu a uložení.

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Coordinatora", existující pozice v modelu, existující směny, naplánované a přiřazené.

Výsledek:

Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|-------------------------------------|----------|----------|
| M3_01_01 | Krok | 1 | Přihlášení do systému jako "Coordinátor" | | Úspěšné přihlášení | Pass | |
| M3_01_02 | Krok | 2 | Zvolení karty "Modul 3" | | Zobrazení GUI pro ovládání modulu 3 | Pass | |
| M3_01_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M3_01_04 | Krok | 4 | Vybrání jedné směny | | Aktivace jedné směny | Pass | |
| M3_01_05 | Krok | 5 | Přepnutí jejího stavu pomocí šipek | | Vizuální změna stavu směny | Pass | |
| M3_01_06 | Ověření | 6 | Kontrola změny stavu směny | | | Pass | |
| M3_01_07 | Krok | 7 | Změna stavu vybrané směny pomocí volby z comboboxu | | Vybrání hodnoty z comboboxu | Pass | |
| M3_01_08 | Krok | 8 | Potvrzení uložení | | Uložení nového stavu | Pass | |
| M3_01_09 | Ověření | 9 | Kontrola změny stavu směny | | Vizuální kontrola změny stavu | Pass | |

M3_02: Vyhledání směny dle filtru podle jména pracovníka a zobrazení jeho statistik

Test ID: M3_02

Vyhledání a zobrazení směny, podle jména pracovníka, kontrola zobrazování jeho statistik

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Coordinatora", existující pozice v modelu, existující směny, naplánované a přiřazené.

Výsledek:

Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|-------------------------------------|----------|----------|
| M3_02_01 | Krok | 1 | Přihlášení do systému jako "Coordinátor" | | Úspěšné přihlášení | Pass | |
| M3_02_02 | Krok | 2 | Zvolení karty "Modul 3" | | Zobrazení GUI pro ovládání modulu 3 | Pass | |
| M3_02_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M3_02_04 | Krok | 4 | Zadání jména hledaného pracovníka do filtru | | Vyplnění jména pracovníka | Pass | |
| M3_02_05 | Krok | 5 | Potvrzení vyhledání filtrem | | Vyhledání výsledků | Pass | |
| M3_02_06 | Ověření | 6 | Kontrola nalezených výsledků | | Zobrazení výsledků | Pass | |
| M3_02_07 | Ověření | 7 | Kontrola zobrazených statistik pracovníka - seznam směn | | Kontrola seznamu směn | Pass | |
| M3_02_08 | Ověření | 8 | Přepočtová kontrola hodinového souhrnu | | Celkový hodinový součet | Pass | |
| M3_02_09 | Ověření | 9 | Kontrola přepočtu hodin z časových úseků | | Výpočet délky směn z časových úseků | Pass | |

M3_03: Kontrola a nastavení watchdog funkce hlídání délky směn a přestávek

Test ID: M3_03

Nastavení a kontrola mechanismu pro hlídání extrémních směn a pauz. Nastavení extrémů je uživatelsky definované.

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Supervisor", existující pozice v modelu, existující směny, naplánované a přiřazené s extrémními časovými úseky, extrémní délka směny a přestávky

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|-------------------------------------|----------|-----------------------------------|
| M3_03_01 | Krok | 1 | Přihlášení do systému jako "Supervisor" | | Úspěšné přihlášení | Pass | |
| M3_03_02 | Krok | 2 | Zvolení karty "Modul 3" | | Zobrazení GUI pro ovládání modulu 3 | Pass | |
| M3_03_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M3_03_04 | Krok | 4 | Zadání jména hledaného pracovníka do filtru | | Vyplnění jména pracovníka | Pass | |
| M3_03_05 | Krok | 5 | Potvrzení vyhledání filtrem | | Vyhledání výsledků | Pass | |
| M3_03_06 | Ověření | 6 | Kontrola nalezených výsledků | | Zobrazení výsledků | Pass | |
| M3_03_07 | Ověření | 7 | Zadání hranic pro vyhledání extrémních směn a přestávek | | Zadání nových hodnot pro extrém | Pass | |
| M3_03_08 | Ověření | 8 | Kontrola nalezených výsledků | | Kontrola seznamu extrémů | Warning | Chybné formátování časových úseků |
| M3_03_09 | Ověření | 9 | Kontrola přepočtu hodin z časových úseků | | Výpočet délky směn z časových úseků | Pass | |

M3_04: Zobrazení aktuálně pracujících pracovníků

Test ID: M3_04

Kontrola zobrazení aktuálně pracujících lidí na akci.

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Security managera", existující pozice v modelu, existující směny, v různých stavech včetně "ve výkonu".

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|-------------------------------------|----------|----------|
| M3_04_01 | Krok | 1 | Přihlášení do systému jako "Security manager" | | Úspěšné přihlášení | Pass | |
| M3_04_02 | Krok | 2 | Zvolení karty "Modul 3" | | Zobrazení GUI pro ovládání modulu 3 | Pass | |
| M3_04_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M3_04_04 | Krok | 4 | Vybrání možnosti "Ve výkonu" z comboboxu typů směn | | Vybrání typu stavu směny | Pass | |
| M3_04_05 | Krok | 5 | Spuštění příkazu "Zobrazit ve filtru" | | Vyhledání výsledků | Pass | |
| M3_04_06 | Ověření | 6 | Kontrola nalezených výsledků | | Zobrazení výsledků | Pass | |

M3_04: Test role Viewera pro modul 3

Test ID: M3_05

Kontrola zobrazení dat pro roli Viewera

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Viewer", existující pozice v modelu, existující směny, v různých stavech.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|-------------------------------------|------|-------------------------------------|----------|----------|
| M3_05_01 | Krok | 1 | Přihlášení do systému jako "Viewer" | | Úspěšné přihlášení | Pass | |
| M3_05_02 | Krok | 2 | Zvolení karty "Modul 3" | | Zobrazení GUI pro ovládání modulu 3 | Pass | |
| M3_05_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M3_05_04 | Krok | 4 | Spuštění filtru se zadanými údaji | | Zobrazení filtru dle zadaných údajů | Pass | |
| M3_05_05 | Ověření | 5 | Kontrola nalezených výsledků | | Kontrola výsledků | Pass | |
| M3_05_06 | Ověření | 6 | Kontrola nalezených statistik | | Kontrola výsledků | Pass | |

M4_01: Zobrazení statistik dle filtru

Test ID: M4_01

Zobrazení statistik dat v aplikaci

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR Manager", existující pozice v modelu, existující směny, v různých stavech. Simulace stavu po skončení akce.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|--|----------|----------|
| M4_01_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M4_01_02 | Krok | 2 | Zvolení karty "Modul 4" | | Zobrazení GUI pro ovládání modulu 4 | Pass | |
| M4_01_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M4_01_04 | Ověření | 4 | Kontrola nalezených výsledků | | Zobrazení listu směn | Pass | |
| M4_01_05 | Krok | 5 | Zadání jména hledaného pracovníka do filtru | | Vyplnění jména pracovníka | Pass | |
| M4_01_06 | Ověření | 6 | Potvrzení vyhledání filtrem | | Zobrazení výsledků | Pass | |
| M4_01_07 | Ověření | 7 | Kontrola zobrazených statistik pracovníka - seznam směn | | Kontrola seznamu směn vybraného pracovníka | Pass | |

M4_02: Export statistik dle filtru

Test ID: M4_02

Export statistik dat z aplikace do formátu CSV

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR Manager", existující pozice v modelu, existující směny, v různých stavech. Simulace stavu po skončení akce.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|-------------------------------------|----------|----------|
| M4_02_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M4_02_02 | Krok | 2 | Zvolení karty "Modul 4" | | Zobrazení GUI pro ovládání modulu 4 | Pass | |
| M4_02_03 | Krok | 3 | Zobrazení kompletního listu směn | | Zobrazení listu směn | Pass | |
| M4_02_04 | Ověření | 4 | Kontrola nalezených výsledků | | | Pass | |
| M4_02_05 | Krok | 5 | Spuštění příkazu "Exportuj dle filtru" | | Zobrazení dialogu ukládání souboru | Pass | |
| M4_02_06 | Krok | 6 | Vybrání místa uložení a názvu souboru | | | Pass | |
| M4_02_07 | Ověření | 7 | Potvrzení exportu + kontrola | | Vytvořený CSV soubor | Pass | |

M4_03: Zobrazení a export materiálu

Test ID: M4_03

Zobrazení a export dat použitého materiálu z aplikace do formátu CSV

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Security Manager", Existující data o použitém materiálu v modelu. Simulace stavu po skončení akce.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|-------------------------------------|----------|----------|
| M4_03_01 | Krok | 1 | Přihlášení do systému jako "Security manager" | | Úspěšné přihlášení | Pass | |
| M4_03_02 | Krok | 2 | Zvolení karty "Modul 4" | | Zobrazení GUI pro ovládání modulu 4 | Pass | |
| M4_03_03 | Krok | 3 | Zobrazení seznamu použitého materiálu | | Zobrazení seznamu materiálu | Pass | |
| M4_03_04 | Ověření | 4 | Kontrola nalezených výsledků | | | Pass | |
| M4_03_05 | Krok | 5 | Spuštění příkazu "Exportuj Materiál" | | Zobrazení dialogu ukládání souboru | Pass | |
| M4_03_06 | Krok | 6 | Vybrání místa uložení a názvu souboru | | | Pass | |
| M4_03_07 | Ověření | 7 | Potvrzení exportu + kontrola | | Vytvořený CSV soubor | Pass | |

M4_04: Zobrazení a export finančního souhrnu

Test ID: M4_04

Zobrazení a export vypočteného finančního souhrnu z aplikace do formátu CSV

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR Manager", existující pozice v modelu, existující směny, v konečných stavech. Simulace stavu po skončení akce.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|--|----------|----------|
| M4_04_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M4_04_02 | Krok | 2 | Zvolení karty "Modul 4" | | Zobrazení GUI pro ovládání modulu 4 | Pass | |
| M4_04_03 | Krok | 3 | Zobrazení finančního souhrnu | | Zobrazení finančního souhrnu | Pass | |
| M4_04_04 | Ověření | 4 | Kontrola nalezených výsledků | | | Pass | |
| M4_04_05 | Krok | 5 | Zadání jména existujícího pracovníka do filtru | | Vyplněné pole jména pracovníka | Pass | |
| M4_04_06 | Krok | 6 | Zobrazení finančního souhrnu pro jednoho pracovníka. | | Zobrazení finančního souhrnu pro určitého pracovníka | Pass | |
| M4_04_07 | Ověření | 7 | Kontrola nalezených výsledků | | | Pass | |
| M4_04_08 | Krok | 8 | Spuštění příkazu "Exportuj finanční souhrn" | | Zobrazení dialogu ukládání souboru | Pass | |
| M4_04_09 | Krok | 9 | Vybrání místa uložení a názvu souboru | | | Pass | |
| M4_04_10 | Ověření | 10 | Potvrzení exportu + kontrola | | Vytvořený CSV soubor | Pass | |

M4_05: Zobrazení a export extrémních směn

Test ID: M4_05

Zobrazení a export extrémních délek směn z aplikace do formátu CSV

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "HR Manager", existující pozice v modelu, existující směny s extrémními parametry, nastavené extrémní parametry. Simulace stavu po skončení akce.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|--|------|-------------------------------------|----------|----------|
| M4_05_01 | Krok | 1 | Přihlášení do systému jako "HR manager" | | Úspěšné přihlášení | Pass | |
| M4_05_02 | Krok | 2 | Zvolení karty "Modul 4" | | Zobrazení GUI pro ovládání modulu 4 | Pass | |
| M4_05_03 | Krok | 3 | Zvolení výběru min/max délka směny | | Zobrazení extrémních směn | Pass | |
| M4_05_04 | Krok | 4 | Spuštění příkazu "Zobraz extrémní směny" | | Zobrazení extrémních směn | Pass | |
| M4_05_05 | Ověření | 5 | Kontrola nalezených výsledků | | | Pass | |
| M4_05_06 | Krok | 6 | Přepnutí výběru min/max délka směny | | Vizuální zobrazení radiobuttonu | Pass | |
| M4_05_07 | Krok | 7 | Spuštění příkazu "Zobraz extrémní směny" | | Zobrazení extrémních směn | Pass | |
| M4_05_08 | Ověření | 8 | Kontrola nalezených výsledků | | | Pass | |
| M4_05_09 | Krok | 9 | Spuštění příkazu "Exportuj extrémní směny" | | Zobrazení dialogu ukládání souboru | Pass | |
| M4_05_10 | Krok | 10 | Vybrání místa uložení a názvu souboru | | | Pass | |
| M4_05_11 | Krok | 11 | Potvrzení exportu | | Vytvořený CSV soubor | Pass | |
| M4_05_12 | Ověření | 12 | Kontrola exportovaného CSV souboru | | Data v CSV formátu | Pass | |

M4_06: Uzavření a otevření akce

Test ID: M4_06

Test uzavření a otevření akce

Vstupní podmínka: Založený a aktivovaný model akce. Založená role "Security managera", existující pozice v modelu, existující směny, v konečných stavech. Simulace stavu po skončení akce.

Výsledek: Pass

| ID | Typ | Krok | Akce | Data | Očekávaný výsledek | Výsledek | Poznámka |
|----------|---------|------|---|------|--|----------|----------|
| M4_04_01 | Krok | 1 | Přihlášení do systému jako "Security manager" | | Úspěšné přihlášení | Pass | |
| M4_04_02 | Krok | 2 | Zvolení karty "Modul 4" | | Zobrazení GUI pro ovládání modulu 4 | Pass | |
| M4_04_03 | Krok | 3 | Zvolení příkazu "Uzavři akci" | | Uzavření modelu akce | Pass | |
| M4_04_04 | Ověření | 4 | Test všech tabů aplikace, kontrola zobrazení | | Deaktivace edit a create příkazů a UC | Pass | |
| M4_04_05 | Krok | 5 | Zvolení Domovské karty | | Zobrazení domovské obrazovky | Pass | |
| M4_04_06 | Krok | 6 | Zvolení příkazu "Otevři akci" | | Otevření modelu akce | Pass | |
| M4_04_07 | Ověření | 7 | Test všech tabů aplikace, kontrola zobrazení | | Přístupnost edit a create příkazů a UC | Pass | |