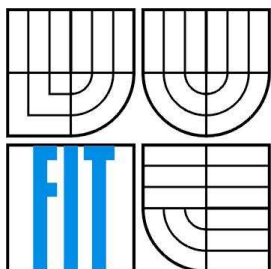




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
ÚSTAV INFORMAČNÍCH SYSTÉMŮ  
FACULTY OF INFORMATION TECHNOLOGY  
DEPARTMENT OF INFORMATION SYSTEMS

# ANALÝZA A NÁVRH INFORMAČNÍHO SYSTÉMU ELEKTRONICKÉHO VZDĚLÁVÁNÍ

ANALYSIS AND DESIGN OF E-LEARNING INFORMATION SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. JAN ŠTACHA

VEDOUCÍ PRÁCE

SUPERVISOR

RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2007

## **Abstrakt**

Práce se podrobně zabývá metodikou IBM Rational Unified Process, která představuje komplexní a robustní přístup k vývoji a životnímu cyklu software. Tato metodika je velmi dobře zdokumentována a každý krok životního cyklu vývoje lze dopředu předpovědět, proto se stává postupně standardem mnoha organizací vyvíjejících komerční software.

Cílem práce je podrobný popis této metodiky a vypracování obvyklých výstupů fáze zahájení a fáze rozpracování při vytváření informačního systému elektronického vzdělávání.

Vzhledem k tomu, že IBM Rational Unified Process je tzv. use-case driven přístupem, je v práci kladen největší důraz na podrobný popis jednotlivých případů užití.

## **Klíčová slova**

IBM Rational Unified Process, UML, životní cyklus software, případy použití, analytický model, model návrhu, model nasazení, plánování projektu, dokument vize, metodika vývoje software, architektura systému, informační systém, analýza, návrh

## **Abstract**

This thesis is focused on IBM Rational Unified Process methodology, which represents complex and robust approach to software development and software lifecycle. This methodology is well described and every step of software lifecycle is predictable. That's the reason why it is becoming used in many software development organizations.

The main goal of this thesis is deep description of this methodology and creation common outputs of inception and elaboration phase of e-learning information system.

IBM Rational Unified Process is called use-case driven approach, that's the reason why it is emphasized description of all use-cases.

## **Keywords**

IBM Rational Unified Process, UML, software lifecycle, use-cases, analytical model, structural model, deployment model, project planning, the vision document, software development methodology, system architecture, information system, analysis, design

## **Citace**

ŠTACHA JAN: Analýza a návrh systému elektronického vzdělávání. Brno, 2007, diplomová práce, FIT VUT v Brně.

# **Analýza a návrh informačního systému elektronického vzdělávání**

© Bc. Jan Štacha, 2006

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

## **Prohlášení**

Prohlašuji, že jsem tuto práci vypracoval samostatně pod odborným vedením mého konzultanta ze společnosti Unicorn a.s., pana Ing. Marka Beránka.

Další informace mi poskytla paní RNDr. Jitka Kreslíková, CSc.

Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Brně dne 22.5.2006

Jan Štacha

# Obsah

<b>ÚVOD A CÍL PRÁCE</b> .....	<b>3</b>
<b>1 SEZNÁMENÍ S IBM RATIONAL UNIFIED PROCESS</b> .....	<b>4</b>
1.1. RATIONAL UNIFIED PROCESS .....	4
1.2. STRUČNÁ HISTORIE RATIONAL UNIFIED PROCESS.....	5
1.3. NEJLEPŠÍ PRAKTIKY SOFTWAREVÉHO VÝVOJE .....	5
1.3.1. Iterativní vývoj.....	5
1.3.2. Správa požadavků.....	6
1.3.3. Komponentová architektura.....	6
1.3.4. Vizuelní modelování .....	7
1.3.5. Ověřování kvality.....	7
1.3.6. Řízení změn .....	7
1.4. JAZYK UNIFIED MODELLING LANGUAGE .....	7
<b>2 ŽIVOTNÍ CYKLUS A CHARAKTERISTIKA FÁZÍ</b> .....	<b>9</b>
2.1. ŽIVOTNÍ CYKLUS PROJEKTU V RUP.....	9
2.1.1. Fáze zahájení.....	9
2.1.2. Fáze rozpracování .....	10
2.1.3. Fáze konstrukce .....	10
2.1.4. Fáze předání .....	10
2.2. MILNÍKY JEDNOTLIVÝCH FÁZÍ.....	11
2.2.1. Rozsah systému .....	11
2.2.2. Definice architektury .....	11
2.2.3. Beta-verze .....	12
2.2.4. Nasazení.....	12
<b>3 ZÁKLADNÍ MODEL RUP A JEHO DISCIPLÍNY</b> .....	<b>13</b>
3.1. ZÁKLADNÍ MODEL.....	13
3.2. DISCIPLÍNY RUP .....	14
3.2.1. Obchodní modelování (Business modelling).....	15
3.2.2. Správa požadavků.....	15
3.2.3. Analýza a návrh.....	15
3.2.4. Implementace.....	16
3.2.5. Testování.....	16
3.2.6. Dodání a nasazení .....	17
<b>4 ARCHITEKTURA SYSTÉMU V RUP</b> .....	<b>18</b>
4.1. PRACOVNÍ DEFINICE.....	18
4.2. MODEL ARCHITEKTURY 4+1 .....	18
4.2.1. Případy užití.....	18
4.2.2. Logický pohled.....	19
4.2.3. Procesní pohled .....	19
4.2.4. Implementační pohled.....	19
4.2.5. Pohled nasazení.....	19
<b>5 FÁZE ZAHÁJENÍ</b> .....	<b>20</b>
5.1. DOKUMENT VIZE.....	20
5.1.1. Úvod.....	20
5.1.2. Rozsah vypracování .....	20

5.1.3.	<i>Cíl a rozsah projektu</i> .....	20
5.1.4.	<i>Demografická specifikace</i> .....	20
5.1.5.	<i>Vymezení problému</i> .....	21
5.1.6.	<i>Vymezení produktu</i> .....	21
5.1.7.	<i>Přehled zainteresovaných osob</i> .....	22
5.1.8.	<i>Prostředí uživatelů</i> .....	22
5.1.9.	<i>Profily uživatelů</i> .....	23
5.2.	POČÁTEČNÍ OBCHODNÍ PŘÍPAD .....	26
5.2.1.	<i>Perspektiva systému</i> .....	26
5.2.2.	<i>Přehled vlastností systému</i> .....	26
5.3.	USE-CASE MODEL SYSTÉMU .....	27
5.3.1.	<i>Globální pohled na systém</i> .....	27
5.3.2.	<i>Správa kurzů z pohledu zaměstnance</i> .....	28
5.3.3.	<i>Prohlížení podrobných informací o kurzu z pohledu zaměstnance</i> .....	32
5.3.4.	<i>Správa kurzů z pohledu lektora</i> .....	36
5.3.5.	<i>Ostatní případy užití z pohledu lektora</i> .....	43
5.3.6.	<i>Případy použití z pohledu garanta</i> .....	48
5.3.7.	<i>Případy použití z pohledu manažera</i> .....	53
5.3.8.	<i>Případy použití z pohledu administrátora</i> .....	58
5.3.9.	<i>Společné případy použití pro všechny aktéry</i> .....	63
5.4.	ÚVODNÍ GLOSÁŘ .....	65
5.5.	PŘEDBĚŽNÝ PLÁN ITERACÍ .....	66
5.5.1.	<i>Přehled funkcí realizovaných v jednotlivých etapách</i> .....	66
5.5.2.	<i>Výstupy realizace a její časový harmonogram</i> .....	66
5.6.	POČÁTEČNÍ OHODNOCENÍ RIZIK .....	68
<b>6</b>	<b>FÁZE ROZPRACOVÁNÍ</b> .....	<b>69</b>
6.1.	DODATEČNÉ POŽADAVKY NA SYSTÉM.....	69
6.1.1.	<i>Rozsah systému</i> .....	69
6.1.2.	<i>Reference</i> .....	69
6.1.3.	<i>Použitelnost</i> .....	69
6.1.4.	<i>Spolehlivost</i> .....	69
6.1.5.	<i>Výkonnost</i> .....	70
6.1.6.	<i>Omezení architektury</i> .....	70
6.2.	POPIS SOFTWARE ARCHITEKTURY .....	70
6.2.1.	<i>Logický pohled</i> .....	70
6.2.2.	<i>Procesní pohled</i> .....	80
6.2.3.	<i>Pohled nasazení</i> .....	80
6.2.4.	<i>Implementační pohled</i> .....	81
6.3.	REVIDOVANÝ SEZNAM RIZIK .....	82
6.4.	PROTOTYP SYSTÉMU .....	83
6.5.	VÝVOJOVÝ PLÁN PROJEKTU .....	83
	<b>ZÁVĚR</b> .....	<b>86</b>
	<b>SEZNAM POUŽITÉ LITERATURY</b> .....	<b>87</b>
	<b>SEZNAM PŘÍLOH</b> .....	<b>88</b>

# Úvod a cíl práce

V současné době je software vyvíjen pod stále větším tlakem zadavatelů na rychlost jejich uvedení do provozu, zároveň s požadavky na udržení a zvyšování kvality. Navíc rostou nároky na integraci aplikací s již současnými firemními systémy, se kterými musí aplikace komunikovat. Jedním ze základních požadavků managementu zadavatelských firem je také snadná a relativně levná údržba.

Zkušenosti z praxe ukazují, že u větších projektů nelze tyto nároky uspokojit tradičním sekvenčním přístupem, který před sebou obvykle tlačí často dosud netušená rizika, jejichž odstranění je postupem času obvykle velmi nákladné. Jedním z řešení je použití iterativního a inkrementálního přístupu k vývoji software, který vychází z Boehmova spirálového modelu a dochází u něj k průběžné detekci rizik. Mezi výhody tohoto přístupu lze zařadit i snazší správu změn, lepší znovupoužitelnost, pevnější vazbu mezi zadavatelem a řešitelem, atd.

Před tím, než je vůbec možné vytvořit software pro řešení nějakého problému, je nutné zjistit všechny důležité údaje o daném problému. Jedná se tedy o popis přesně vymezené části reality, která nás zajímá, která bude obsahem aplikace. K tomuto popisu se v průběhu vývoje v oblasti analýzy a návrhu informačních systémů vyvinulo mnoho modelů, nástrojů a postupů, které pak byly postupem času standardizovány. V současnosti tedy máme komplexní systém modelovacích nástrojů a pravidel, s využitím kterých je možné pracovat, aniž bychom museli vymýšlet vlastní metodiku vývoje.

Tato diplomová práce se zabývá metodikou IBM Rational Unified Process (dále jen RUP), která je v současnosti v praxi nejpoužívanější metodikou.

Cílem práce je popis metodiky RUP, včetně vypracování všech výstupů fáze zahájení a fáze rozpracování informačního systému elektronického vzdělávání, který by měl uspokojit potřeby interního vzdělávání zaměstnanců ve firmě Unicorn a.s., největší české firmě poskytující komplexní služby v oblasti informačních systémů a informačních a komunikačních technologií.

Práce je stavěna logickým způsobem a každá kapitola navazuje na předchozí kapitolu. Abstrahuje záměrně od technických detailů.

V první kapitole obecně popisují metodiku RUP, dále uvádím historické aspekty jejího vzniku, zmiňuji nejlepší praktiky softwarového vývoje, které byly motivací vzniku metodiky, a dále popisují jazyk UML jako její hlavní výrazový prostředek.

Ve druhé kapitole jsou popsány jednotlivé fáze životního cyklu, plánovaného pomocí metodiky RUP, včetně popisu jejich milníků.

Ve třetí kapitole je popsán základní model RUP, jeho základní prvky a disciplíny, které jsou spojené s projektem.

Čtvrtá kapitola definuje architekturu a popisuje jednotlivé části modelu architektury 4+1 využívaného v RUP.

Pátá kapitola je vypracováním výstupů v rámci dříve popsané fáze zahájení. Konkrétně se jedná o vizi, počáteční obchodní případ, model případů použití, počáteční glosář a počáteční ohodnocení rizik.

Šestá kapitola je vypracováním výstupů fáze rozpracování, kde jsou výstupy tvořeny dodatečnými požadavky na systém, popisem architektury, revidovaným seznamem rizik a vývojovým plánem projektu. Dále je v rámci této fáze vypracován prototyp systému určený k testování architektury a jeho instalační příručka.

Předmětem semestrálního projektu předcházejícímu této práci byly první čtyři hlavní kapitoly práce, ve kterých jsem shrnul základní informace o metodice RUP. Tyto kapitoly jsem v diplomové práci dále zpřesnil.

# 1 Seznámení s IBM Rational Unified Process

## 1.1. Rational Unified Process

Metodika IBM Rational Unified Process® neboli RUP® je flexibilní platformou pro proces vývoje software. Díky její přizpůsobitelné architektuře umožňuje volit jednotlivé komponenty dle specifických potřeb v jednotlivých fázích konkrétního projektu.[13]

Z tohoto pohledu můžeme RUP považovat jako framework (obr.1) pro určování úkolů a odpovědnosti v příslušné organizaci, jehož hlavním cílem je zajistit tvorbu produktů, které budou vyhovovat všem požadavkům zadavatele, zároveň bude tento produkt dodán včas a v odpovídající kvalitě bez překročení celkových nákladů.

RUP je relativně mladá metodika, podnětem pro její vznik byla neúspěšná bilance úspěšnosti softwarových projektů. Metodika je založena na rozsáhlých praktických poznatcích z mnoha renomovaných firem. Analýzou nalezených příčin neúspěšnosti zkoumaných projektů byly identifikovány postupy a opatření umožňující efektivně tyto příčiny eliminovat nebo alespoň omezit jejich dopad. Tato opatření byla shrnuta do šesti ucelených souborů doporučení, které se nazývají Nejlepší praktiky softwarového vývoje. RUP zahrnuje zapracování šesti nejlepších praktik do konkrétních návodů a postupů. Metodika jako taková pokrývá globálně proces softwarového vývoje a současně poskytuje návody a doporučení na detailní úrovni. Při zpracování a pro implementaci metodických postupů uživatelé poskytují i odpovídající konstrukce vycházející ze standardního jazyka pro modelování – Unified Modelling Language (dále jen UML). [1]



obr. 1 – Pohled na Rational Unified Process jako na framework

*zdroj: převzato z [4]*

## 1.2. Stručná historie Rational Unified Process

Roku 1988 byla Ivarem Jacobsonem z firmy Objectory AB definována metodika Objectory v1.0, ze které se metodika firmy Rational přímo vyvinula a v roce 1995 tyto firmy fúzí. (obr.2)

V roce 1996 byl vytvořen Rational Objectory Process (ROP) v4.0, ve kterém firma Rational rozšířila metodiku Objectory o iterativní přístup. [16]

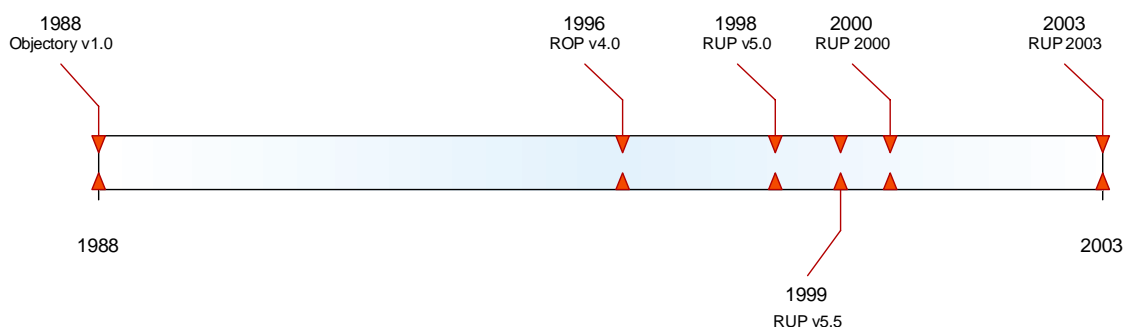
V květnu 1998 vzniká Rational Unified Process (RUP) v5.0, což je ve skutečnosti přejmenovaný ROP, rozšířený o další prvky procesu modelování ze zakoupených společností a také o materiál připravený vývojovou skupinou vedenou Phillipe Kruchtenem. Tato verze popisuje aplikaci UML diagramů při vývoji objektově orientovaných systémů.

O rok později v květnu 1999 je vypuštěna RUP v5.5. Hlavní vylepšení spočívají v rozšíření metodiky o vývoj real-time a webových aplikací.

RUP 2000 byla vypuštěna v únoru roku 2000. Hlavní vylepšení tkví v přidání technik podnikového návrhu, podnikového modelování a daleko užšímu přístupu k požadavkům.

Srpen 2002 zakoupila firma IBM vývojáře tzv. Summit methodology, firmu PriceWaterhouseCooper (PWC) tato metodologie je také později zahrnuta do RUP.

Prosinec 2002 IBM zakoupila i Rational Corporation a do května 2003 vydala novou verzi RUP (2003), kterou obohatila o některé koncepty agilního vývoje softwaru a také vylepšila metodologii testování. [14]



obr. 2 – přehled vývoje a verzí RUP

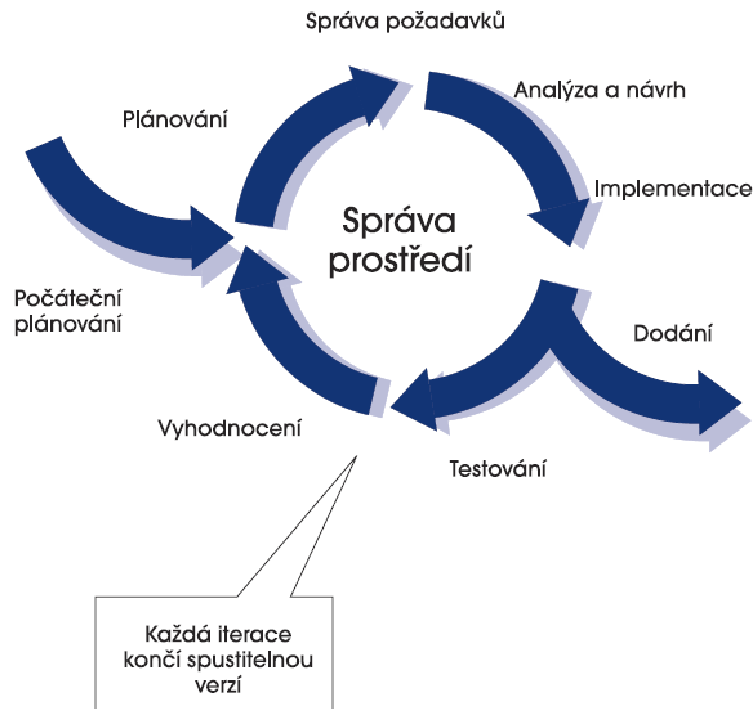
## 1.3. Nejlepší praktiky softwarového vývoje

### 1.3.1. Iterativní vývoj

U dnešních rozsáhlých systémů již není možné na začátku přesně definovat celý problém, navrhnout řešení, provést implementaci a až na závěr, kdy je spotřebována většina přidělených finančních prostředků, celý systém otestovat. Iterativní přístup spočívá v rozdělení celého projektu na čtyři fáze, z nichž každá se skládá z několika iterací se životním cyklem typu „vodopád“. Výsledkem každé této iterace je spustitelná verze.

Iterativní vývoj (obr.3) představuje konkurenční výhodu pro řídicí management. V rámci iterativního vývoje se lze soustředit přímo na klíčové problémy a průběžně odstraňovat rizika. Mezi výhody taktéž patří možnost objektivního posouzení stavu projektu a včasné odhalení nesrovnalostí mezi požadavky, návrhem a implementací. Pracovní zatížení je rovnoměrněji rozloženo. [1]





obr. 3 – Iterativní vývoj s přírůstky

*zdroj: převzato z [1]*

### 1.3.2. Správa požadavků

Požadavek je definován jako schopnost, kterou musí systém splňovat (syndrom IKIWISI – I Will Know It When I See It – „budu to vědět, až to uvidím“, s nadsázkou vyjádřená neschopnost zákazníka specifikovat dopředu všechny požadavky na vyvíjený produkt). [1] Sběr požadavků probíhá hned na začátku vývoje. Před započítím jakékoli další činnosti musí být shromážděny všechny požadavky, které jsou následně vyhodnoceny a zpracovány do projektové dokumentace. V rámci správy požadavků je nutné zjistit požadovanou funkčnost, zpracovat detailní dokumentaci, průběžně vyhodnocovat změny požadavků a jejich důsledky a dokumentovat jednotlivá rozhodnutí.

### 1.3.3. Komponentová architektura

Komponentová architektura spolu s iterativním vývojem softwaru přispívá k postupné tvorbě systémové architektury. Takový přístup usnadňuje identifikaci rizik a jejich odstranění již v samotném procesu vývoje. Softwarovou komponentu lze definovat jako netriviální část software (modul, balíček či subsystém), která plní určitou funkci, má jasné hranice a může být součástí definované architektury. [1]

Proces RUP poskytuje systematický přístup k definování architektury využívající nové, či již existující komponenty.

Tyto komponenty jsou vzájemně propojovány v rámci korektně definované architektury buď případ od případu (ad-hoc) nebo prostřednictvím komponentní architektury využívající Internet, infrastrukturu CORBA (Common Object Request Broker Architecture) nebo COM (Component Object Model). Již dnes existuje celá řada znovupoužitelných komponent a je zřejmé, že softwarový průmysl věnuje velké úsilí dalšímu vývoji v této oblasti. Vývoj softwaru se tak přesouvá do oblasti skládání produktu z prefabrikovaných komponent. [9]

### 1.3.4. Vizuální modelování

Model představuje zjednodušení reality. Vizuální modelování pomáhá zpřehlednit, specifikovat, zkonstruovat a zdokumentovat strukturu a chování systémové architektury. K jednomu systému se vytváří zpravidla více modelů (například z různých pohledů). Jako standardní jazyk pro modelování slouží UML, díky němuž mohou jednotliví členové týmu mezi sebou srozumitelně komunikovat a předávat si informace bez ohledu na fázi vývoje.[1]

### 1.3.5. Ověřování kvality

Kvalitu produktu s ohledem na funkčnost, spolehlivost a výkon je třeba sledovat nepřetržitě v průběhu celého procesu vývoje. Po dodání softwaru už je zpravidla pozdě na jakékoli změny a dodatečné opravy nalezených chyb jsou obtížné. Z tohoto důvodu je nutné průběžné testování a kontrola. Pro nejvyšší kvalitu produktu je nutné zaměřit se na oblasti s nejvyšším rizikem.

Princip ověřování kvality vytvářeného produktu je součástí softwarového procesu, je obsažen ve všech jeho aktivitách, týká se všech účastníků vývoje softwarového řešení. Využívají se objektivní měření a kriteria (metriky) kvantifikující kvalitu výsledného produktu. Zajištění kvality tak není považováno za něco, co stojí mimo hlavní linii vývoje produktu a není to záležitost zvláštní aktivity realizované speciální skupinou. [1]

### 1.3.6. Řízení změn

Důležitou podmínkou úspěchu při vývoji softwaru je efektivní koordinace všech aktivit, aby bylo možné opakovaně využívat standardní pracovní metody a reagovat na změny s cílem zajištění lepší alokace zdrojů. RUP se ve svých jednotlivých oblastech snaží řídit všemi uvedenými obecnými praktikami a přibližuje se k nim pomocí mnoha definovaných aktivit. [9]

Proces RUP také popisuje jak řídit, sledovat a monitorovat změny a umožňuje tak úspěšný iterativní vývoj. Nedílnou součástí této problematiky je vytvoření bezpečného pracovního prostředí, které poskytuje maximální možnou ochranu před změnami vznikajícími v jiném pracovním prostředí.

## 1.4. Jazyk Unified Modelling Language

UML je jazyk pro vizualizaci, specifikaci, stavbu a dokumentaci softwarových systémů. Na světě již existují různé metodické postupy, které vychází z modelovacích technik UML a rozšiřují je o vlastní doporučené postupy, další diagramy a techniky. Nejznámější takováto metodika je právě IBM Rational Unified Process.

Modelovací jazyk Unified Modelling Language (UML) je výsledkem snažení analytiků a návrhářů, kteří v průběhu 80. a 90. let vytvářeli metody umožňující popsat objektově orientovanou analýzu a návrh. V polovině 90. let byly velmi rozšířené metody OMT (Object Modelling Technique), jejichž autory byli Booch a Rumbaugh, a dále metodika Ivara Jacobsona – Objectory.

V roce 1995 byly zahájeny práce na sjednocení různých metod a syntaxí pro modelování pod záštitou firmy Rational. Výsledkem tohoto snažení bylo v roce 1997 vytvoření první verze modelovacího jazyka UML. Tento souhrn metod se stal průmyslovým standardem a postupně se vyvíjí až do aktuálně používané verze 2.0. Verze UML 2.0 představuje největší změnu, k jaké v UML vůbec kdy došlo. V rámci samotného UML byl změněn samotný metamodel. Patrně nejviditelnější změnou je uvedení nových typů diagramů. Diagramy objektů a seskupení se sice obecně používaly, ale nebyly oficiálním typem diagramu. Nyní již jsou. Dále došlo ke změně názvu diagramu objektové spolupráce na komunikační diagramy. UML 2.0 uvádí další typy nových diagramů, jako je diagram

přehledu interakce, diagram načasování a diagram smíšené struktury. Další významnou změnou je zavedení notace orámovaných interakcí pro ošetření iterativních, podmíněných, nebo dalších typů řízení chování v sekvenčních diagramech. [2]

Modelovací jazyk UML je souhrnem především grafických notací k vyjádření analytických a návrhových modelů. UML je jazyk, který umožňuje modelovat jednoduché i složité aplikace pomocí stejné formální syntaxe. Tím umožňuje výsledky práce sdílet s ostatními návrháři. Vybrané modely jsou pochopitelné i pro zadavatele aplikace a umožní kvalitní vyjasnění požadavků uživatelů na vytvářený systém. Žádný diagram nezachycuje navrhovaný systém jako celek, ale soustředí se vždy právě na jeden pohled na vyvíjený systém.

## 2 Životní cyklus a charakteristika fází

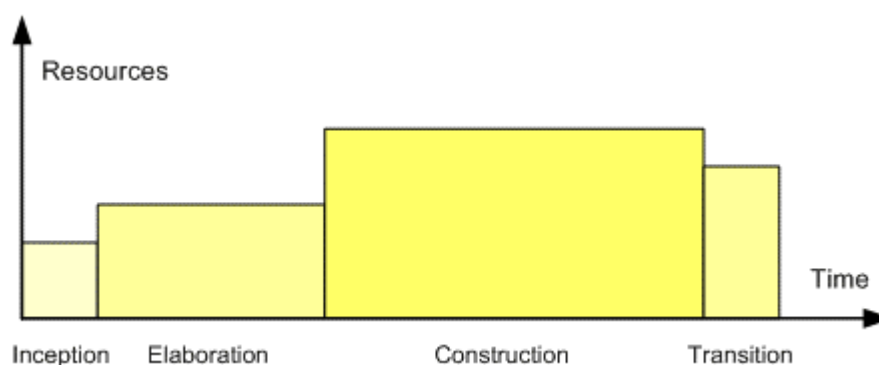
### 2.1. Životní cyklus projektu v RUP

Životním cyklem v RUP je ve skutečnosti implementace spirálového modelu, který je tvořen z polouspořádaných sekvencí sestavených z obsahu elementů. RUP je poté dostupný jako rozkladové schéma, které může být přizpůsobeno specifickým požadavkům každého projektu. [15]

RUP rozděluje projekt do 4 základních fází (obr.4) – zahájení (inception), rozpracování (elaboration), konstrukce (construction) a fáze předání (transition). Každá z těchto fází je ukončena tzv. milníkem. Po dokončení každé fáze se provede zhodnocení, zda byly dodrženy předepsané cíle. Další fáze začíná splněním všech požadovaných kritérií z předchozí fáze.

Náročnost jednotlivých fází projektu na čas a zdroje závisí na typu vyvíjeného software, existenci nástrojů pro automatizaci vývoje a tzv. generaci software.

Vývoj produktu obvykle není definitivně ukončen uvedením první verze do provozu. Pokud software definitivně „neodumře“, vytvářejí se na základě nově vznikajících potřeb jeho další verze. Vývoj následující generace produktu se z pohledu řízení projektu skládá opět ze čtyř fází. U dalších vývojových cyklů se ale zpravidla podstatně zkracuje délka prvních dvou fází, neboť analytické činnosti byly již z větší části provedeny u předchozích verzí. [16]



obr. 4. – Náročnost jednotlivých fází RUP

*zdroj: převzato z [12]*

#### 2.1.1. Fáze zahájení

Hlavním cílem této fáze je vytvořit obchodní případ a určit rozsah projektu. Je třeba identifikovat prvky, s nimiž bude systém komunikovat, a definovat povahu těchto prvků.

To vše se děje na základě vyhodnocení požadavků a omezení systému, stanovení kritérií, která musí systém splňovat, zvážení možných alternativ v poměru cena/termín/výkon, definování použitelných architektur (a komponent) a rozhodnutí, co se vyrobí/koupí/znovupoužije. [1]

Vstupy v této fázi tvoří původní vize, úvodní požadavky na systém a případně již existující systém. Výstupem této fáze bude vize ve formě dokumentu obsahujícího obecnou vizi klíčových požadavků projektu, včetně hlavních rysů a omezení. Dalšími výstupy jsou model případu použití, počáteční glosář projektu, počáteční obchodní případ (obchodní souvislosti, kritéria úspěchu, rozpočet), počáteční ohodnocení rizik projektu a přehled fází včetně iterací.

### **2.1.2. Fáze rozpracování**

Cílem této fáze je analyzovat klíčové problémy, vyvinout základy architektury, vytvořit plán projektu a odhadnout jeho nejrizikovější prvky. Součástí rozpracování je vytvoření funkčního prototypu.

Fáze rozpracování navazuje na fázi zahájení projektu, na jejím začátku musí být k dispozici úvodní specifikace požadavků a úvodní verze plánu vývoje software, zahrnující stručný plán iterací pro tuto fázi. [1]

V této fázi by se měl návrhář zaměřit především na kritické případy užití identifikované v počáteční fázi, které typicky zahrnují hlavní rizika projektu. Dá se říci, že tato fáze je nejrizikovější ze všech čtyř fází. Na jejím konci nastává důležitý okamžik, kdy se rozhodne o pokračování projektu.

Výstupem jsou v této fázi dopracovaný model případu užití, dodatečné požadavky (nefunkční požadavky a požadavky, které nejsou spojeny s případem užití), popis softwarové architektury, spustitelný prototyp architektury, revidovaný seznam rizik a vývojový plán pro fázi implementace včetně hodnotících kritérií pro každou iteraci. Dále je výstupem předběžný instalační manuál.

### **2.1.3. Fáze konstrukce**

V průběhu konstrukční fáze jsou navrženy všechny zbývající komponenty a vlastnosti aplikace jsou vyvinuty a integrovány do produktu. Všechny vlastnosti systému jsou důkladně otestovány. Fáze konstrukce je vlastně výrobním procesem v němž se klade důraz na řízení zdrojů a kontrolu kvality, kvantity a termínů.

Hlavním cílem této fáze je dokončení návrhu, vytvoření a otestování první funkční verze systému. Základním problémem je efektivní řízení zdrojů - stejně jako v předchozích fázích je vhodné projekt rozdělit na relativně nezávislé části a ty pak přidělit jednotlivým vývojářským týmům. V této fázi projektu by měly být požadavky na systém již stabilní, správa požadavků by měla řešit pouze případné zapracování dodatečných požadavků na změny.[1]

Mezi výstupy této fáze můžeme zařadit popis současné verze, uživatelský manuál a především softwarový produkt, který je integrovaný na odpovídajících platformách.

### **2.1.4. Fáze předání**

Tato fáze se soustředí na činnosti potřebné k předávání software do provozního prostředí. Zpravidla zahrnuje několik iterací (např. uživatelské akceptační testy, pilotní provoz, rutinní provoz), včetně servisních sestavení (buildů) a opravných balíčků (patchů) řešících chyby. Velké úsilí je vynaloženo na přípravu a školení uživatelů a na přípravu uživatelské a servisní dokumentace.

Hlavním cílem této fáze je předání finální verze systému koncovým uživatelům. V jejím průběhu se provádí beta testování systému a jeho drobné úpravy na základě připomínek uživatelů. V této fázi by již nemělo docházet k žádným zásadním změnám funkcionality software - požadavky zadavatele by se měly týkat pouze instalace, konfigurace a odlaďování drobných chyb zjištěných při testovacím provozu. [1]

Výstupem této fáze jsou software (konečná funkční verze splňující všechny požadavky zadání), dále pak manuály (kompletní uživatelská dokumentace k dané verzi projektu) a také materiály pro školení uživatelů v rámci aktuální verze projektu.

## 2.2. Milníky jednotlivých fází

### 2.2.1. Rozsah systému

Na konci fáze zahájení se objevuje první milník projektu - Rozsah systému. Jeho smyslem je navodit konsensus mezi osobami zodpovědnými za realizaci projektu a objektivně prokázat, že projekt je realizovatelný v navržené kvalitě, kvantitě, termínu a rozpočtu.

Jako podklady zde slouží výstupy z fáze zahájení. Ty musí mapovat všechny veličiny, které podstatně ovlivňují zmíněné atributy. V případě, že se nepodaří dosáhnout tohoto milníku, musí být projekt přehodnocen, popřípadě zastaven.

#### **Kritéria pro posouzení počáteční fáze jsou:**

- souhlas všech zúčastněných s cenovým a časovým odhadem
- porozumění požadavků
- odhad nákladů, časový rozvrh, priority, rizika
- šířka a hloubka prototypu architektury
- aktuální náklady vs. náklady plánované [1]

### 2.2.2. Definice architektury

Na konci fáze rozpracování se prověřují detailní vlastnosti systému a jeho rozsah. Je zaveden další milník projektu - Definice architektury. Klíčovými úkoly jsou výběr architektury a odstranění hlavních rizik vyřešením technologických problémů nebo zvolením alternativních postupů.

Jako podklady slouží výstupy z fáze rozpracování, které musí obsahovat prototyp architektury, konkrétní plán pro další fázi a revizi známých rizik. Nepodaří-li se dosáhnout tohoto milníku, je projekt přehodnocen, případně zastaven.

#### **Kritéria pro posouzení fáze rozpracování:**

- vize produktu je neměnná
- bylo zajištěno alespoň 80% případů užití, alespoň polovina tohoto počtu je zanalyzována
- návrh a implementace proveden u 10% případů užití
- návrh architektury je konečný, nebude docházet k výrazným změnám
- jsou definovány postupy pro hodnocení rizik
- testování prototypů prokázalo, že byla podchycena rizika architektury
- plán iterací pro fázi konstrukce je dostatečně přesný a podrobný
- zadavatelé souhlasí s realizací za použití navrhované architektury
- skutečná spotřeba zdrojů nepřevyšuje plán

### **2.2.3. Beta-verze**

Po zakončení fáze konstrukce následuje třetí milník projektu - Beta-verze. V tomto okamžiku se prověřuje, zda software, provozní prostředí a uživatelé jsou připraveni k nasazení systému do provozu, aniž bychom zúčastněné strany vystavovali velkým rizikům.

V případě, že se nepodaří dosáhnout tohoto milníku, je třeba odložit zavedení a naplánovat náhradní verzi odstraňující rizika nasazení systému do provozu. Případně musí být upraven Plán nasazení.

#### **Kritéria pro posouzení fáze konstrukce:**

- produkt je dostatečně stabilní, aby jej bylo možné poskytnout uživatelům
- zadavatel je připraven k nasazení systému
- porovnání přijatelnosti skutečných nákladů s plánovanými

### **2.2.4. Nasazení**

V tomto okamžiku se rozhodne, zda byly dosaženy stanovené cíle a zda je možno začít práci na jiném vývojovém cyklu. V některých případech se tento milník kryje s ukončením počáteční fáze dalšího cyklu. Důležitá je zde správná funkčnost komunikačního kanálu mezi servisním týmem a koncovými uživateli. Do servisního týmu je vhodné zařadit několik lidí z vývoje, ti zaškolí ostatní a poté se mohou vrátit zpět do realizace.

#### **Kritéria pro posouzení fáze předání:**

- uživatelé jsou spokojeni
- porovnání přijatelnosti výsledných nákladů s plánovanými

## 3 Základní model RUP a jeho disciplíny

### 3.1. Základní model

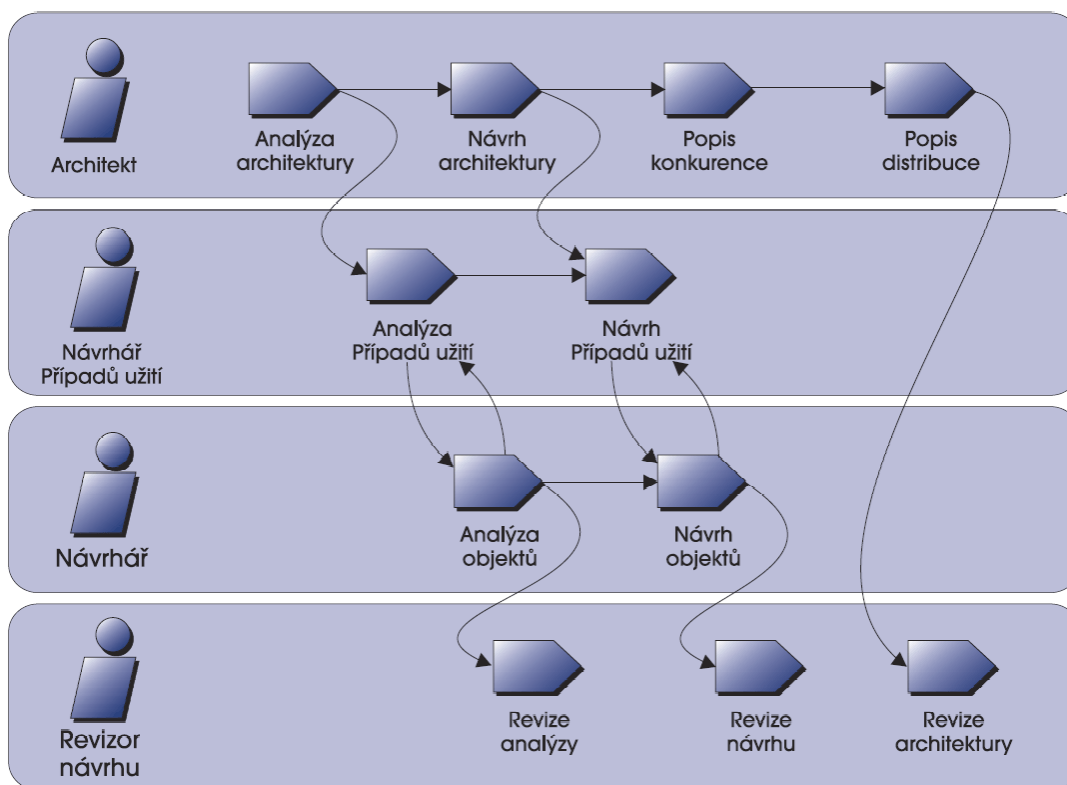
Základní model Rational Unified Process definuje kdo, jak, kdy a co dělá za použití čtyř základních prvků - osoby (role), aktivity (činnosti), artefakty a pracovní postupy.

Osoby definují zodpovědnosti za artefakty a aktivity. Osobu si lze představit jako roli v divadelní hře. Jeden člověk může reprezentovat více rolí a zároveň do jedné role může být obsazeno více lidí. V Rational Unified Process osoba definuje, jak má jednotlivec dělat svou práci a jakých artefaktů je vlastníkem. Toto rozdělení má na starosti projektový manager, který plánuje projekty a jejich personální obsazení.

Aktivita je jednotka práce, která může být osobě přidělena. Aktivita má jasný cíl, většinou se jedná o tvorbu či správu artefaktů. Každá aktivita je přidělena specifické osobě. Aktivita musí být použitelná, jako prvek plánování. Pokud je příliš malá, může být opomenuta, naopak, je-li příliš velká, lze pokrok vyjádřit po částech aktivity. V objektově orientované terminologii je osoba aktivní objekt a aktivity, které osoba vykonává jsou pak operace vykonané tímto objektem.

Artefakt představuje informaci, která je vytvořena, modifikována a používána v procesu vývoje. Je hmatatelným výsledkem projektu. Artefakty jsou používány jako vstup pro vykonávání aktivity určitou osobou a jsou cílem nebo výstupem takovéto aktivity.

Pracovní postup je způsob, jakým lze popsat posloupnost aktivit, které přinášejí předem definované výstupy a ukazují interakce mezi osobami (obr.5).



obr. 5 – Typické vyobrazení jednotlivých symbolů a příklad pracovního postupu

*zdroj: převzato z [1]*



## 3.2. Disciplíny RUP

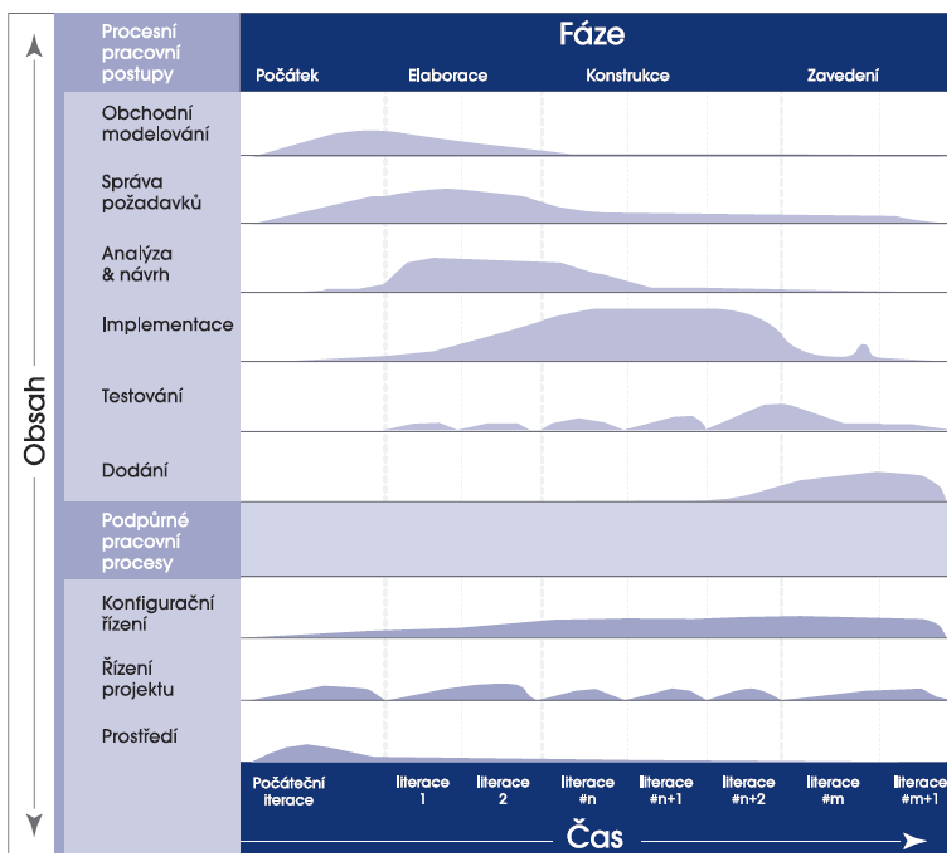
Rational Unified Process rozlišuje mezi tzv. procesními pracovními postupy a podpůrnými pracovními procesy (obr.6). Procesní pracovní postupy zajišťují vlastní práci na projektu, zatímco podpůrné pracovní procesy slouží k jejich řízení a koordinaci.

### Procesní pracovní postupy

- Obchodní modelování
- Správa požadavků
- Analýza a návrh
- Implementace
- Testování
- Dodání a nasazení

### Podpůrné pracovní procesy

- Konfigurační řízení
- Řízení projektu
- Prostředí



obr. 6 – Schématické vyjádření procesu RUP

*zdroj: převzato z [1]*

### 3.2.1. Obchodní modelování (Business modelling)

Problémem řady projektů bývá špatná komunikace mezi pracovníky zodpovědnými za obchodní modelování a mezi softwarovými návrháři. Metodika RUP propojuje obě činnosti na úrovni řízení projektu. Pro lepší komunikaci se používá společná terminologie – tzv. významové slovníky neboli glosáře.

Tato disciplína je definována jako volitelná, tzn. můžeme ji vypustit, pokud systém není vázán na konkrétní obchodní model.

Modely, které jsou v rámci dané disciplíny vytvářeny jsou dvou typů:

- **Modely podnikových procesů** popisující množinu vzájemně propojených procedur a aktivit vedoucí ke splnění podnikatelského cíle.
- **Doménový model** postihující nejdůležitější objekty vyskytující se v kontextu daného systému. Doménovými objekty rozumíme entity existující v prostředí, ve kterém funguje daný systém.

### 3.2.2. Správa požadavků

Cílem specifikace požadavků je popsat, co má softwarový systém dělat prostřednictvím specifikace jeho funkcionality. Modely specifikace požadavků slouží k odsouhlasení zadání mezi vývojovým týmem a zadavatelem. [9]

Požadavky jsou kritéria, jejichž splnění podmiňuje úspěch projektu. Požadavky je proto třeba systematicky evidovat a odpovídajícím způsobem dokumentovat a zapracovávat jejich případné změny.

Správu požadavků bychom tedy mohli definovat jako systematický přístup ke zjišťování a dokumentaci požadavků na systém a zároveň jako proces, který zajišťuje shodu mezi zadavatelem a dodavatelem, v případě změny požadavků na systém.

### 3.2.3. Analýza a návrh

Analýza a návrh propojuje požadavky na systém s jeho implementací. Zatímco modely případů užití vytvářené v rámci správy požadavků odpovídají na otázku co se bude vytvářet, návrh by měl řešit jakým způsobem se to bude vytvářet - tj. nadefinovat architekturu systému, včetně jeho členění na komponenty.

Modely, které jsou v rámci analýzy a návrhu vytvářeny jsou následující:

- **Model analýzy**, který zkoumá specifikované požadavky z pohledu objektů, které lze nalézt v problémové doméně
- **Model návrhu** dále upřesňuje model analýzy ve světle skutečného implementačního prostředí.
- **Model nasazení** specifikuje topologii technických prostředků určených ke spuštění výsledného softwarového produktu

### 3.2.4. Implementace

Cílem implementace je doplnit navrženou architekturu (kostru) aplikace o programový kód a vytvořit tak kompletní systém. Implementační model je v architektuře specifikován implementačním pohledem ve smyslu, jak jsou jednotlivé elementy (objekty a třídy) vytvořené v etapě návrhu implementovány v podobě softwarových komponent. Komponenty obvykle představují zdrojové kódy, spustitelné kódy, data a podobně. [9]

Softwarová komponenta je definována jako fyzicky existující a zaměnitelná část systému vyhovující požadované množině rozhraní a poskytující jejich realizaci.

Podle typu softwarových komponent hovoříme o:

- zdrojovém kódu, částech systému zapsaném v programovacím jazyce
- binárním (přeloženém do strojové kódu procesoru) a spustitelném kódu
- ostatních částech reprezentovaných databázovými tabulkami, dokumenty apod.

Jestliže jsme ve fázi analýzy a návrhu pracovali pouze s abstrakcemi dokumentovanými v podobě jednotlivých diagramů, pak v průběhu implementace dochází k jejich fyzické realizaci. Implementační model se tedy také zaměřuje na specifikaci toho, jak budou tyto komponenty fyzicky organizovány podle implementačního prostředí a programovacího jazyka poskytujícího konkrétní mechanismus strukturování a modularizace. [9]

### 3.2.5. Testování

Cílem testování je vytvořit a provést soubor testů pro ověření funkčnosti komponent systému a jejich správné integrace. Testování se provádí z pohledu tří základních dimenzí reprezentovaných kvalitou, funkcionalitou a výkonností systému. Testování se týká všech vytvářených modelů a jejich diagramů.

Testy se plánují pro každou iteraci a zahrnují integrační testy a testy systémové. Integrační testy se provádí pro každý vytvořený produkt během iterace, zatímco systémový test se provádí pouze na konci iterace, kdy vzniká spustitelná verze vyvíjeného produktu. [9]

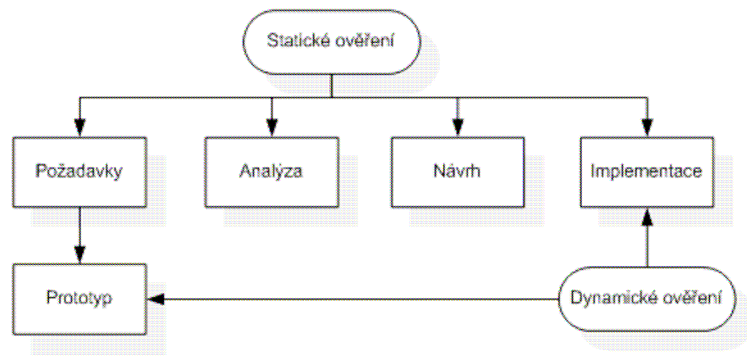
Testy se navrhují a následně implementují v podobě testovacích úloh, které jednoznačně definují co se má ověřit. Z tohoto pohledu hovoříme o testovacích procedurách, které specifikují, jak se má test provádět, nebo se vytváří spustitelné testovací komponenty umožňující automatizaci procesu ověřování.

Výsledky provedených testů jsou systematicky zpracovávány a vadné části jsou opakovaně testovány a případně zaslány zpět do toků činností jako je analýza, návrh nebo implementace s cílem nedostatky odstranit.

Testování softwarových systémů probíhá dvěma způsoby nazývanými verifikace a validace. Verifikace je proces testování hledající odpověď na otázku, zda-li softwarový produkt je vytvářen správně. Jinými slovy řečeno, hledáme nedostatky v samotném softwarovém systému a validace je proces testování hledající odpověď na otázku, zda-li vytvářený software je správný. Jinými slovy, zda-li implementuje požadovanou funkcionalitu. [9]

**Techniky testování můžeme dále rozdělit na statické a dynamické (obr.7):**

- statické techniky – ověřují korespondenci mezi vytvořeným systémem a jeho specifikací
- dynamické techniky – spouštějí výsledný produkt a ověřují i jeho užitečnost a kvalitu



obr. 7 – Statické a dynamické testování

*zdroj: převzato z [9]*

Oproti předchozím modelům, které byly definovány různými typy diagramů, nemá jazyk UML pro testování žádný definovaný diagram. Testování je definováno souborem testovacích úloh definujících co se má u systému testovat, testovacích procedur specifikujících postupy pro provedení testovacích úloh a testovacích komponent, které automatizují testovací procedury.

Testovací úloha specifikuje způsob testování systému zahrnující informace o tom, co se má testovat s jakými vstupy a za jakých podmínek. Testovací úlohy lze rozdělit na testovací úlohy určené k ověření případů užití nebo jejich specifických scénářů. Takové úlohy ověřují výsledky interakce aktéra se systémem. Tento tzv. black-box test ověřuje z vnějšku pozorovatelné chování systému. Dále pak na testovací úlohy specifikující, jak ověřovat realizaci případů užití. Tyto úlohy ověřují interakce mezi komponentami implementující případ užití. Tento tzv. white-box test je určen k ověření vnitřních interakcí daného systému. [9]

Testovací procedura specifikuje jak provést jednu nebo více testovacích úloh nebo jejich částí. Testovací procedury lze specifikovat pomocí instrukcí popisujících jak ručně provést danou testovací úlohu, popisu jak vytvořit spustitelnou testovací komponentu, specifikace jak použít nástroj pro automatické testování.

Testovací komponenty automatizují jednu nebo více testovacích procedur nebo jejich částí. Testovací komponenty jsou používány k ověření komponent tvořících implementační model cestou poskytnutí požadovaných vstupů, řízení a sledování běhu ověřované komponenty a vytvoření výsledné zprávy o průběhu testu. K vytvoření testovacích komponent se mohou používat různé skriptovací jazyky nebo k tomuto účelu vytvořené systémy automatizované verifikace. [9]

### 3.2.6. Dodání a nasazení

Účelem toku činností dodání a nasazení je úspěšně vytvořit výsledný produkt a dodat jej koncovým uživatelům. [9] Tento tok pokrývá celou řadu činností:

- vytvoření výsledného produktu či jeho verzí
- kompletace softwarového systému
- distribuce softwarového systému
- instalace softwarového systému u uživatele
- poskytnutí asistenční služby uživatelům
- plánování a řízení beta testování
- migrace již existujících dat a softwarových produktů.

## 4 Architektura systému v RUP

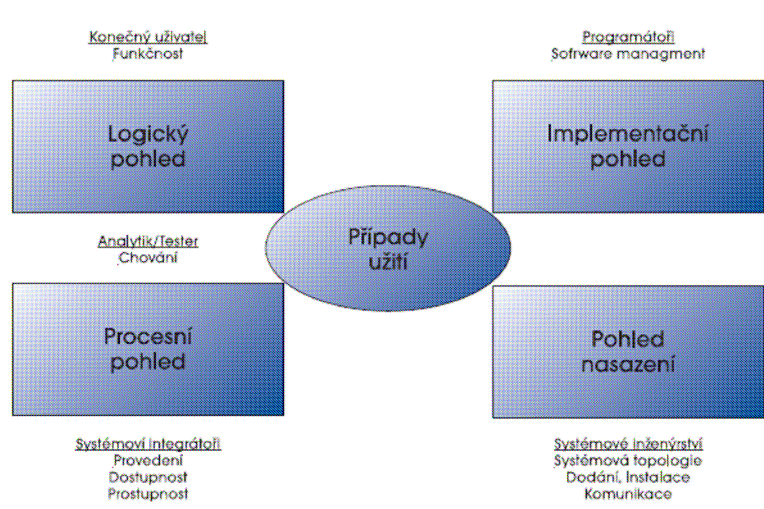
### 4.1. Pracovní definice

Pojmem architektura můžeme označit popis a způsob uspořádání komponent systému a rozhraní, pomocí kterých navzájem komunikují. Architektura je tedy součástí návrhu systému. Zachycuje nejdůležitější prvky, které mají vliv na kvalitu a výkon systému.

Velká část RUP je zaměřena na vizuální modelování. Modely nám pomáhají lépe pochopit problém a znázornit jej spolu s jeho řešením. Model můžeme považovat za zjednodušení reality, které nám pomáhá zvládnout složité a rozsáhlé systémy.

Právě výběr modelů a techniky k jejich vyjádření má velký vliv na náš pohled na celý systém. Žádný jednoduchý model nedokáže pojmut všechny aspekty softwarového vývoje. Potřebujeme tedy několik modelů k vyjádření různých pohledů. Modely musejí být dostatečně propojeny, aby byly konzistentní a nevznikaly zbytečné redundance.

### 4.2. Model architektury 4+1



obr. 8 – Model architektury 4+1

*zdroj: převzato z [1]*

Rational Unified Process používá pro znázornění architektury systému pět pohledů (obr.8). Jednotlivé pohledy řeší různé aspekty fungování systému, nejsou však na sobě nezávislé a do určité míry se překrývají. Navíc tyto pohledy nejsou povinné pro všechny projekty.

#### 4.2.1. Případy užití

Případy užití představují pohled na systém z hlediska jeho funkcionality pro koncového uživatele. Tento pohled je jádrem architektury systému, zbývající pohledy zajišťují realizaci zde specifikovaných požadavků. [16]

V případech užití definuje Rational Unified Process čtyři klíčové prvky. Aktér, což je reprezentace uživatele systému, konkrétní případ užití, který značí posloupnost akcí které poskytují

aktérům určitou hodnotu, vztahy include/extend které vyjadřují závislé nebo rozšiřující funkčnosti a vztah asociace která zachycuje komunikaci mezi aktérem a případem užití.

Typický postup návrhu případu užití vyžaduje obvykle workshop pro ustavení základních aktérů a případů užití, poté se model reviduje a zakreslují se další požadavky na model. V další fázi se jednotlivé případy užití textově popíší a provádí se návrh architektury.

Při rozšiřování systému je třeba brát zřetel na správné užívání popisných výrazů definovaných v počátečním glosáři.

#### **4.2.2. Logický pohled**

Znázorňuje logickou strukturu systému z hlediska výsledné funkcionality. Jedná se o abstraktní model návrhu, který popisuje hlavní analytická seskupení (packages), subsystémy a třídy. [16]

Hlavním prostředkem je zde diagram tříd znázorňující objektové třídy a vztahy mezi nimi (asociace, dědičnost).

Při tvorbě diagramu objektových tříd hledáme ve scénáři tzv. klíčová slova, která nám pomohou detekovat budoucí objekty systému. Je velmi vhodné nalezené objektové třídy spojovat do analytických seskupení, tzv. balíčků (packages). Balíčky spolu komunikují pomocí svých rozhraní (interface), což je specifikované chování, které musejí implementátoři dodržovat. Realizací rozhraní musejí třídy garantovat podporu požadovaného chování, které umožňuje systému spravovat neasociované elementy přes toto společné rozhraní.

Diagram objektových tříd by neměl být chápán jako model tříd ve významu tříd objektově orientovaného programování. Třídou v analytickém modelu se rozumí evidovaný pojem. Ten nemusí být ve výsledném programu nutně realizován přímo třídou, ale například tabulkou, funkcí a podobně. Velmi často se ale programové třídy a analytické třídy překrývají.

#### **4.2.3. Procesní pohled**

Tento pohled zachycuje software jako množinu navzájem komunikujících procesů (proces je chápán jako soubor úloh). Řeší problém konkurence a paralelismu procesů. Modelování procesů a meziprocsových komunikace se provádí pomocí procesního diagramu. [16]

#### **4.2.4. Implementační pohled**

Obvykle zobrazuje organizaci statických softwarových komponent (exe, dll, html...). Znázorňuje rozčlenění systému do jednotlivých modulů. Strukturu pak definuje aplikační framework. Hlavním grafickým prostředkem je diagram komponent.

#### **4.2.5. Pohled nasazení**

Definuje souvislost procesů s hardwarovou architekturou systému. Je zaměřen na návaznost spustitelných programů a jednotlivých komponent systému na topologii hardwarových a dalších softwarových komponent. Typickým formálním prostředkem je diagram nasazení v UML.

## 5 Fáze zahájení

### 5.1. Dokument Vize

#### 5.1.1. Úvod

Hlavním cílem dokumentu vize je shromáždit, specifikovat a zanalyzovat klíčové potřeby a vlastnosti Informačního systému vnitřního vzdělávání zaměstnanců.

Tento dokument je zaměřen na definování klíčových požadavků na systém, které jsou vyžadovány zainteresovanými osobami a koncovými uživateli.

#### 5.1.2. Rozsah vypracování

Rozsah vypracování je omezen rozsahem diplomové práce a velikosti týmu (jednočlenný) pouze na „fázi zahájení“ a „fázi rozpracování“.

#### 5.1.3. Cíl a rozsah projektu

Cílem projektu je analyzovat a navrhnout Informační systém vnitřního vzdělávání zaměstnanců, který bude vypracován a využíván společností Unicorn a.s.

**Tento systém bude umožňovat:**

- organizaci výukových kurzů, včetně vazby na outsourcing
- evidenci historie absolvovaných kurzů jednotlivými zaměstnanci
- přihlašování zaměstnanců na kurzy
- hodnocení zaměstnanců z hlediska výsledků v kurzech
- příjemné uživatelské rozhraní pro všechny zainteresované osoby
- jasné časové rozvržení termínů v podobě rozvrhů a časových linek

Informační systém vnitřního vzdělávání zaměstnanců bude dostupný z internetu i intranetu a má za úkol zefektivnit práci jednotlivých lektorů, poskytnout aktuální informace a pohodlí školeným zaměstnancům a nakonec také informace manažerům o jednotlivých zaměstnancích.

#### 5.1.4. Demografická specifikace

Tento informační systém je určen pro potřeby největší české společnosti poskytující komplexní služby v oblasti informačních systémů a informačních a komunikačních technologií. Společnost byla založena v roce 1990 a v současné době působí především v České republice. Dále se orientuje zejména na evropský region a Spojené státy americké.

Za více než 15 let jejího působení vzrostl počet zaměstnanců na 900. Zaměstnanci jsou pečlivě vybíráni z velkého počtu uchazečů a poté neustále vzděláváni. Společnost si plně uvědomuje, že její síla spočívá především ve vysoké kvalitaci a profesní způsobilosti jejích zaměstnanců.

V současnosti společnost také výrazně investuje do vývoje vlastního podnikového informačního systému UES (Unicorn Enterprise System).

V průběhu roku 2005 společnost Unicorn zrealizovala 78 projektů v oblasti zakázkového vývoje softwaru, 42 team-leasingových projektů, 25 projektů v oblasti konzultací, 57 v oblasti servisu a podpory a 133 vzdělávacích kurzů.

### 5.1.5. Vymezení problému

<b>Problémem je:</b>	Nevhodně zvolená architektura stávajícího systému, pomalost stávajícího systému, neefektivní komunikace mezi lektory a školenými zaměstnanci, bez výstupu pro management.
<b>Dopad na:</b>	Pracovníky firmy Unicorn a.s., pracovníky externích školících firem.
<b>Následky problému:</b>	Nízká efektivita práce lektorů, slabá zpětná vazba mezi managementem a pracovníky, časté využívání outsourcingových firem.
<b>Úspěšné řešení:</b>	Zvýší efektivitu školení, usnadní práci lektorům i managementu.

tab. 1 – Vymezení problému

### 5.1.6. Vymezení produktu

<b>Pro:</b>	Zaměstnance Unicorn a.s., lektory, externí školící firmy, management.
<b>Co:</b>	Organizovat výukové kurzy, poskytovat rozhraní externím školícím firmám, evidovat historii školení, poskytovat přehledné rozhraní pro zaměstnance i lektory, poskytovat informace managementu.
<b>Kategorie:</b>	Softwarový produkt typu informační systém.
<b>Přínos našeho produktu:</b>	Zvýší efektivitu školení, poskytne příjemné uživatelské komunikační rozhraní pomocí sítě Internet.
<b>Konkurenční produkt:</b>	Stávající zastaralý intranetový/nástěnkový systém.

tab. 2 – Vymezení produktu



### 5.1.7. Přehled zainteresovaných osob

Název	Reprezentován	Role
Školený zaměstnanec	Zaměstnanci společnosti Unicorn	Vybírá si z přehledu nabízených výukových kurzů, přihlašuje se na kurzy, listuje v informacích kurzu, stahuje materiály, prohlíží své výsledky, odevzdává úlohy, přihlašuje se na termíny zkoušení.
Lektor	Vyškolení zaměstnanci společnosti Unicorn	Sestavuje kurz, vkládá studijní materiály a podklady, vytváří a hodnotí úlohy, vyhodnocuje zkoušení, vypisuje termíny zkoušení.
Externí lektor	Zaměstnanci externích školících firem	Sestavuje kurz, vkládá studijní materiály a podklady, vytváří a hodnotí úlohy, vyhodnocuje zkoušení, vypisuje termíny zkoušení, předkládá kurz ke schválení, konzultuje s garanty z Unicornu.
Garant kurzu	Pověřený zaměstnanec s vysokou kvalifikací	Dohlíží na kvalitu kurzu, schvaluje nabídky kurzů externích firem, řeší připomínky po skončení kurzu
Dohlízející manager	Pověřený personální manager	Kontroluje počty školení u jednotlivých zaměstnanců, analyzuje získaná data o úspěšnosti zaměstnanců, sestavuje programy školení podle potřeb jednotlivých oddělení společnosti.
Administrátor	Správce systému	Spravuje dotazy uživatelů, přiděluje práva, spravuje uživatelské účty.

tab. 3 – Přehled zainteresovaných osob a jejich rolí v systému

### 5.1.8. Prostředí uživatelů

Uživatelé mají nejméně středoškolské vzdělání a měli by zvládat základní úkony v prostředí systému Microsoft Windows. Společnost Unicorn disponuje systémem Microsoft Windows XP nainstalovaným na všech pracovních stanicích.

V blízké budoucnosti se nepředpokládá změna operačního systému. Uživatelé využívají pro svou práci vnitropodnikový systém UES, do kterého bude integrován námi navržený systém vnitřního vzdělávání zaměstnanců.

## 5.1.9. Profily uživatelů

### 5.1.9.1. Školený zaměstnanec

<b>Popis:</b>	Zaměstnanec Unicorn a.s.
<b>Typ:</b>	Uživatel
<b>Odpovědnosti:</b>	Odpovědný za splnění vypsanych ročních kvót na školení.
<b>Kritéria úspěchu:</b>	Snadno použitelné a intuitivní prostředí kurzů, snadná dostupnost zveřejněných materiálů, jasný rozvrh.
<b>Spoluzodpovědnost:</b>	Reviduje systém z hlediska ergonomie uživatelského rozhraní, dále reviduje funkcionality a použitelnost částí související s prohlížením kurzů, materiálů, zkušebních termínů.

tab. 4.1 – Školený zaměstnanec

### 5.1.9.2. Lektor

<b>Popis:</b>	Vyškolený zaměstnanec Unicorn a.s. určený ke školení dalších zaměstnanců firmy.
<b>Typ:</b>	Uživatel
<b>Odpovědnosti:</b>	Odpovědný za vyučované kurzy, dostatek materiálů ke kurzům, hodnocení a zkušební proces.
<b>Kritéria úspěchu:</b>	Snadné editace zkušebních termínů, zadávání úloh, vkládání učebních podkladů, hodnocení zaměstnanců.
<b>Spoluzodpovědnost:</b>	Reviduje systém z hlediska ergonomie uživatelského rozhraní lektorské části, dále reviduje funkcionality a použitelnost částí související s vkládáním kurzů, materiálů, zkušebních termínů.

tab. 4.2 – Lektor

### 5.1.9.3. Externí lektor

<b>Popis:</b>	Zaměstnanec externí školící firmy, určený ke spolupráci s Unicorn a.s.
<b>Typ:</b>	Uživatel
<b>Odpovědnosti:</b>	Odpovědný za vyučované kurzy v rámci outsourcingu, dostatek materiálů ke kurzům, hodnocení a zkušební proces.
<b>Kritéria úspěchu:</b>	Stejně jako u lektora, navíc export požadavků na kurzy ve formě XML
<b>Spoluzodpovědnost:</b>	Reviduje systém z hlediska ergonomie uživatelského rozhraní části pro outsourcingové firmy, dále reviduje funkčnost komunikačního rozhraní.

tab. 4.3 – Externí lektor

### 5.1.9.4. Garant kurzu

<b>Popis:</b>	Pověřený zaměstnanec Unicorn a.s. s vysokou kvalifikací
<b>Typ:</b>	Uživatel
<b>Odpovědnosti:</b>	Odpovědný za kvalitu vyučovaných kurzů, schvaluje nabídky externích firem, odpovědný za řešení připomínek po skončení kurzu.
<b>Kritéria úspěchu:</b>	Jednoduché porovnávání struktury kurzu dle metodického plánu, dostupný přehled statistických výsledků a připomínek zaměstnanců.
<b>Spoluzodpovědnost:</b>	Reviduje systém z hlediska správnosti statistických dat, přehledu metodických plánů.

tab. 4.4 – Garant kurzu

### 5.1.9.5. Dohlížející manager

<b>Popis:</b>	Pověřený personální manager Unicorn a.s.
<b>Typ:</b>	Uživatel
<b>Odpovědnosti:</b>	Odpovědný za splnění počtu školení u jednotlivých zaměstnanců, dále odpovědný za analýzu dat úspěšnosti zaměstnanců a odpovědný za sestavování plánů školení dle potřeb oddělení.
<b>Kritéria úspěchu:</b>	Jednoduché porovnávání struktury kurzu dle metodického plánu, dostupný přehled statistických výsledků a připomínek zaměstnanců.
<b>Spoluzodpovědnost:</b>	Reviduje systém z hlediska správnosti statistických dat, přehledu metodických plánů.

tab. 4.5 – Dohlížející manager

### 5.1.9.6. Administrátor

<b>Popis:</b>	Pověřený pracovník Unicorn a.s.
<b>Typ:</b>	Administrátor
<b>Odpovědnosti:</b>	Odpovědný za správné přidělení uživatelských práv, vytváření nových uživatelských účtů. Odpovědný za správu FAQ dotazů.
<b>Kritéria úspěchu:</b>	Bezproblémové přidělování práv a pohodlná správa účtů. Pravidelné a pohotové zodpovídání dotazů.
<b>Spoluzodpovědnost:</b>	Kontroluje správné provozní vlastnosti systému.

tab. 4.6 – Administrátor

## **5.2. Počáteční obchodní případ**

### **5.2.1. Perspektiva systému**

System nahradí stávající systém společnosti. Nový systém bude poskytovat rozhraní pro komunikaci s outsourcingovými firmami a bude poskytovat vybraným firmám nabídky kurzů k sestavení.

System se bude skládat ze serverové a klientské části. Serverová část bude umístěna na webovém serveru společnosti Unicorn a.s., který běží na platformě FreeBSD. Serverová část bude komunikovat s informačními systémy školících firem pomocí protokolu SOAP. Tato služba bude umístěna na serveru běžícím také na FreeBSD serveru. Klientská část bude využívána na osobním počítači. Tento počítač bude vybaven prohlížečem Internet Explorer 6 / Mozilla Firefox 1.5 / Opera 8 či vyšším. Pomocí tohoto prohlížeče bude uživatel se systémem komunikovat. Na klientský počítač nejsou kladeny žádné další nároky na software.

### **5.2.2. Přehled vlastností systému**

#### **Předpoklady a závislosti**

- systémy školících firem musejí být přizpůsobeny pro námi navrženou webovou službu
- současný systém vzdělávání je založen na ASP.NET, ale je přizpůsoben pouze pro použití v intranetu
- předpokládá se že Unicorn a.s. nebude měnit operační systém nejméně do roku 2008

#### **Náklady a cenová kalkulace**

Finanční omezení bylo nastaveno na 500 000 Kč, lze předpokládat, že stávající počítače spol. Unicorn budou využity jako klientské stanice a že nebude třeba nakupovat další klientský hardware.

#### **Licencování**

Nebyly ustaveny žádné požadavky na licencování první verze systému. System bude součástí vnitřního informačního systému UES.

#### **Instalace**

Klientská aplikace nevyžaduje instalaci žádného programu.

#### **Vhodné standardy**

Uživatelské rozhraní by se mělo držet všeobecných standardů definovaných konsorciem W3C, konkrétně XHTML1.0 Strict.

#### **Předběžné požadavky na výkon**

System by měl podporovat až 500 současně pracujících uživatelů vůči centrální databázi a lokálnímu serveru v jakýkoliv čas. Doba odezvy by neměla přesáhnout 1 sekunda.

## Požadavky na prostředí

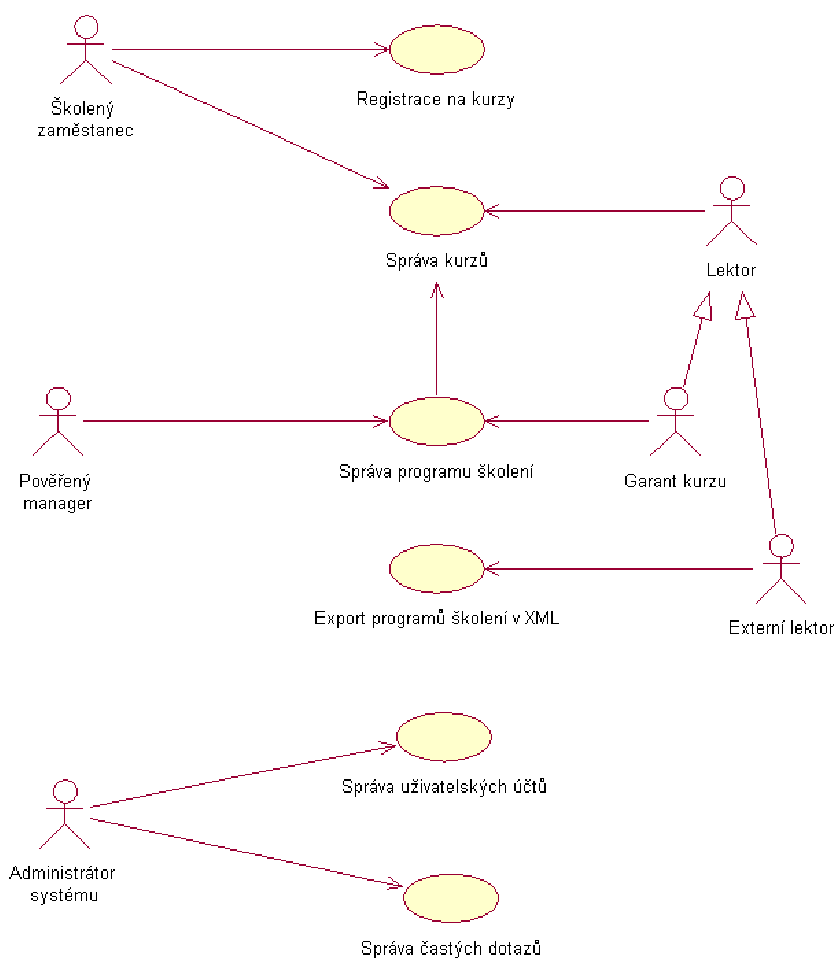
Servery provozující informační systém elektronického vzdělávání by měly být umístěny v klimatizované místnosti, bez přístupu vlhkosti.

## 5.3. Use-case model systému

V této části diplomové práce uvádím kompletní use-case model systému, ve kterém byly identifikovány a popsány veškeré případy použití vyskytující se v systému. Jedná se o již zpřesněnou verzi, kterou jsem několikrát přepracovával, proto by se správně měla znovu uvést jako zpřesněná forma případů použití ve fázi elaborace. Z důvodu velkého rozsahu tuto část uvádím zde již celou a dopracovanou. Neuvádím jí již ve fázi elaborace, protože z ní vycházejí ostatní podkapitoly této fáze.

### 5.3.1. Globální pohled na systém

Globální pohled označený jako UC01 (obr. 9) identifikuje aktéry a hlavní případy použití. Mezi hlavní případy použití patří - Správa programů školení, Správa kurzů, Správa uživatelských účtů, Registrace na kurzy, Export programů školení, Správa častých dotazů. Tyto jednotlivé případy použití jsou dále rozpracovány a označeny dle návaznosti.



obr. 9 – Globální pohled na systém

## UC01 – Globální pohled na systém

### Anotace

**Název:** Globální pohled na systém

**Identifikátor:** UC01

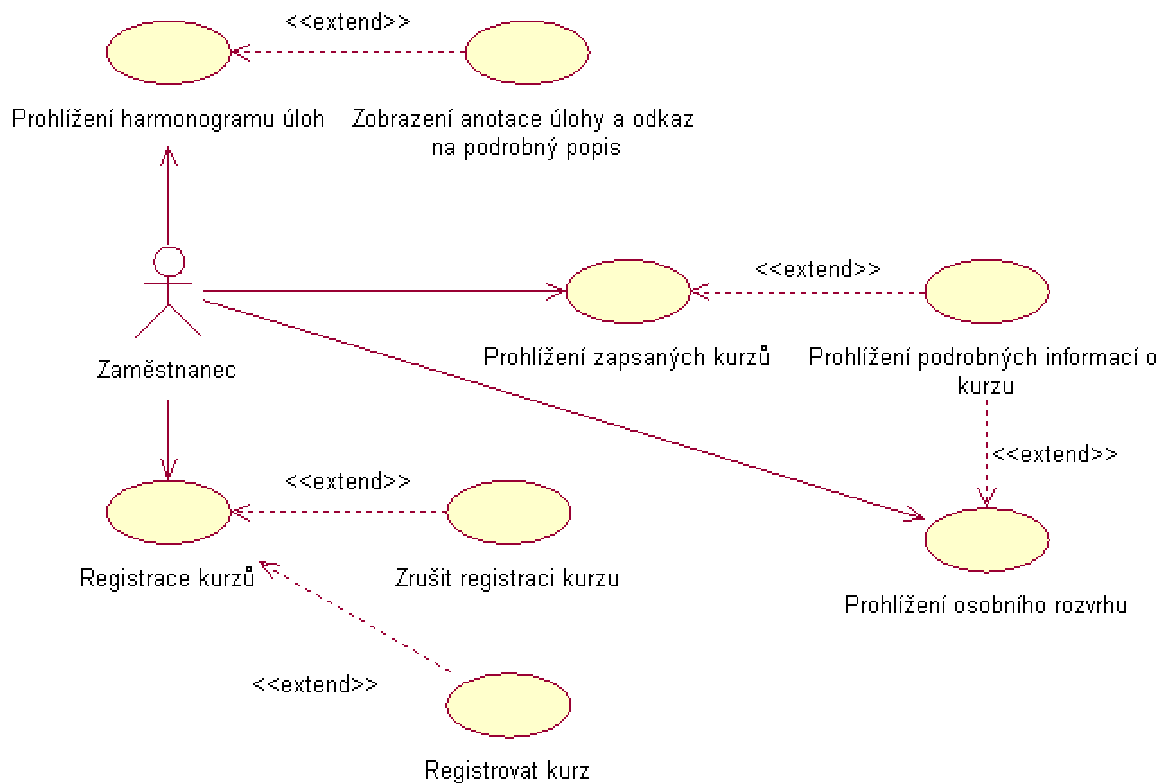
**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

**Poznámka:** Jedná se o pohled na zainteresované osoby a hlavní případy užití, jednotlivé případy užití jsou dále rozpracovány v následujících podkapitolách

### 5.3.2. Správa kurzů z pohledu zaměstnance

Toto schéma popisuje základní možnosti aktéra ve správě úloh.



obr. 10 – Identifikace případů použití související se správou kurzů z pohledu zaměstnance

## UC02 – Prohlížení zapsaných kurzů

### Anotace

**Název:** Prohlížení zapsaných kurzů

**Identifikátor:** UC02

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) aktér zvolí v menu odkaz „přehled kurzů“
- 2) systém vygeneruje seznam zapsaných a absolvovaných kurzů
- 3) systém vyznačí šedou barvou absolvované kurzy a seřadí prvky v seznamu od nejpozději zapsaného, absolvované kurzy zařadí na konec seznamu
- 4) kliknutím na název kurzu, nebo na ikonu lupy s popisem detaily systém přejde do bodu rozšíření UC08 - Prohlížení podrobných informací o kurzu

### **Body rozšíření**

- UC08 „*Prohlížení podrobných informací o kurzu*“

## **UC03 – Prohlížení osobního rozvrhu**

### **Anotace**

**Název:** Prohlížení osobního rozvrhu

**Identifikátor:** UC03

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) aktér klikne na odkaz menu „osobní rozvrh“
- 2) systém vygeneruje časový harmonogram zapsaných kurzů v přehledné tabulce
- 3) aktér klikne na název či zkratku v harmonogramu zapsaných kurzů
- 4) systém přejde do bodu rozšíření UC08 - Prohlížení podrobných informací o kurzu

### **Body rozšíření**

- UC08 „*Prohlížení podrobných informací o kurzu*“

## **UC04 – Prohlížení harmonogramu úloh**

### **Anotace**

**Název:** Prohlížení harmonogramu úloh

**Identifikátor:** UC06

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)



## Hlavní tok

- 1) aktér klikne na odkaz menu „harmonogram úloh“
- 2) systém vygeneruje seznam zadaných úloh
- 3) systém seřadí prvky seznamu chronologicky dle data ukončení úlohy
- 4) systém vyznačí šedou barvou úlohy po aktuálním datu
- 5) kliknutím na název úlohy nebo na ikonu lupy s popisem detaily systém přejde do bodu rozšíření UC13 - Správa detailů úlohy

## UC05 – Registrace kurzů

### Anotace

**Název:** Registrace kurzů

**Identifikátor:** UC05

**Podmínky před spuštěním:** kurzy mají status „k zápisu“

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) aktér klikne v seznamu na „registrace kurzů“
- 2) systém vygeneruje přehlednou tabulku obsahující jednotlivé kurzy seřazené dle programů školení
- 3) systém u každé položky, která nemá naplněnou kapacitu vygeneruje tlačítko „registrovat“
- 4) aktér klikne na tlačítko „registrovat“
- 5) systém přejde do bodu rozšíření UC06 – Registrovat kurz
- 6) po dokončení UC06 změní status kurzu na zapsáno

### Alternativní tok

- 1) má-li aktér zapsán kurz, zobrazí systém u tohoto kurzu tlačítko „zrušit registraci“
- 2) aktér klikne na tlačítko zrušit registraci
- 3) systém přejde do bodu rozšíření UC07 – Zrušit registraci kurzu

### Body rozšíření

- UC06 „*Registrovat kurz*“
- UC07 „*Zrušit registraci kurzu*“

## UC06 – Registrovat kurz

### Anotace

**Název:** Registrovat kurz

**Identifikátor:** UC06

**Podmínky před spuštěním:** aktér kliknul na tlačítko registrovat

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení akce
- 2) aktér akci potvrdí / zamítne
- 3) systém změni ubere jedno místo z kapacity kurzu
- 4) systém přiřadí aktérovi kurz do jeho osobního rozvrhu

## **UC07 – Zrušit registraci kurzu**

### **Anotace**

**Název:** Zrušit registraci kurzu

**Identifikátor:** UC07

**Podmínky před spuštěním:** aktér kliknul na tlačítko zrušit registraci

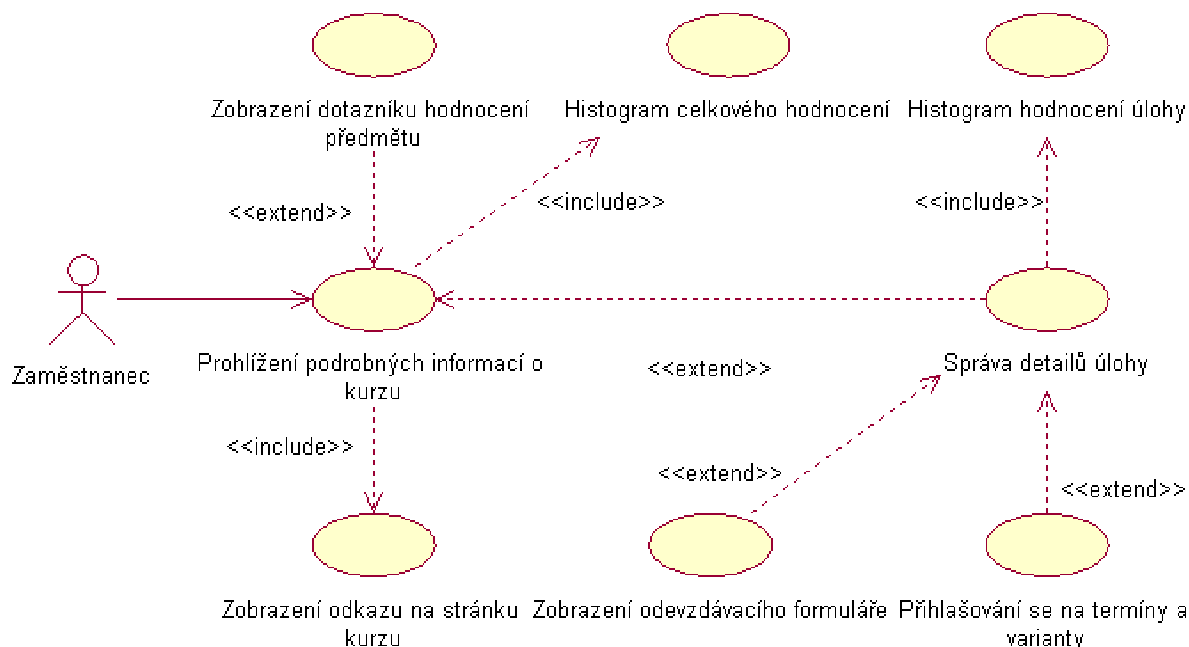
**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení akce
- 2) aktér akci potvrdí / zamítne
- 3) systém změni přidá jedno místo do kapacity kurzu
- 4) systém odstraní z aktérova rozvrhu záznam o kurzu

### 5.3.3. Prohlížení podrobných informací o kurzu z pohledu zaměstnance

Toto schéma popisuje možnosti aktéra při získávání informací o úlohách a také umožňuje aktérovi spravovat zadané úlohy z hlediska uživatele. Do tohoto stavu systému se aktér dostal buď kliknutím na odkaz v osobním rozvrhu nebo přehledu absolvovaných kurzů.



obr. 11 – Identifikace případů užití při prohlížení podrobných informací o kurzu

### UC08 – Prohlížení podrobných informací o kurzu

#### Anotace

**Název:** Prohlížení podrobných informací kurzu

**Identifikátor:** UC08

**Podmínky před spuštěním:** aktér se nachází v UC03 nebo UC05, je-li aktérem zaměstnanec provádí se hlavní tok, jinak alternativní

**Podmínky po ukončení:** (žádné)

#### Hlavní tok

- 1) aktér klikne v seznamu na „zkratku předmětu“ nebo „název kurzu“
- 2) systém zahrne UC10 (Zobrazení odkazu na stránku kurzu), UC11 (Histogram celkového hodnocení)
- 3) systém vygeneruje přehlednou tabulku obsahující chronologicky seřazené názvy úloh, včetně případného bodového hodnocení
- 4) aktér klikne na název úlohy
- 5) systém přejde do bodu rozšíření UC12 - Správa detailů úlohy

- 6) je-li kurz uzavřen a uživatel klikne na odkaz „dotazník hodnocení“, systém přejde do bodu rozšíření UC09 - Zobrazení dotazníku hodnocení předmětu

### **Alternativní tok**

- 1) aktér klikne v seznamu na „zkratku předmětu“ nebo „název kurzu“
- 2) systém vygeneruje přehlednou tabulku obsahující chronologicky seřazené názvy úloh, včetně případného bodového hodnocení
- 3) systém vytvoří dvě nové tlačítka „upravit“, „smazat“ u každé úlohy

### **Body rozšíření**

- UC09 „Zobrazení dotazníku hodnocení předmětu“
- UC12 „Správa detailů úlohy“

## **UC09 – Zobrazení dotazníku hodnocení předmětu**

### **Anotace**

**Název:** Zobrazení dotazníku hodnocení předmětu

**Identifikátor:** UC09

**Podmínky před spuštěním:** kurz musí mít status uzavřen

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje formulář dotazníku hodnocení předmětu
- 2) uživatel vyplní tento formulář a stiskne tlačítko „odeslat“
- 3) systém uloží informace z dotazníku a nastaví status kurzu na potvrzený

## **UC10 – Zobrazení odkazu na stránku kurzu**

### **Anotace**

**Název:** Zobrazení odkazu na stránku kurzu

**Identifikátor:** UC10

**Podmínky před spuštěním:** kurz má přiřazený veřejné nebo soukromé webové stránky

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje odkaz na stránky kurzu
- 2) uživatel vyplní tento formulář a stiskne tlačítko „odeslat“
- 3) systém uloží informace z dotazníku a nastaví status kurzu na potvrzený

## UC11 – Histogramu celkového hodnocení kurzu

### Anotace

**Název:** Histogram celkového hodnocení kurzu

**Identifikátor:** UC11

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém získá informace o bodovém hodnocení všech úloh všech účastníků kurzu
- 2) systém vygeneruje přehledný graf histogramu aktuálního hodnocení
- 3) systém vypočítá bodové pořadí zaměstnance v rámci školených zaměstnanců v tomto kurzu

## UC12 – Správa detailů úlohy

### Anotace

**Název:** Správa detailů úlohy

**Identifikátor:** UC12

**Podmínky před spuštěním:** aktér kliknul na název úlohy

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje anotaci úlohy
- 2) systém vygeneruje další informace o úloze, včetně podrobného popisu
- 3) systém zahrne UC13 - Histogram hodnocení úlohy
- 4) jsou-li úloze přiřazeny termíny či varianty k zapisování a uživatel klikne na odkaz „přihlásit na termín“ systém přejde do bodu rozšíření UC14 - Přihlašování se na termíny a varianty
- 5) má-li úloha vlastnosti odevzdávání a uživatel klikne na odkaz „odevzdání souborů“, systém přejde do UC15 - Zobrazení odevzdávacího formuláře

### Body rozšíření

- UC14 „Přihlašování se na termíny a varianty“
- UC15 „Zobrazení odevzdávacího formuláře“

## UC13 – Histogram hodnocení úlohy

### Anotace

**Název:** Histogram hodnocení úlohy

**Identifikátor:** UC13

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém získá informace o bodovém hodnocení úlohy všech účastníků kurzu
- 2) systém vygeneruje přehledný graf histogramu hodnocení

## UC14 – Přihlašování se na termíny a varianty

### Anotace

**Název:** Přihlašování se na termíny a varianty

**Identifikátor:** UC14

**Podmínky před spuštěním:** aktér kliknul na odkaz „přihlásit se na termíny“, úloha tuto volbu umožňuje

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje seznam dostupných termínů / variant
- 2) aktér klikne na název vybraného termínu
- 3) systém přiřadí uživatele k termínu

## UC15 – Zobrazení odevzdávacího formuláře

### Anotace

**Název:** Zobrazení odevzdávacího formuláře

**Identifikátor:** UC15

**Podmínky před spuštěním:** aktér kliknul na odkaz „odevzdat soubory“, úloha tuto volbu umožňuje

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje odkaz přiložit soubor
- 2) systém vygeneruje seznam aktuálně přiložených souborů s tlačítkem odstranit

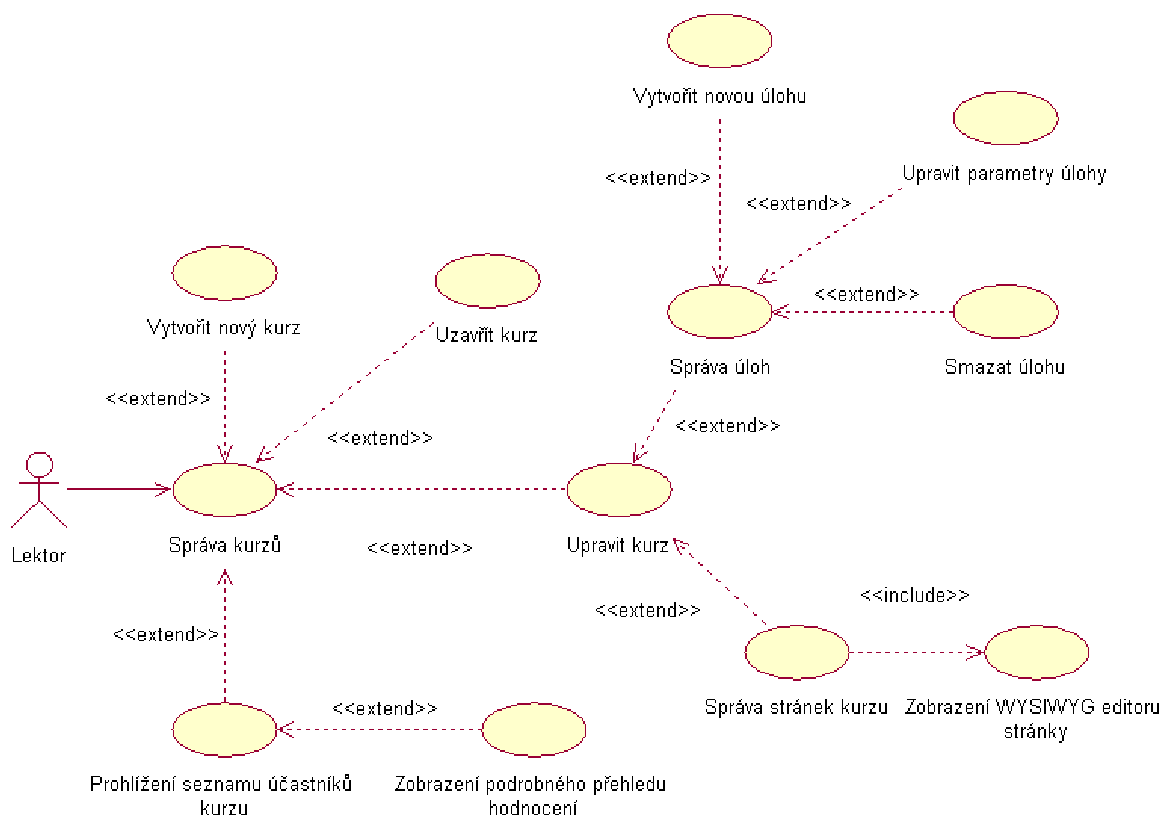
- 3) aktér klikne na odkaz přiložit soubor (lze i opakovaně)
- 4) systém vygeneruje příslušný počet polí formuláře pro přiložení souboru
- 5) aktér vybere přes standardní okno procházení konkrétní soubor
- 6) aktér klikne na tlačítko „odeslat“
- 7) systém uloží soubory do systému

### Alternativní tok

- 1) systém vygeneruje odkaz přiložit soubor
- 2) systém vygeneruje seznam aktuálně přiložených souborů s tlačítkem odstranit
- 3) aktér klikne na tlačítko odstranit
- 4) systém odstraní odkaz na soubor z databáze
- 5) systém odstraní soubor, včetně adresářového zanoření ze serveru

## 5.3.4. Správa kurzů z pohledu lektora

Toto schéma popisuje možnosti aktéra při vytváření a úpravě kurzů, tvorbě webových stránek kurzu a zadávání jednotlivých úloh včetně jejich hodnocení.



obr. 12 – Identifikace případů použití vztahujících se ke správě kurzů z pohledu lektora

## UC16 – Správa kurzů

### Anotace

**Název:** Správa kurzů

**Identifikátor:** UC16

**Podmínky před spuštěním:** aktér je přihlášen jako lektor

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) aktér klikne na položku v menu s názvem „správa kurzů“
- 2) systém získá seznam vyučovaných kurzů aktérem
- 3) systém zobrazí tento seznam ve formě přehledné tabulky
- 4) systém kurzy se statutem uzavřeno umístí na konec seznamu, vyznačí šedou barvou a seřadí dle data sestupně od nejpozději založeného
- 5) aktér klikne na tlačítko „vytvořit nový kurz“
- 6) systém přejde do bodu rozšíření UC17 – Vytvořit nový kurz

### **Alternativní tok č. 1**

- 1) aktér klikne na tlačítko „uzavřít“
- 2) systém přejde do bodu rozšíření UC18 – Uzavřít kurz

### **Alternativní tok č. 2**

- 1) aktér klikne na tlačítko „hodnocení“
- 2) systém přejde do bodu rozšíření UC19 – Prohlížení seznamu účastníků kurzu

### **Alternativní tok č. 3**

- 1) aktér klikne na tlačítko „upravit“
- 2) systém přejde do bodu rozšíření UC21 – Upravit kurz

### **Body rozšíření**

- UC17 „*Vytvořit nový kurz*“
- UC18 „*Uzavřít kurz*“
- UC19 „*Prohlížení účastníků kurzu*“
- UC21 „*Upravit kurz*“

## **UC17 – Vytvořit nový kurz**

### **Anotace**

**Název:** Vytvořit nový kurz

**Identifikátor:** UC17

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)



### **Hlavní tok**

- 1) systém vygeneruje formulář obsahující všechny položky kurzu
- 2) aktér vyplní tento formulář
- 3) aktér stiskne tlačítko „odeslat“
- 4) systém zkontroluje vstupní data
- 5) systém uloží data o kurzu

### **UC18 – Uzavřít kurz**

#### **Anotace**

**Název:** Uzavřít kurz

**Identifikátor:** UC18

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém zobrazí formulář s přepínačem statutu kurzu ve formě comboboxu
- 2) uživatel přepne přepínač ze stavu otevřeno na stav uzavřeno
- 3) uživatel stiskne tlačítko „uložit“
- 4) systém uloží nový status kurzu

### **UC19 – Prohlížení seznamu účastníků kurzu**

#### **Anotace**

**Název:** Prohlížení seznamu účastníků kurzu

**Identifikátor:** UC19

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém získá seznam účastníků kurzu
- 2) systém získá součet celkového hodnocení úloh účastníka v aktuálním kurzu
- 3) systém vygeneruje seznam účastníků včetně hodnocení v přehledné tabulce
- 4) aktér klikne na tlačítko „hodnotit“
- 5) systém přejde do bodu rozšíření UC20 – Zobrazení podrobného přehledu hodnocení

#### **Body rozšíření**

- UC20 „Zobrazení podrobného přehledu hodnocení“

## **UC20 – Zobrazení podrobného přehledu hodnocení**

### **Anotace**

**Název:** Zobrazení podrobného přehledu hodnocení

**Identifikátor:** UC20

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém získá výsledky hodnocení jednotlivých úloh
- 2) systém vygeneruje seznam úloh a odevzdaných souborů včetně hodnocení do formuláře
- 3) uživatel upravuje hodnocení jednotlivých úloh
- 4) uživatel klikne na tlačítko „uložit změny“
- 5) systém uloží informace o hodnocení jednotlivých úloh

## **UC21 – Upravit kurz**

### **Anotace**

**Název:** Upravit kurz

**Identifikátor:** UC21

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém zobrazí formulář editace vlastností kurzu
- 2) systém zobrazí dvě tlačítka: „správa úloh“, „správa stránek kurzu“
- 3) aktér vyplní tento formulář
- 4) aktér klikne na tlačítko odeslat ke schválení
- 5) systém uloží informace se změnami

### **Alternativní tok č. 1**

- 1) aktér klikne na tlačítko „správa stránek kurzu“
- 2) systém přejde do bodu rozšíření UC22 – Správa stránek kurzu

### **Alternativní tok č. 2**

- 1) aktér klikne na tlačítko „správa úloh“
- 2) systém přejde do bodu rozšíření UC24 – Správa úloh

### **Body rozšíření**

- UC22 „Správa stránek kurzu“

## UC22 – Správa stránek kurzu

### Anotace

**Název:** Správa stránek kurzu

**Identifikátor:** UC22

**Podmínky před spuštěním:** aktér kliknul na tlačítko správa stránek kurzu

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje dvě záložky, „vytvořit veřejnou stránku kurzu“ a „vytvořit soukromou stránku kurzu“
- 2) aktér klikne na některé z těchto tlačítek
- 3) systém zahrne UC23

## UC23 – Zobrazení WYSIWYG editoru stránky

### Anotace

**Název:** Zobrazení WYSIWYG editoru stránky

**Identifikátor:** UC23

**Podmínky před spuštěním:** hlavní tok – aktér se nachází v záložce veřejné stránky, alternativní tok č.1 – aktér se nachází v záložce soukromé stránky, alternativní tok č. 2 – aktér vytváří osobní stránku v UC38

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje okno dynamického HTML editoru
- 2) aktér vytvoří v tomto editoru webovou stránku
- 3) systém uloží tuto stránku jako veřejnou

### Alternativní tok č.1

- 1) systém vygeneruje okno dynamického HTML editoru
- 2) aktér vytvoří v tomto editoru webovou stránku
- 3) systém uloží tuto stránku jako soukromou

### Alternativní tok č.2

- 1) systém vygeneruje okno dynamického HTML editoru
- 2) aktér vytvoří v tomto editoru webovou stránku
- 3) systém uloží tuto stránku jako osobní

## UC24 – Správa úloh

### Anotace

**Název:** Správa úloh

**Identifikátor:** UC24

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje tlačítko „vytvořit úlohu“
- 2) systém zahrne UC10 (Zobrazení odkazu na stránku kurzu)
- 3) systém vygeneruje přehlednou tabulku obsahující chronologicky seřazené názvy úloh, včetně případného bodového hodnocení
- 4) aktér klikne na tlačítko „vytvořit úlohu“
- 5) systém přejde do bodu rozšíření UC25 – Vytvořit novou úlohu

### Alternativní tok č. 1

- 1) aktér klikne na tlačítko „upravit parametry úlohy“
- 2) systém přejde do bodu rozšíření UC26 – Upravit parametry úlohy

### Alternativní tok č. 2

- 1) aktér klikne na tlačítko „smazat úlohu“
- 2) systém přejde do bodu rozšíření UC27 – Smazat úlohu

### Body rozšíření

- UC25 „Vytvořit novou úlohu“
- UC26 „Upravit parametry úlohy“
- UC27 „Smazat úlohu“

## UC25 – Vytvořit novou úlohu

### Anotace

**Název:** Vytvořit novou úlohu

**Identifikátor:** UC25

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém zobrazí formulář s editačními položkami úlohy
- 2) systém zobrazí přepínací tlačítko „zveřejnit“ / „nezveřejnit“ ve formě comboboxu
- 3) aktér vyplní tento formulář a vybere možnost z comboboxu
- 4) aktér klikne na tlačítko odeslat
- 5) systém uloží informace o nové úloze

### **UC26 – Upravit parametry úlohy**

#### **Anotace**

**Název:** Upravit parametry úlohy

**Identifikátor:** UC26

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém zobrazí formulář s editačními položkami úlohy předvyplněný informacemi o aktuální úloze
- 2) systém zobrazí přepínací tlačítko „zveřejnit“ / „nezveřejnit“ ve formě comboboxu
- 3) aktér pozmění obsah tohoto formuláře a vybere možnost z comboboxu
- 4) aktér klikne na tlačítko odeslat
- 5) systém uloží informace o změně úlohy

### **UC27 – Smazat úlohu**

#### **Anotace**

**Název:** Smazat úlohu

**Identifikátor:** UC27

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém zkontroluje, jedná-li se o úlohu zveřejněnou
- 2) systém zobrazí okno s výstrahou smazání úlohy
- 3) systém v tomto okně vypíše upozornění na případné zapsané hodnocení
- 4) aktér potvrdí volbu tlačítkem „opravdu smazat“
- 5) systém vymaže informace o úloze a zruší bodové hodnocení

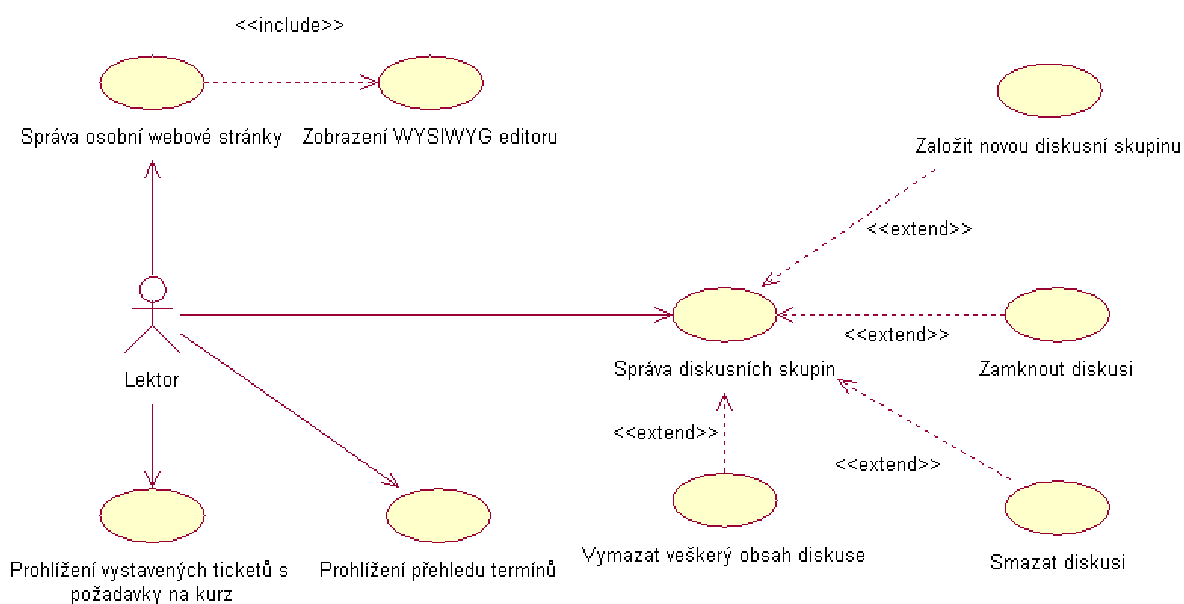
#### **Alternativní tok**

- 1) systém zkontroluje jedná-li se o úlohu nezveřejněnou

- 2) systém zobrazí okno s výstrahou smazání úlohy
- 3) aktér potvrdí volbu tlačítkem „opravdu smazat“
- 4) systém vymaže informace o úloze

### 5.3.5. Ostatní případy užití z pohledu lektora

Toto schéma popisuje možnosti další možnosti aktéra mimo správu a hodnocení kurzů.



obr. 13 – Identifikace ostatních případů použití z pohledu lektora

## UC28 – Správa diskusních skupin

### Anotace

**Název:** Správa diskusních skupin

**Identifikátor:** UC28

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) aktér klikne na odkaz „správa diskusních skupin“
- 2) systém vygeneruje seznam všech diskusních skupin
- 3) systém vygeneruje u každé skupiny tři tlačítka v podobě ikon „smazání“, „vymazání obsahu“, „zamčení“
- 4) aktér klikne na tlačítko „založit novou diskusní skupinu“
- 5) systém přejde do bodu rozšíření UC32

### **Alternativní tok č. 1**

- 1) aktér klikne na ikonu zámku v přehledu diskusních skupin
- 2) systém přejde do bodu rozšíření UC29 – Zamknout diskusi

### **Alternativní tok č. 2**

- 1) aktér klikne na ikonu koše v přehledu diskusních skupin
- 2) systém přejde do bodu rozšíření UC30 – Vymazat diskusi

### **Alternativní tok č. 3**

- 1) aktér klikne na ikonu vysypaného koše v přehledu diskusních skupin
- 2) systém přejde do bodu rozšíření UC31 – Vymazat veškerý obsah diskuse

### **Body rozšíření**

- UC29 „Zamknout diskusi“
- UC30 „Vymazat diskusi“
- UC31 „Vymazat veškerý obsah diskuse“
- UC32 „Založit novou diskusní skupinu“

## **UC29 – Zamknout diskusi**

### **Anotace**

**Název:** Zamknout diskusi

**Identifikátor:** UC29

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje javascriptový alert žádající potvrzení akce
- 2) uživatel akci potvrdí / zamítne
- 3) systém zamkne skupinu – přepne ikonu zámku z odemčeno na zamčeno

## **UC30 – Vymazat diskusi**

### **Anotace**

**Název:** Vymazat diskusi

**Identifikátor:** UC30

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje javascriptový alert žádající potvrzení akce
- 2) uživatel akci potvrdí / zamítne
- 3) systém odstraní celou diskusi
- 4) systém odstraní příslušné odkazy na diskusi v ostatních případech použití

### **UC31 – Vymazat obsah diskuse**

#### **Anotace**

**Název:** Vymazat obsah diskuse

**Identifikátor:** UC31

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení akce
- 2) uživatel akci potvrdí / zamítne
- 3) systém vymaže obsah diskusní skupiny

### **UC32 – Vytvoření nové diskusní skupiny**

#### **Anotace**

**Název:** Vytvoření nové diskusní skupiny

**Identifikátor:** UC32

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém vygeneruje formulář založení diskusní skupiny
- 2) aktér vyplní formulář, kde také vybere ke kterému kurzu si přeje diskusní skupinu přiřadit
- 3) po odeslání formuláře aktérem systém vytvoří odkaz na diskusní skupinu v příslušných případech použití

### **UC33 – Správa osobní webové stránky**

#### **Anotace**

**Název:** Správa osobní webové stránky



**Identifikátor:** UC33

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje odkaz „vytvořit osobní webovou stránku“
- 2) aktér klikne na tento odkaz
- 3) systém zahrne UC23 – Zobrazení WYSIWYG editoru

## **UC34 – Prohlížení přehledu termínů**

### **Anotace**

**Název:** Prohlížení přehledu termínů

**Identifikátor:** UC34

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) aktér klikne na tlačítko „přehled termínů“
- 2) systém získá informace o kurzech a jejich jednotlivých úlohách
- 3) systém seřadí kurzy od nejbližšího termínu odevzdání zadané úlohy
- 4) systém vygeneruje přehledný harmonogram jednotlivých kurzů
- 5) aktér klikne na název kurzu termínu
- 6) systém přejde do bodu rozšíření UC19 – Prohlížení seznamu účastníků kurzu

### **Body rozšíření**

- UC19 „*Prohlížení seznamu účastníků kurzu*“

## **UC35 – Prohlížení ticketů s požadavky na kurz**

### **Anotace**

**Název:** Prohlížení ticketů s požadavky na kurz

**Identifikátor:** UC35

**Podmínky před spuštěním:** (žádné)

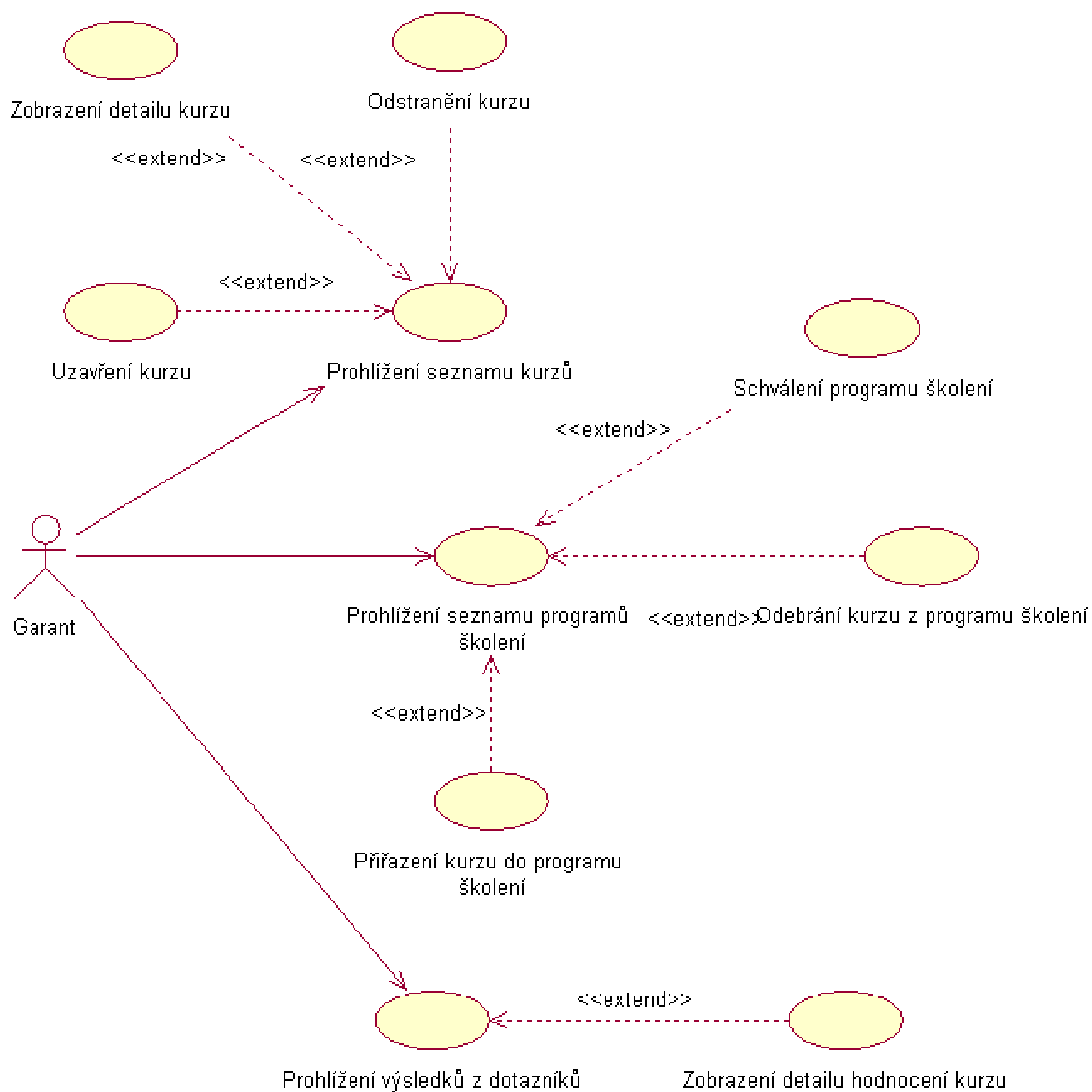
**Podmínky po ukončení:** (žádné)

## Hlavní tok

- 1) systém připojí k tlačítku „prohlížet nabídku ticketů“ údaj o počtu nových ticketů od poslední kontroly aktérem
- 2) aktér klikne na tlačítko „prohlížet nabídku ticketů“
- 3) systém získá data o vytvořených ticketech
- 4) systém vygeneruje seznam všech ticketů
- 5) systém vyznačí šedou barvou tickety jimž byl lektorem přiřazen některý z vytvořených kurzů
- 6) aktér klikne na konkrétní ticket
- 7) systém vygeneruje přehled informací v konkrétním ticketu

### 5.3.6. Případy použití z pohledu garanta

Toto schéma popisuje možnosti aktéra při správě programů školení, správě kurzů a při vyhodnocování zpětné vazby z dotazníků.



obr. 14 – Identifikace všech případů použití z pohledu garanta

#### UC36 – Prohlížení seznamu kurzů

##### Anotace

**Název:** Prohlížení seznamu kurzů

**Identifikátor:** UC36

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

## Hlavní tok

- 1) aktér klikne na položku v menu s názvem „prohlížet všechny kurzy“
- 2) systém vygeneruje seznam všech diskusních skupin
- 3) systém vygeneruje u každé skupiny tři tlačítka v podobě ikon „smazání“, „vymazání obsahu“, „zamčení“
- 4) aktér vybere řádek s kurzem
- 5) aktér klikne na tlačítko „detaily“
- 6) systém přejde do bodu rozšíření UC37 – Zobrazení detailu kurzu

## Alternativní tok č. 1

- 1) aktér klikne na tlačítko „uzavřít“
- 2) systém přejde do bodu rozšíření UC18 – Uzavřít kurz

## Alternativní tok č. 2

- 1) aktér klikne na tlačítko „odstranit“
- 2) systém přejde do bodu rozšíření UC38 – Odstranit kurz

## Body rozšíření

- UC18 „Uzavřít kurz“
- UC37 „Zobrazení detailu kurzu“
- UC38 „Odstranit kurz“

## UC37 – Zobrazení detailu kurzu

### Anotace

**Název:** Zobrazení detailu kurzu

**Identifikátor:** UC37

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

## Hlavní tok

- 1) systém získá informace o zvoleném kurzu
- 2) systém zobrazí tyto informace v podrobném přehledu
- 3) systém zobrazí formulář obsahující přepínač schváleno / neschváleno, výchozí status je neschváleno
- 4) aktér přepne status na schváleno
- 5) aktér klikne na tlačítko odeslat
- 6) systém uloží status

## **UC38 – Odstranit kurz**

### **Anotace**

**Název:** Odstranit kurz

**Identifikátor:** UC38

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení aktérem
- 2) aktér potvrdí odstranění
- 3) systém přepne status kurzu na odstraněno

## **UC39 – Prohlížení seznamu programů školení**

### **Anotace**

**Název:** Prohlížení seznamu programů školení

**Identifikátor:** UC39

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém získá informace o programech školení
- 2) systém vygeneruje přehlednou tabulku s programem školení, včetně přiřazených kurzů
- 3) systém vygeneruje u každého programu tlačítko přidat kurz, schválit a u každého přiřazeného kurzu tlačítko odebrat
- 4) aktér klikne na tlačítko přidat kurz
- 5) systém přejde do bodu rozšíření UC40 – Přidání kurzu do programu školení

### **Alternativní tok č.1**

- 1) systém získá informace o programech školení
- 2) systém vygeneruje přehlednou tabulku s programem školení včetně přiřazených kurzů
- 3) systém vygeneruje u každého programu tlačítko přidat kurz, schválit a u každého přiřazeného kurzu tlačítko odebrat
- 4) aktér klikne na tlačítko „odebrat“
- 5) systém přejde do bodu rozšíření UC41 – Odebrání kurzu z programu školení

### **Alternativní tok č.2**

- 1) systém získá informace o programech školení

- 2) systém vygeneruje přehlednou tabulku s programem školení, včetně přiřazených kurzů
- 3) systém vygeneruje u každého programu tlačítko přidat kurz, schválit a u každého přiřazeného kurzu tlačítko odebrat
- 4) aktér klikne na tlačítko „schválit“
- 5) systém přejde do bodu rozšíření UC42 – Schválení programu školení

### **Body rozšíření**

- UC40 „Přidání kurzu do programu školení“
- UC41 „Odebrání kurzu z programu školení“
- UC42 „Schválení programu školení“

## **UC40 – Přidání kurzu do programu školení**

### **Anotace**

**Název:** Přidání kurzu do programu školení

**Identifikátor:** UC40

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje přehled všech kurzů, mimo status odstraněno
- 2) aktér vybere z přehledu kurz, který si přeje do programu školení zařadit
- 3) systém uloží informace o přiřazení

## **UC41 – Odebrání kurzu z programu školení**

### **Anotace**

**Název:** Odebrání kurzu z programu školení

**Identifikátor:** UC41

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení aktérem
- 2) aktér potvrdí odebrání kliknutím na tlačítko „potvrdit“
- 3) systém odstraní záznam o přiřazení kurzu k programu školení

## UC42 – Schválení programu školení

### Anotace

**Název:** Schválení programu školení

**Identifikátor:** UC42

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje alert okno žádající potvrzení aktérem
- 2) aktér potvrdí schválení kliknutím na tlačítko „potvrdit“
- 3) systém zruší možnost odebrání kurzu v UC39 a vyznačí schválené záznamy šedou barvou
- 4) systém změní tlačítko u programu ze „schválit“ na „zrušit schválení“

## UC43 – Prohlížení výsledků z dotazníků

### Anotace

**Název:** Prohlížení výsledků z dotazníků

**Identifikátor:** UC43

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) aktér klikne na položku v menu s názvem „prohlížet výsledky dotazníků“
- 2) systém vygeneruje seznam všech kurzů se statutem uzavřeno
- 3) systém vygeneruje u každého kurzu tlačítko detaily
- 4) aktér vybere řádek s kurzem
- 5) aktér klikne na tlačítko „detaily“
- 6) systém přejde do bodu rozšíření UC44 – Zobrazení detailu hodnocení kurzu

### Body rozšíření

- UC44 „Zobrazení detailu hodnocení kurzu“

## UC44 – Zobrazení detailu hodnocení kurzu

### Anotace

**Název:** Zobrazení detailu hodnocení kurzu

**Identifikátor:** UC44

**Podmínky před spuštěním:** (žádné)

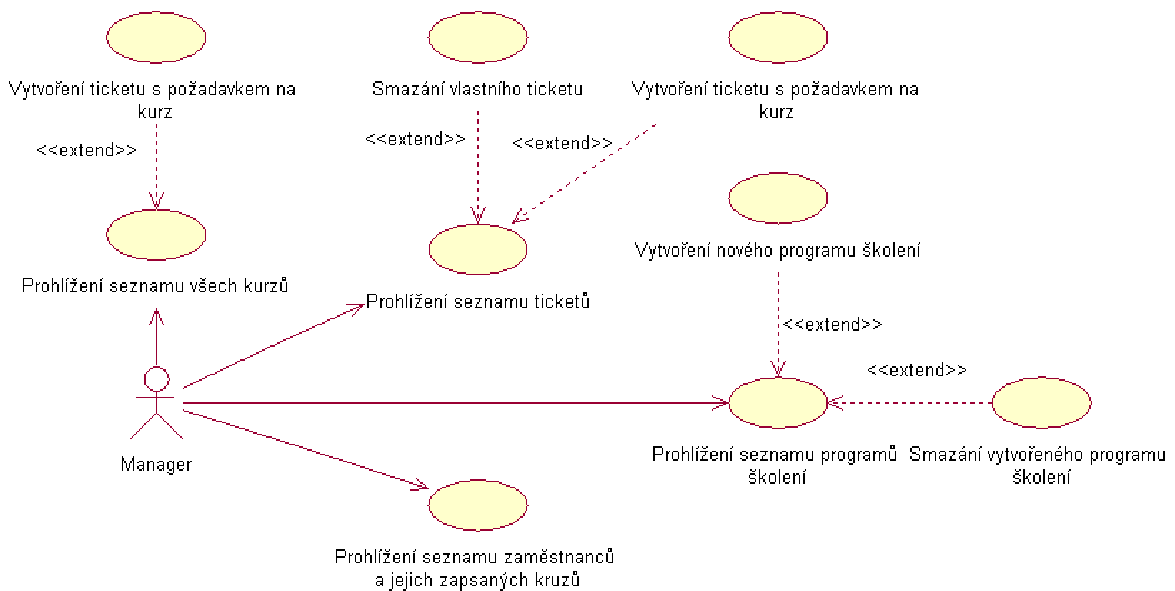
**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje seznam zodpovězených otázek v podobě výsledných grafů
- 2) aktér kliknutím na ikonu tiskárny vytiskne výsledné grafy

## 5.3.7. Případy použití z pohledu managera

Toto schéma popisuje možnosti aktéra při správě vytváření ticketů, sestavování programů školení a kontrolu zapsaných kurzů u jednotlivých zaměstnanců.



obr. 15 – Identifikace všech případů použití z pohledu managera

### UC45 – Prohlížení seznamu ticketů

#### Anotace

**Název:** Prohlížení seznamu ticketů

**Identifikátor:** UC45

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### Hlavní tok

- 1) systém získá informace o všech ticketech
- 2) systém barevně odliší vlastní tickety od ticketů ostatních aktérů
- 3) systém vygeneruje tlačítko „vytvořit ticket“



- 4) systém u vlastních ticketů vygeneruje tlačítko „smazat ticket“
- 5) aktér klikne na tlačítko „vytvořit ticket“
- 6) systém přejde do bodu rozšíření UC46 – Vytvoření ticketu s požadavkem na kurz

### **Alternativní tok č.1**

- 1) systém získá informace o všech ticketech
- 2) systém barevně odliší vlastní tickety od ticketů ostatních aktérů
- 3) systém vygeneruje tlačítko „vytvořit ticket“
- 4) systém u vlastních ticketů vygeneruje tlačítko „smazat ticket“
- 5) aktér klikne na tlačítko „smazat ticket“
- 6) systém přejde do bodu rozšíření UC47 – Smazání vlastního ticketu

### **Body rozšíření**

- UC46 „Vytvoření ticketu s požadavkem na kurz“
- UC47 „Smazání vlastního ticketu“

## **UC46 – Vytvoření ticketu s požadavkem na kurz**

### **Anotace**

**Název:** Vytvoření ticketu s požadavkem na kurz

**Identifikátor:** UC46

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje formulář pro vytvoření nového ticketu
- 2) aktér vyplní formulář, kde uvede veškeré informace o představě nového kurzu
- 3) systém po odeslání vytvoří nový ticket a umístí jej do seznamu ticketů

## **UC47 – Smazání vlastního ticketu**

### **Anotace**

**Název:** Smazání vlastního ticketu

**Identifikátor:** UC47

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení aktérem

- 2) aktér potvrdí smazání kliknutím na tlačítko „smazat“
- 3) systém odstraní ticket ze seznamu ticketů

## **UC48 – Prohlížení seznamu všech kurzů**

### **Anotace**

**Název:** Prohlížení seznamu všech kurzů

**Identifikátor:** UC48

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém získá informace o všech schválených kurzech
- 2) systém vygeneruje přehlednou tabulku obsahující všechny schválené kurzy
- 3) systém vygeneruje tlačítko „vytvořit ticket“
- 4) aktér klikne na tlačítko „vytvořit ticket“
- 5) systém přejde do bodu rozšíření UC46 – Vytvoření ticketu s požadavkem na kurz

### **Body rozšíření**

- UC46 „Vytvoření ticketu s požadavkem na kurz“

## **UC49 – Prohlížení seznamu programů školení**

### **Anotace**

**Název:** Prohlížení seznamu programů školení

**Identifikátor:** UC49

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém získá informace o všech vytvořených programech školení a přiřazených kurzech
- 2) systém barevně odliší vlastní programy od programů ostatních managerů
- 3) systém zobrazí aktuálně přiřazené kurzy
- 4) systém u vlastních vytvořených programů vygeneruje tlačítko „smazat program“
- 5) systém vygeneruje tlačítko „nový program“
- 6) aktér klikne na tlačítko „nový program“
- 7) systém přejde do bodu rozšíření UC50 – Vytvoření nového programu školení

### **Alternativní tok č.1**

- 1) systém získá informace o všech vytvořených programech školení a přiřazených kurzech

- 2) systém barevně odliší vlastní programy od programů ostatních managerů
- 3) systém zobrazí aktuálně přiřazené kurzy
- 4) systém u vlastních vytvořených programů vygeneruje tlačítko „smazat program“
- 5) systém vygeneruje tlačítko „nový program“
- 6) aktér klikne na tlačítko „smazat program“
- 7) systém přejde do bodu rozšíření UC51 – Smazání vytvořeného programu školení

### **Body rozšíření**

- UC50 „Vytvoření nového programu školení“
- UC51 „Smazání vytvořeného programu školení“

## **UC50 – Vytvoření nového programu školení**

### **Anotace**

**Název:** Vytvoření nového programu školení

**Identifikátor:** UC50

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje formulář pro vytvoření nového ticketu
- 2) aktér vyplní formulář, kde uvede veškeré informace o představě nového kurzu
- 3) systém po odeslání vytvoří nový ticket a umístí jej do seznamu ticketů

## **UC51 – Smazání vytvořeného programu školení**

### **Anotace**

**Název:** Smazání vytvořeného programu školení

**Identifikátor:** UC51

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení aktérem
- 2) aktér potvrdí smazání kliknutím na tlačítko „smazat“
- 3) systém odstraní program ze seznamu programů školení

## **UC52 – Prohlížení seznamu zaměstnanců a jejich zapsaných kurzů**

### **Anotace**

**Název:** Prohlížení seznamu zaměstnanců a jejich zapsaných kurzů

**Identifikátor:** UC52

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

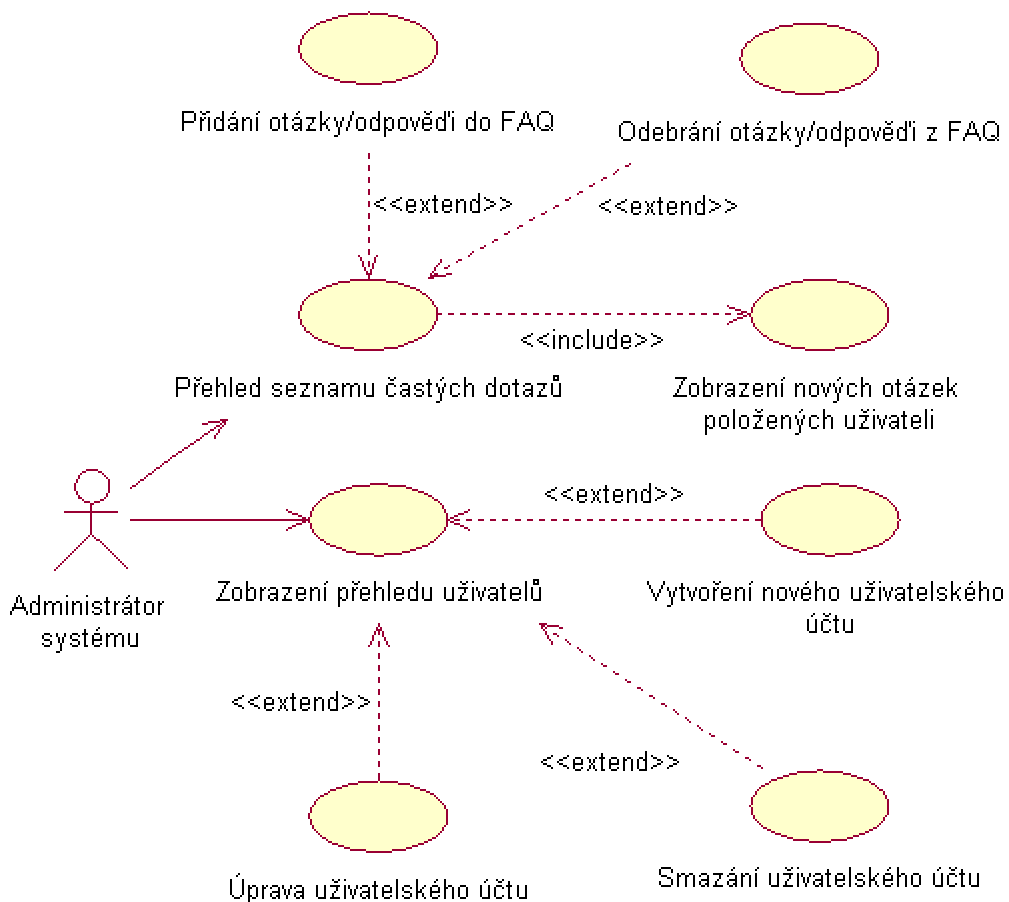
- 1) systém získá informace o všech zaměstnancích a jejich kurzech
- 2) systém zobrazí seznam všech zaměstnanců
- 3) systém zobrazí formulář pro vyhledávání v seznamu zaměstnanců
- 4) aktér klikne na jméno v seznamu
- 5) systém rozbálí pod jménem tabulku s přehledem zapsaných kurzů

### **Alternativní tok**

- 1) systém získá informace o všech zaměstnancích a jejich kurzech
- 2) systém zobrazí seznam všech zaměstnanců
- 3) systém zobrazí formulář pro vyhledávání v seznamu zaměstnanců
- 4) aktér vyplní políčko hledat ve formuláři vyhledávání a odešle požadavek
- 5) systém vyhledá příslušné uživatele dle jména / loginu / role
- 6) systém zobrazí výsledek v podobě redukce seznamu

### 5.3.8. Případy použití z pohledu administrátora

Toto schéma popisuje společné možnosti administrátora při správě často kladených dotazů a administraci uživatelských účtů.



obr. 16 – Identifikace všech případů použití z pohledu administrátora systému

### UC53 – Přehled seznamu častých dotazů

#### Anotace

**Název:** Přehled seznamu častých dotazů

**Identifikátor:** UC53

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

#### Hlavní tok

- 1) systém získá seznam všech zodpovězených často kladených dotazů
- 2) existují-li nově položené dotazy z webového formuláře, systém zahrne UC54 (Zobrazení nových otázek položených uživateli)
- 3) systém u každé položky vygeneruje tlačítko „vymazat položku“

- 4) systém vygeneruje tlačítko „nová položka“
- 5) aktér klikne na tlačítko „nová položka“
- 6) systém přejde do bodu rozšíření UC55 – Přidání otázky/odpovědi do FAQ

#### **Alternativní tok**

- 1) systém získá seznam všech zodpovězených často kladených dotazů
- 2) existují-li nově položené dotazy z webového formuláře, systém zahrne UC54 (Zobrazení nových otázek položených uživateli)
- 3) systém u každé položky vygeneruje tlačítko „vymazat položku“
- 4) systém vygeneruje tlačítko „nová položka“
- 5) aktér klikne na tlačítko „vymazat položku“
- 6) systém přejde do bodu rozšíření UC56 – Odebrání otázky/odpovědi z FAQ

#### **Body rozšíření**

- UC55 „Přidání otázky/odpovědi do FAQ“
- UC56 „Odebrání otázky/odpovědi z FAQ“

### **UC54 – Zobrazení nových otázek položených uživateli**

#### **Anotace**

**Název:** Přidání otázky/odpovědi do FAQ

**Identifikátor:** UC54

**Podmínky před spuštěním:** systém obsahuje nové nezodpovězené otázky

**Podmínky po ukončení:** (žádné)

#### **Hlavní tok**

- 1) systém získá seznam všech nově položených otázek, které mají status „nezodpovězeno“
- 2) systém zobrazí tento přehled v tabulce
- 3) systém vytvoří u každé položky tlačítka „odpovědět“, „neodpovídat“
- 4) aktér klikne na tlačítko „odpovědět“
- 5) systém přejde do bodu rozšíření UC55 – Přidání otázky/odpovědi do FAQ

#### **Alternativní tok**

- 1) systém získá seznam všech nově položených otázek, které mají status „nezodpovězeno“
- 2) systém zobrazí tento přehled v tabulce
- 3) systém vytvoří u každé položky tlačítka „odpovědět“, „neodpovídat“
- 4) aktér klikne na tlačítko „neodpovídat“
- 5) systém nastaví nové otázce status „zodpovězeno“

## **UC55 – Přidání otázky/odpovědi do FAQ**

### **Anotace**

**Název:** Přidání otázky/odpovědi do FAQ

**Identifikátor:** UC55

**Podmínky před spuštěním:** alternativní tok proběhne, je-li tento případ zavolán z UC54

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje formulář pro zadání dvojice otázka – odpověď
- 2) aktér vyplní formulář
- 3) systém po odeslání uloží nový záznam do seznamu FAQ

### **Alternativní tok**

- 1) systém vygeneruje formulář pro zadání dvojice otázka – odpověď, který je předvyplněn dle hodnot z UC54
- 2) aktér vyplní formulář
- 3) systém po odeslání uloží nový záznam do seznamu FAQ
- 4) systém přiřadí nové otázce status „zodpovězeno“

## **UC56 – Odebrání otázky/odpovědi z FAQ**

### **Anotace**

**Název:** Odebrání otázky/odpovědi z FAQ

**Identifikátor:** UC56

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém vygeneruje alert okno žádající potvrzení aktérem
- 2) aktér potvrdí smazání kliknutím na tlačítko „smazat“
- 3) systém odstraní záznam ze seznamu FAQ

## **UC57 – Zobrazení přehledu uživatelů**

### **Anotace**

**Název:** Zobrazení přehledu uživatelů

**Identifikátor:** UC57

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém získá seznam všech uživatelů
- 2) systém vygeneruje seznam uživatelů v přehledné tabulce
- 3) systém u každého uživatele vytvoří tlačítko „upravit“, „smazat“
- 4) systém vygeneruje tlačítko „vytvořit nový účet“
- 5) aktér klikne na tlačítko „vytvořit nový účet“
- 6) systém přejde do bodu rozšíření UC58 – Vytvoření nového uživatelského účtu

### **Alternativní tok č.1**

- 1) systém získá seznam všech uživatelů
- 2) systém vygeneruje seznam uživatelů v přehledné tabulce
- 3) systém u každého uživatele vytvoří tlačítko „upravit“, „smazat“
- 4) systém vygeneruje tlačítko „vytvořit nový účet“
- 5) aktér klikne na tlačítko „upravit“
- 6) systém přejde do bodu rozšíření UC59 – Úprava uživatelského účtu

### **Alternativní tok č.2**

- 1) systém získá seznam všech uživatelů
- 2) systém vygeneruje seznam uživatelů v přehledné tabulce
- 3) systém u každého uživatele vytvoří tlačítko „upravit“, „smazat“
- 4) systém vygeneruje tlačítko „vytvořit nový účet“
- 5) aktér klikne na tlačítko „smazat“
- 6) systém přejde do bodu rozšíření UC60 – Smazání uživatelského účtu

### **Body rozšíření**

- UC58 „*Vytvoření nového uživatelského účtu*“
- UC59 „*Úprava uživatelského účtu*“
- UC60 „*Smazání uživatelského účtu*“

## **UC58 – Vytvoření nového uživatelského účtu**

### **Anotace**

**Název:** Úprava uživatelského účtu

**Identifikátor:** UC58

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)



## Hlavní tok

- 1) systém vygeneruje formulář uživatelských údajů
- 2) systém zobrazí přepínač práv, ve kterém jsou uvedena všechna ostatní oprávnění
- 3) aktér upraví položky ve formuláři a nastaví správně přepínač na požadované oprávnění
- 4) systém po odeslání uloží informace o uživateli a vytvoří nový uživatelský účet
- 5) systém vygeneruje specifické uživatelské heslo
- 6) systém poté odešle na zadaný email informace o vytvoření uživatelského účtu s přihlašovacími údaji

## UC59 – Úprava uživatelského účtu

### Anotace

**Název:** Úprava uživatelského účtu

**Identifikátor:** UC59

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje formulář pro úpravu uživatelských údajů a předvyplní jej aktuálními daty
- 2) systém zobrazí přepínač práv, ve kterém jsou uvedena všechna ostatní oprávnění
- 3) aktér upraví položky ve formuláři a nastaví správně přepínač na požadované oprávnění
- 4) systém po odeslání uloží informace o uživateli

## UC60 – Smazání uživatelského účtu

### Anotace

**Název:** Smazání uživatelského účtu

**Identifikátor:** UC60

**Podmínky před spuštěním:** (žádné)

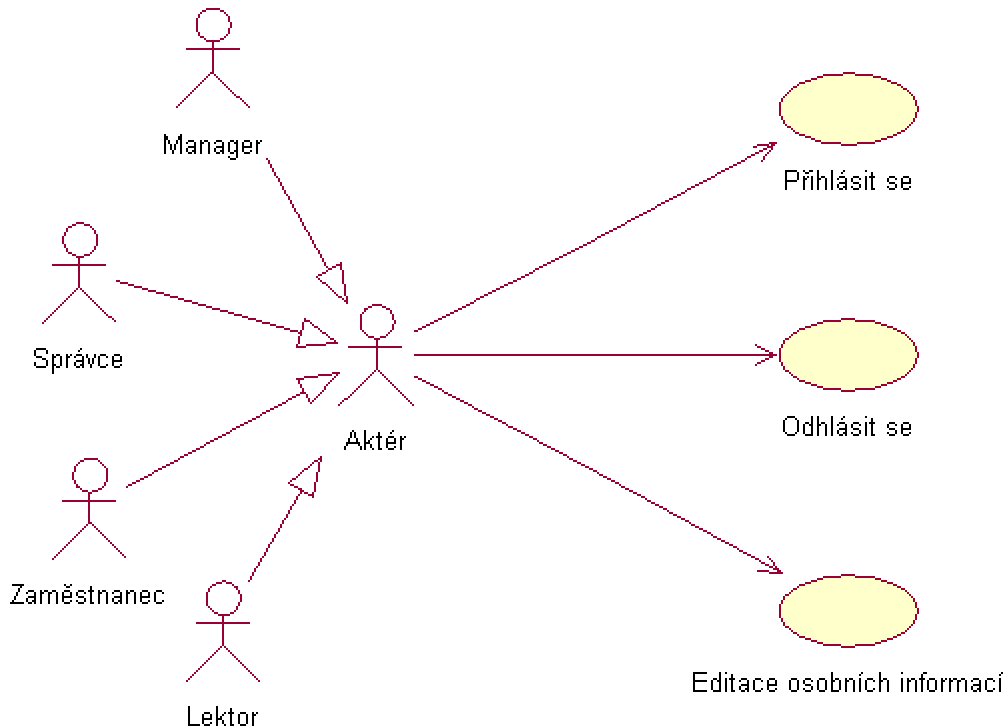
**Podmínky po ukončení:** (žádné)

### Hlavní tok

- 1) systém vygeneruje alert okno žádající potvrzení aktérem
- 2) aktér potvrdí smazání kliknutím na tlačítko „smazat“
- 3) systém odstraní uživatelský účet ze systému

### 5.3.9. Společné případy použití pro všechny aktéry

Toto schéma popisuje společné možnosti všech aktérů, jedná se o základní funkční požadavky mimo problémovou doménu.



obr. 17 – Identifikace společných případů použití

#### UC61 – Přihlásit se

##### Anotace

**Název:** Přihlásit se

**Identifikátor:** UC61

**Podmínky před spuštěním:** (žádné)

**Podmínky po ukončení:** systém uloží informace o uživateli do globálního pole sezení

##### Hlavní tok

- 1) systém zobrazí pod hlavní nabídkou formulář pro přihlášení uživatele
- 2) aktér vyplní tento formulář a klikne na tlačítko odeslat
- 3) systém zkontroluje zadaná data, jsou-li v pořádku, uloží informace o přihlášení do globálního pole sezení (sessions)

##### Alternativní tok

- 1) systém zobrazí pod hlavní nabídkou formulář pro přihlášení uživatele
- 2) aktér vyplní tento formulář a klikne na tlačítko odeslat

- 3) systém zkontroluje zadaná data, nejsou-li v pořádku, oznámí tuto skutečnost vypsáním dialogu o neúspěšném přihlášení na obrazovku

## **UC62 – Odhlásit se**

### **Anotace**

**Název:** Odhlásit se

**Identifikátor:** UC62

**Podmínky před spuštěním:** aktér je přihlášen a existují jeho informace v poli sezení

**Podmínky po ukončení:** systém zruší informace z globálního pole sezení

### **Hlavní tok**

- 1) systém v uživatelském menu zobrazí tlačítko „odhlásit se“
- 2) aktér klikne na toto tlačítko
- 3) systém odstraní informace o aktérovi z globálního pole sezení a přesměruje na úvodní stranu systému

## **UC63 – Editace osobních informací**

### **Anotace**

**Název:** Editace osobních informací

**Identifikátor:** UC63

**Podmínky před spuštěním:** aktér je přihlášen a existují jeho informace v poli sezení

**Podmínky po ukončení:** (žádné)

### **Hlavní tok**

- 1) systém v uživatelském menu zobrazí tlačítko „osobní nastavení“
- 2) aktér klikne na toto tlačítko
- 3) systém vygeneruje formulář změn osobních informací
- 4) aktér upraví tento formulář a potvrdí změnu stávajícím heslem
- 5) systém, souhlasí-li heslo, uloží tyto informace

### **Alternativní tok**

- 1) systém v uživatelském menu zobrazí tlačítko „osobní nastavení“
- 2) aktér klikne na toto tlačítko
- 3) systém vygeneruje formulář změn osobních informací
- 4) aktér upraví tento formulář a potvrdí změnu stávajícím heslem
- 5) systém, nesouhlasí-li heslo, zobrazí o tomto hlášení v pozici nad formulářem

## 5.4. Úvodní glosář

Tento dokument slouží ke specifikaci pojmů užitých v průběhu celé inepční fáze, důležitý je z hlediska vzájemného porozumění mezi klientem a realizátorem.

Pojem	Vysvětlení
aktér	zobecnění celého názvu uživatele v rámci modelu usecase
alert	javascriptové okno typu alert se dvěma přepínači (potvrzení, storno)
FAQ	frequently asked questions, často kladené dotazy
globální pole sezení	asociativní pole ve smyslu sezení - session
histogram	grafická forma zobrazení rozložení
manager	zkrácený pojem pověřený manager
osobní profil	obecné informace o uživateli – např. jméno, login, foto
on-line test	časově omezený test, spouštěný v určitou dobu, možnosti odpovědí jsou a,b,c...
combobox	formulářový prvek typu přepínač
ticket	požadavek na kurz v písemné formě předávaný v rámci vytváření kurzů
správce	zkrácený pojem administrátor systému
úloha	obecný pojem pro úkol s časovým termínem (odevzdání, splnění, počátku) v rámci kurzu, může se jednat i o zkoušku, termíny jsou zahrnuty v harmonogramu
vypracování	ve smyslu odevzdávané části úlohy
vygenerovat	zobrazit na obrazovku
zaměstnanec	zkrácený pojem školený zaměstnanec
zapsaný kurz	kurz, který má uživatel zaregistrován

tab. 5 – Počáteční glosář

## 5.5. Předběžný plán iterací

Analýza systému vnitřního vzdělávání zaměstnanců prokázala, že se jedná o středně rozsáhlý projekt, který vyžaduje více než jednu iterační etapu.

Etapy jsou plánovány s důrazem na logickou návaznost jednotlivých celků, od nejvyšších úrovní oprávnění až po nejnižší, v podobě školeného zaměstnance. Např. pokud zaměstnanec nebude mít vytvořen účet, pak se logicky nemůže přihlásit, neměl-li by vytvořeny kurzy k zápisu, logicky by se nemohl registrovat na předměty.

### 5.5.1. Přehled funkcí realizovaných v jednotlivých etapách

#### První etapa

- V rámci této etapy budou řešeny společné prvky jednotlivým aktérům. Jedná se o přihlášení, odhlášení správa osobních informací.
- Dále pak prvky systému mimo oblast problémové domény jako např. zabezpečené připojení k databázi a knihovny funkcí.

#### Druhá etapa

- V rámci této etapy budou řešeny funkčnosti z pohledu administrátora systému. Konkrétně správa uživatelských účtů, správa často kladených dotazů.

#### Třetí etapa

- V rámci této etapy budou řešeny funkčnosti z pohledu pověřeného manažera.

#### Čtvrtá etapa

- V rámci této etapy budou řešeny funkčnosti z pohledu lektora a garanta.

#### Pátá etapa

- V rámci této etapy budou řešeny funkčnosti z pohledu školeného zaměstnance.

### 5.5.2. Výstupy realizace a její časový harmonogram

Výstupem každé etapy bude informační systém se specifikovanou funkcí. Aplikace bude předána v den ukončení realizace, resp. zahájení uživatelských akceptačních testů.

Výstupem realizace bude také uživatelská dokumentace, instalační příručka, on-line nápověda. Dále bude probíhat školení v rozsahu 2 hodin, ve kterém budou vybraní zaměstnanci školeni. Předání dokumentů a školení proběhne v rámci uživatelských akceptačních testů.

#### Časový harmonogram

Rozsah prototypu je obsažen v prvních dvou fázích. Pokud by se jednalo o realizaci celého projektu, včetně fáze implementace a testování, pak harmonogram musí obsahovat všechna předběžná data realizací jednotlivých fází.

Tento harmonogram je zde uváděn proto, že z něj ve ve fázi rozpracování vychází plán projektu, který je jako jeden z výstupů vytvořen pro potřeby navazující fáze implementace.

<b>Fáze</b>	<b>Od</b>	<b>Do</b>
<b><i>první etapa</i></b>		
Realizace	7. 5. 2007	11. 5. 2007
Akceptační testování	11. 5. 2007	15. 5. 2007
<b><i>druhá etapa</i></b>		
Realizace	16. 5. 2007	18. 5. 2007
Akceptační testování	19. 5. 2007	23. 5. 2007
<b><i>třetí etapa</i></b>		
Realizace	23. 5. 2007	25. 5. 2007
Akceptační testování	26. 5. 2007	30. 5. 2007
<b><i>čtvrtá etapa</i></b>		
Realizace	1. 6. 2007	7. 6. 2007
Akceptační testování	8. 6. 2007	15. 6. 2007
<b><i>pátá etapa</i></b>		
Realizace	16. 6. 2007	30. 6. 2007
Akceptační testování	1. 7. 2007	5. 7. 2007

tab. 6 – Časový harmonogram

## 5.6. Počáteční ohodnocení rizik

V seznamu ohodnocení rizik jsou uvedeny vybrané zobecněné případy užití, u kterých je vhodné popsat míru rizika a prioritu z hlediska implementace.

Hodnota míry rizika a implementační priorita jsou označeny stupnicí od 1 do 10 kde 1 znamená mírné riziko a vysoká priorita, 10 nejvyšší riziko a nejmenší priorita.

Případ užití	Riziko	Priorita v rámci implementace	Poznámka
přihlášení	6	1	přihlášení je hlavní prvek řešené funkcionality, zabezpečení přihlášení zvyšuje jeho riziko
odhlášení	4	1	odhlášení je jedním ze základních prvků funkcionality, zabezpečení odhlášení zvyšuje jeho riziko
zobrazení přehledu	1	2	přehled je využíván u většiny případů použití, proto má velkou prioritu
vytvoření tlačítek	1	3	vytvoření tlačítek v přehledech je navázáno dále na formuláře úprav, proto má vyšší prioritu
formuláře úprav	2	4	formuláře úprav jsou základním prvkem, jejich ověřování pomocí reg. výrazů tvoří mírné riziko
WYSIWYG editor	9	10	připojení externího javascriptového editoru představuje vysoké riziko a není hlavním prvkem systému
histogramy	4	9	histogramy hodnocení jsou pouze doplňkovou funkcí k přehledům, kreslení grafů tvoří mírné riziko
kontextová nápověda	3	8	kontextová nápověda je doplňkovou funkčností, s mírným rizikem při zobrazování

tab. 7 – Počáteční ohodnocení rizik

## 6 Fáze rozpracování

### 6.1. Dodatečné požadavky na systém

Cílem této podkapitoly je definovat požadavky informačního systému elektronického vzdělávání, které nesouvisí s funkcionalitou.

Tyto tzv. Supplementary specifications nejsou zahrnuty v případech použití a spolu s modelem případů použití tvoří úplnou množinu požadavků na systém.

#### 6.1.1. Rozsah systému

Systém bude umožňovat organizaci výukových kurzů, včetně výstupů v XML formátu pro systémy třetích stran poskytujících outsourcingové služby. Sestavování metodických plánů ve smyslu programů školení, vytváření kurzů, přehled registrovaných kurzů zaměstnanci a jejich hodnocení. Příjemné uživatelské rozhraní pro správu osobních kurzů zaměstnance pro řešení zadaných úloh, ukládání souborů do systému, zabezpečení přístupu k databázi.

Systém bude volně dostupný z internetu a má za úkol zefektivnit práci jednotlivých lektorů, poskytnout aktuální informace a pohodlí školeným zaměstnancům a nakonec také informace manažerům o jednotlivých zaměstnancích.

#### 6.1.2. Reference

Tento dokument vychází z těchto výstupů fáze zahájení:

- Vize
- Počáteční obchodní případ
- Model případů užití
- Počáteční glosář

#### 6.1.3. Použitelnost

Veškerá funkcionalita by měla být přístupna tenkým klientem (ve formě prohlížeče) pomocí vzdáleného přístupu přes síť internet.

Uživatelské rozhraní by mělo být navrženo intuitivně, z ohledem na průměrného uživatele, bez nutnosti školení. Dále by uživatelské rozhraní mělo dodržovat zásady přístupného webu popsané v dokumentu vydaném Ministerstvem informatiky ČR. Tento dokument je veřejně dostupný na adrese: [http://www.micr.cz/files/1548/mi-FINAL-BP\\_web\\_pristupnost-20040722.pdf](http://www.micr.cz/files/1548/mi-FINAL-BP_web_pristupnost-20040722.pdf)

Systém bude obsahovat vestavěnou on-line uživatelskou nápovědu, která bude obsahovat instrukce nutné pro správné používání systému. Obsah nápovědy se bude dynamicky měnit podle aktuálního stavu systému.

#### 6.1.4. Spolehlivost

Systém by měl být dostupný 24 hodin 7 dní v týdnu, kde za špičku můžeme považovat dobu mezi 8h ranní a 17h odpolední. Minimální doba mezi pády systému by měla být okolo 300 hodin.



Všechny chyby generované systémem by měly být logovány. Fatální chyby by měly vést k bezpečnému ukončení práce se systémem. Chybová hlášení budou obsahovat popis chyby, číslo chyby, datum a čas vzniku chyby, číslo a název modulu ve kterém došlo k chybě.

Veškeré operace prováděné v systému může provádět pouze uživatel vlastníci uživatelský účet a je k tomuto účtu řádně přihlášen.

### **6.1.5. Výkonnost**

Systém by měl ve špičce podporovat 500 současně pracujících uživatelů vůči centrální databázi a webovému serveru. Mimo špičku by měl podporovat 100 současně pracujících uživatelů vůči centrální databázi a serveru. Doba odezvy systému by neměla přesáhnout 1000ms.

Z hlediska objemu dat se předpokládá, že bude zpracovávat nejméně 20 000 záznamů v databázi ročně.

### **6.1.6. Omezení architektury**

Systém by měl produkovat výstupy pro vybrané outsourcingové firmy ve formě XML dokumentu specifikovaném veřejným DTD.

Klientská část systému by měla být spustitelná na libovolném počítači minimálně s procesorem Intel Pentium 90MHz a vyšším, klientský počítač by měl obsahovat minimálně 32MB RAM a měl by být vybaven internetovým prohlížečem Internet Explorer verze 6 a vyšším, Mozilla Firefox 1.5 nebo Opera 8.

Serverová část poběží na operačním systému UNIXového typu, jako webový server byl zvolen Apache 2 a databáze poběží na MySQL v5.

## **6.2. Popis softwarové architektury**

### **6.2.1. Logický pohled**

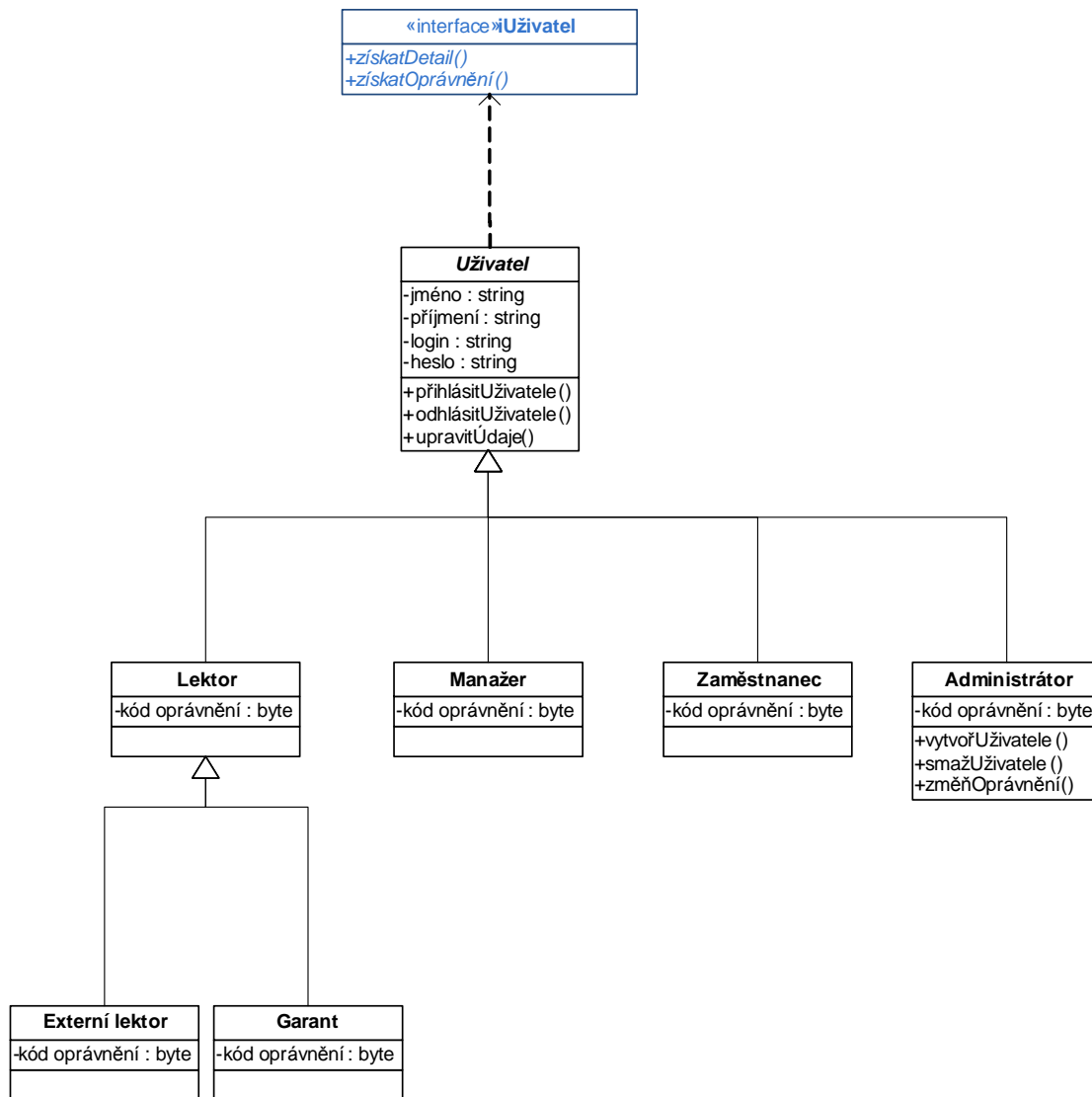
Tento pohled zachycuje slovník domény řešeného problému jako množinu tříd objektů. Důraz je kladen na to, jakým způsobem třídy tvořící systém implementují požadované chování

#### **6.2.1.1. Diagramy tříd**

Hlavním výrazovým prostředkem logického pohledu je diagram objektových tříd. Z hlediska fází nelze přesně určit kdy začneme tuto techniku používat. Zahájení obvykle závisí na firemní kultuře, či individuálních preferencích analytika. Někteří analytici upřednostňují kompletní dokončení fáze zahájení, včetně popsání všech případů použití, jiní preferují jejich souběžné vytváření. Osobně jsem použil první přístup z důvodu jasného rozčlenění do jednotlivých fází.

## Balíček problémové domény č.1 – Uživatel

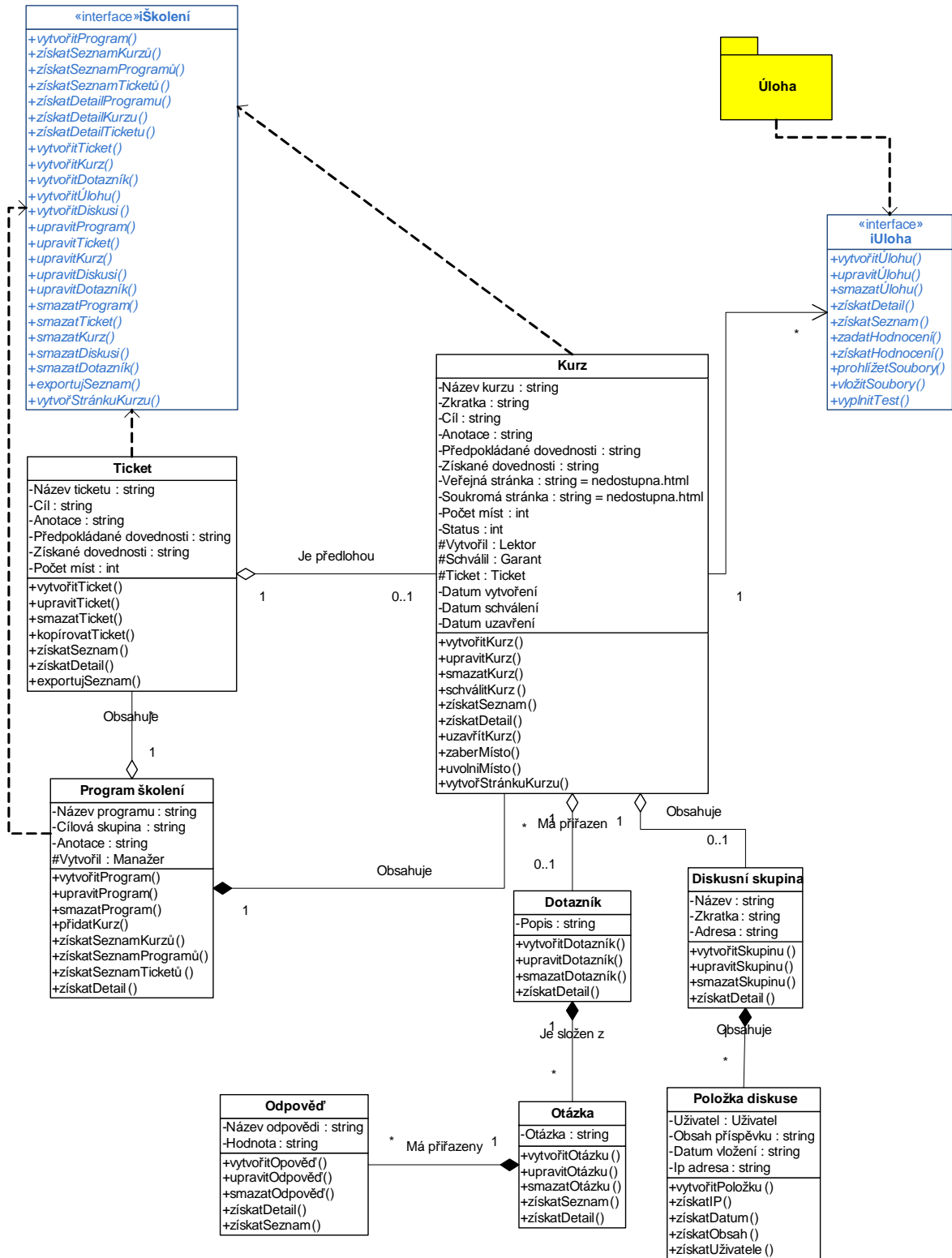
Tento balíček zachycuje strukturu uživatelských účtů a jednotlivé případy generalizace aktérů. Aktér je při komunikaci s ostatními balíčky jednoznačně identifikován dle kódu oprávnění.



obr. 18 – Přehled objektových tříd zařazených do balíčku uživatel

## Balíček problémové domény č.2 – Program školení

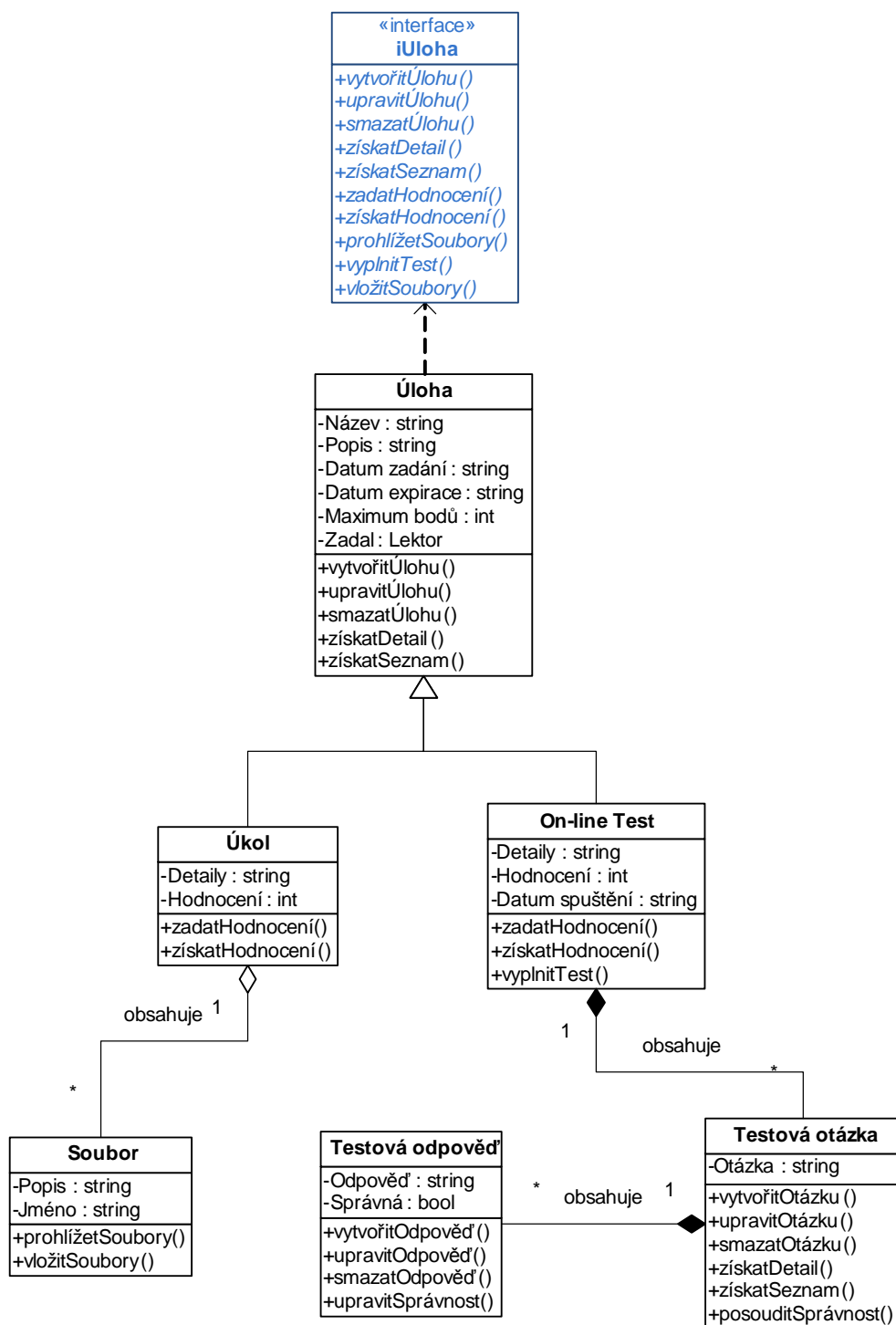
Tento balíček tvoří jádro systému. Jeho podbalíčkem je balíček úlohy, který je navázán na třídu s kurzem pomocí svého interface. V tomto balíčku jsou identifikovány objektové třídy sloužící k vytváření programů školení, tvorbě kurzu a jemu závislých tříd.



obr. 19 – Přehled objektových tříd zařazených do balíčku program školení

## Balíček problémové domény č.2 - podbalík Úloha

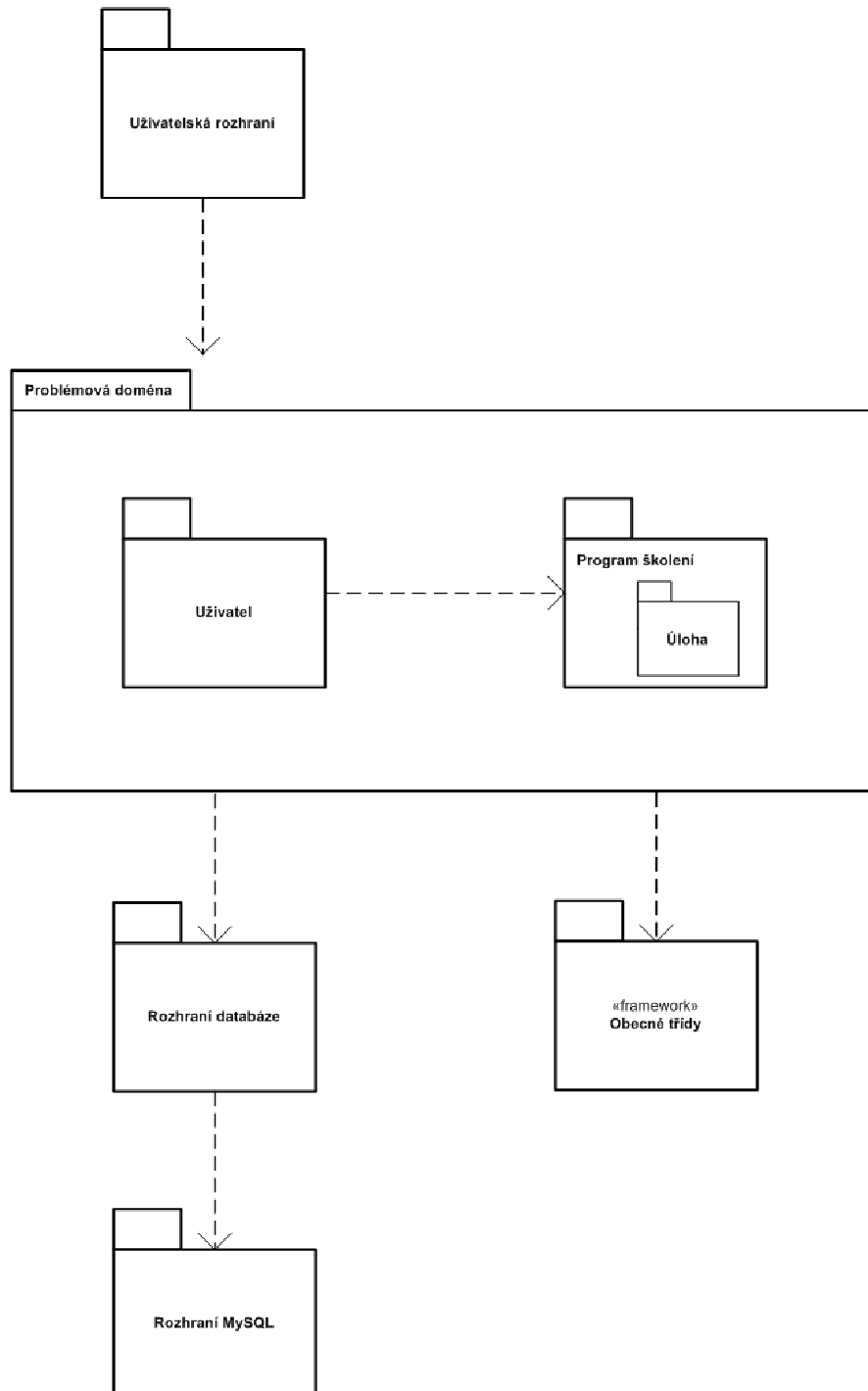
Tento balíček zachycuje strukturu jednotlivých úloh. Úloha je připojena ke konkrétnímu kurzu a je spravována přes uvedený interface.



obr. 20 – Přehled objektových tříd zařazených do balíčku úloha

### 6.2.1.2. Diagram balíčků

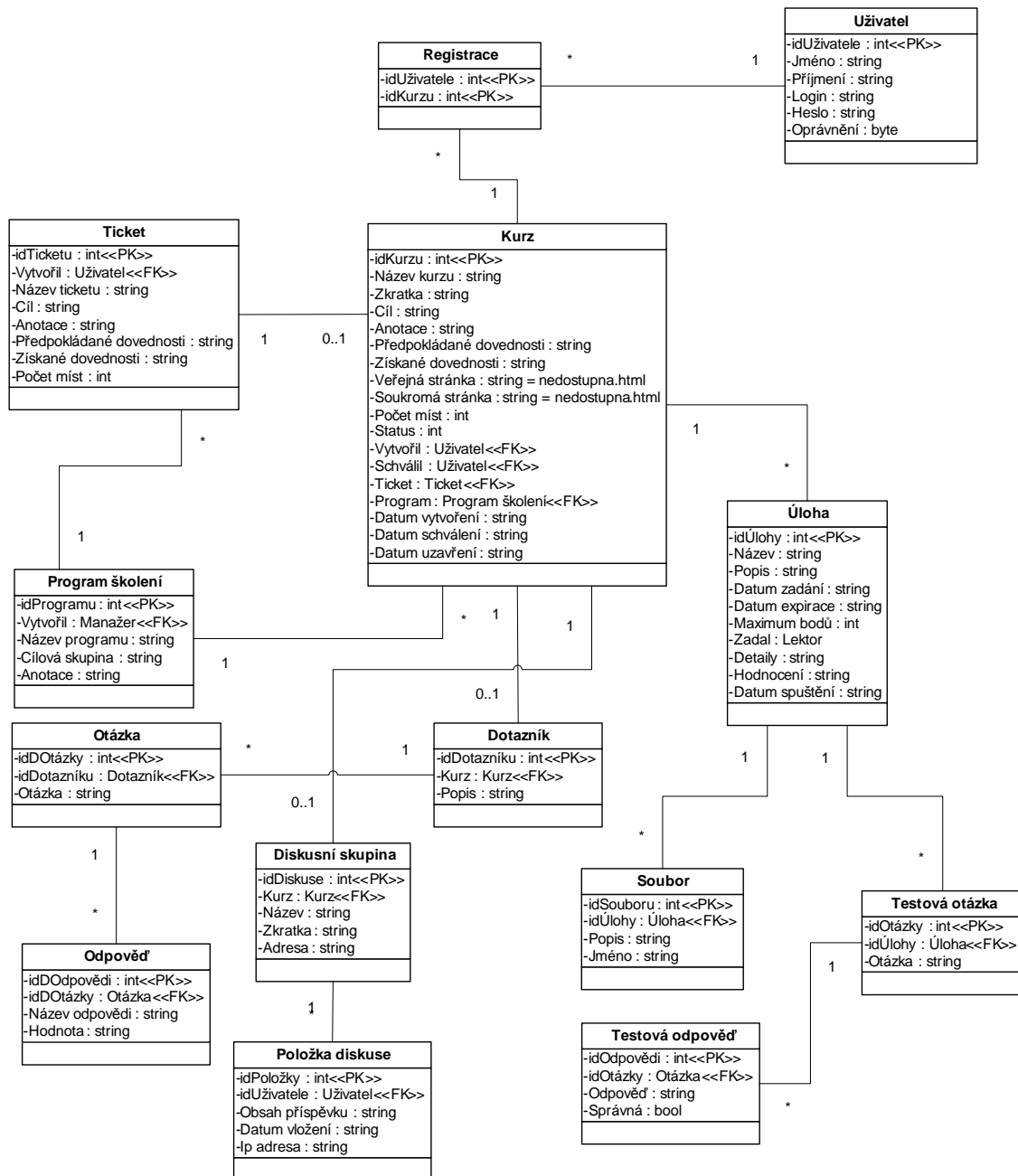
Tento pohled návrhového modelu jasně ukazuje, jakým způsobem budou propojeny jednotlivé části systému z hlediska balíčků naplněných třídami. V analytickém modelu jsou obvykle specifikovány pouze balíčky problémové domény.



obr. 21 – Pohled na balíčky systému

### 6.2.1.3. Datový model domény

Vytvořený datový model je navržen pro cílovou databázi MySQL. Vyladění datového modelu obvykle provádí odborník na daný databázový systém, protože špatně navržená a vyladěná databáze může znehodnotit celou analýzu a návrh vyvíjeného software.

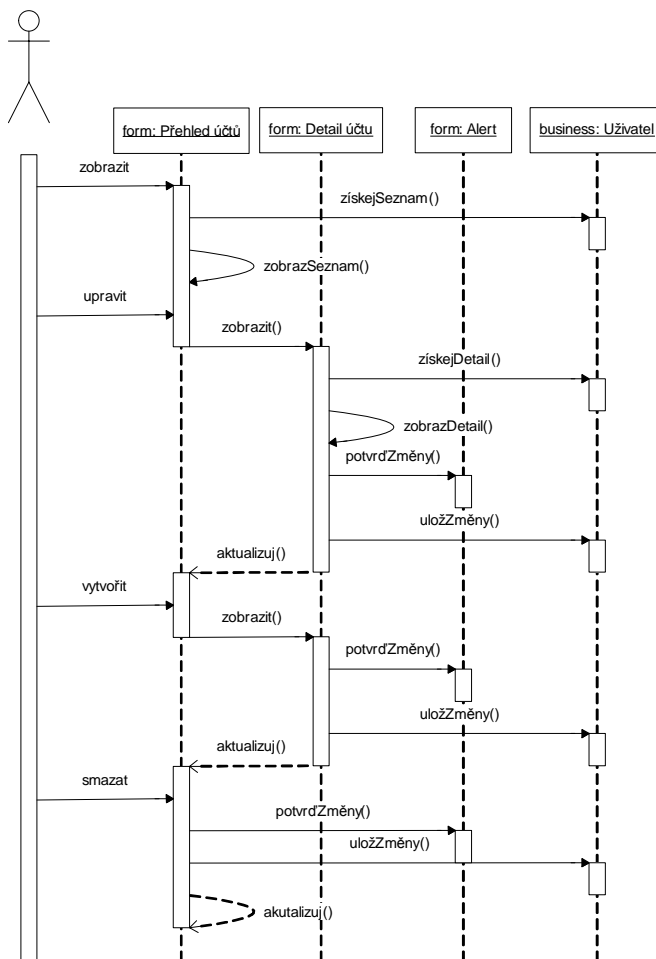


obr. 22 – Návrh databáze

### 6.2.1.4. Sekvenční diagramy vybraných případů použití

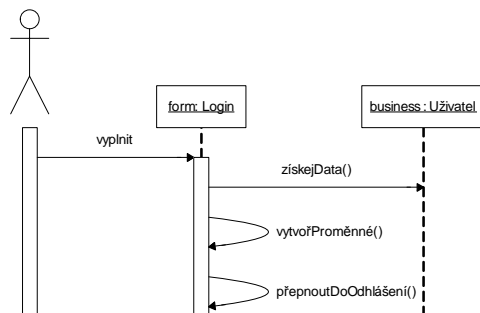
Tyto diagramy slouží k upřesnění toků a vazeb jednotlivých případů použití z hlediska časové návaznosti akcí. Pro náš systém je vhodné upřesnit některé klíčové diagramy použití na rozhraní balíčků uživatel a program školení.

#### Správa uživatelských účtů z pohledu administrátora



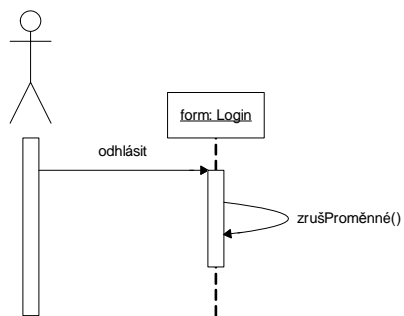
obr. 23 - Sekvenční diagram popisující vytváření uživatelských účtů

#### Přihlášení aktéra do systému



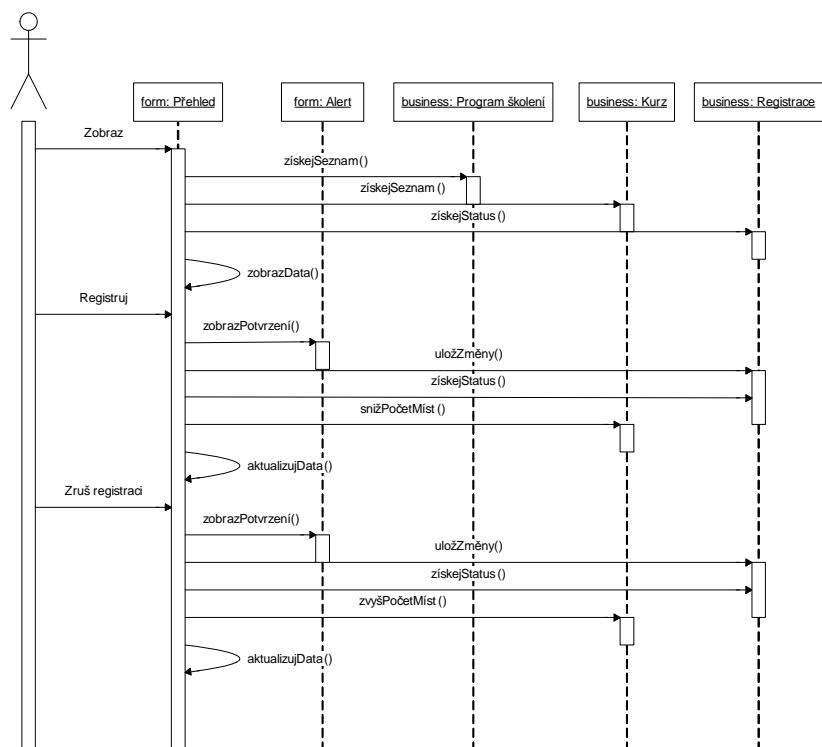
obr. 24 – Sekvenční diagram popisující přihlášení uživatele

## Odhlášení aktéra ze systému



obr. 25 – Sekvenční diagram popisující odhlášení aktéra ze systému

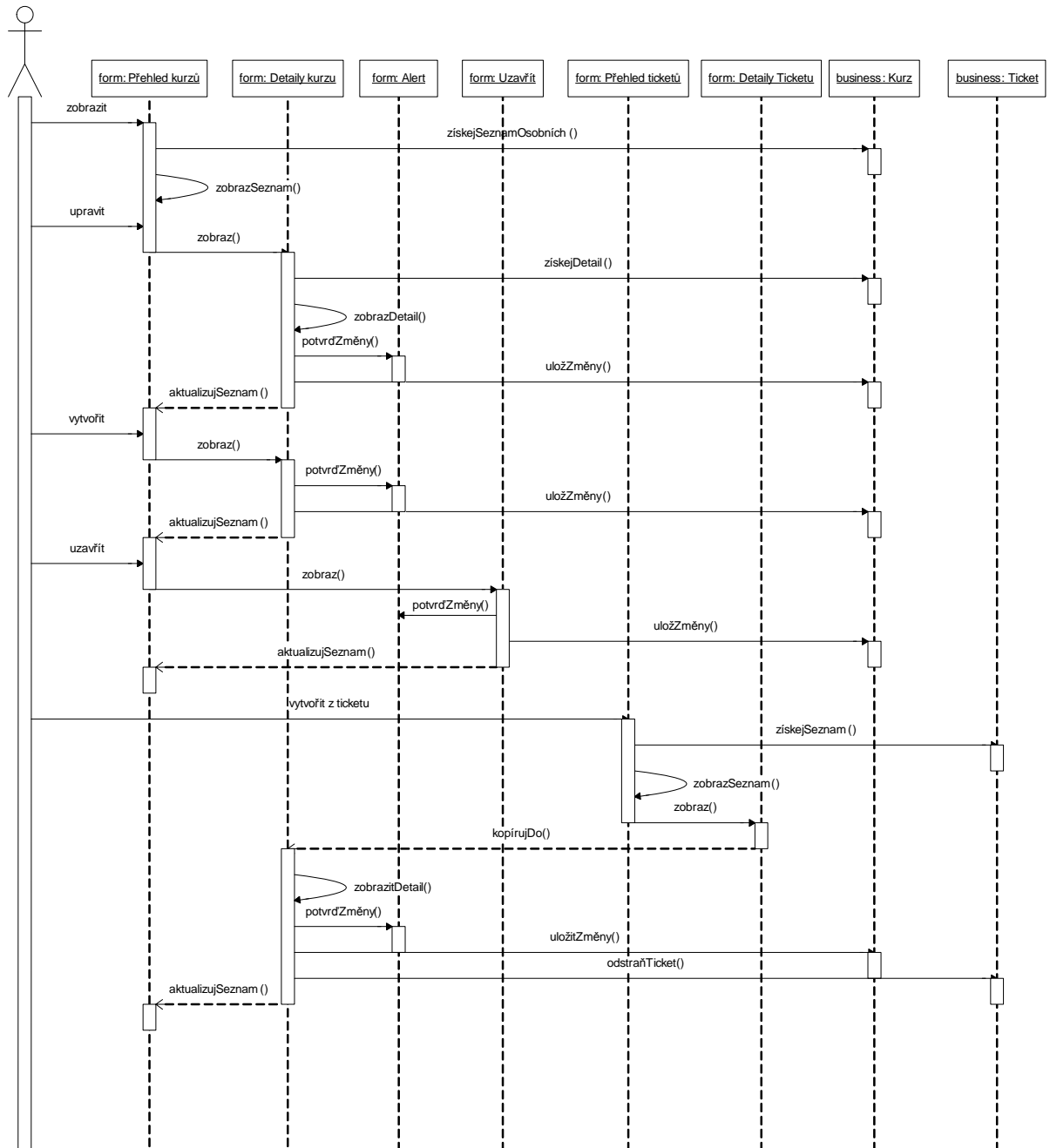
## Registrace kurzů zaměstnance



obr. 26 – Sekvenční diagram popisující registraci kurzů zaměstnance

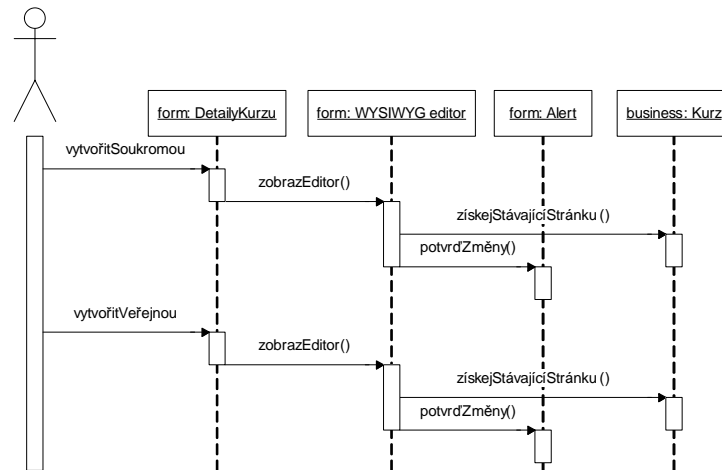


## Základní prvky správy kurzů z pohledu lektora



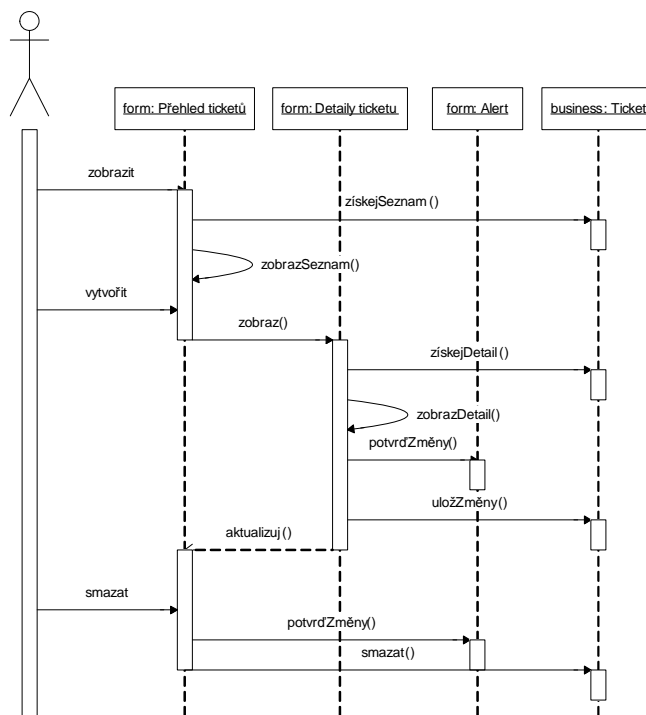
obr. 27 – Sekvenční diagram popisující základní prvky správy kurzů

## Prvek úpravy kurzu – vytvoření HTML stránky kurzu lektorem



obr. 28 – Sekvenční diagram popisující tvorbu HTML stránky pomocí externího editoru

## Správa ticketů z pohledu manažera



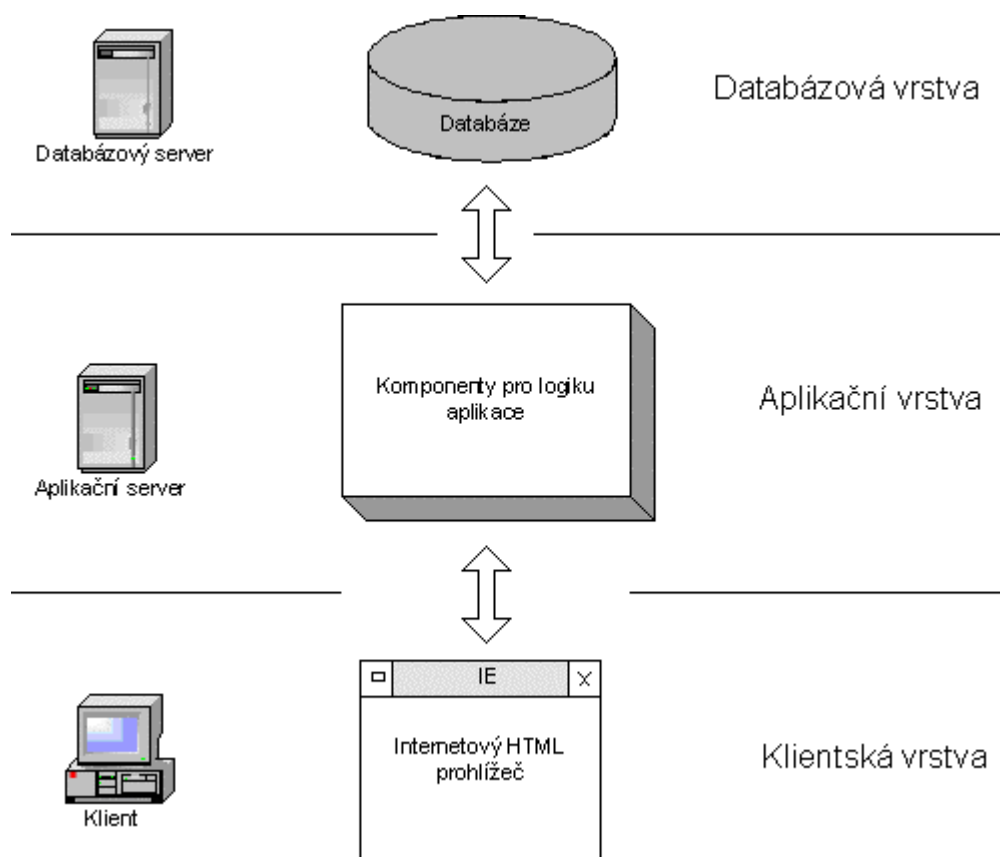
obr. 29 – Sekvenční diagram popisující tvorbu ticketů

## 6.2.2. Procesní pohled

Tento pohled mapuje jednotlivé třídy do souboru vzájemně komunikujících procesů a vláken. Uplatňuje se tam, kde je využívání více vláken nevyhnutelné. Z pohledu RUP je tento pohled nepovinný proto jej nebudu uvádět. Obvykle je řešen systémovými analytiky.

## 6.2.3. Pohled nasazení

V současné době je nejpoužívanějším způsobem pro implementace IS třívrstvá architektura. Ta má řadu podstatných výhod. Třívrstvá architektura spočívá v rozdělení celé aplikace na tři úrovně, jak ukazuje následující obrázek.



obr. 30 – Schéma tříúrovňového informačního systému

### 6.2.3.1. Databázová vrstva

Hlavním úkolem této vrstvy je poskytovat bezpečné a spolehlivé úložiště dat informačního systému. Kromě toho také zajišťuje základní integritu dat. Dále poskytuje základní zabezpečení přístupu k datům a také prostředky pro rychlý přístup k datům a jejich vyhledávání.

### 6.2.3.2. Aplikační vrstva

Druhou vrstvou je tzv. aplikační vrstva. Tato vrstva reprezentuje vlastní aplikační logiku celého informačního systému. Základním úkolem aplikační vrstvy je poskytovat všechny potřebné funkce pro klienty z klientské vrstvy.

Data získává komunikací s databázovou vrstvou a s případnými ostatními instancemi (uzly) vlastního informačního systému.

### **6.2.3.3. Klientská vrstva**

Poslední vrstvou je klientská vrstva. Vrstva má za úkol prezentovat data informačního systému uživatelům a poskytovat jim zároveň přístup k jednotlivým službám systému.

V informačním systému existují dva základní druhy klientských aplikací. Prvním z nich jsou standardní aplikace navržené pro použitou platformu (tlustý klient). Tyto aplikace jsou použity tam, kde je požadováno komplexní a výkonné uživatelské rozhraní.

Druhým typem klientské aplikace je rozhraní pro standardní prohlížeč HTML obsahu (tenký klient). Toto rozhraní je použito všude tam, kde nejsou kladeny velké nároky na uživatelské rozhraní a zejména tam, kde je potřeba toto uživatelské rozhraní často měnit nebo upravovat. V tomto případě byl zvolen právě tenký klient.

### **6.2.3.4. Komunikace mezi vrstvami**

Jednotlivé vrstvy vždy komunikují pouze se sousední vrstvou. Veškerá komunikace probíhá na základě standardních internetových komunikačních protokolů skupiny TCP/IP.

Komunikace mezi klientskou a aplikační vrstvou probíhá výměnou HTML obsahu pomocí protokolu HTTP. Jako vstupní bod pro komunikaci na straně aplikační vrstvy slouží webový server. Všichni klienti tedy komunikují s tímto serverem výhradně pomocí protokolu HTTP.

## **6.2.4. Implementační pohled**

Tento model je typicky řešen ve fázi implementace, proto není v této fázi uváděn. Jeho výrazovým prostředkem je obvykle diagram komponent, který znázorňuje jednotlivé komponenty systému a jejich závislosti.

### 6.3. Revidovaný seznam rizik

Revize seznamu rizik se provádí po provedení návrhu jednotlivých pohledů. Důvodem je možná změna priorit a rizik. Některá rizika se projeví jako nadhodnocená, nová mohou naopak vzniknout.

Hodnocení je prováděno stejným způsobem jako u prvotního odhadu rizik, v závorce je uvedena hodnota z prvotního odhadu.

Případ užití	Riziko	Priorita v rámci implementace	Zhodnocení po rozpracování
přihlášení	<b>4 (6)</b>	<b>1</b>	přihlášení se neprojevilo jako výrazné riziko a proto bylo jeho hodnocení sníženo
odhlášení	<b>2 (3)</b>	<b>1</b>	odhlášení se taktéž neprojevilo jako výraznější riziko a jeho hodnocení bylo mírně sníženo
zobrazení přehledu	<b>1</b>	<b>2</b>	riziko ani priorita nezměněny
vytvoření tlačítek	<b>1</b>	<b>3</b>	riziko ani priorita nezměněny
formuláře úprav	<b>2</b>	<b>4</b>	riziko ani priorita nezměněny
WYSIWYG editor	<b>3 (9)</b>	<b>10</b>	připojení externího editoru se neprojevilo jako riziko díky velmi dobré poskytnuté dokumentaci
histogramy	<b>7 (4)</b>	<b>9</b>	riziko bylo zvýšeno, protože zvolená architektura neobsahuje rozhraní pro kresbu grafů
kontextová nápověda	<b>3</b>	<b>5 (6)</b>	priorita byla po první iteraci zvýšena z důvodu zrychlení zaškolení uživatelů u akceptačních testů

tab. 8 - Revidovaný seznam rizik

## 6.4. Prototyp systému

Jedním z výstupů fáze rozpracování je tzv. prototyp systému. Jedná se o softwarový prvek, který obsahuje cca 10% specifikované funkčnosti systému. Tento prvek je tvořen jako podklad celkového systému tak, aby se na něm dala otestovat zvolená architektura a zároveň, aby zákazník dokázal posoudit zda-li je s analýzou a návrhem systému spokojen.

Pokud je prototyp systému zákazníkem schválen, nastává následující fáze životního cyklu – fáze implementace, ve které jsou vytvořeny ostatní části systému. Pokud není zákazník s prototypem spokojen, je nutné provést znovu fázi zahájení a rozpracování, ve které se systém znovu podrobně navrhne. Druhá možnost je zastavení prací na projektu a ukončení.

V rámci tvorby prototypu je tvořena také podrobná instalační příručka, která se včetně vypracovaného prototypu nachází na přiloženém DVD.

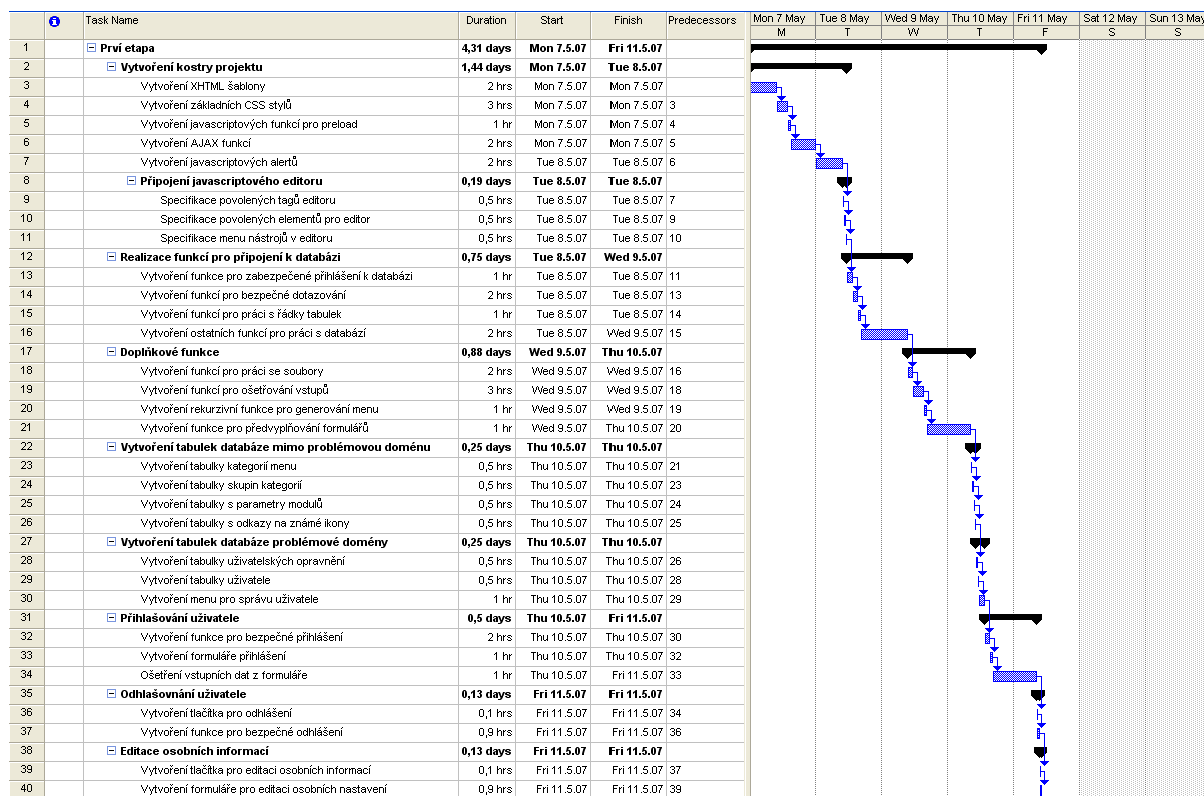
## 6.5. Vývojový plán projektu

Vzhledem k jednočlennému teamu, jsou vazby ganttova diagramu pouze typu FS (finish-to-start), čímž jednotlivé úkoly na sebe chronologicky navazují.

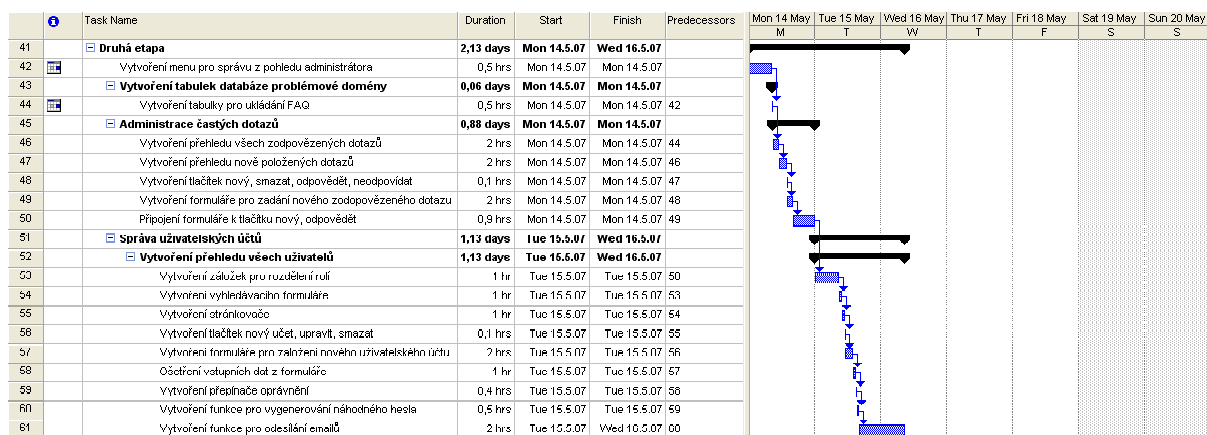
Rozsah prací je omezen jak pracovním týdnem (nepočítají se víkendy), tak i obvyklou osmihodinovou pracovní dobou.

Vývojový plán projektu je celý uložen ve formátu MS Project 2003 (.mpp) na DVD přiloženém k diplomové práci.

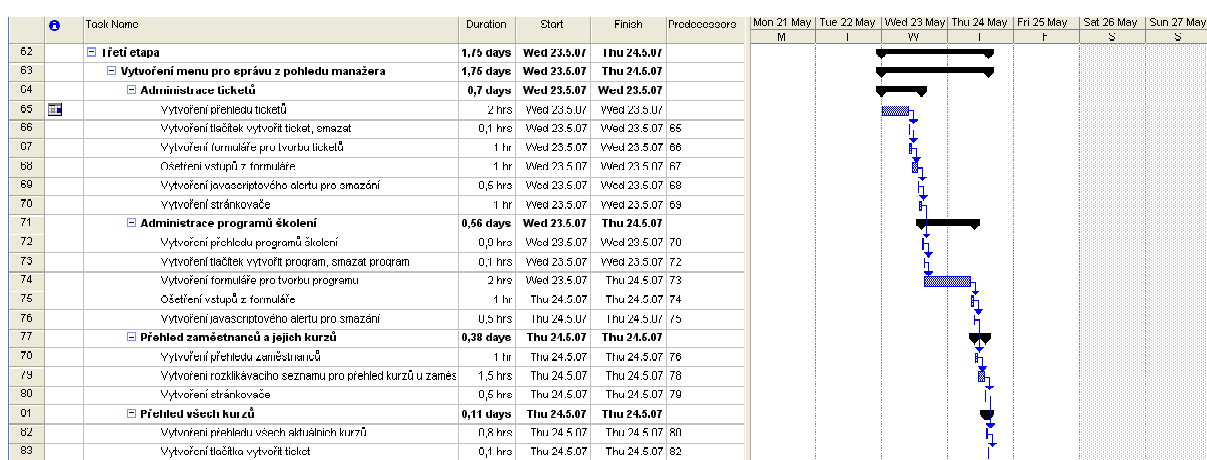
### Plány etap



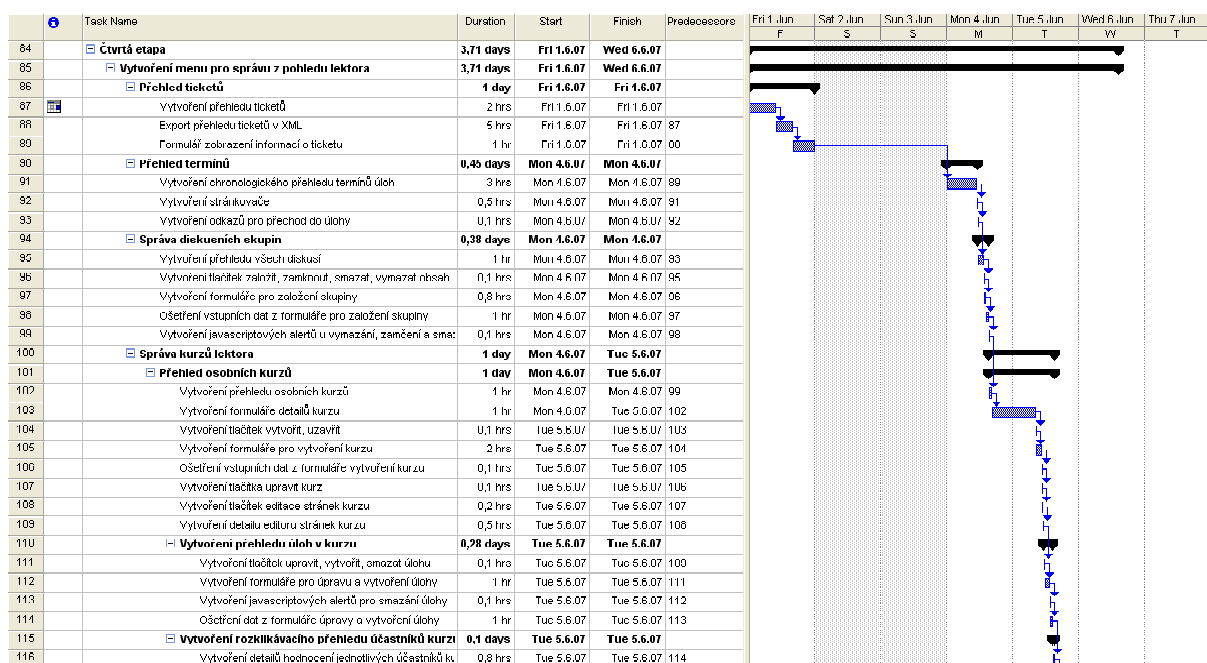
obr. 31 – Plán první etapy



obr. 32 – Plán druhé etapy



obr. 33 – Plán třetí etapy



obr. 34 – Plán čtvrté etapy – první část

Task ID	Task Name	Duration	Start	Finish	Predecessors	Mon 4 Jun	Tue 5 Jun	Wed 6 Jun	Thu 7 Jun	Fri 8 Jun	Sat 9 Jun	Sun 10 Jun
117	<b>Správa kurzů garanta</b>	<b>0,89 days</b>	<b>Tue 5.6.07</b>	<b>Wed 6.6.07</b>								
118	<b>Přehled všech kurzů</b>	<b>0,4 days</b>	<b>Tue 5.6.07</b>	<b>Wed 6.6.07</b>								
119	Vytvoření přehledu všech kurzů	2 hrs	Tue 5.6.07	Wed 6.6.07	116							
120	Vytvoření tlačítek uzavřít, odstranit, detaily v přehledu vš	0,1 hrs	Wed 6.6.07	Wed 6.6.07	119							
121	Vytvoření formuláře s detaily kurzu	1 hr	Wed 6.6.07	Wed 6.6.07	120							
122	Vytvoření javascriptových alertů u odstranění a uzavření	0,1 hrs	Wed 6.6.07	Wed 6.6.07	121							
123	<b>Přehled programů školení</b>	<b>0,24 days</b>	<b>Wed 6.6.07</b>	<b>Wed 6.6.07</b>								
124	Vytvoření přehledu seznamu školení	1 hr	Wed 6.6.07	Wed 6.6.07	122							
125	Vytvoření tlačítek schválit, odebrat, přiřadit	0,1 hrs	Wed 6.6.07	Wed 6.6.07	124							
126	Vytvoření formuláře přiřazení kurzů	0,7 hrs	Wed 6.6.07	Wed 6.6.07	125							
127	Vytvoření javascriptových alertů u odebrání a schválení k	0,1 hrs	Wed 6.6.07	Wed 6.6.07	126							
128	<b>Přehled výsledků dotazníků</b>	<b>0,25 days</b>	<b>Wed 6.6.07</b>	<b>Wed 6.6.07</b>								
129	Zobrazení detailu dotazníků	1 hr	Wed 6.6.07	Wed 6.6.07	127							
130	Zobrazení hodnocení kurzu	1 hr	Wed 6.6.07	Wed 6.6.07	129							

obr. 35 – Plán čtvrté etapy – druhá část

Task ID	Task Name	Duration	Start	Finish	Predecessors	Mon 18 Jun	Tue 19 Jun	Wed 20 Jun	Thu 21 Jun	Fri 22 Jun	Sat 23 Jun	Sun 24 Jun
131	<b>Pátá etapa</b>	<b>3,74 days</b>	<b>Mon 18.6.07</b>	<b>Thu 21.6.07</b>								
132	<b>Vytvoření menu zaměstnance</b>	<b>3,74 days</b>	<b>Mon 18.6.07</b>	<b>Thu 21.6.07</b>								
133	<b>Přehled harmonogramu úloh</b>	<b>0,51 days</b>	<b>Mon 18.6.07</b>	<b>Mon 18.6.07</b>								
134	Vytvoření chronologického přehledu úloh	3 hrs	Mon 18.6.07	Mon 18.6.07	133							
135	Vytvoření tlačítka popis	0,1 hrs	Mon 18.6.07	Mon 18.6.07	134							
136	Vytvoření formuláře s detaily úlohy	1 hr	Mon 18.6.07	Mon 18.6.07	135							
137	<b>Registrace kurzů</b>	<b>0,51 days</b>	<b>Mon 18.6.07</b>	<b>Tue 19.6.07</b>								
138	Vytvoření přehledu kurzů	2 hrs	Mon 18.6.07	Mon 18.6.07	136							
139	Vytvoření dynamického zobrazování počtu volných míst	0,5 hrs	Mon 18.6.07	Mon 18.6.07	138							
140	Vytvoření tlačítek registrovat, zrušit	0,1 hrs	Mon 18.6.07	Mon 18.6.07	139							
141	Vytvoření formuláře registrace kurzů	1 hr	Mon 18.6.07	Mon 18.6.07	140							
142	Ošetření dat z formuláře registrace	0,4 hrs	Mon 18.6.07	Tue 19.6.07	141							
143	Vytvoření javascriptového alertu u zrušení registrace	0,1 hrs	Tue 19.6.07	Tue 19.6.07	142							
144	<b>Zapsané kurzy</b>	<b>2,71 days</b>	<b>Tue 19.6.07</b>	<b>Thu 21.6.07</b>								
145	Vytvoření přehledu zapsaných kurzů	2 hrs	Tue 19.6.07	Tue 19.6.07	143							
146	Vytvoření tlačítka detaily	0,1 hrs	Tue 19.6.07	Tue 19.6.07	145							
147	Vytvoření chronologického rozvrhu kurzů	3 hrs	Tue 19.6.07	Tue 19.6.07	146							
148	<b>Detaily kurzu</b>	<b>2,08 days</b>	<b>Tue 19.6.07</b>	<b>Thu 21.6.07</b>								
149	Vytvoření dynamického histogramu hodnocení	4 hrs	Tue 19.6.07	Wed 20.6.07	147							
150	Vytvoření odkazu na zobrazování dotazníku	0,1 hrs	Wed 20.6.07	Wed 20.6.07	149							
151	Formulář dotazníku	1 hr	Wed 20.6.07	Wed 20.6.07	150							
152	Logika formuláře dotazníku	0,5 hrs	Wed 20.6.07	Wed 20.6.07	151							
153	<b>Zobrazení přehledu úloh</b>	<b>1,38 days</b>	<b>Wed 20.6.07</b>	<b>Thu 21.6.07</b>								
154	Vytvoření odevzdávacího formuláře	0,5 hrs	Wed 20.6.07	Wed 20.6.07	152							
155	Ošetření dat z odevzdávacího formuláře	0,5 hrs	Wed 20.6.07	Wed 20.6.07	154							
156	Vytvoření úložiště pro odevzdané soubory	2 hrs	Wed 20.6.07	Wed 20.6.07	155							
157	Vytvoření dynamického histogramu hodnocení úlohy	3 hrs	Wed 20.6.07	Thu 21.6.07	156							
158	Vytvoření přiřazování se na testy, varianty a termíny	1 hr	Thu 21.6.07	Thu 21.6.07	157							
159	Vytvoření formuláře on-line testu	2 hrs	Thu 21.6.07	Thu 21.6.07	158							
160	Ošetření dat z formuláře on-line testu	2 hrs	Thu 21.6.07	Thu 21.6.07	159							

obr. 36 – Plán páté etapy



## Závěr

Novým a logickým trendem v IT se očividně pomalu, ale jistě stává orientace na tvorbu komponentového softwaru a orientace na služby. Zdá se, že brzy bude tvorba software daleko více podobná jiným průmyslovým odvětvím, jako třeba stavitelství, a na jeho konstrukci se bude nadále využívat inženýrských postupů.

Notací budoucnosti je zcela jistě UML a metodiky založené na nich, jako je právě Rational Unified Process. Již z první části práce zabývající se představením RUP je zřejmé, že klade největší důraz na vysokoúrovňové činnosti, jakou je právě analýza a návrh. Implementaci, která zabere většinu času potřebného na provedení projektu, prakticky neřeší a omezuje ji téměř výhradně na doporučení týkající se plánování a řízení zdrojů.

RUP bývá prezentován jako iterativní proces. Pokud se ovšem zamyslíme nad jeho podstatou, je zřejmé, že se jedná o modifikaci klasického „vodopádu“. Při bližším zkoumání lze popírat i samotný dvojrozměrný přístup RUP. Je zde zřejmá korelace mezi fázemi projektu a disciplínami. Specifikace požadavků se provádí, kromě mírných odchylek, výhradně ve fázi zahájení a na začátku fáze rozpracování. Analýza jen ve fázi rozpracování, atp. Jednotlivé disciplíny na sebe navíc velmi často přímo navazují, což je typická vlastnost klasického „vodopádu“.

Lze polemizovat i s tvrzením, že RUP je proces zaměřený na architekturu, neboť iterace jsou v RUP chápány jako dílčí fáze vývoje systému. Komponenty by, naopak, měly být konstruovány jako relativně nezávislé prvky a problém s jednou komponentou by neměl ovlivnit celý projekt.

Je dobré uvést, že se metodika RUP nehodí na menší softwarové projekty a projekty řešené týmy s malým počtem členů. Tato metodika je právě kvůli její robustnosti a propracovanosti velmi rozsáhlá a její přizpůsobení konkrétnímu projektu může být poměrně náročné a dost často i zbytečné. Pro menší projekty se obecně spíše doporučují techniky Extremního programování, Test-driven development či některé další novodobé techniky tzv. agilního programování.

# Seznam použité literatury

- 1) ŠTORK R., VITOUŠ O., Rational Unified Process: Stručný průvodce, Unicorn multimedia 2000, ISBN: 80-238-6358-4
- 2) KANISOVÁ H., MÜLLER M., UML srozumitelně, Computer Press a.s. 2006, ISBN: 80-251-1083-4
- 3) ARLOW J., NEUSTADT I., UML a unifikovaný proces vývoje aplikací, Computer Press a.s. 2003, ISBN: 80-7226-947-X
- 4) GALIC M., MACISAAC B., POPESCU D., Using a Single Business Pattern with the Rational Unified Process, IBM Rational Software Redbook
- 5) BERGSTRÖM S., RÅBERG L., Adopting the Rational Unified Process: Success with the RUP, Addison-Wesley Professional 2003, ISBN: 0321202945
- 6) BOOCH G., RUMBAUGH J., JACOBSON I., The Unified Modelling Language User Guide, Addison-Wesley Professional 2005, ISBN: 0321267974
- 7) KRUCHTEN P., The Rational Unified Process: An Introduction, Addison-Wesley Professional 2003, ISBN: 0321197704
- 8) ZENDULKA J., BÁRTÍK V., KVĚTOŇOVÁ Š., Studijní opora předmětu Analýza a návrh informačních systémů, VUT Brno 2006
- 9) VONDRÁK I., Úvod do softwarového inženýrství, VŠB TU Ostrava, 2003
- 10) ALLEN'S M., Guide to e-Learning, John Wiley & Sons 2003, ISBN: 0471203025
- 11) KADLEC V., Rational Unified Process: Základní pojmy, článek na portálu [www.zive.cz](http://www.zive.cz) ze dne 05.08.2003
- 12) Internetové stránky [http://www.sparxsystems.com.au/resources/uml2\\_tutorial/](http://www.sparxsystems.com.au/resources/uml2_tutorial/), informace o diagramech jazyka UML 2.0
- 13) Internetové stránky <http://www.ibm.com>, promo článek IBM Rational Unified Process
- 14) Internetové stránky <http://www.enterpriseunifiedprocess.com>, článek o historii RUP
- 15) Internetové stránky <http://www.wikipedia.com>, položka Rational Unified Process
- 16) Internetové stránky <http://objekty.vse.cz/>, článek o Rational Unified Process

# Seznam příloh

- 1) Seznam obrázků
- 2) Seznam tabulek
- 3) Náhled na uživatelské rozhraní systému
- 4) Příložené DVD obsahující - realizovaný prototyp, instalační příručku, plán projektu, vybrané diagramy ve formátu MS Visio 2007

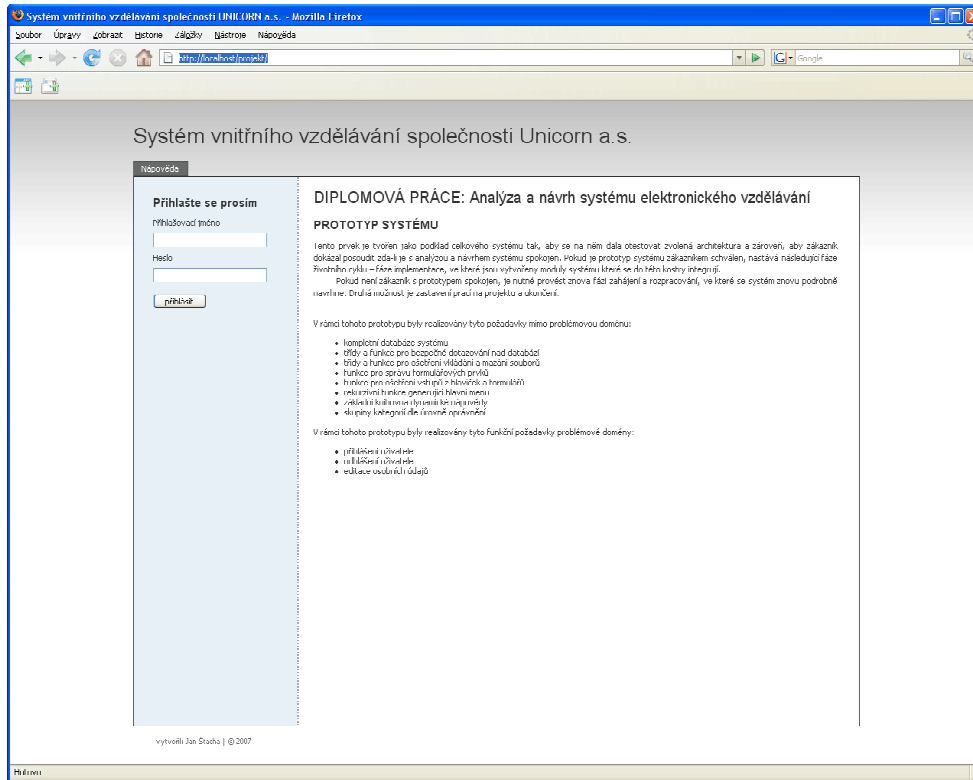
# Seznam obrázků

obr. 1 – Pohled na Rational Unified Process jako na framework .....	4
obr. 3 – Iterativní vývoj s přírůstky .....	6
obr. 4. – Náročnost jednotlivých fází RUP .....	9
obr. 5 – Typické vyobrazení jednotlivých symbolů a příklad pracovního postupu .....	13
obr. 6 – Schématické vyjádření procesu RUP .....	14
obr. 7 – Statické a dynamické testování .....	17
obr. 8 – Model architektury 4+1 .....	18
obr. 9 – Globální pohled na systém .....	27
obr. 10 – Identifikace případů použití související se správou kurzů z pohledu zaměstnance.....	28
obr. 11 – Identifikace případů užití při prohlížení podrobných informací o kurzu .....	32
obr. 12 – Identifikace případů použití vztahujících se ke správě kurzů z pohledu lektora .....	36
obr. 13 – Identifikace ostantích případy užití z pohledu lektora.....	43
obr. 14 – Identifikace všech případů použití z pohledu garanta .....	48
obr. 15 – Identifikace všech případů použití z pohledu manažera .....	53
obr. 16 – Identifikace všech případů použití z pohledu administrátora systému .....	58
obr. 17 – Identifikace společných případů použití.....	63
obr. 18 – Přehled objektových tříd zařazených do balíčku uživatel .....	71
obr. 19 – Přehled objektových tříd zařazených do balíčku program školení.....	72
obr. 20 – Přehled objektových tříd zařazených do balíčku úloha .....	73
obr. 21 – Pohled na balíčky systému .....	74
obr. 22 – Návrh databáze .....	75
obr. 23 - Sekvenční diagram popisující vytváření uživatelských účtů .....	76
obr. 24 – Sekvenční diagram popisující přihlášení uživatele .....	76
obr. 25 – Sekvenční diagram popisující odhlášení aktéra ze systému .....	77
obr. 26 – Sekvenční diagram popisující registraci kurzů zaměstnance .....	77
obr. 27 – Sekvenční diagram popisující základní prvky správy kurzů .....	78
obr. 28 – Sekvenční diagram popisující tvorbu HTML stránky pomocí externího editoru.....	79
obr. 29 – Sekvenční diagram popisující tvorbu ticketů .....	79
obr. 30 – Schéma tříúrovňového informačního systému .....	80
obr. 31 – Plán první etapy .....	83
obr. 32 – Plán druhé etapy .....	84
obr. 33 – Plán třetí etapy.....	84
obr. 34 – Plán čtvrté etapy – první část .....	84
obr. 35 – Plán čtvrté etapy – druhá část.....	85
obr. 36 – Plán páté etapy.....	85

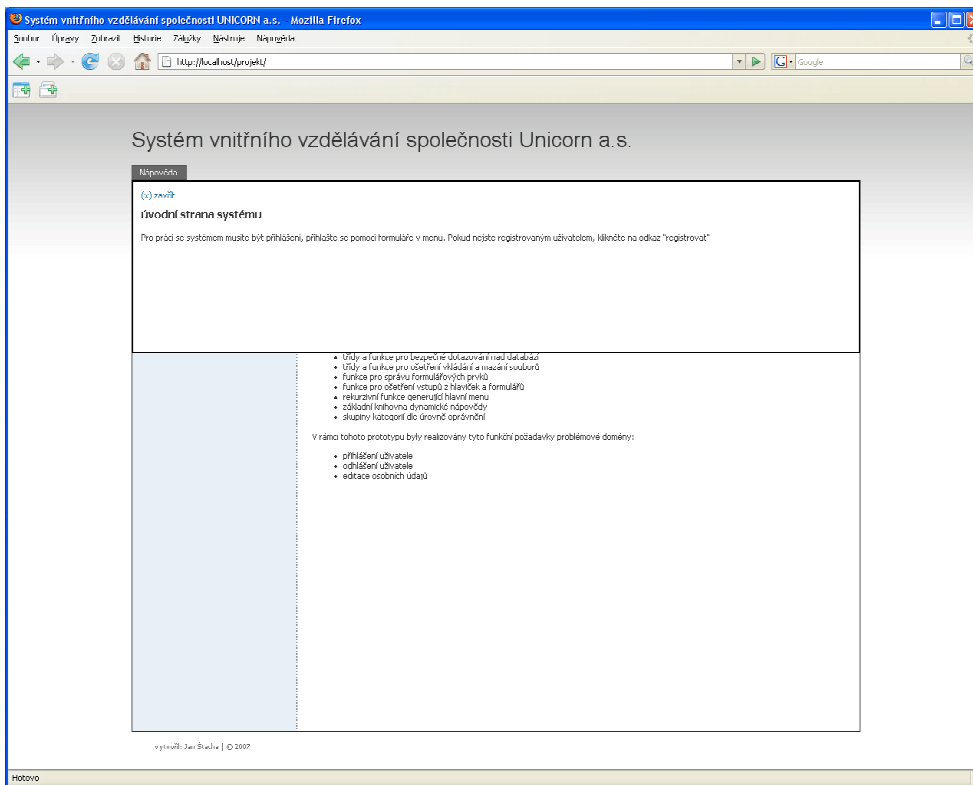
# Seznam tabulek

tab. 1 – Vymezení problému.....	21
tab. 2 – Vymezení produktu.....	21
tab. 3 – Přehled zainteresovaných osob a jejich rolí v systému.....	22
tab. 4.1 – Školený zaměstnanec.....	23
tab. 4.2 – Lektor.....	23
tab. 4.3 – Externí lektor .....	24
tab. 4.4 – Garant kurzu .....	24
tab. 4.5 – Dohlížející manager.....	25
tab. 4.6 – Administrátor .....	25
tab. 5 – Počáteční glosář .....	65
tab. 6 – Časový harmonogram.....	67
tab. 7 – Počáteční ohodnocení rizik.....	68
tab. 8 - Revidovaný seznam rizik .....	82

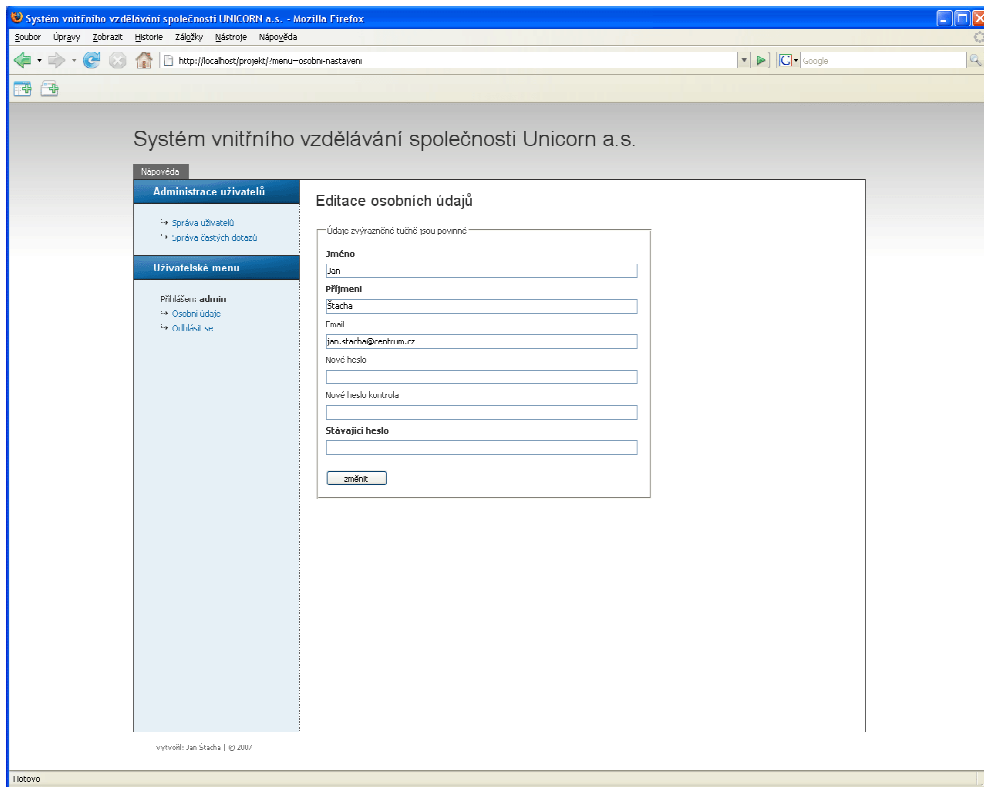
# Náhled na uživatelské rozhraní systému



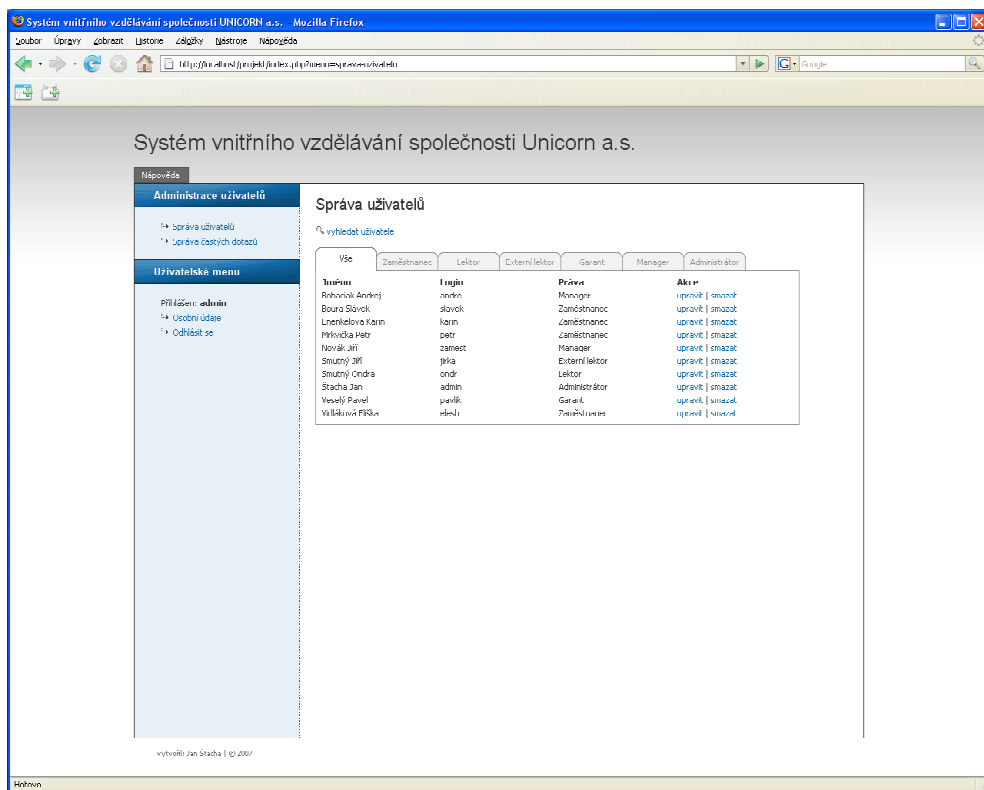
úvodní obrazovka s formulářem pro přihlášení



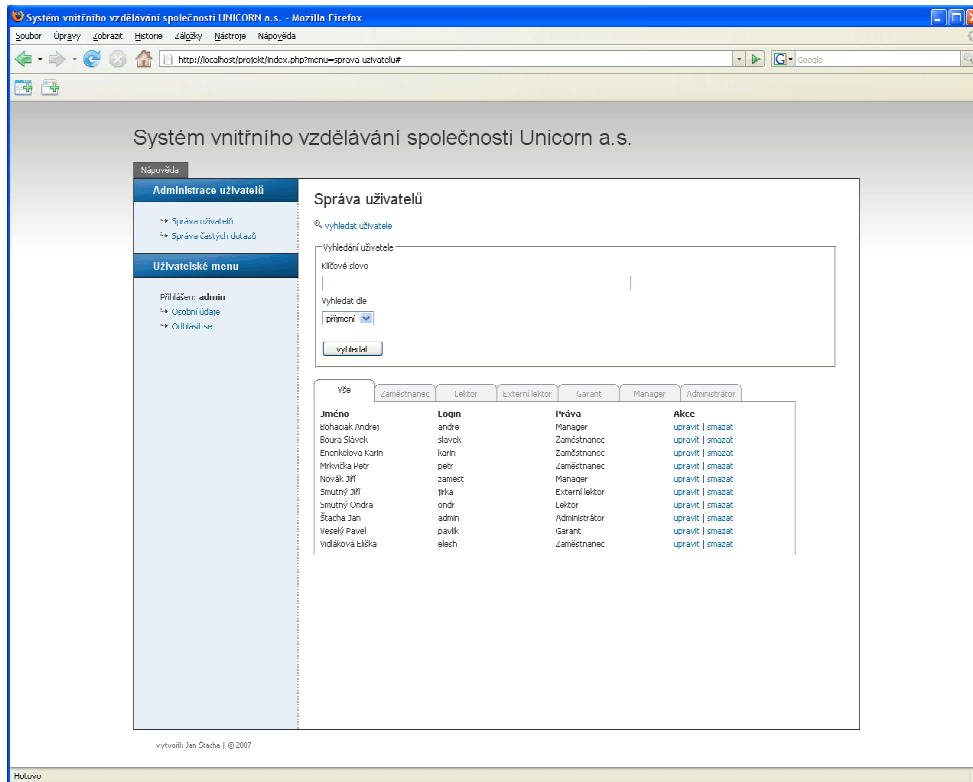
AJAXový prvek - dynamická kontextová nápověda



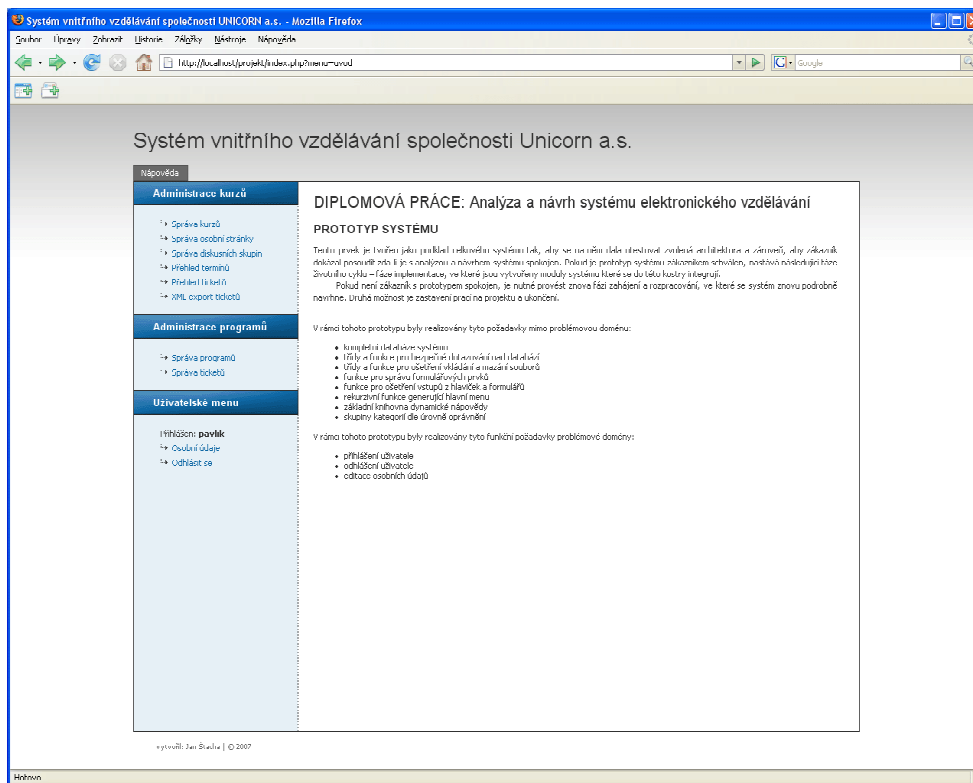
formulář editace osobních údajů



formulář správy uživatelů



AJAXový prvek – „vyskakující“ a „skrývající“ vyhledávací formulář



vzhled levého menu z pohledu garanta