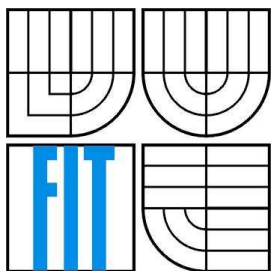




VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

ANALÝZA A NÁVRH INFORMAČNÍHO SYSTÉMU S VYUŽITÍM METODIKY RUP

ANALYSIS AND DESIGN OF INFORMATION SYSTEM USING METHODOLOGY RUP

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. ONDŘEJ SEMAN

VEDOUCÍ PRÁCE

SUPERVISOR

RNDr. JITKA KRESLÍKOVÁ, CSc.

BRNO 2007

Abstrakt

IBM Rational Unified Process je rozsáhlá iterativní metodologie vývoje software. Cílem diplomové práce je podrobný popis aspektů této metodiky a provedení analýzy a návrhu fiktivního informačního systému v rámci „incepční“ a „elaborační“ fáze.

Klíčová slova

IBM Rational Unified Process, metodika vývoje software, incepční fáze, elaborační fáze, dokument vize, případy užití, glosář, plán projektu, rizika projektu, UML, logický pohled, pohled nasazení, diagram tříd, ER diagram.

Abstract

IBM Rational Unified Process is a robust iterative software development methodology. The main goal of master's thesis is to describe the aspects of this methodology and to analyze and design a fictional information system within the scope of "inception" and "elaboration" phase.

Keywords

IBM Rational Unified Process, software development methodology, inception phase, elaboration phase, the vision document, use-cases, glossary, master plan, project risks, UML, logical view, deployment view, class diagram, ER diagram

Citace

Seman Ondřej: Analýza a návrh informačního systému s využitím metodiky RUP. Brno, 2007, diplomová práce, FIT VUT v Brně.

Analýza a návrh informačního systému s využitím metodiky RUP

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením paní RNDr. Jitky Kreslíkové, CSc.

Další informace mi poskytl odborný konzultant pan Ing. Marek Beránek ze společnosti Unicorn a.s.

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

.....
Bc. Ondřej Seman
22.května 2007

Poděkování

Rád bych poděkoval vedoucí mé diplomové práce, paní RNDr. Jitce Kreslíkové, CSc. a také mému konzultantovi z firmy Unicorn a.s., panu Ing. Marku Beránkovi za odborný dohled nad diplomovou prací.

© Bc. Ondřej Seman, 2007.

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

Obsah

| | |
|--|----|
| Obsah | 1 |
| 1 Úvod..... | 4 |
| 2 Rational Unified Process (RUP) | 5 |
| 2.1 Základní filosofie | 6 |
| 2.1.1 Iterativní vývoj..... | 6 |
| 2.1.2 Řízení požadavků..... | 7 |
| 2.1.3 Architektura založená na komponentách | 8 |
| 2.1.4 Vizualní modelování | 8 |
| 2.1.5 Ověřování kvality..... | 8 |
| 2.1.6 Řízení změn | 8 |
| 2.2 Základní stavební kameny | 9 |
| 2.2.1 Pracovníci (workers)..... | 9 |
| 2.2.2 Činnosti (Activities)..... | 9 |
| 2.2.3 Meziprodukty (artifacts) | 10 |
| 2.2.4 Pracovní procesy (workflows) | 11 |
| 2.3 Fáze životního cyklu vývoje projektu v RUP | 11 |
| 2.3.1 Inepční fáze (zahájení) | 12 |
| 2.3.2 Elaborační fáze (rozpracování) | 13 |
| 2.3.3 Konstrukční fáze | 14 |
| 2.3.4 Fáze nasazení (přechod)..... | 15 |
| 2.4 Disciplíny životní cyklu vývoje projektu | 16 |
| 2.4.1 Modelování organizace (business modeling)..... | 17 |
| 2.4.2 Požadavky | 17 |
| 2.4.3 Analýza a návrh | 17 |
| 2.4.4 Implementace | 18 |
| 2.4.5 Testování..... | 18 |
| 2.4.6 Nasazení..... | 18 |
| 2.4.7 Správa konfigurace | 18 |
| 2.4.8 Řízení projektu..... | 19 |
| 2.4.9 Prostředí | 19 |
| 2.5 Architektura systému..... | 19 |
| 2.5.1 Logický pohled | 20 |
| 2.5.2 Implementační pohled..... | 21 |
| 2.5.3 Procesní pohled..... | 21 |

| | | |
|--------|--|----|
| 2.5.4 | Pohled nasazení..... | 21 |
| 2.5.5 | Případy užití..... | 23 |
| 3 | Incepční fáze | 24 |
| 3.1 | Dokument vize | 24 |
| 3.1.1 | Úvod..... | 24 |
| 3.1.2 | Obecný popis | 24 |
| 3.1.3 | Cíle projektu..... | 24 |
| 3.1.4 | Výchozí stav..... | 24 |
| 3.1.5 | Analýza problémů..... | 25 |
| 3.1.6 | Vymezení produktu..... | 25 |
| 3.1.7 | Identifikování uživatelé | 26 |
| 3.1.8 | Profily uživatelů..... | 27 |
| 3.1.9 | Prostředí uživatelů | 28 |
| 3.1.10 | Perspektiva produktu | 28 |
| 3.1.11 | Základní funkce systému | 29 |
| 3.1.12 | Předpoklady a závislosti | 29 |
| 3.1.13 | Náklady a cenová kalkulace | 29 |
| 3.1.14 | Licencování | 29 |
| 3.1.15 | Vhodné standardy | 29 |
| 3.1.16 | Předběžné požadavky na výkon..... | 29 |
| 3.1.17 | Požadavky na prostředí..... | 30 |
| 3.1.18 | Instalace..... | 30 |
| 3.2 | Glosář pojmů..... | 31 |
| 3.3 | Use-Case model..... | 32 |
| 3.3.1 | Pohled na celý systém (hlavní případy užití) | 32 |
| 3.3.2 | Správa zaměstnanců z pohledu Administrátora | 33 |
| 3.3.3 | Správa zaměstnanců z pohledu Vedoucího směny | 37 |
| 3.3.4 | Správa služeb z pohledu Manažera hotelu..... | 44 |
| 3.3.5 | Správa služeb z pohledu Personálu na recepci..... | 47 |
| 3.3.6 | Správa klientů | 50 |
| 3.3.7 | Správa rezervací..... | 54 |
| 3.3.8 | Správa ubytování | 59 |
| 3.3.9 | Správa partnerů | 63 |
| 3.3.10 | Správa získaných dat | 67 |
| 3.3.11 | Autentizace uživatelů | 70 |
| 3.4 | Počáteční plán vývoje softwaru..... | 72 |
| 3.4.1 | Specifikace jednotlivých etap | 72 |

| | | |
|-------|---|----|
| 3.5 | Počáteční identifikace a ohodnocení rizik..... | 74 |
| 4 | Elaborační fáze..... | 75 |
| 4.1 | Ostatní požadavky (požadavky mimo funkcionalitu)..... | 75 |
| 4.1.1 | Rozsah systému..... | 75 |
| 4.1.2 | Reference | 75 |
| 4.1.3 | Použitelnost..... | 75 |
| 4.1.4 | Spolehlivost..... | 76 |
| 4.1.5 | Výkonnost..... | 76 |
| 4.1.6 | Omezení architektury..... | 76 |
| 4.2 | Architektura systému..... | 76 |
| 4.2.1 | Logický pohled | 76 |
| 4.2.2 | Pohled nasazení..... | 81 |
| 4.2.3 | Implementační pohled..... | 82 |
| 4.3 | Upravený seznam rizik..... | 82 |
| 4.4 | Plán vývoje softwaru..... | 83 |
| 4.5 | Prototyp systému | 86 |
| 5 | Závěr | 87 |
| | Literatura | 88 |
| | Seznam příloh | 89 |

1 Úvod

V současnosti existuje na trhu nespočet organizací zabývajících se vývojem softwaru. Aby byly konkurence schopné, je pro ně důležité vytvářet kvalitní produkt splňující požadavky uživatelů. Takový výrobek musí být flexibilní, musí být schopen přizpůsobit se budoucím požadavkům a musí se snadno udržovat. To však není vše. Musí být dodán včas a nesmí nijak výrazně překročit stanovený rozpočet. Pokud ale vývoj postrádá jasný a opakovatelný proces, který by jednoznačně vedl všechny členy vývojového týmu jednotlivými kroky projektu, jsou tyto cíle nedosažitelné. Především během vývoje rozsáhlých informačních systémů se vyskytují problémy spojené s neadekvátní délkou projektu. Rational Unified Process (dále jen RUP) nabízí řešení těchto problémů.

RUP je rozsáhlá objektově orientovaná iterativní metodologie vývoje softwarových produktů, kterou vytvořila společnost Rational Software (v současné době společnost Rational Software spadá pod společnost IBM). Jedná se o nesmírně propracovaný, detailně zdokumentovaný rigorózní přístup k vývoji software. Tato celosvětově používaná metodika představuje jakousi normu softwarového vývoje a vyznačuje se velkou robustností a komplexním přístupem k vývoji. Každý krok ve vývojovém cyklu lze dopředu předpovědět a lze o něm najít mnoho informací.

Sjednocení nejlepších postupů z mnoha oblastí do konzistentního procesu, který pokrývá celý životní cyklus vývoje software, umožňuje vývojovému týmu zkrátit dobu potřebnou pro dodání produktu a zároveň zvýšit kvalitu vytvářeného systému. Rational Unified Process díky integraci s vývojovými nástroji Rational a díky Unified Modeling Language (UML) podporuje celý realizační tým detailními návody a důležitými informacemi. Jinak řečeno, RUP představuje disciplinovaný přístup k přidělování úkolů a odpovědností v rámci vývojové organizace. [10]

Cílem diplomové práce je seznámit se s metodikou IBM Rational Unified Process a to především s inepční a elaborační fází a provést inepční a elaborační fází pro zvolenou business problematiku.

Druhá kapitola popisuje základní filosofii, základní pojmy této metodiky a ostatní postupy vývoje v rámci RUP. Třetí kapitola je zaměřena na inepční fází a čtvrtá kapitola obsahuje vypracovanou elaborační fází modelovaného systému. Jako problémovou business doménu jsem si zvolil informační systém hotelu.

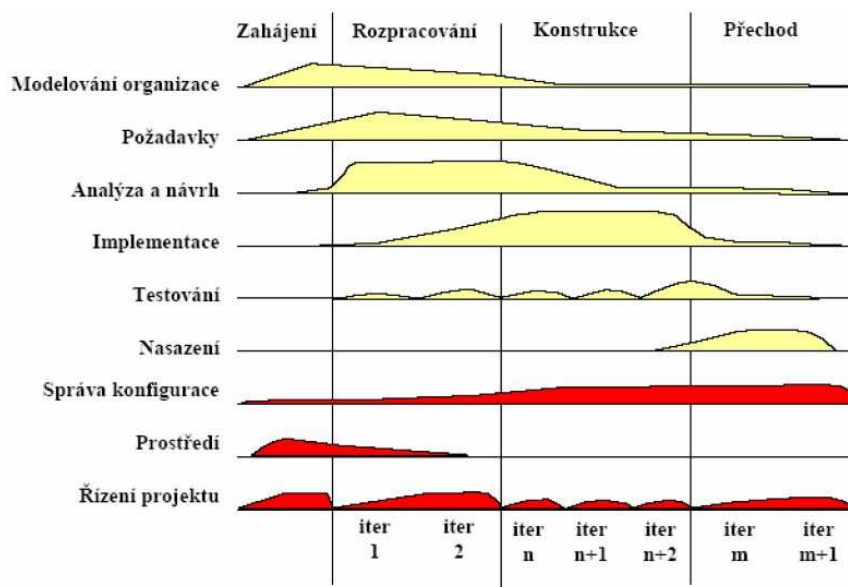
2 Rational Unified Process (RUP)

Proces vývoje softwaru (SDP, software development process) známý rovněž jako metoda tvorby softwarového vybavení (SEP, software engineering process) definuje při vývoji softwaru otázky kdo, co, kdy a jak. SEP je proces, v němž jsou uživatelské požadavky realizovány vytvořeným softwarem. Metodika USDP (Unified Software Development Process) je průmyslovým standardem SEP (procesu tvorby softwarového vybavení) pocházejícím od autorů jazyka UML (Unified Modeling Language). Tento standard je běžně označován jako Unified Process (UP). RUP je nejrozšířenější komerční variantou metodiky UP (Unified Process). Byl vyvinut společností Rational Software Corporation, dnes známý jako IBM Rational Unified Process. [2]

RUP můžeme chápat jako prostředí (platforma RUP), které slouží vývojářům. To má podobu HTML a jiných dokumentů poskytujících online nápovědu, šablony dokumentace a průvodce. Podpůrné prostředí je součástí balíku CASE nástrojů dodávaných firmou IBM (dříve firmou Rational Corporation), ale RUP lze použít jako model životního cyklu pro vývoj software obecně. [4]

Jako metodika vývoje software si klade za cíl vytvoření vysoce kvalitního produktu, který bude vyhovovat všem požadavkům uživatele a zároveň bude odpovídat normám a dodržovat daný rozpočet. Jde o procesní strukturu, která může být adoptována a rozšířena podle potřeby organizace. V metodice jsou definovány čtyři základní stavební kameny, na nichž je celý RUP postaven a z nichž sestává celý vývoj podle této metodiky. Jedná se o pracovníky (Workers), činnosti (Activities), pracovní procesy (Workflows) a meziprodukty (Artifacts). RUP pokrývá celý životní cyklus vývoje softwaru a skoro vše, co RUP obsahuje, lze označit jedním z těchto základních pojmů. [3]

RUP bývá prezentován jako dvoudimenzionální, jak je vidět na obrázku č.1. Horizontální osa reprezentuje jednotlivé fáze vývoje projektu, jak následují v čase za sebou (dynamický aspekt životního cyklu). Na vertikální ose jsou naneseny jednotlivé disciplíny, podílející se na životním cyklu (statický aspekt).



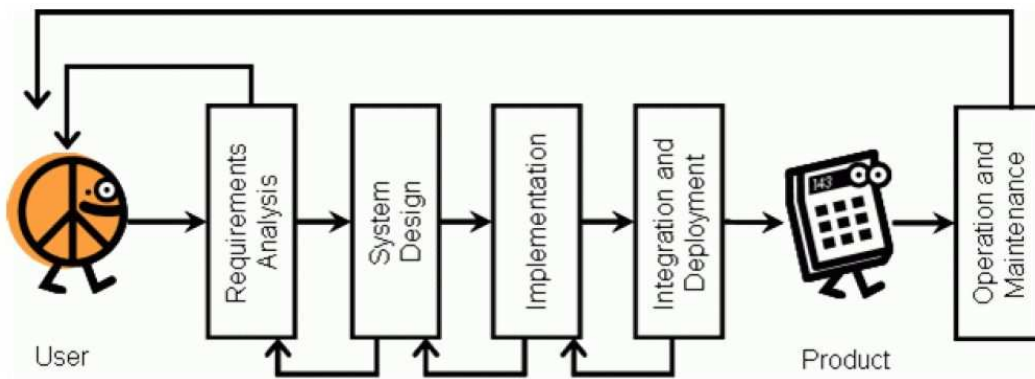
Obr. 1: Dvojdímenzionální znázornění RUP (převzato z [Lit4])

2.1 Základní filosofie

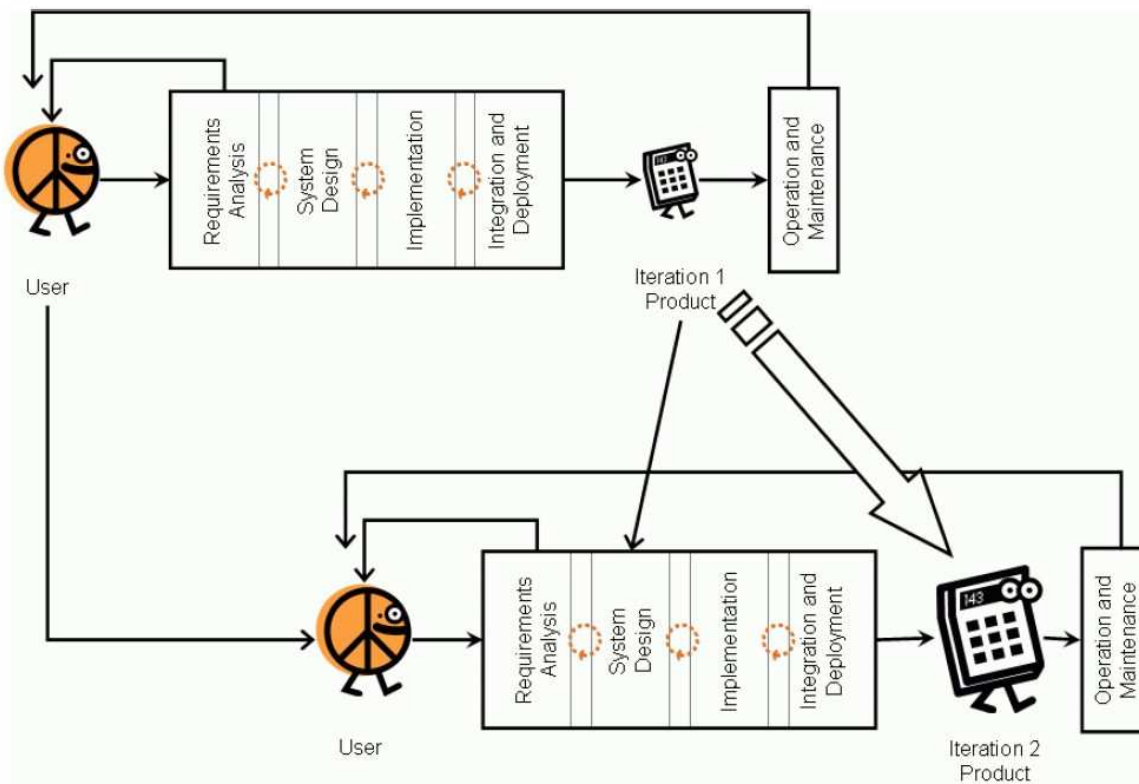
Základní filosofií metodiky Rational Unified Process je šest obecných praktik, používaných při vývoji software. Tyto praktiky mají při svém použití zajistit efektivnější vývoj, propracovanější řízení kvality a lepší výsledky.

2.1.1 Iterativní vývoj

U současných rozsáhlých systémů není již 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.[5] Pro iterativní vývoj software je charakteristické, že zahrnuje tzv. iterace. Iterací se zde rozumí opakovaný průchod fázemi vytvářeného systému s cílem obohatit ho v každé iteraci o nějaké rozšíření či vylepšení. Výsledkem každé této iterace je spustitelná verze (tzv. build). Každá iterace je ve skutečnosti malým životním cyklem „vodopád“ (Obr. 2) s tím, že uživatel je opakovaně zapojen do procesu vývoje během analýzy požadavků v rámci následující iterace. Využívá již zkušeností z práce s konstrukcí, která vznikla při předchozí iteraci. Rovněž fáze návrhu navazuje na návrh vytvořený v iteraci předchozí (Obr. 3) [4].



Obr. 2: Životní cyklus vodopád (převzato z [1])



Obr. 3: Iterativní životní cyklus (převzato z [1])

2.1.2 Řízení požadavků

Řízení požadavků je systematický přístup k zjišťování, organizování a dokumentaci měnících se požadavků. Takovýto přístup umožňuje adekvátní kontrolu celého projektu. Požadavky jsou monitorovány a zapracovány do procesu v průběhu celého vývoje softwaru.[6]

2.1.3 Architektura založená na komponentách

Komponenty v tomto kontextu označují netriviální části systému zajišťující určitou jeho funkcionalitu.[5] Je-li vyvíjený systém otevřený k použití jiných komponent, může být vývoj efektivnější. Znovupoužití hotové komponenty znamená podstatnou úsporu zdrojů. RUP poskytuje metodickou, systematickou cestu návrhu, vývoje a ověření správnosti architektury. Nabízí šablony, architektonické styly a pravidla designu.[3] Přínosem komponentové architektury je možnost rozhodnout, zda se bude požadovaná komponenta vyvíjet, kupovat, nebo se použije již hotová (odpovídající požadavkům na komponentu). Výhody přináší také do oblasti testování. Jako první se testují nejjednodušší komponenty a následně složitější celky.

2.1.4 Vizualní modelování

Pro účely vizuálního modelování nabízí RUP standardní vyjadřovací mechanismy. Unified Modeling Language (UML) je grafický jazyk pro vizualizaci, specifikaci, konstrukci a dokumentaci artefaktů. Poskytuje standardní prostředky pro znázornění systémových objektů, databázových schémat, použitých softwarových komponent apod. UML je společný jazyk k vyjádření různých modelů, ale neříká nám, jak vyvíjet software. Rational Unified Process se snaží tento jazyk co nejlépe využít k modelování. Popisuje, které modely jsou potřeba, proč je potřebujeme a jak je zhotovit.[6] Použití vizuálního modelování znamená zjednodušení komunikace vývojových týmů.

2.1.5 Ověřování kvality

Pokud se vyskytne problém až po předání vyvíjeného softwaru, je jeho nalezení a odstranění několikanásobně dražší, než v úvodních fázích vývoje. Této situaci se snažíme předejít neustálou kontrolou správné funkcionality, dostatečné spolehlivosti a vysokého výkonu vyvíjené aplikace. Hlavním nástrojem v RUP pro zajištění těchto vlastností je testování. Dokáže odhalit nesoulady mezi požadavky, návrhem a implementací. Efektivitu kontroly a testování lze zvýšit tím, že se provádí v oblastech s nejvyšším rizikem.

2.1.6 Řízení změn

Klíčovým úkolem při vývoji softwaru je dosáhnout efektivní koordinace všech aktivit a artefaktů tak, aby bylo možné opakovaně využívat standardní pracovní metody a reagovat na změny. Tím dosáhneme lepší alokace zdrojů, práce je řízena dle priorit a rizik. RUP popisuje jak kontrolovat, sledovat a monitorovat změny a umožnit úspěšný iterativní vývoj. Řeší také paralelní vývoj různých verzí softwaru a správu tzv. „buildů“.[6]

2.2 Základní stavební kameny

RUP definuje kdo, co, kdy a jak dělá. Jak bylo výše uvedeno, používá k tomu čtyři základní stavební kameny modelování.

2.2.1 Pracovníci (workers)

Odpovídají na otázku „kdo?“. Pracovník definuje chování a odpovědnost jedince nebo skupiny. Chování pracovníka je popsáno pomocí činností (activities). Každý pracovník je asociován s množinou činností. Odpovědnost je obvykle definována ve vztahu k meziproduktům (artifacts), které pracovník vytváří, modifikuje nebo kontroluje. Pracovníkem není ani tak fyzická osoba, jako spíše jakýsi „klobouk“, který může být v průběhu projektu „nasazován“ různým fyzickým osobám (jedna osoba může postupně nosit více klobouků). Pracovníka je vhodné vidět jako roli [3] (např. v divadelní hře: každý pracovník může hrát několik rolí a naopak do jedné role je možno obsadit několik pracovníků).

Jako příklad takového pracovníka mohu uvést například Databázového návrháře. Ten definuje strukturu databáze (tabulky, omezení, indexy, trigger, atd.) a další databázové konstrukce nutné ke správě a práci s daty.

2.2.2 Činnosti (Activities)

Odpovídají na otázku „jak?“. Činnost je jednotkou práce, kterou má provést jednotlivec nebo skupina, a která má vyústit ve smysluplný výsledek v kontextu projektu. Činnost má jasně definovaný účel, obvykle vyjádřený jako vytvoření nebo modifikace meziproduktu (např. modelu, třídy, plánu apod.). Rozsah činností je různý, od několikahodinových po několikadenní. Obvykle se však týká jen jednoho pracovníka a ovlivňuje nejvýše několik málo meziproduktů. Každá činnost má definovány své vstupní a výstupní meziprodukty.

| Příklady činností | |
|--------------------|------------------------------|
| Pracovník | Činnost |
| Návrhář | Návrh projektu |
| Architekt | Analýza a návrh architektury |
| Projektový manažer | Plán a integrace |

Tab. 1: Příklady činností

Činnosti se dále dělí do kroků. Obecné dělení kroků spočívá ve stanovení tří základních kategorií:

- Úvahy (Thinking Steps).
- Provádění (Performing Steps).
- Přezkoumání (Reviewing Steps).

Konkrétní činnost má vždy stanovenou podrobnou posloupnost kroků, které vedou k jejímu uspokojivému provedení. Jako příklad lze uvést např. činnost návrh databázového designu (Activity: Database Design), která se skládá z následující posloupnosti kroků:

- provést mapování tříd z designu do datového modelu
- optimalizace datového modelu (datových struktur) z hlediska výkonnosti
- optimalizace přístupu k datům (použití indexů a jejich typů apod.)
- definice charakteristik uložení dat (velikost stránky apod.)
- definice referenčních tabulek a přednastavených hodnot atributů (např. uložení nejpoužívanější tabulky na nejrychlejší disk, optimalizace vyrovnávacích pamětí apod.)
- definice pravidel datové a referenční integrity
- distribuce chování třídy do databáze (uložené procedury apod.)
- přezkoumání výsledku (ověření kvality a integrity datového modelu)

Každý z uvedených kroků je v rámci RUP velmi podrobně dokumentován. RUP poskytuje řadu tipů, návodů, pomůcek a rad potřebných ke zvládnutí daného kroku. [3]

2.2.3 Meziprodukty (artifacts)

Odpovídají na otázku „co?“. Představují informaci, která je vytvořena, modifikována a používána v procesu vývoje. Je hmatatelným výsledkem projektu. Meziprodukty jsou používány jako vstup pro vykonání činnosti určitou osobou a jsou cílem nebo výstupem takovéto činnosti. Za vytvoření a správnost meziproduktu odpovídá definovaný pracovník. Formy meziproduktů mohou být různé a pro jejich vytváření lze použít nejrůznější nástroje (např. modely pomocí Rational Rose, plány projektů pomocí Microsoft Project apod.).

| Příklady forem meziproduktů | |
|-----------------------------|-----------------------------------|
| Typ | Příklady |
| Model | model případu užití, model návrhu |
| Element modelu | třída, případ užití, subsystem |
| Dokument | část specifikace, plán |

Tab. 2: Příklady forem meziproduktů

Samotné RUP definuje velké množství meziproduktů, rozdělených do několika základních skupin. V konkrétním projektu není typicky nutné použít všechny. RUP je z principu určen k tomu, aby byl přizpůsoben konkrétním potřebám projektu. Je tak možné využít pouze podmnožinu meziproduktů, stejně jako podmnožinu pracovníků a činností. [3]

2.2.4 Pracovní procesy (workflows)

Odpovídají na otázku „kdy?“. Pouhý výčet všech pracovníků, činností a meziproductů ještě nepředstavuje proces. Zbývá definovat smysluplnou posloupnost činností a stanovit interakce mezi pracovníky.

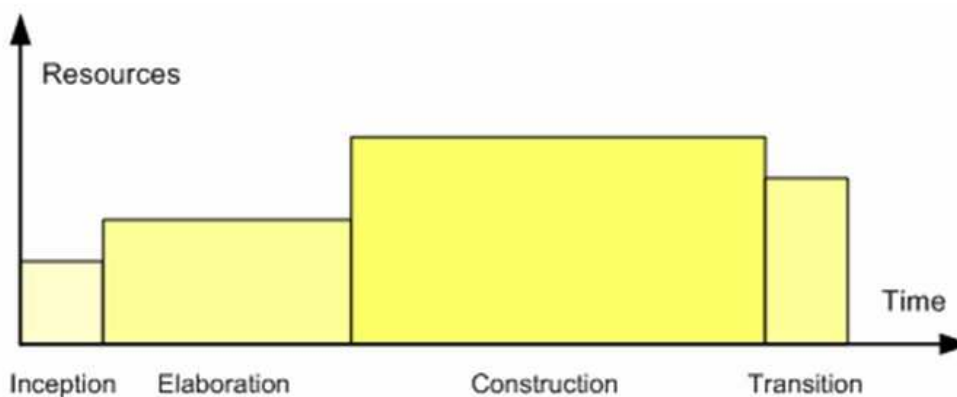
Pracovní proces je posloupnost činností vedoucí k vytvoření požadovaného výsledku. Může být s výhodou modelován v UML jako sekvenční diagram (sequence diagram), diagram spolupráce (collaboration diagram) nebo diagram činnosti (activity diagram). Phillipe Kruechten však ve své knize *The Rational Unified Process: An Introduction* (Addison-Wesley, Boston, USA 2000) upozorňuje, že obvykle není možné dokonale postihnout veškeré (i skryté) závislosti mezi jednotlivými činnostmi. Činnosti bývají navzájem značně protkány. Diagram pracovního procesu tedy nesmí být interpretován mechanicky. Každý pracovní proces se skládá z několika (typicky nejvýše z desíti) činností. RUP definuje devět klíčových pracovních procesů (Core Workflows).

Protože každý z devíti klíčových procesů pokrývá širokou oblast, dělí je dále RUP na tzv. podrobnosti pracovních procesů (Workflow Details). Ty se používají k vyjádření specifické skupiny úzce souvisejících činností. Například klíčový pracovní proces Workflow: Implementation se skládá z pěti podrobných pracovních procesů:

- vytvoření struktury implementace (Workflow Detail: Structure the Implementation Model)
- plánování integrace (Workflow Detail: Plan the Integration)
- implementace komponent (Workflow Detail: Implement Components)
- integrace každého podsystému (Workflow Detail: Integrate Each Subsystem)
- integrace systému (Workflow Detail: Integrate the System) [3]

2.3 Fáze životního cyklu vývoje projektu v RUP

V metodice Rational Unified Process je vývoj projektu rozdělen do čtyř základních fází (Obr. 4). Incepční fáze (Inception), elaborační fáze (Elaboration), konstrukční fáze (Construction) a fáze nasazení (Transition). Všechny fáze končí definovanými hlavními milníky (což jsou indikátory pokroku v projektu). Milníky stanovují určité cíle, které musí být dosaženy, abychom mohli považovat milníky za dosažené. Každá fáze může obsahovat jednu nebo více iterací. Přesný počet iterací v jednotlivých fázích závisí na velikosti projektu. Obrázek č. 6 naznačuje běžnou časovou náročnost těchto fází. Každá z těchto fází je zakončena milníkem. V něm se rozhodne, jestli je potřeba ve vývoji pokračovat, změnit postup vývoje nebo vývoj ukončit.



Obr. 4: Běžná časová náročnost fází životního cyklu projektu v RUP (převzato z [7])

2.3.1 Incepční fáze (zahájení)

V této fázi jde především o vymezení požadavků na systém, na němž se shodnou všechny zainteresované strany. Je třeba vymezit rozsah vytvářeného systému, identifikovat prvky, s nimiž bude systém komunikovat, a definovat povahu těchto prvků. To vede k nalezení všech případů užití a k vytvoření modelů případů užití pro klíčové činnosti.

Do této fáze však mohou spadat rovněž určité návrhářské či implementační práce – je-li rozhodnuto vytvořit technický prototyp, jenž by potvrdil správnost koncepce. Obvykle nedochází během inepční fáze k pracovnímu postupu testování, protože jedinými softwarovými artefakty jsou prototypy, které budou stejně zahozeny. [2]

U projektů, při nichž se pouze upravuje stávající systém, je tato fáze podstatně méně náročná, neboť výše uvedené kroky byly již dříve zčásti provedeny. [5]

Vstupy

- Původní vize
- Stávající systém (pokud existuje)
- Úvodní požadavky na systém

Výstupy

- Vize – dokument obsahující obecnou vizi klíčových požadavků projektu, jeho hlavní rysy a omezení.
- Počáteční model případů užití (kompletní přibližně z 10 až 20 %)
- Počáteční plán vývoje softwaru. V této fázi nemusí být údaj ještě příliš přesný, v dalších iteracích se bude postupně zpřesňovat.
- Počáteční ohodnocení rizik.
- U složitějších projektů též úvodní návrh architektury - může se jednat o “nezávazný” prototyp, který se později přepracuje.

Milník

Na závěr inepční fáze nastává důležitý milník projektu, který je také nazýván „Rozsah systému“. Jeho smyslem je prokázat, že projekt je realizovatelný v navržené kvalitě, kvantitě, termínu a rozpočtu. Na základě splnění následujících podmínek se rozhodne, zda se bude ve vývoji projektu pokračovat, nebo zda je třeba celý projekt přehodnotit či zastavit.

- Všechny zúčastněné strany souhlasí s rozsahem projektu, časovým harmonogramem a cenovým odhadem.
- Všechny zúčastněné strany souhlasí se zachycenými klíčovými požadavky.
- Výstupní dokumenty musí prokázat, že dodavatel pochopil zadání.
- Odhady rizik, nákladů a dob provádění musí být reálné.

2.3.2 Elaborální fáze (rozpracování)

Cílem této fáze je navrhnout stabilní model architektury systému, který poslouží jako základ pro jeho implementaci. Model architektury vychází ze základních požadavků na systém, vytváří se jeden nebo několik prototypů. Navazuje na inepční fázi, ve které se zaměřuje především na kritické případy užití, které typicky zahrnují technická rizika projektu. Elaborace je klíčovou a nejrizikovější fází celého projektu.

Výsledkem úvodní iterace by měl být první funkční prototyp architektury. Další iterace v této fázi slouží k rozšíření modelu návrhu a implementace o další případy užití (na základě stanovených priorit). Počet těchto iterací závisí mj. na rozsahu projektu a zkušenostech dodavatele s danou oblastí.[5]

Výstupy

- Jeden či více prototypů architektury software.
- Model případů užití (nejméně z 80% kompletní). Byli identifikováni všichni aktéři a všechny klíčové případy užití, k většině případů užití existuje popis.
- Model návrhu: proveden návrh alespoň pro 10% případů užití.
- Datový model: byly identifikovány klíčové prvky datového modelu (tabulky, relace).
- Další požadavky netýkající se funkcionality a požadavky, které nelze přiřadit jednotlivým případům užití.
- Upravený seznam rizik.
- Plán vývoje softwaru.
- U systémů se složitým uživatelským rozhraním lze v této fázi vytvořit jeho prototyp.[5]

Milník

Na závěr elaborační fáze nastává druhý milník projektu, často nazýván „Definice architektury“. Zde se prověřují detailní vlastnosti systému a jeho rozsah. Výběr architektury a odstranění hlavních rizik jsou nejdůležitější úkoly tohoto milníku. V této fázi by měl projekt splňovat následující podmínky:

- Vize produktu je již neměnná.
- Bylo identifikováno více než 80% všech případů užití, alespoň polovina z tohoto počtu byla již analyzována.
- Návrh a implementace byly provedeny alespoň u 10% případů užití.
- Návrh architektury je již konečný, nebude docházet k zásadním změnám.
- Jsou definovány postupy pro hodnocení a testování.
- Testování prototypů prokázalo, že byly správně identifikovány a podchyceny rizikové oblasti.
- Plán iterací pro fázi konstrukce je dostatečně podrobný a kvalitní, aby bylo možné v projektu pokračovat. Tyto plány jsou podpořeny důvěryhodnými odhady dalšího vývoje.
- Zadavatelé souhlasí s realizací software, s použitím navrhované architektury a plánu vývoje. Shodují se na tom, že takový software splní požadavky vytyčené ve vizi.
- Skutečná spotřeba zdrojů nepřevyšuje plán.[5]

2.3.3 Konstrukční fáze

V této fázi dojde k dokončení návrhu a vytvoření všech zbývajících komponent a jejich integraci do produktu. Vznikne první funkční verze systému, která je důkladně otestována. Na rozdíl od předchozích dvou fází je tato zaměřena převážně na implementaci. Konstrukční fáze je výrobním procesem, v němž se klade důraz na efektivní řízení zdrojů a kontrolu kvality, kvantity, termínů a rozpočtu.

V tomto okamžiku by měly být požadavky na systém již stabilní, správa požadavků se omezuje pouze na případné zapracování dodatečných požadavků na změny. Iterace slouží převážně k návrhu a implementaci dalších funkcí systému. Konečným výstupem z konstrukční fáze je produkt, který je připraven k předání zadavatelům.

Výstupy

- Software - první funkční verze vyvíjeného systému.
- Plán nasazení - první verze.
- Model implementace - soubor komponent vytvořených v této fázi.
- Model testování - soubor testů vyvinutý pro ověřování verzí systému vyvinutých ve fázi konstrukce.
- Uživatelská dokumentace - první verze manuálů. Je třeba hlavně u systémů s rozsáhlým uživatelským rozhraním.

- Model návrhu - po dokončení této fáze by měly být identifikovány již všechny požadavky a zapracovány do modelu návrhu.
- Plán iterací pro fázi předávání.
- Datový model - úplná specifikace datového modelu (tabulky, relace, indexy...).

Milník

Na konci konstrukční fáze nastává třetí milník projektu, nazývaný „Beta verze“. V této fázi prověřujeme, zda software, provozní prostředí a uživatelé jsou, bez velkých rizik, připraveni k nasazení do provozu. Kromě „alfa“ či „beta“ verze software musí být k dispozici uživatelský manuál včetně aktuální dokumentace. Pokud nebudou splněny následující podmínky, je nutné nasazení produktu odložit a vytvořit novou verzi, která odstraní veškerá rizika.

- Verze je dostatečně stabilní a zralá na to, aby byla nasazena do provozu mezi běžné uživatele.
- Zadavatel je připraven k nasazení systému.
- Skutečné náklady vůči plánovaným jsou stále ještě přijatelné.

2.3.4 Fáze nasazení (přechod)

Fáze nasazení je poslední fází životního cyklu RUP. Smyslem této fáze je předání finální verze systému koncovým uživatelům. Typicky se skládá z více jak jedné iterace, zahrnující opravu všech chyb nalezených v beta-verzi a přípravu pro přenesení systému na všechny počítače uživatele. Před začátkem fáze nasazení je důležité vytvořit její plán a jeho kontrolu.

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. [5]

Zde uvádím shrnutí činností spadajících do této fáze:

- „Beta testování“ pro validaci nového systému vzhledem k požadavkům uživatele.
- Příprava uživatelského pracoviště na přijetí nového softwaru.
- Paralelní nasazení společně s nahrazovaným systémem.
- Konverzi operačních databází.
- Školení administrátorů, správců a uživatelů.
- Předání systému do rutinního provozu.

Výstupy

- Software - konečná funkční verze splňující všechny požadavky zadání, jsou k dispozici instalační programy a média.
- Manuály - uživatelská dokumentace k aktuální verzi produktu.
- Materiály pro školení uživatelů k aktuální verzi produktu.

Milník

Čtvrtým a posledním milníkem, nazývaným též „Nasazení“, končí vývoj produktu. Ten je předán zadavateli a nasazuje se do běžného provozu. Pro hodnocení úspěšnosti projektu se používají tyto kritéria: spotřeba zdrojů (např. finanční) a spokojenost uživatelů. Po nasazení systému do běžného provozu, obdrží zadavatel zpravidla řadu připomínek a návrhů na zlepšení. Tyto připomínky řeší servisní tým (podpora ze strany dodavatele). V závislosti na závažnosti připomínek se nyní rozhodne, zda se zahájí nový projekt a vytvoří další verze celého software či zda bude pouze stačit provést úpravy některých komponent.

2.4 Disciplíny životního cyklu vývoje projektu

V této kapitole si popíšeme statický aspekt životního cyklu vývoje projektu v RUP. Jsou jim disciplíny, které představují vertikální osu na obrázku č.1, uvedeném na začátku kapitoly Rational Unified Process. Je zde devět disciplín, které jsou rozděleny do dvou skupin. První skupinu představují tzv. „hlavní disciplíny“, které zajišťují vlastní práci na projektu. Druhou skupinou jsou tzv. „podpurné disciplíny“, které slouží k řízení a koordinaci hlavních disciplín.

Hlavní disciplíny

- Modelování organizace
- Požadavky
- Analýza a návrh
- Implementace
- Testování
- Nasazení

Podpurné disciplíny

- Správa konfigurace
- Prostředí
- Řízení projektu

2.4.1 Modelování organizace (business modeling)

Komunikace mezi komunitou odpovědnou za „business modeling“ a komunitou softwarových návrhářů je často hlavním problémem během vývoje většiny projektů. Proto jsou v metodice Rational Unified Process propojeny obě činnosti na úrovni řízení projektu. Pro zlepšení komunikačních kanálů mezi těmito komunitami se využívá společná terminologie.

Pro účely modelování organizace se využívají prostředky UML. Jednotlivé scénáře organizace jsou znázorněny pomocí tzv. „business modelu“ případů užití. Jejich realizaci popisuje tzv. „business object-model“.

Tato disciplína se během vývoje v některých projektech vypouští.

2.4.2 Požadavky

Určení požadavků je jedním z prvních kroků životního cyklu vyvíjeného systému. Jejich smyslem je popsat co by měl systém dělat (jaké by měly být jeho funkce). V zásadě rozlišujeme dva typy požadavků. Prvním typem jsou funkční požadavky, jež určují, jaké chování bude systém nabízet. Druhým typem jsou nefunkční požadavky, které definují omezení kladená na systém nebo proces vývoje.

Požadavky by měly odpovídat pouze na otázku co by měl systém dělat, nikoli jak by to měl dělat. Je to velmi důležitý rozdíl. Můžeme určit, co by měl systém dělat a jaké chování by měl poskytovat, aniž bychom cokoli říkali o způsobu, jak bude dané funkce dosaženo.

Úspěch projektu je podmíněn splněním požadavků formulovaných zákazníkem. Proto je musíme systematicky evidovat a odpovídajícím způsobem dokumentovat a zpracovávat jejich případné změny.

2.4.3 Analýza a návrh

Tato disciplína propojuje požadavky na systém s jeho implementací. Analýza je zaměřena hlavně na tvorbu logického modelu připravovaného systému (analytický model), který zachycuje funkce, jež tento systém musí poskytovat. Smyslem návrhu je přesná specifikace způsobu, jak takové funkce implementovat, tj. nadefinovat architekturu systému včetně jeho členění na komponenty.

Návrh je primární modelovací aktivitou v poslední etapě elaborační fáze a v počátcích konstrukční fáze. Výsledkem analýzy a návrhu je tzv. návrhový model, který je založen na analytickém modelu. Návrhový model je jakousi předlohou pro zdrojový kód vyvíjeného systému.

Návrhový model obsahuje

- Jeden návrhový systém.
- Návrhové podsystémy.
- Návrh realizace případů užití.
- Rozhraní.
- Návrhové třídy.
- První verzi diagramu nasazení.

2.4.4 Implementace

Implementace spočívá v převodu komponent vymezených v návrhovém modelu do spustitelné podoby - tj. vytvořit programy, knihovny a všechny potřebné datové soubory. Je hlavní disciplínou prováděnou během konstrukční fáze. V rámci implementace se též provádí organizace zdrojového kódu, integrace komponent a základní testy jejich funkčnosti.

RUP popisuje jak použít existující komponenty, nebo jak implementovat nové komponenty tak aby bylo možno systém jednoduše udržovat a tyto komponenty znovu použít.

2.4.5 Testování

Smyslem disciplíny testování je vytvořit a provést soubor testů pro ověření interakce mezi objekty, funkčnosti komponent systému a jejich správné integrace.

Postup v rámci testování

- Určení předmětu testů.
- Ověření způsobu testování.
- Ověření stability systému.
- Tvorba testů
- Testování a hodnocení.

2.4.6 Nasazení

Hlavní náplní je úspěšně předat vytvořený produkt koncovým uživatelům. Disciplína nasazení systému je hlavní činností prováděnou ve fázi nasazení, ale mnoho činností v rámci této disciplíny je potřeba zahrnout do předcházejících fází kvůli připravovanému nasazení produktu na konci konstrukční fáze.

2.4.7 Správa konfigurace

Správa konfigurace je disciplína nezbytná pro zachování konzistence systému. Jejím hlavním úkolem je předcházení problémům jako jsou současné update (dva a více pracovníků vyvíjí odděleně stejný

meziprodukt a vzájemně si přepíší svou práci), omezená informovanost (je opravena chyba v meziproduktu sdíleném vývoji a někteří z nich o tom nejsou informováni), nešíření oprav do všech verzí (nalezení a opravení chyby u verze systému, která je např. ve fázi vývoje, ale tato oprava není distribuována do všech ostatních verzí systému).

2.4.8 Řízení projektu

Řízení projektu je hlavní podpůrnou disciplínou zajišťující plánování, koordinaci prací na projektu a řešení překážek pro úspěšné předání produktu. Významným úkolem této disciplíny je též zajištění potřebných personálních i materiálních zdrojů a řízení možných rizik.

2.4.9 Prostředí

Zatímco řízení projektu řeší otázky plánování a koordinace - tj. "kdo kdy a co bude dělat", úkolem správy prostředí je určit "jak" (stanovit pravidla pro jednotlivé činnosti) a "čím" (dát k dispozici příslušné nástroje). [5]

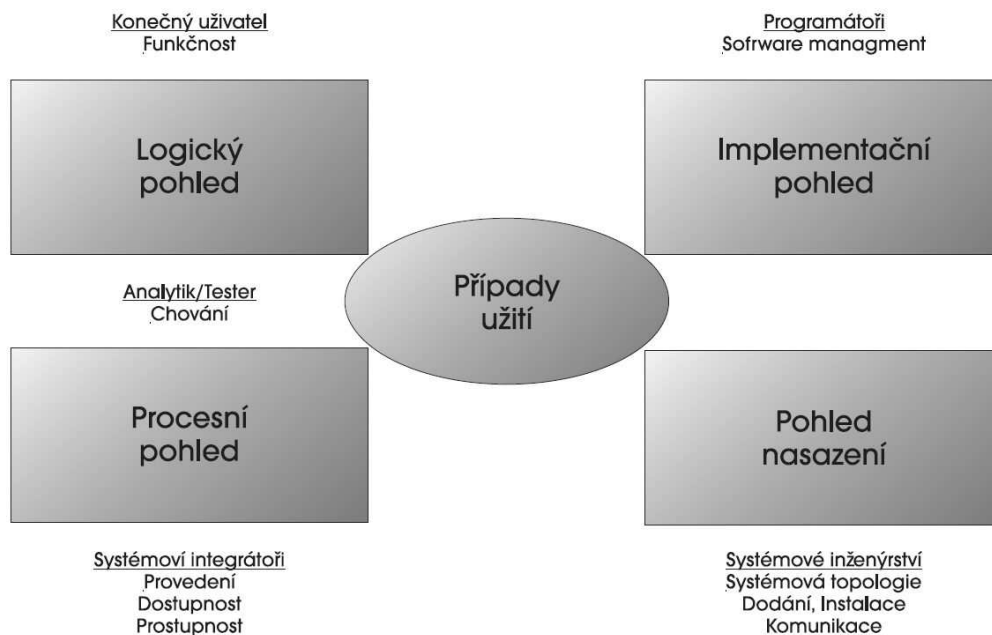
2.5 Architektura systému

Jednou ze základních filozofií metodiky RUP je tvorba softwaru pomocí komponent. Architektura systému představuje popis o struktuře systému, použitých komponentách a jejich vzájemných vztazích, způsobu spolupráce apod.

Metodika Rational Unified Process je především zaměřena na vizuální modelování. Využití modelů, zobrazujících zjednodušeně realitu, pomáhá pochopit problém a znázornit ho spolu s jeho řešením.

Na systémovou architekturu je možné nahlížet různými pohledy. A to z toho důvodu, že nás v některých případech zajímají odlišné stupně podrobností. Pohledy jsou rozděleny podle účelu a říkáme jim architektonické pohledy (vrstvy). Takovýto pohled značí zjednodušený popis systému (abstrakci) z různých perspektiv a pozic, zahrnující jednotlivé oblasti zájmu.

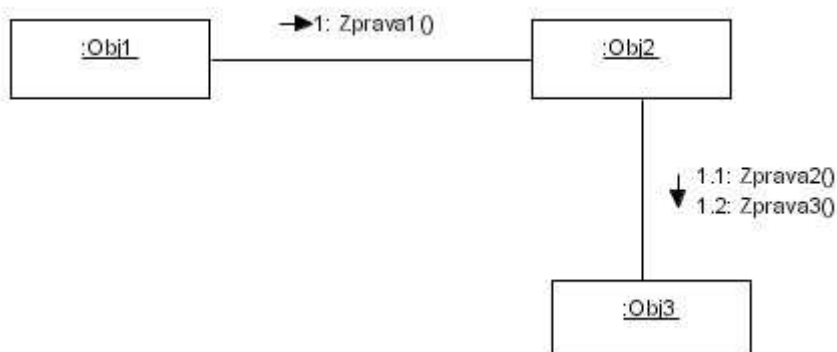
V RUP je navrženo pět základních pohledů na architekturu (Obr. 5). Tento model se nazývá Model architektury 4+1. Znázorňuje architektonicky významné prvky, ovlivňující strukturu systému, jeho robustnost, výkon, pružnost a rozlišitelnost. Jednotlivé jeho pohledy řeší různé aspekty fungování systému, nejsou na sobě však nezávislé a do určité míry se překrývají.



Obr. 5 Model architektury 4+1 (převzato z [6])

2.5.1 Logický pohled

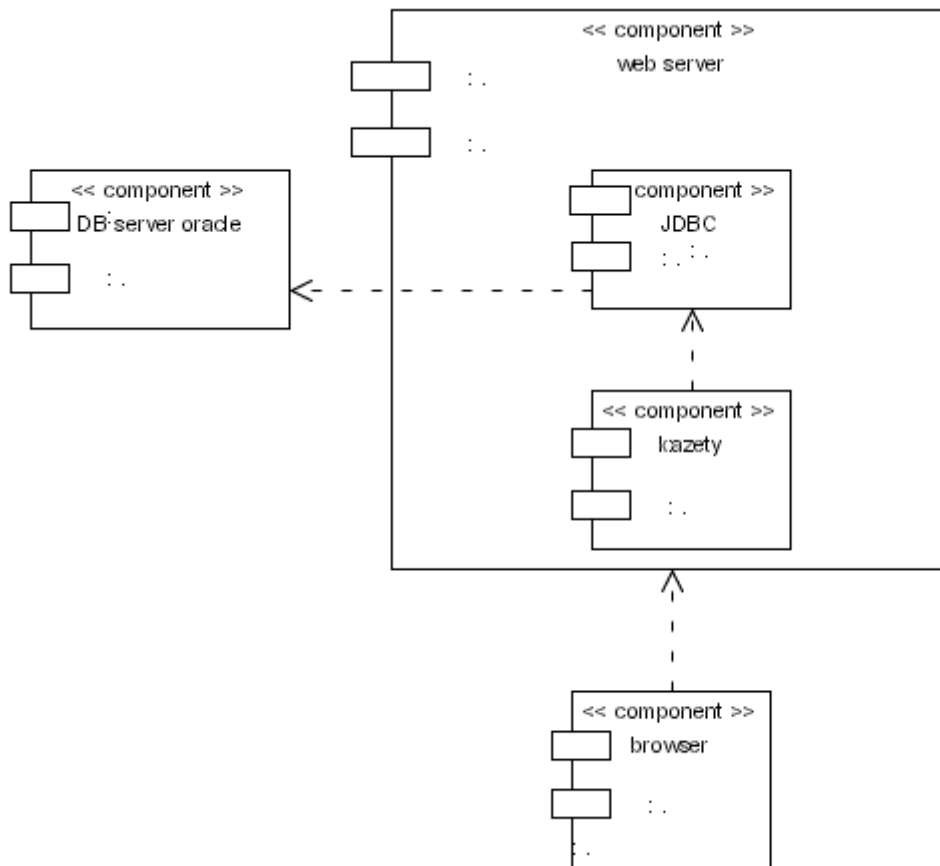
Znázorňuje logickou strukturu systému z hlediska výsledné funkcionality. Jde o abstraktní model návrhu, který popisuje hlavní moduly, subsystémy a třídy. Hlavním prostředkem je zde diagram tříd znázorňující objektové třídy a vztahy mezi nimi (asociace, dědičnost) a objektový diagram popisující vztahy mezi konkrétními instancemi jednotlivých tříd. [5] Pro popis funkcionality určitého objektu lze použít stavové diagramy. Komunikace mezi jednotlivými objekty (tj. vzájemné volání metod) se zpravidla znázorňuje pomocí diagramů spolupráce (Obr. 6), popřípadě sekvenčním diagramem. Sekvenční diagram se použije v případě, že u komunikace objektů je důležitým aspektem posloupnost jednotlivých kroků.



Obr. 6: Diagram spolupráce (převzato z [Lit8])

2.5.2 Implementační pohled

Zobrazuje organizaci statických softwarových komponent (exe, dll, html...). Zachycuje také rozdělení systému do jednotlivých modulů. Diagram komponent (Obr. 7) představuje hlavní prostředek znázorňující tento pohled.



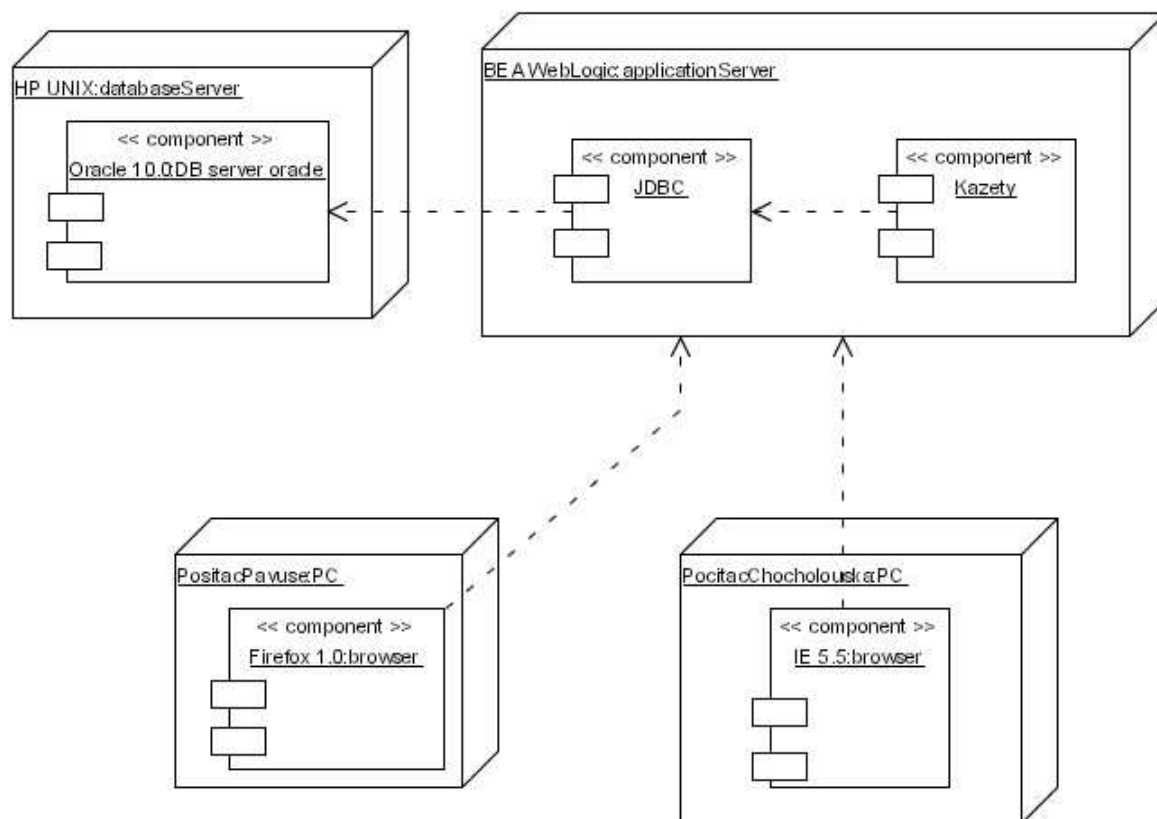
Obr. 7: Diagram komponent (převzato z [8])

2.5.3 Procesní pohled

Tento pohled zachycuje software jako množinu navzájem komunikujících procesů (proces je chápán jako soubor úloh). Odpovídá na otázku konkurence a paralelismu procesů. Modelování procesů a meziprocesové komunikace se provádí pomocí procesního diagramu. [5] Tento pohled je z hlediska RUP nepovinný.

2.5.4 Pohled nasazení

Soustředí se na návaznost spustitelných programů a ostatních komponent, na topologii hardwarových a dalších softwarových komponent. Modelování se typicky provádí pomocí diagramu nasazení (Obr. 8) v UML. Tento diagram znázorňuje jednotlivé hardwarové komponenty a jejich propojení (v terminologii UML “procesory” a “zařízení”).

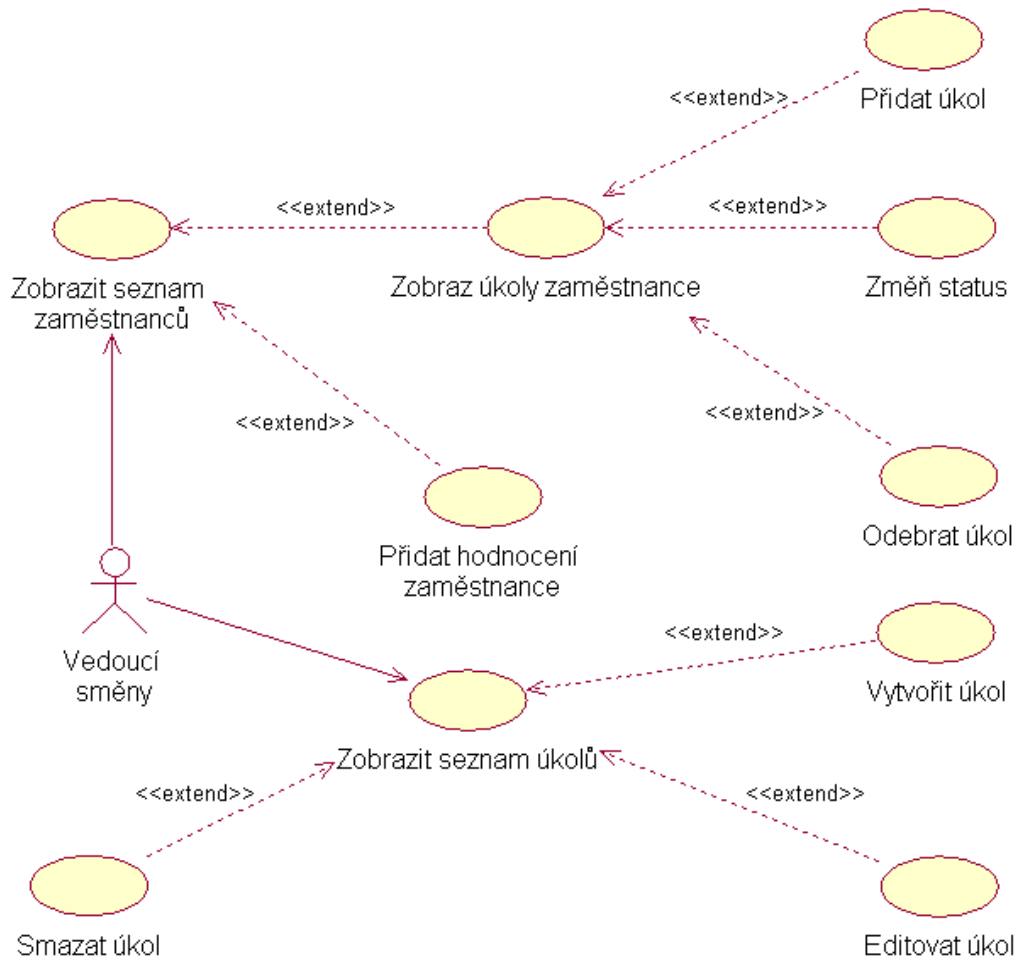


Obr. 8: Diagram nasazení (převzato z [8])

2.5.5 Případy užití

Představují pohled na systém z hlediska jeho funkcionality pro koncového uživatele. Obsahuje několik klíčových scénářů nebo případů užití. Ty se používají k objevení a následnému návrhu architektury (ve fázi inepční a fázi elaborační).

Tento pohled je jádrem architektury systému, zbývající pohledy zajišťují realizaci zde specifikovaných požadavků. Diagram případů užití (Obr. 9) představuje hlavní prostředek znázorňující tento pohled. Využívá se pro znázornění statické struktury požadavků.



Obr. 9: Diagram případů užití

3 Incepční fáze

Incepční fáze, jinak nazývaná fáze zahájení, vymezuje především požadavky na systém, na nichž se shodnou všechny zainteresované strany. Dalé specifikuje rozsah vytvářeného systému, identifikuje prvky, s nimiž bude systém komunikovat, a definuje povahu těchto prvků. Podstatou této fáze je nalezení všech případů užití a jejich aktérů a vytvoření modelů případů užití pro klíčové činnosti.

3.1 Dokument vize

V tomto dokumentu specifikuji obecnou vizi klíčových požadavků projektu, jeho hlavní rysy a omezení.

3.1.1 Úvod

Hlavním cílem této části je především vymezení požadavků na systém. Je třeba vymežit rozsah vytvářeného systému, identifikovat prvky, s nimiž bude systém komunikovat, a definovat povahu těchto prvků

3.1.2 Obecný popis

Dle zadání je předmětem diplomové práce zvolit si business problematiku, na které lze vhodně demonstrovat analýza a návrh informačního systému s využitím metodiky RUP. Pro tyto potřeby jsem zvolil jako modelovanou doménu informační systém hotelu. Tento systém budu navrhovat pro fiktivní firmu.

3.1.3 Cíle projektu

Předmětem projektu je navrhnout Informační systém hotelu (dále jen IS hotelu).

IS hotelu umožní

- Veškerou evidenci a správu zákazníků a zaměstnanců.
- Správu hotelových pokojů a jejich rezervací.
- Správu služeb poskytovaných hotelem.
- Správu získaných dat o obsazenosti hotelu.

3.1.4 Výchozí stav

Firma, která provozuje hotel, nepoužívá žádný rozsáhlejší informační systém, pouze jakési samostatné evidence stálých zákazníků, rezervací a zaměstnanců. To zabraňuje efektivnímu využití zaměstnanců a plánování chodu hotelu.

3.1.5 Analýza problémů

Problémy

- Neefektivní využití zaměstnanců.
- Nejednotný a nesrozumitelný výstup historie chodu hotelu.
- Nepřehledná evidence zákazníků.
- Neefektivní správa služeb.
- Neefektivní a náročné plánování chodu hotelu.

Dopad problémů na

- Zaměstnance celé firmy.
- Zákazníky hotelu.

Následky problémů

- Neefektivní chod celého hotelu.
- Možné uniky zisků.
- Špatná komunikace mezi vedením a jednotlivými zaměstnanci.

Úspěšné vyřešení problémů

- Usnadnění práce všem zaměstnancům.
- Zpřístupnění a zpřehlednění chodu hotelu pro management podniku.
- Zajištění efektivní evidence rezervovaných ubytovacích míst a optimální přístup při rezervaci pokojů.
- Zajištění optimálního rozdělení pracovních činností, což umožní dynamičtější chod podniku.
- Zajištění efektivní správy služeb poskytovaných hotelem.
- Zvýšení zisků firmy provozující hotel.
- Celkové zrychlení a zefektivnění řízení chodu hotelu.

3.1.6 Vymezení produktu

Pro koho je produkt vytvářen

- Zaměstnance a management hotelu.

Co by měl umožňovat

- Evidovat zákazníky a zaměstnance hotelu.
- Spravovat rezervaci pokojů a ubytování klientů.
- Řídit rezervace poskytovaných služeb.
- Poskytovat informace o chodu hotelu pro management.

Kategorie produktu

- Softwarový produkt

3.1.7 Identifikování uživatelé

Tato podkapitola identifikuje všechny potencionální uživatele IS hotelu.

Administrátor

Zřizuje uživatelské účty, přiděluje a edituje práva. Dohlíží na správný chod systému, stará se o údržbu aplikace a pravidelně zálohuje databázi systému, instaluje updaty. V případě potřeby má možnost procházet soubor, ve kterém jsou zaznamenány posloupnosti kroků jednotlivých uživatelů, tzv. log soubor. Pomáhá zaměstnancům s ovládáním systému.

Personál na recepci

Má na starosti ubytování klientů a komunikaci s nimi, vyřizuje rezervace, stará o pokladnu na recepci. Rezervuje pro ubytované klienty poskytování hotelových služeb. Také se stará o administrativu klientů.

Vedoucí směny

Rozděluje úkoly ostatním zaměstnancům směny, jako např. pokojským, údržbářům, personálu na recepci. Vyřizuje rozsáhlejší rezervace (kontingenty, firmy). Spravuje veškeré informace o klientech, firmách a partnerech. Stará se také o správu služeb. Hodnotí jednotlivé zaměstnance.

Manažer hotelu

Analyzuje data o chodu hotelu, stará se o přijímání zaměstnanců a vyhodnocuje efektivitu jejich práce. Zřizuje partnerské smlouvy s firemními subjekty (např. cestovní kancelář) a to ve smyslu poskytnutí kontingentu na určitý počet ubytovacích míst v hotelu v daném termínu. Manažer také zřizuje nové služby poskytované hotelem.

3.1.8 Profily uživatelů

V této podkapitole detailně rozpracuji profily uživatelů.

3.1.8.1 Administrátor systému

Popis

Speciálně školený uživatel. Kvalifikovaný administrátor systému.

Typ

Administrátor.

Odpovědnosti

Bezproblémový a bezpečný chod systému, bezchybné přidělování práv uživatelům, rychlá odezva při nahlášení poruchy a dostatečně častá záloha systému.

Kritéria úspěchu

Kompletní historie veškerých kroků uživatelů v systému, jednoduchá a přehledná editace uživatelů, snadná a rychlá záloha databáze.

3.1.8.2 Personál na recepci

Popis

Proškolený zaměstnanec hotelu, určený ke komunikaci s jednotlivými klienty.

Typ

Uživatel.

Odpovědnosti

Bezproblémové a rychlé ubytování příchozích klientů, dobrá komunikace se zákazníkem, efektivní využití ubytovacích míst hotelu, efektivní rezervace hotelových služeb a bezchybná administrativa klientů.

Kritéria úspěchu

Snadné a přehledné prohlížení a zadávání údajů o rezervacích, klientech a poskytovaných službách. Dostatečně přehledné zobrazení obsazenosti hotelu.

3.1.8.3 Vedoucí směny

Popis

Vysoce kvalifikovaný zaměstnanec hotelu, určený k řízení lidských zdrojů a ke komunikaci s prestižními klienty.

Typ

Uživatel.

Odpovědnosti

Efektivní rozdělení úkolů jednotlivým zaměstnancům směny. Poskytnutí slíbených rezervací. Kontrola rozdělení poskytovaných služeb. Dohled nad přípravou pokojů pro příchozí zákazníky a vyřízením všech aktuálních požadavků klientů.

Kritéria úspěchu

Dokonalý přehled o stavu všech pokojů a obsazenosti hotelu. Snadné a přehledné prohlížení a zadávání údajů o rezervacích, klientech a poskytovaných službách. Jednoduché přidělování úkolů jednotlivým zaměstnancům směny.

3.1.8.4 Manažer hotelu

Popis

Personální manažer hotelu.

Typ

Uživatel.

Odpovědnosti

Co nejefektivnější chod hotelu, zakládání nových a rušení starých pracovních poměrů, zřizování výhodných partnerských smluv, zřizování nových hotelových služeb dle požadavků klientů hotelu.

Kritéria úspěchu

Existence přehledných dat o chodu hotelu a výkonu zaměstnanců. Dostupná historie obsazenosti hotelu, vztahená k jednotlivým sezónám a k významným akcím konaným v místě hotelu.

3.1.9 Prostředí uživatelů

Požadavkem na uživatele systému je minimálně středoškolské vzdělání a schopnost používat aplikace v prostředí operačního systému Microsoft Windows.

Počítače hotelu jsou provozovány na operačním systému Microsoft Windows XP. Produkt je sice vytvářen na zakázku pro konkrétní firmu provozující hotel, avšak bude navržen univerzálně, aby jej bylo možné nasadit i v jiných hotelech, ve kterých disponují operačním systémem Microsoft Windows XP.

3.1.10 Perspektiva produktu

Produkt bude nasazen do prostředí, ve kterém doposud nebyl žádný informační systém podporující chod hotelu.

Informační systém hotelu se bude skládat ze dvou částí, klientská aplikace a www server. Jako www server bude použit systém Apache. Databáze pro potřeby IS hotelu bude postavena na platformě

relační databáze MySQL a samotná aplikace bude implementována pomocí skriptovacího jazyka PHP. Samotnou klientskou aplikaci bude možno spustit na kterémkoli počítači v hotelu, který bude vybaven internetovým prohlížečem Internet Explorer (minimálně verze 6.0), Mozilla Firefox nebo Opera (minimálně verze 7). Žádný jiný software není potřeba pro chod systému instalovat. Do systému nebude možno přistupovat ze sítě internet, pouze z počítačů připojených k lokální síti.

3.1.11 Základní funkce systému

Toto jsou základní funkce systému:

- Zadávání, aktualizace a prohlížení klientů a partnerů.
- Rezervování pokojů.
- Vytváření a rezervování služeb.
- Vytváření a přidělování úkolů.

3.1.12 Předpoklady a závislosti

Ovládání systému a design aplikace musí být přizpůsoben pro práci v internetovém prohlížeči. Uživatelské rozhraní by nemělo nijak vybočovat ze zaběhnutých standardů v aplikacích používaných v internetových prohlížečích.

3.1.13 Náklady a cenová kalkulace

Finanční rozpočet pro vytvoření systému byl stanoven na 40 000 Kč. Tato cena se týká pouze softwaru a nezahrnuje náklady spojené na technické vybavení.

3.1.14 Licencování

Použité technologie pro implementaci systému nevyžadují žádné licence.

3.1.15 Vhodné standardy

Pro uživatelské rozhraní, tvořené pomocí XHTML a CSS jsou definované konsorciem W3C obecné standardy. Vytvářený systém by se jich měl držet.

3.1.16 Předběžné požadavky na výkon

Systém by neměl být z hlediska současného přístupu nijak zvlášť vytížen. Je to dáno místem nasazení. V hotelu bude do systému přistupovat maximálně několik uživatelů současně. Tím pádem není potřeba vybavit prostředí pro nasazení systému výkonnými počítači, ale stačí pouze standardní kancelářské přístroje.

3.1.17 Požadavky na prostředí

Server provozující databázi informačního systému hotelu by měl být umístěn v oddělené místnosti nejlépe s nízkou prašností a vlhkostí a bez veřejného přístupu.

3.1.18 Instalace

Pro klientské počítače není potřeba instalovat žádnou speciální aplikaci. Postačí pouze prohlížeč webových stránek.

3.2 Glossář pojmů

V inepční fázi je nejdůležitější vzájemná komunikace mezi zadavatelem zakázky a realizačním týmem projektu. Během této fáze se mohou vyskytnout pojmy, kterým by zadavatel nemusel rozumět. Proto se v Dokumentu vize uvádí glosář pojmů (Tab. 3).

| Glosář pojmů | |
|---------------|--|
| Pojem | Popis |
| Use-case | Jeden případ užití, konkrétní akce v systému. |
| Role (aktér) | Objekt, který provádí akci (případ užití) v modelu případů užití. |
| Recepční | Zkrácený název role Personál na recepci. |
| Vedoucí | Zkrácený název role Vedoucí směny. |
| Manažer | Zkrácený název role Manažer hotelu. |
| Administrátor | Správce systému a speciální uživatel, který má možnost vytvářet ostatní uživatele systému |
| Partner | Firma, která má uzavřenou smlouvu s hotelem v rámci tzv. kontingentu. Například cestovní kancelář. |
| Kontingent | Rezervaci určitého počtu pokojů vztažené ke konkrétnímu termínu (datum začátku a datum konce). |
| Sezóna | Určitá část roku, významná pro měřítko obsazenosti hotelu. Například letní sezóna. |
| Login | Přihlašovací jméno uživatele systému. |
| Zaměstnanec | Totéž co v databázi uživatelský účet. Je to tentýž objekt. |
| Hodnocení | Zkrácený název hodnocení zaměstnance. |

Tab. 3: Glosář pojmů

3.3 Use-Case model

Tato kapitola specifikuje pohled na systém z hlediska jeho funkcionality. Obsahuje případy užití vztažené k jednotlivým aktérům.

Podkapitola 3.3.1 zobrazuje pohled na celý systém a kapitoly 3.3.2 – 3.3.11 popisují podrobně konkrétní případy užití.

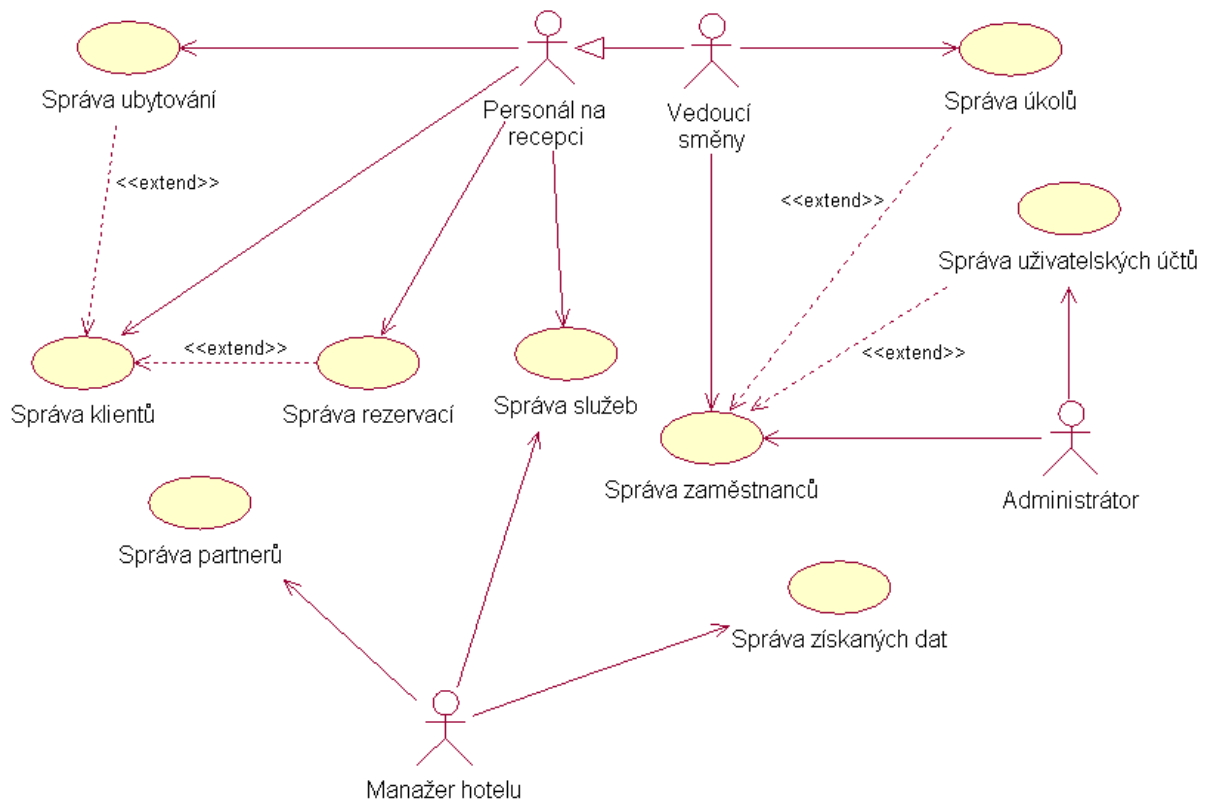
3.3.1 Pohled na celý systém (hlavní případy užití)

Seznam případů užití

- Správa ubytování.
- Správa rezervací.
- Správa klientů.
- Správa služeb.
- Správa zaměstnanců
- Správa získaných dat.
- Správa partnerů
- Správa úkolů
- Správa uživatelských účtů

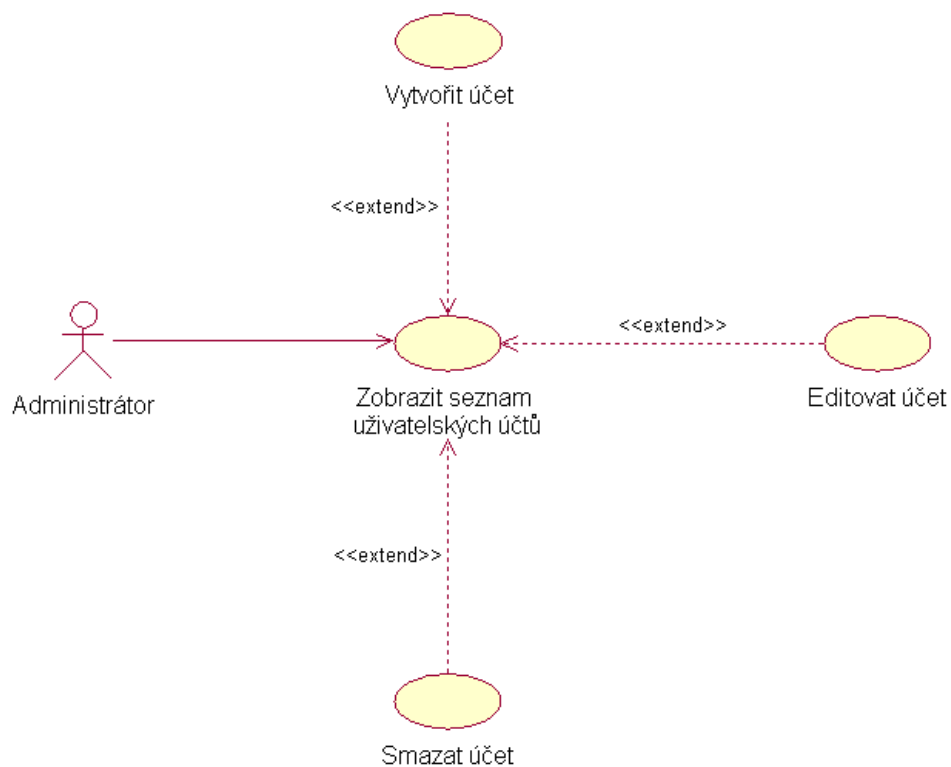
Seznam aktérů

- Vedoucí směny
- Personál na recepci
- Manažer hotelu
- Administrátor



Obr. 10: Diagram případů užití UC01 – Hlavní případy užití

3.3.2 Správa zaměstnanců z pohledu Administrátora



Obr. 11: Diagram případů užití UC02 – Správa zaměstnanců z pohledu Administrátora

3.3.2.1 Stručný popis

Diagram případů užití na obr. 11 umožňuje Administrátorovy spravovat zaměstnanecké účty. Systém umožní zobrazení seznamu uživatelských účtů, vytváření nových, editování stávajících a mazání účtů.

3.3.2.2 Příklad užití: Zobrazit seznam uživatelských účtů

ID

UC02.1

Účastníci

Administrátor

Vstupní podmínky

1. Administrátor je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „seznam účtů“.
2. Systém zobrazí seznam uživatelských účtů.

Alternativní tok 1

1. Administrátor může kdykoli opustit systém.

Následné podmínky AT 1

1. Administrátor se musí odhlásit ze systému.

3.3.2.3 Příklad užití: Vytvořit účet

ID

UC02.2

Účastníci

Administrátor

Vstupní podmínky

1. Administrátor je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „vytvoř nový“.
2. Systém zobrazí prázdný formulář vytvoření nového uživatelského účtu.
3. Administrátor zadá tyto informace: jméno, příjmení, titul, adresa bydliště, rodné číslo, telefonní číslo na mobil, emailová adresa, název souboru s fotografií a životopisem, login, heslo, typ oprávnění.
4. Administrátor zvolí „odeslat“.
5. Systém zkontroluje, zda jsou zadána všechna data ve formuláři a ve správném formátu.
6. Systém zkontroluje, zda již není účet v databázi (dle rodného čísla).
7. KDYŽ účet není v databázi:

- 7.1 Vytvoří se nová položka účet (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Administrátor může kdykoli opustit systém.

Následné podmínky AT 1

1. Administrátor se musí odhlásit ze systému.

Alternativní tok 2

1. Nejsou zadána všechna požadovaná data nebo nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. Rodné číslo již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Systém doplní do formuláře doposud zadané informace.

3.3.2.4 Příklad užití: Editovat účet

ID

UC02.3

Účastníci

Administrátor

Vstupní podmínky

1. Administrátor je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „editovat“ na řádku konkrétního účtu ve výpisu seznamu všech účtů.
2. Systém zobrazí formulář, vyplněný stávajícími informacemi o účtu.
3. Administrátor změní požadované údaje a zadá volbu „odeslat“.
4. Systém zkontroluje, zda jsou zadána všechna data ve formuláři a ve správném formátu.
5. KDYŽ administrátor změní rodné číslo:
 - 5.1 Systém zkontroluje, zda již toto rodné číslo není v databázi.
6. Systém uloží do databáze změněné údaje o účtu.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Administrátor může kdykoli opustit systém.

Následné podmínky AT 1

1. Administrátor se musí odhlásit ze systému.

Alternativní tok 2

1. Nejsou zadána všechna požadovaná data nebo nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. Rodné číslo již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Systém doplní do formuláře doposud zadané informace.

3.3.2.5 Příklad užití: Smazat účet

ID

UC02.4

Účastníci

Administrátor

Vstupní podmínky

1. Administrátor je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „smazat“ na řádku konkrétního účtu ve výpisu seznamu všech účtů.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétního účtu.
3. KDYŽ administrátor potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku účtu z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Administrátor může kdykoli opustit systém.

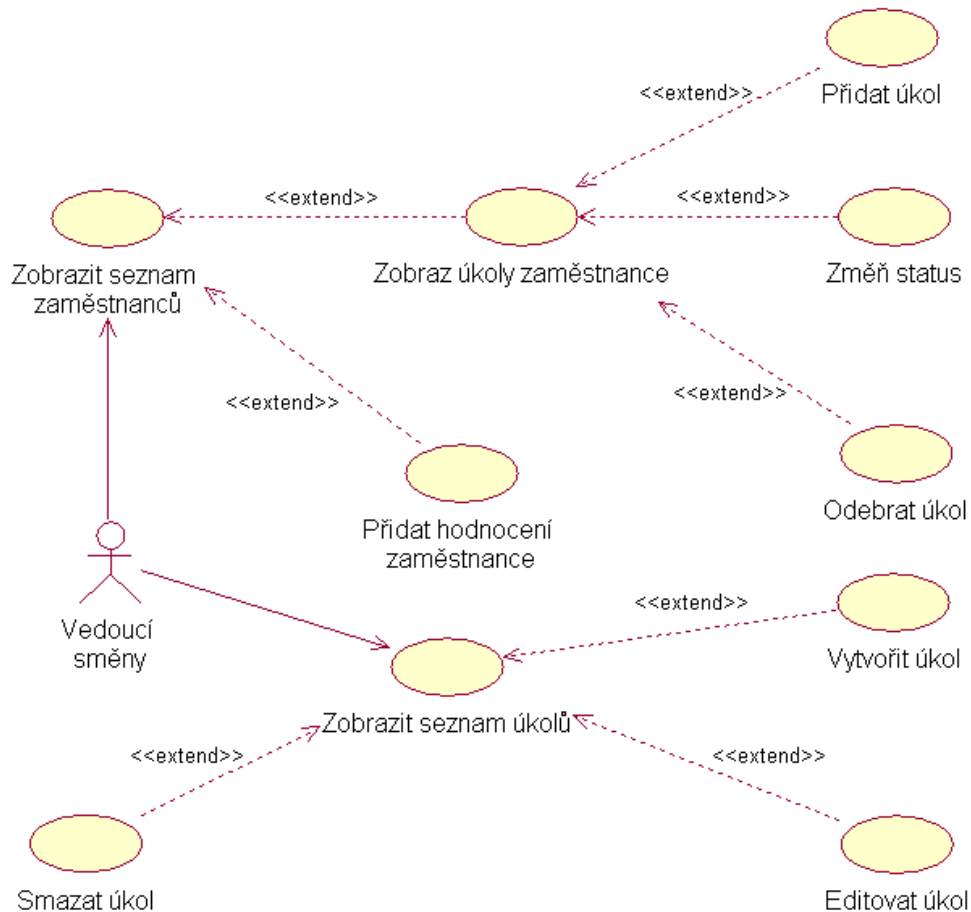
Následné podmínky AT 1

1. Administrátor se musí odhlásit ze systému.

Alternativní tok 2

1. Administrátor nepotvrdí v dialogovém okně smazání konkrétního účtu.
2. Příklad užití končí.

3.3.3 Správa zaměstnanců z pohledu Vedoucího směny



Obr. 12: Diagram případů užití UC03 – Správa úkolů z pohledu Vedoucího směny

3.3.3.1 Stručný popis

Diagram případů užití na obr. 12 ukazuje správu nad úkoly z pohledu Vedoucího směny. Systém mu umožní zobrazení seznamu úkolů, vytváření nových úkolů, editování stávajících úkolů a mazání úkolů. Dále umožní zobrazit seznam zaměstnanců, ve kterém systém poskytne možnost přidat či odebrat každému zaměstnanci požadovaný úkol.

3.3.3.2 Příklad užití: Zobrazit seznam úkolů

ID

UC03.1

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „seznam úkolů“.
2. Systém zobrazí seznam úkolů, včetně všech informací.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

3.3.3.3 Příklad užití: Vytvořit úkol

ID

UC03.2

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „vytvoř nový“.
2. Systém zobrazí prázdný formulář vytvoření nového úkolu.
3. Vedoucí směny zadá tyto informace: název úkolu, popis, čas splnění.
4. Vedoucí směny zadá volbu „odeslat“.
5. Systém zkontroluje, zda jsou zadaná všechna data ve formuláři.
6. Systém zkontroluje, zda již není úkol téhož názvu v databázi.
7. KDYŽ není úkol ještě v databázi:
 - 7.1 Vytvoří se nová položka úkol (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

Alternativní tok 2

1. Nejsou zadaná všechna data, požadovaná ve formuláři.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. Název úkolu již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Systém doplní do formuláře doposud zadané informace.

3.3.3.4 Případ užití: Editovat úkol

ID

UC03.3

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „editovat“ na řádku konkrétního úkolu ve výpisu seznamu všech úkolů.
2. Systém zobrazí formulář, vyplněný stávajícími informacemi o úkolu.
3. Vedoucí směny změní požadované údaje a zadá volbu „odeslat“.
4. KDYŽ vedoucí směny změní název úkolu:
 - 4.1 Proběhne kontrola, zda již není úkol téhož názvu v databázi.
5. Systém uloží do databáze změněné údaje o účtu.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

Alternativní tok 2

1. Nejsou zadaná všechna data, požadovaná ve formuláři.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. Úkol téhož názvu již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Systém doplní do formuláře doposud zadané informace.

3.3.3.5 Případ užití: Smazat úkol

ID

UC03.4

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „smazat“ na řádku konkrétního úkolu ve výpisu seznamu všech úkolů.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétního úkolu.
3. KDYŽ vedoucí směny potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku úkolu z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

Alternativní tok 2

1. Vedoucí směny nepotvrdí v dialogovém okně smazání konkrétního úkolu.
2. Případ užití končí.

3.3.3.6 Případ užití: Zobraz seznam zaměstnanců

ID

UC03.5

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „seznam zaměstnanců“.
2. Systém zobrazí seznam všech zaměstnanců, včetně informací o nich.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

3.3.3.7 Příklad užití: Zobraz úkoly zaměstnance

ID

UC03.6

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „úkoly“ na řádku konkrétního zaměstnance ve výpisu seznamu všech zaměstnanců.
2. Systém zobrazí seznam všech úkolů, přiřazených ke konkrétnímu zaměstnanci.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

3.3.3.8 Příklad užití: Přidat úkol

ID

UC03.7

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou konkrétního úkolu z roletové nabídky
2. Stiskem volby „přidat úkol“ potvrdí přiřazení úkolu ke konkrétnímu zaměstnanci.
3. Systém zobrazí dialogové okno s potvrzením.
4. KDYŽ vedoucí směny potvrdí kladně nabídku v dialogovém okně:
 - 4.1 Systém uloží data do databáze.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

Alternativní tok 2

1. Vedoucí směny nepotvrdí nabídku v dialogovém kladně.
2. Příklad užití končí.

3.3.3.9 Příklad užití: Odebrat úkol

ID

UC03.8

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „odebrat“ na řádku konkrétního úkolu ve výpisu seznamu všech úkolů zaměstnance.
2. Systém zobrazí dialogové okno s potvrzením.
3. KDYŽ vedoucí směny potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém odebere položku konkrétního úkolu ze seznamu úkolů zaměstnance z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

Alternativní tok 2

1. Vedoucí směny nepotvrdí nabídku v dialogovém kladně.
2. Příklad užití končí.

3.3.3.10 Příklad užití: Změň status

ID

UC03.9

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „změň status“ na řádku konkrétního úkolu ve výpisu seznamu všech úkolů zaměstnance.
2. Systém zobrazí dialogové okno s potvrzením.
3. KDYŽ vedoucí směny potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém provede změny v databázi.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

Alternativní tok 2

1. Vedoucí směny nepotvrdí nabídku v dialogovém kladně.
2. Příklad užití končí.

3.3.3.11 Příklad užití: Přidat hodnocení zaměstnance

ID

UC03.10

Účastníci

Vedoucí směny

Vstupní podmínky

1. Vedoucí směny je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „přidat hodnocení“.
2. Systém zobrazí prázdný formulář pro přidání hodnocení zaměstnance.
3. Vedoucí směny zadá hodnocení (textový popis) do formuláře.
4. Vedoucí směny zadá volbu „odeslat“.
5. Systém uloží do databáze hodnocení ke konkrétnímu zaměstnanci.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

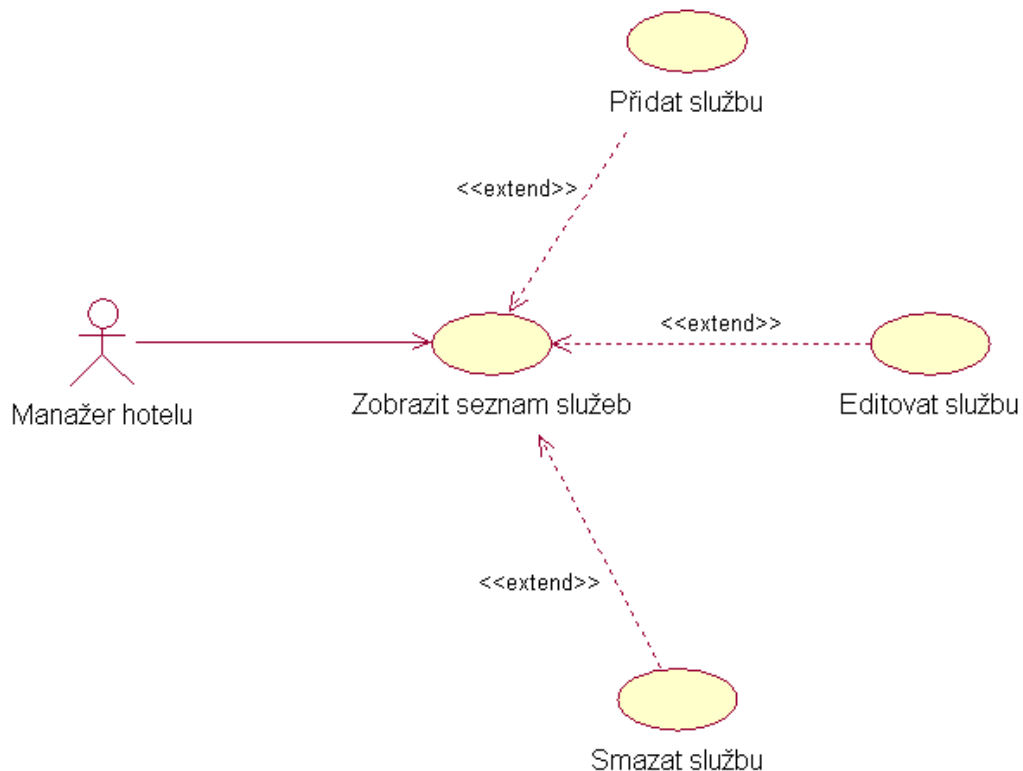
Alternativní tok 1

1. Vedoucí směny může kdykoli opustit systém.

Následné podmínky AT 1

1. Vedoucí směny se musí odhlásit ze systému.

3.3.4 Správa služeb z pohledu Manažera hotelu



Obr. 13: Diagram případů užití UC04 – Správa služeb z pohledu Manažera hotelu

3.3.4.1 Stručný popis

Na obr. 13 je diagram případů užití, který umožňuje Manažerovi hotelu spravovat služby. To zahrnuje zobrazení seznamu služeb, vytváření nových služeb, editování služeb a mazání služeb. Hotelové služby jsou například kadeřnictví, masáže, kosmetické programy apod., poskytované klientům hotelu.

3.3.4.2 Příklad užití: Zobrazit seznam služeb

ID

UC04.1

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer hotelu je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „zobrazit seznam služeb“.
2. Systém zobrazí seznam všech služeb, včetně informací o nich.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

3.3.4.3 Příklad užití: Přidat službu

ID

UC04.2

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer hotelu je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „přidat službu“.
2. Systém zobrazí prázdný formulář pro přidání služby.
3. Manažer zadá následující informace o službě: název, popis, cena.
4. Manažer zadá volbu „odeslat“.
5. Systém zkontroluje data vložená manažerem do formuláře.
6. Proběhne kontrola, zda již není služba v databázi (dle názvu).
7. KDYŽ není služba v databázi
 - 7.1 Vytvoří se nová položka služby (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře pro přidání služby doposud zadané informace.

Alternativní tok 3

1. Služba se zadaným názvem již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře pro přidání služby se vyplní doposud zadané informace.

3.3.4.4 Případ užití: Editovat službu

ID

UC04.3

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer hotelu je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „editovat“ na řádku konkrétní služby ve výpisu seznamu všech služeb.
2. Systém zobrazí formulář vyplněný údaji o zvolené službě.
3. Manažer změní požadované údaje a zadá volbu „odeslat“.
4. Systém zkontroluje data vložená manažerem do formuláře.
5. KDYŽ manažer změní název služby:
 - 5.1 Proběhne kontrola, zda již není služba v databázi (dle názvu).
6. Systém uloží do databáze změněné údaje o službě.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Manažer hotelu může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer hotelu se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře pro vytvoření služby doposud zadané informace.

Alternativní tok 3

1. Služba se zadaným názvem již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře pro vytvoření služby se vyplní doposud zadané informace.

3.3.4.5 Případ užití: Smazat službu

ID

UC04.4

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer hotelu je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „smazat“ na řádku konkrétní služby ve výpisu seznamu všech služeb.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétní služby.
3. KDYŽ manažer potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku služby z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

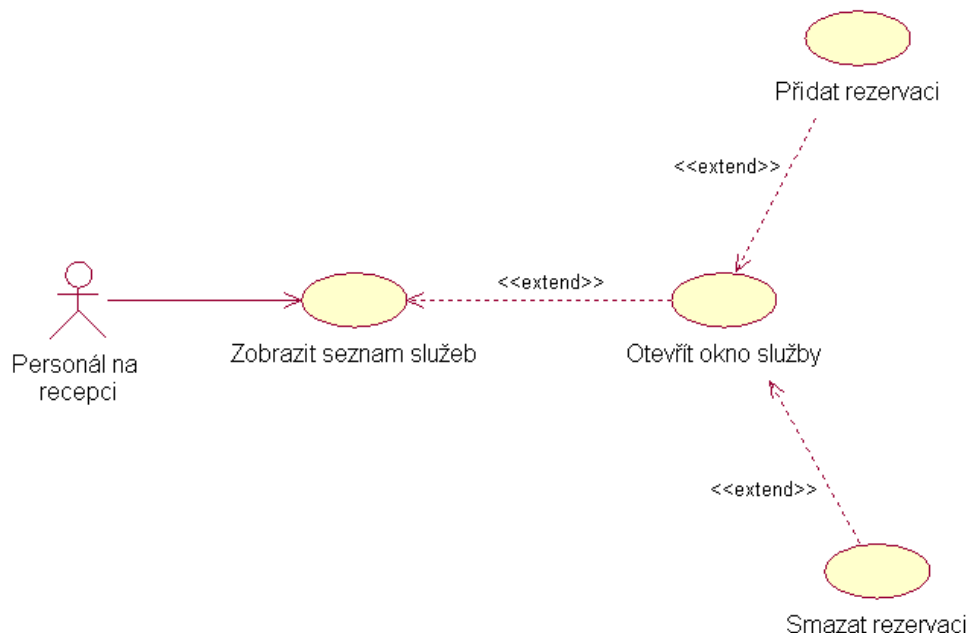
Alternativní tok 1

1. Manažer hotelu může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer hotelu se musí odhlásit ze systému.

3.3.5 Správa služeb z pohledu Personálu na recepci



Obr. 14: Diagram případů užití UC05 – Správa služeb z pohledu Personálu na recepci

3.3.5.1 Stručný popis

Na obr. 14 je diagram případů užití, který umožňuje Personálu na recepci spravovat služby. To zahrnuje zobrazení seznamu služeb a detailu každé z nich. Personál na recepci rezervuje zákazníky

hotelu na konkrétní hodinu dané služby. Hotelové služby jsou například kadeřnictví, masáže, kosmetické programy apod., poskytované klientům hotelu.

3.3.5.2 Případ užití: Zobrazit seznam služeb

ID

UC05.1

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „seznam služeb“.
2. Systém zobrazí seznam všech služeb, včetně informací o nich.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

3.3.5.3 Případ užití: Otevřít okno služby

ID

UC05.2

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „detail“ na řádku konkrétní služby ve výpisu seznamu všech služeb.
2. Systém zobrazí detailní informace konkrétní služby a seznam rezervací na aktuální týden.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

3.3.5.4 Případ užití: Přidat rezervaci

ID

UC05.3

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „rezervovat“.
2. Systém zobrazí prázdný formulář rezervace služby.
3. Recepční vyplní datum a čas rezervace (v roletových nabídkách) a zvolí zákazníka hotelu.
4. Recepční zadá volbu „odeslat“.
5. Vytvoří se nová položka v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Pro rezervaci služby nemusí být žádný volný termín.
2. Systém vypíše hlášení o této situaci na obrazovku.
3. Příklad užití končí.

3.3.5.5 Příklad užití: Smazat rezervaci

ID

UC05.3

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „smazat“ na řádku konkrétní rezervace ve výpisu seznamu rezervací konkrétní služby.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétní rezervace.
3. KDYŽ recepční potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku rezervace z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

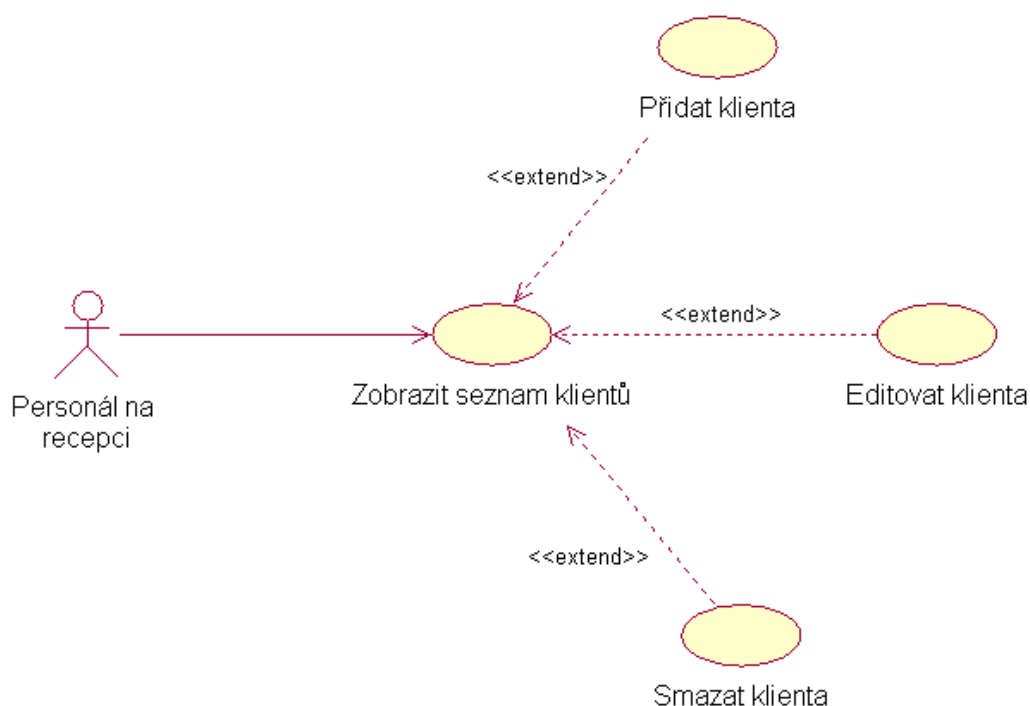
Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční nepotvrdí v dialogovém okně smazání rezervace.
2. Příklad užití končí.

3.3.6 Správa klientů



Obr. 15: Diagram případů užití UC06 – Správa klientů

3.3.6.1 Stručný popis

Na obr. 15 je diagram případů užití, který umožňuje Personálu na recepci spravovat jednotlivé klienty. To zahrnuje zobrazení seznamu klientů, přidávání klientů, editování klientů, mazání klientů. Spolu s přidáváním klientů do databáze se ukládají také všechny potřebné informace o nich.

3.3.6.2 Příklad užití: Zobrazit seznam klientů

ID

UC06.1

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „seznam klientů“.
2. Systém zobrazí seznam všech klientů.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

3.3.6.3 Příklad užití: Přidat klienta

ID

UC06.2

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „přidat klienta“.
2. Systém zobrazí prázdný formulář pro přidání klienta.
3. Recepční zadá tyto informace o klientovi: jméno, příjmení, titul, adresa bydliště, telefonní číslo na mobil, emailová adresa, národnost, číslo OP (nebo číslo pasu), jazyk.
4. Recepční zadá volbu „odeslat“.
5. Systém zkontroluje data vložená do formuláře.
6. Proběhne kontrola, zda konkrétní klient již není v databázi (dle čísla OP).
7. **KDYŽ** klient není v databázi:
 - 7.1 Vytvoří se nová položka klient (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.

2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře pro přidání klienta doposud zadané informace.

Alternativní tok 3

1. Číslo OP klienta již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře pro přidání klienta se vyplní doposud zadané informace.

3.3.6.4 Případ užití: Editovat klienta

ID

UC06.3

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „editovat“ na řádku konkrétního klienta ve výpisu seznamu všech klientů.
2. Systém zobrazí formulář, vyplněný údaji o klientovi.
3. Recepční změní požadované údaje a zadá volbu „odeslat“.
4. Systém zkontroluje data vložená do formuláře.
5. KDYŽ recepční změní číslo OP klienta
 - 5.1 Proběhne kontrola, zda klient již není v databázi (dle čísla OP).
6. Systém uloží do databáze změněné údaje o klientovi.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře pro editaci klienta doposud zadané informace.

Alternativní tok 3

1. Číslo OP klienta již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře pro editaci klienta se vyplní doposud zadané informace.

3.3.6.5 Příklad užití: Smazat klienta

ID

UC06.4

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „smazat“ na řádku konkrétního klienta ve výpisu seznamu všech klientů.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétního klienta.
3. KDYŽ recepční potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku klienta z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

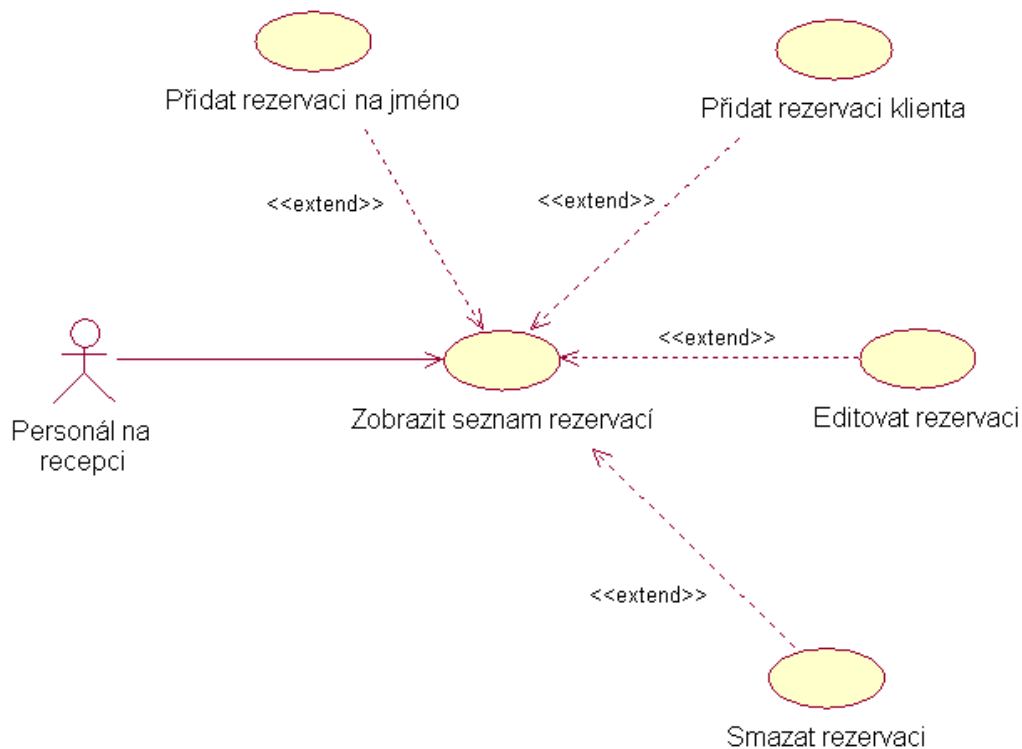
Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční nepotvrdí v dialogovém okně smazání klienta.
2. Příklad užití končí.

3.3.7 Správa rezervací



Obr. 16: Diagram případů užití UC07 – Správa rezervací

3.3.7.1 Stručný popis

Na obr. 16 je diagram případů užití, který umožňuje Personálu na recepci spravovat. To zahrnuje zobrazení seznamu rezervací, přidávání rezervací na jméno, přidávání rezervací pro klienta, editování rezervací, mazání rezervací. O rezervacích se do databáze ukládají také všechny potřebné informace.

3.3.7.2 Příklad užití: Zobrazit seznam rezervací

ID

UC07.1

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „seznam rezervací“.
2. Systém zobrazí seznam všech rezervací v aktuálním týdnu.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční zvolí v roletovém menu požadovaný týden pro zobrazení.
2. Recepční potvrdí výběr volbou „zobrazit“.
3. Systém zobrazí seznam všech rezervací ve vybraném týdnu.

3.3.7.3 Příklad užití: Přidat rezervaci na jméno

ID

UC07.2

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „rezervace na jméno“.
2. Systém zobrazí prázdný formulář pro přidání rezervace.
3. Recepční zadá tyto informace do formuláře: jméno a příjmení rezervujícího, titul, telefonní číslo na mobil, emailová adresa, datum příjezdu, datum odjezdu, počet osob, počet pokojů.
4. Recepční zadá volbu „odeslat“.
5. Systém zkontroluje data vložená do formuláře.
6. Systém zkontroluje, zda je v požadovaném termínu dostatek volných pokojů v hotelu.
7. KDYŽ je v požadovaném termínu dostatek volných pokojů:
 - 7.1 Systém vyčlení volné pokoje pro tuto rezervaci a označí je jako rezervované v daném termínu.
 - 7.2 Vytvoří se nová položka rezervace (s jedinečným ID) v databázi a naplní se údaji z formuláře a čísly přiřazených pokojů.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. V požadovaném termínu není dostatek volných pokojů v hotelu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře se vyplní doposud zadané informace.

3.3.7.4 Příklad užití: Přidat rezervaci klienta

ID

UC07.3

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „rezervace klienta“.
2. Systém zobrazí formulář pro přidání rezervace, ve kterém budou vyplněny informace o klientovi potřebné k rezervaci (viz případ užití UC08.2).
3. Recepční zadá tyto informace do formuláře: datum příjezdu, datum odjezdu, počet pokojů, počet osob.
4. Recepční zadá volbu „odeslat“.
5. Systém zkontroluje data vložená do formuláře.
6. Systém zkontroluje, zda je v požadovaném termínu dostatek volných pokojů v hotelu.
7. KDYŽ je v požadovaném termínu dostatek volných pokojů:
 - 7.1 Vytvoří se nová položka rezervace (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.

2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. V požadovaném termínu není dostatek volných pokojů v hotelu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře se vyplní doposud zadané informace.

3.3.7.5 Případ užití: Editovat rezervaci

ID

UC07.4

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „editovat“ na řádku konkrétní rezervace ve výpisu seznamu všech rezervací.
2. Systém zobrazí formulář, vyplněný informacemi o rezervaci.
3. Recepční změní požadované údaje a zadá volbu „odeslat“.
4. Systém zkontroluje data vložená do formuláře.
5. KDYŽ recepční změní termín rezervace:
 - 5.1 Proběhne kontrola, zda je v požadovaném termínu dostatek volných pokojů v hotelu.
6. Systém uloží do databáze změněné údaje o rezervaci.

Následné podmínky

1. Proběhne úspěšné uložení záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. V požadovaném termínu není dostatek volných pokojů v hotelu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Systém doplní do formuláře doposud zadané informace.

3.3.7.6 Případ užití: Smazat rezervaci

ID

UC07.5

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „smazat“ na řádku konkrétní rezervace ve výpisu seznamu všech rezervací.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétní rezervace.
3. KDYŽ recepční potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku rezervace z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

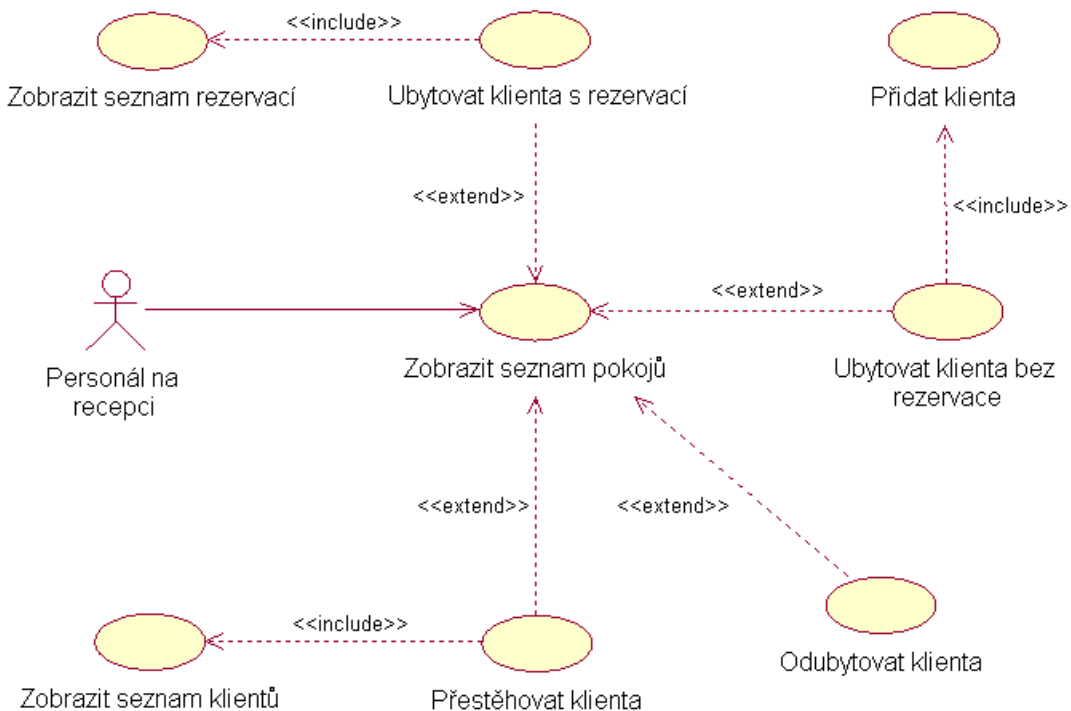
Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční nepotvrdí v dialogovém okně smazání klienta.
2. Případ užití končí.

3.3.8 Správa ubytování



Obr. 17: Diagram případů užití UC08 – Správa ubytování

3.3.8.1 Stručný popis

Na obr. 17 je diagram případů užití, který umožňuje Personálu na recepci spravovat ubytování. To zahrnuje zobrazení seznamu pokojů, ubytování klienta s rezervací, odubytování klienta, přestěhování klienta. Systém umožňuje také ubytování klienta bez rezervace (tzv. WALK IN) a to v případě, že jsou k dispozici volné pokoje bez rezervace.

3.3.8.2 Případ užití: Zobrazit seznam pokojů

ID

UC08.1

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „seznam pokojů“.
2. Systém zobrazí seznam všech pokojů a jejich obsazenost v aktuálním týdnu.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční zvolí v roletovém menu požadovaný týden pro zobrazení.
2. Recepční potvrdí výběr volbou „zobrazit“.
3. Systém zobrazí seznam všech pokojů a jejich obsazenost ve vybraném týdnu.

3.3.8.3 Případ užití: Ubytovat klienta s rezervací

ID

UC08.2

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „ubytovat rezervované“.
2. Zahrnout (Zobrazit seznam rezervací).
3. Recepční zvolí „ubytovat“ na řádku konkrétní rezervace daného klienta.
4. Systém zobrazí dialogové okno s potvrzením ubytování klienta na rezervovaný pokoj.
5. KDYŽ recepční potvrdí kladně nabídku v dialogovém okně:
 - 5.1 Vytvoří se nová položka ubytování (s jedinečným ID) v databázi (spolu s číslem pokoje a id klienta a datem odjezdu).

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční nepotvrdí v dialogovém okně ubytování klienta.
2. Případ užití končí.

3.3.8.4 Případ užití: Ubytovat klienta bez rezervace

ID

UC08.3

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Recepční v seznamu pokojů ověří, zda je dle požadavků klienta dostatek volných míst v hotelu.
2. KDYŽ je v hotelu dostatek volných míst:
 - 2.1 Recepční zvolí „ubytovat příchozí“.
 - 2.2 Zahrnout (Přidat klienta).
 - 2.3 Systém zobrazí prázdný formulář pro zřízení ubytování.
 - 2.4 Recepční zadá tyto informace do formuláře: číslo pokoje, id klienta, datum odjezdu.
 - 2.5 Recepční zadá volbu „odeslat“.
 - 2.6 Systém zkontroluje data vložená do formuláře.
 - 2.7 Vytvoří se nová položka ubytování (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.3.

Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. V požadovaném termínu není dostatek volných pokojů v hotelu.
2. Recepční to oznámí klientovi a nabídne mu jiný termín.

3.3.8.5 Příklad užití: Odubytovat klienta

ID

UC08.4

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „odubytovat“ na řádku konkrétní pokoje ve výpisu seznamu všech pokojů v aktuálním týdnu.
2. Systém zobrazí dialogové okno s potvrzením odubytování klienta z konkrétního pokoje.
3. KDYŽ recepční potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku ubytování z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT 1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Recepční nepotvrdí v dialogovém okně odubytování klienta.
2. Příklad užití končí.

3.3.8.6 Příklad užití: Přestěhovat klienta

ID

UC08.5

Účastníci

Personál na recepci (recepční)

Vstupní podmínky

1. Recepční je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „přestěhovat klienta“.
2. Zahrnout (Zobrazit seznam klientů).
3. Recepční zvolí „přestěhovat“ na řádku konkrétního klienta ve výpisu seznamu všech ubytovaných klientů.
4. Systém zobrazí formulář, vyplněný údaji o ubytování klienta.
5. Recepční změní ve formuláři číslo stávajícího pokoje na číslo nového pokoje, určeného k přestěhování.
6. Recepční zadá volbu „odeslat“.
7. Systém zkontroluje data vložená do formuláře.

8. Systém zkontroluje, zda je pokoj určený k přestěhování volný.
9. KDYŽ je pokoj volný:
 - 9.1 Systém uloží do databáze změněné údaje o ubytování.

Následné podmínky

1. Proběhne úspěšná změna záznamu v databázi.

Alternativní tok 1

1. Recepční může kdykoli opustit systém.

Následné podmínky AT1

1. Recepční se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 4.

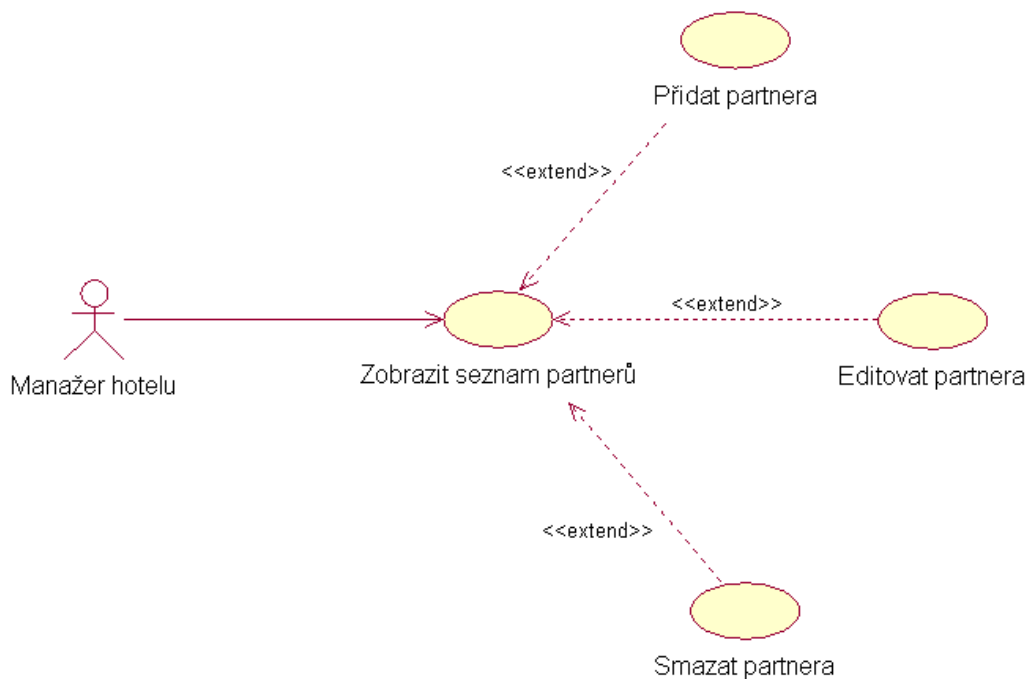
Následné podmínky AT 2

1. Systém doplní do formuláře doposud zadané informace.

Alternativní tok 3

1. Pokoj určený k přestěhování není volný.
2. Vypíše se na obrazovku chybové hlášení.
3. Recepční se pokusí najít jiný pokoj, tok se vrátí k bodu 4.

3.3.9 Správa partnerů



Obr. 18: Diagram případů užití UC09 – Správa partnerů

3.3.9.1 Stručný popis

Na obr. 18 je diagram případů užití, který umožňuje Manažerovy hotelu správu nad partnery. Ta zahrnuje zobrazení seznamu partnerů, přidávání partnerů, editování partnerů a mazání partnerů. Partnerem se myslí firma, která má s hotelem uzavřenou smlouvu typu kontingent (např. cestovní kancelář).

3.3.9.2 Příklad užití: Zobrazit seznam partnerů

ID

UC09.1

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „seznam partnerů“.
2. Systém zobrazí seznam všech partnerů.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

3.3.9.3 Příklad užití: Přidat partnera

ID

UC09.2

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „přidat partnera“.
2. Systém zobrazí prázdný formulář pro přidání partnera.
3. Manažer zadá tyto informace o partnerovi: název subjektu, adresa sídla, telefonní číslo, emailová adresa, datum začátku termínu, datum konce termínu, počet míst.
4. Manažer zvolí „odeslat“.
5. Systém zkontroluje data vložená do formuláře.

6. Proběhne kontrola, zda konkrétní partner již není v databázi (dle názvu subjektu a adresy sídla).
7. KDYŽ partner není v databázi:
 - 7.1 Vytvoří se nová položka partner (s jedinečným ID) v databázi a naplní se údaji z formuláře.

Následné podmínky

1. Proběhne úspěšné uložení nového záznamu do databáze.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře pro přidání partnera doposud zadané informace.

Alternativní tok 3

1. Partner již existuje v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře pro přidání partnera se vyplní doposud zadané informace.

3.3.9.4 Případ užití: Editovat partnera

ID

UC09.3

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „editovat“ na řádku konkrétního partnera ve výpisu seznamu všech partnerů.
2. Systém zobrazí formulář, vyplněný údaji o partnerovi.
3. Manažer změní požadované údaje a zadá volbu „odeslat“.
4. Systém zkontroluje data vložená do formuláře.
5. KDYŽ manažer změní název subjektu:

5.1 Proběhne kontrola, zda konkrétní partner již není v databázi (dle názvu subjektu a adresy sídla).

6. Systém uloží do databáze změněné údaje o partnerovi.

Následné podmínky

1. Proběhne úspěšná změna záznamu v databázi.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

Alternativní tok 2

1. Data nejsou ve správném formátu.

2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 2

1. Systém doplní do formuláře pro editaci partnera doposud zadané informace.

Alternativní tok 3

1. Partner již existuje v databázi.

2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

Následné podmínky AT 3

1. Do formuláře pro editaci partnera se vyplní doposud zadané informace.

3.3.9.5 Příklad užití: Smazat partnera

ID

UC09.4

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „smazat“ na řádku konkrétního partnera ve výpisu seznamu všech partnerů.

2. Systém zobrazí dialogové okno s potvrzením smazání konkrétního partnera.

3. KDYŽ manažer potvrdí kladně nabídku v dialogovém okně:

3.1 Systém smaže konkrétní položku partnera z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu z databáze.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

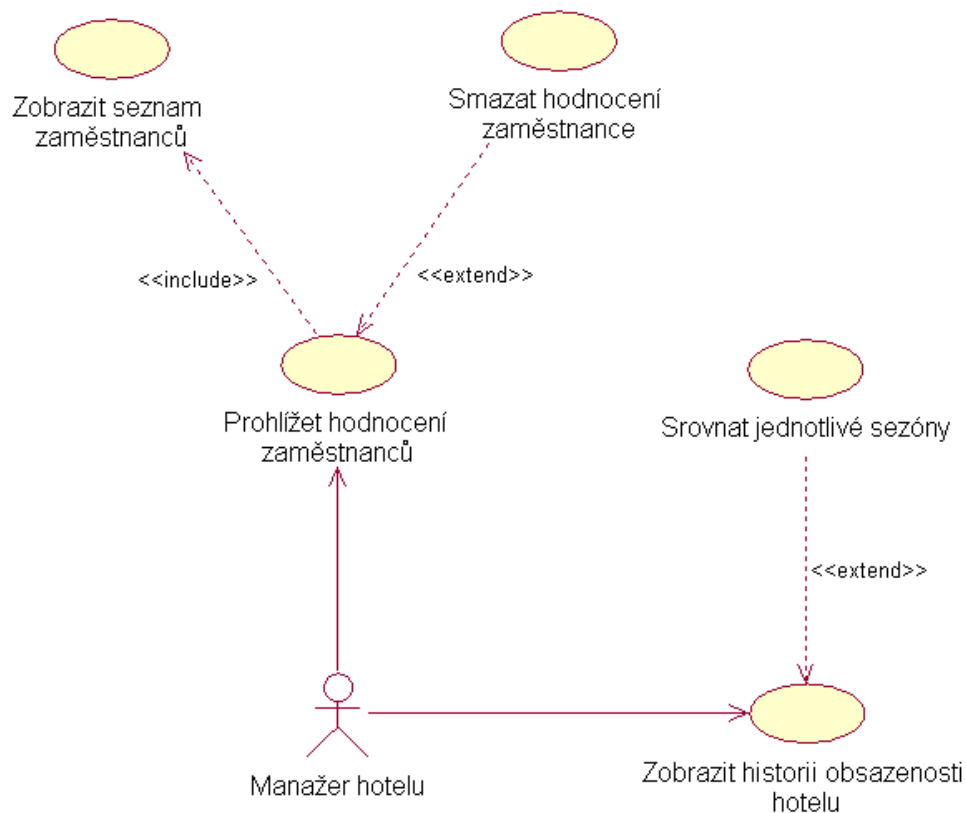
Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

Alternativní tok 2

1. Manažer nepotvrdí v dialogovém okně smazání partnera.
2. Příklad užití končí.

3.3.10 Správa získaných dat



Obr. 19: Diagram případů užití UC10 – Správa získaných dat

3.3.10.1 Stručný popis

Na obr. 19 je diagram případů užití, který umožňuje Manažerovy hotelu spravovat získaná data o obsazenosti hotelu a prohlížet a mazat hodnocení zaměstnanců. Hodnocení zaměstnanců zadává do databáze Vedoucí směny. Manažer má možnost zobrazit a porovnat obsazenost hotelu v rámci jednotlivých sezón.

3.3.10.2 Příklad užití: Prohlížet hodnocení zaměstnanců

ID

UC10.1

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „hodnocení zaměstnanců“.
2. Zahrnout (Zobrazit seznamu zaměstnanců)
3. Systém zobrazí seznam všech zaměstnanců spolu s jejich hodnocením (pouze zaměstnance, kteří mají hodnocení v databázi).

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

3.3.10.3 Příklad užití: Smazat hodnocení zaměstnance

ID

UC10.2

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Příklad užití začíná volbou „smazat hodnocení“ na řádku konkrétního zaměstnance ve výpisu seznamu zaměstnanců.
2. Systém zobrazí dialogové okno s potvrzením smazání konkrétního hodnocení.
3. KDYŽ manažer potvrdí kladně nabídku v dialogovém okně:
 - 3.1 Systém smaže konkrétní položku hodnocení z databáze.

Následné podmínky

1. Proběhne úspěšné smazání záznamu hodnocení z databáze.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

Alternativní tok 2

1. Manažer nepotvrdí v dialogovém okně smazání hodnocení.
2. Příklad užití končí.

3.3.10.4 Případ užití: Zobrazit historii obsazenosti hotelu

ID

UC10.3

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „historie obsazenosti“.
2. Systém zobrazí informace o obsazenosti hotelu v rámci aktuálního měsíce.

Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

Alternativní tok 2

1. Manažer zvolí v roletovém menu požadovaný měsíc pro zobrazení.
2. Manažer potvrdí výběr volbou „zobrazit“.
3. Systém zobrazí informace o obsazenosti hotelu ve vybraném měsíci.

3.3.10.5 Případ užití: Srovnat jednotlivé sezóny

ID

UC10.4

Účastníci

Manažer hotelu

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „srovnat sezóny“.
2. Systém zobrazí průměrné hodnoty obsazenosti hotelu v rámci jednotlivých sezón.

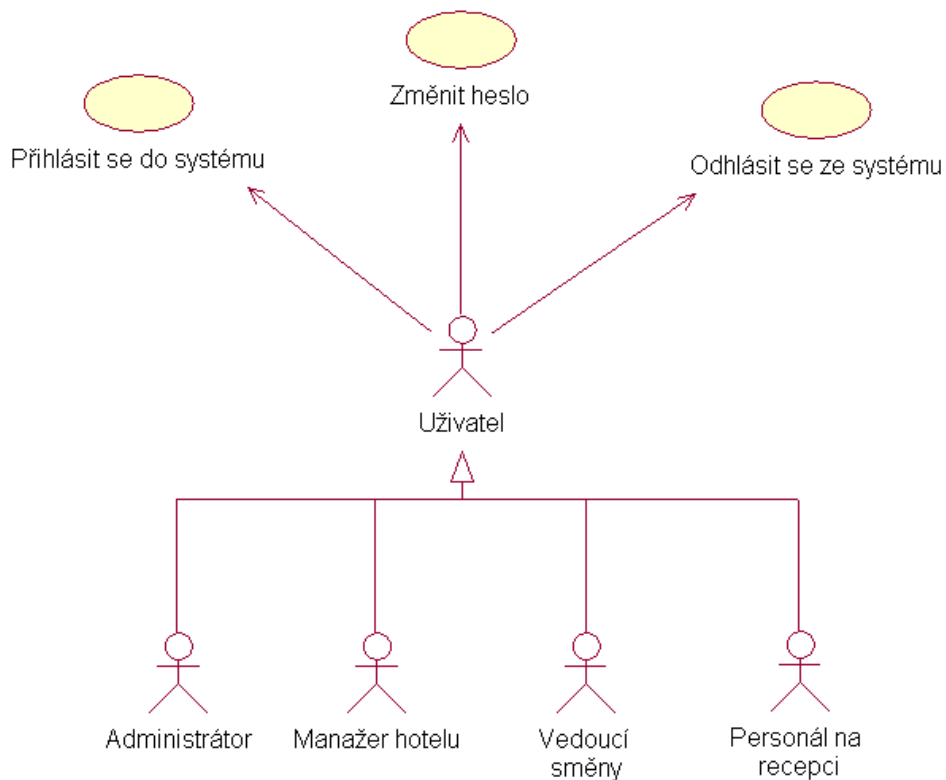
Alternativní tok 1

1. Manažer může kdykoli opustit systém.

Následné podmínky AT 1

1. Manažer se musí odhlásit ze systému.

3.3.11 Autentizace uživatelů



Obr. 20: Diagram případů užití UC11 – Autentizace uživatelů

3.3.11.1 Stručný popis

Na obr. 20 je diagram případů užití, který umožňuje všem uživatelům systému přihlášení, odhlášení a změnu hesla.

3.3.11.2 Příklad užití: Přihlásit se do systému

ID

UC11.1

Účastníci

Uživatel

Vstupní podmínky

Tok událostí

1. Příklad užití začíná volbou „přihlášení do systému“.
2. Systém zobrazí prázdný formulář pro přihlášení do systému.
3. Uživatel zadá přihlašovací jméno (login) a heslo.
4. Uživatel zvolí „přihlásit“.
5. Systém zkontroluje login a heslo
6. KDYŽ login a heslo odpovídají některému záznamu z databáze účtů:

- 6.1 System zobrazí menu, odpovídající typu oprávnění.
- 6.2 Vytvoří se proměnná „session“, která v okně prohlížeče identifikuje konkrétního uživatele pro práci v systému.

Následné podmínky

1. Proběhne úspěšné přihlášení do systému.

Alternativní tok 1

1. Login a heslo neodpovídají záznamu v databázi.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

3.3.11.3 Případ užití: Odhlásit se ze systému

ID

UC11.2

Účastníci

Uživatel

Vstupní podmínky

Tok událostí

1. Případ užití začíná volbou „odhlášení“.
2. System zobrazí dialogové okno s potvrzením odhlášení uživatele.
3. KDYŽ uživatel potvrdí kladně nabídku v dialogovém okně:
 - 3.1 System odhlásí uživatele ze systému.

Následné podmínky

1. Zruší se proměnná „session“.

Alternativní tok 1

1. Uživatel nepotvrdí v dialogovém okně kladně nabídku.
2. Případ užití končí.

3.3.11.4 Případ užití: Změnit heslo

ID

UC11.3

Účastníci

Uživatel

Vstupní podmínky

1. Manažer je přihlášen do systému.

Tok událostí

1. Případ užití začíná volbou „změnit heslo“.
2. System zobrazí prázdný formulář pro změnu hesla.
3. Uživatel zadá staré heslo, nové heslo a nové heslo podruhé.

4. Uživatel zvolí „odeslat“.
5. Systém zkontroluje zda se shoduje dvakrát zadané nové heslo.
6. KDYŽ se hesla shodují:
 - 6.1 Systém uloží změněné údaje do databáze.

Alternativní tok 1

1. Nové heslo prvně zadané se neshoduje s novým heslem zadaným podruhé.
2. Vypíše se na obrazovku chybové hlášení a tok se vrátí k bodu 2.

3.4 Počáteční plán vývoje softwaru

Na základě specifikace požadavků a případů užití je vidět, že vývoj tohoto softwaru bude vyžadovat více než jednu iteraci (etapu vývoje).

V první etapě se vytvoří informační systém, s prvotními funkcemi. Následující etapy budou tento systém rozvíjet o další funkce specifikované v případech užití. Etapy budou na sebe logicky navazovat. To znamená, že dokud nebude hotová například správa služeb, není možné pracovat na rezervaci služeb pro klienty.

V poslední etapě bude probíhat závěrečné testování celého systému. Pokud se dále nevyskytnou žádné problémy a systém bude fungovat dle specifikace požadavků, proběhne předání produktu zadavateli zakázky. Součástí produktu bude uživatelský manuál a instalační příručka. Pro zaměstnance hotelu, kteří budou se systémem pracovat, proběhne jednorázové školení, zaměřené na ovládání systému.

3.4.1 Specifikace jednotlivých etap

Zde je uvedena náplň jednotlivých etap a předběžný časový harmonogram jejich realizace a testování.

Etapa 1

Náplň:

V této etapě se vytvoří základní uživatelské rozhraní systému a funkce autentizace uživatelů (přihlašování, odhlašování).

Termín realizace:

10. 05. 2007 - 12. 05. 2007

Dílčí testování:

12. 05. 2007 – 13. 05. 2007

Etapa 2

Náplň:

V této etapě se vytvoří funkce pro správu uživatelských účtů (tj. správa zaměstnanců z pohledu administrátora systému).

Termín realizace:

14. 05. 2007 - 15. 05. 2007

Dílčí testování:

15. 05. 2007 – 16. 05. 2007

Etapa 3

Náplň:

Ve třetí etapě se budou řešit funkce systému, vztahující se ke správě klientů a partnerů.

Termín realizace:

17. 05. 2007 - 23. 05. 2007

Dílčí testování:

23. 05. 2007 – 24. 05. 2007

Etapa 4

Náplň:

Ve čtvrté etapě se budou řešit funkce systému, vztahující se ke správě rezervací a ubytování.

Termín realizace:

25. 05. 2007 - 01. 06. 2007

Dílčí testování:

01. 06. 2007 – 02. 06. 2007

Etapa 5

Náplň:

V páté etapě se budou řešit funkce systému, vztahující se ke správě úkolů, služeb a hodnocení zaměstnanců.

Termín realizace:

03. 06. 2007 - 05. 06. 2007

Dílčí testování:

05. 05. 2007 – 06. 06. 2007

Etapa 6

Náplň:

V poslední šesté etapě se vytvoří funkce systému, vztahující se ke správě získaných dat a bude probíhat závěrečné testování produktu.

Termín realizace:

07. 06. 2007 - 10. 06. 2007

Dílčí testování:

3.5 Počáteční identifikace a ohodnocení rizik

Identifikace rizik spočívá v určení případů užití, která mohou ovlivnit funkci systému, při nesprávné implementaci.

Ohodnocení rizik je subjektivní činnost, která se zabývá určováním rizikových událostí, na které je třeba reagovat. Ohodnocení je komplikováno řadou faktorů, protože rizika se mohou vzájemně ovlivňovat neočekávanými způsoby. Následující tabulka (Tab. 4) ukazuje identifikované rizika a jejich ohodnocení. Hodnoty vážnost rizika jsou v rozsahu od 0 do 1,0. Nejzávažnější riziko má hodnotu 1,0. Hodnoty priority jsou v rozsahu od 1 do 10 (1 – nejvyšší priority) a jsou vyjádřením důležitosti v rámci implementace.

| Počáteční identifikace a ohodnocení rizik | | | |
|--|--|---------|----------|
| Riziko | Popis | Vážnost | Priorita |
| Přihlášení | Je nutná správná autentizace uživatelů kvůli bezpečnosti systému a pro oddělení pole působnosti. | 1,0 | 1 |
| Odhlášení | Pro případ neodhlášení ze systému je nutné řešit automatické odhlášení. | 0,9 | 1 |
| Vícenásobná rezervace stejného pokoje | Kvůli rezervaci pokojů vztažených k datu je nutné řešit riziko překrývání. | 0,7 | 2 |
| Kontrola dat ve formuláři | Je potřeba kontrolovat správnost dat zadávaných ve formulářích. | 0,5 | 6 |
| Oddělení nabídky vzhledem k typu oprávnění | Je nutné oddělit pole působnosti jednotlivých uživatelů, vztažené k typu oprávnění. | 0,7 | 3 |
| Uživatelské rozhraní | Je dobré vytvořit přehledné a jednoduché uživatelské rozhraní pro práci se systémem. | 0,2 | 8 |

Tab. 4: Počáteční identifikace a ohodnocení rizik

4 Elaborační fáze

Hlavním cílem této fáze je navrhnout stabilní model architektury systému na základě požadavků na systém. Tento model architektury poslouží jako základ pro implementaci systému.

4.1 Ostatní požadavky (požadavky mimo funkcionalitu)

Tyto požadavky můžeme označit jako další požadavky, netýkající se funkcionality a požadavky, které nelze přiřadit jednotlivým případům užití. Metodika RUP je nazývá „Supplementary specifications“ a jsou posledním chybějící částí v úplné množině požadavků na systém.

4.1.1 Rozsah systému

Tento produkt bude napomáhat s organizací rezervací jak hotelových pokojů tak i služeb. Informační systém hotelu bude umožňovat evidenci a správu zákazníků a zaměstnanců hotelu, hotelových pokojů a jejich rezervací, služeb poskytovaných hotelem a získaných dat o obsazenosti hotelu. Databáze bude zabezpečena proti přístupu zvenčí. Budou se do ní také ukládat soubory s fotografiemi zaměstnanců a s jejich životopisy.

Systém nebude volně dostupný z internetu, ale pouze z lokální sítě. Naopak v lokální síti bude dostupný z jakéhokoli počítače. To znamená, že uživatel má možnost si připojit v lokální síti svůj přenosný počítač a vše bude fungovat jak má. Stačí jen aby měl nainstalovaný standardní internetový prohlížeč a samozřejmě uživatelský účet pro přístup do systému.

4.1.2 Reference

Dokument vychází z následujících výstupů inceptní fáze:

- Dokument vize
- Glosář pojmů
- Use-Case model

4.1.3 Použitelnost

Aplikace bude přístupná z lokální sítě tenkým klientem v podobě internetového prohlížeče. Sama o sobě bude spuštěna na serveru připojenému k této lokální síti.

Pro snadné a rychlé ovládání by mělo být vhodně navrženo a realizováno grafické uživatelské rozhraní. Při postupu návrhu tohoto uživatelského rozhraní by se mělo vycházet z nepsaných pravidel pro tvorbu univerzálních webových aplikací. Při splnění těchto vlastností se nový uživatel systému rychle zorientuje v jeho ovládaní a nebudou potřeba rozsáhlá školení těchto uživatelů.

4.1.4 Spolehlivost

Vzhledem k tomu, že recepce hotelu je otevřena nonstop 24 hodin denně, musí být systém schopen běžet nepřetržitě. Musí být stanovena minimální doba mezi pády a také maximální doba pro provedení údržby systému.

Veškeré informace o chybách (identifikační číslo chyby, čas chyby, činnost, při které došlo k chybě, část systému, kde chyba vznikla) a pádech systému by měli být ukládány do speciálního souboru, tzv „log“ souboru.

Musí být zajištěna spolehlivost aplikace při autentizaci uživatelů, vlastních uživatelský účet pro přihlášení a odhlášení ze systému. Bez tohoto účtu nelze využívat žádnou z funkcí systému.

4.1.5 Výkonnost

Systém by měl podporovat minimálně 10 současně pracujících uživatelů. Pro zajištění pohodlné práce by neměla doba odezvy na požadavky uživatele překročit 2 sekundy. Od systému se očekává, že bude zpracovávat v řádech desetitisíce záznamů ročně.

4.1.6 Omezení architektury

Klientskou část aplikace bude možno spustit na libovolném počítači, který bude vybaven síťovou kartou pro přístup k lokální síti. Počítač by měl být vybaven alespoň procesorem o taktu 300 MHz a operační paměti 128 MB. Pro zobrazení aplikace je potřeba mít nainstalován internetový prohlížeč Internet Explorer (minimálně verze 6.0), Mozilla Firefox nebo Opera (minimálně verze 7). Žádný jiný software není potřeba pro chod systému instalovat.

Serverová část aplikace bude nainstalována na vhodném operačním systému (UNIX), pro použití s webovým serverem Apache. Databáze pro potřeby IS hotelu bude postavena na platformě relační databáze MySQL.

4.2 Architektura systému

Tato kapitola specifikuje softwarovou architekturu ze čtyř různých pohledů. Jsou to logický pohled, procesní pohled, pohled nasazení, implementační pohled.

4.2.1 Logický pohled

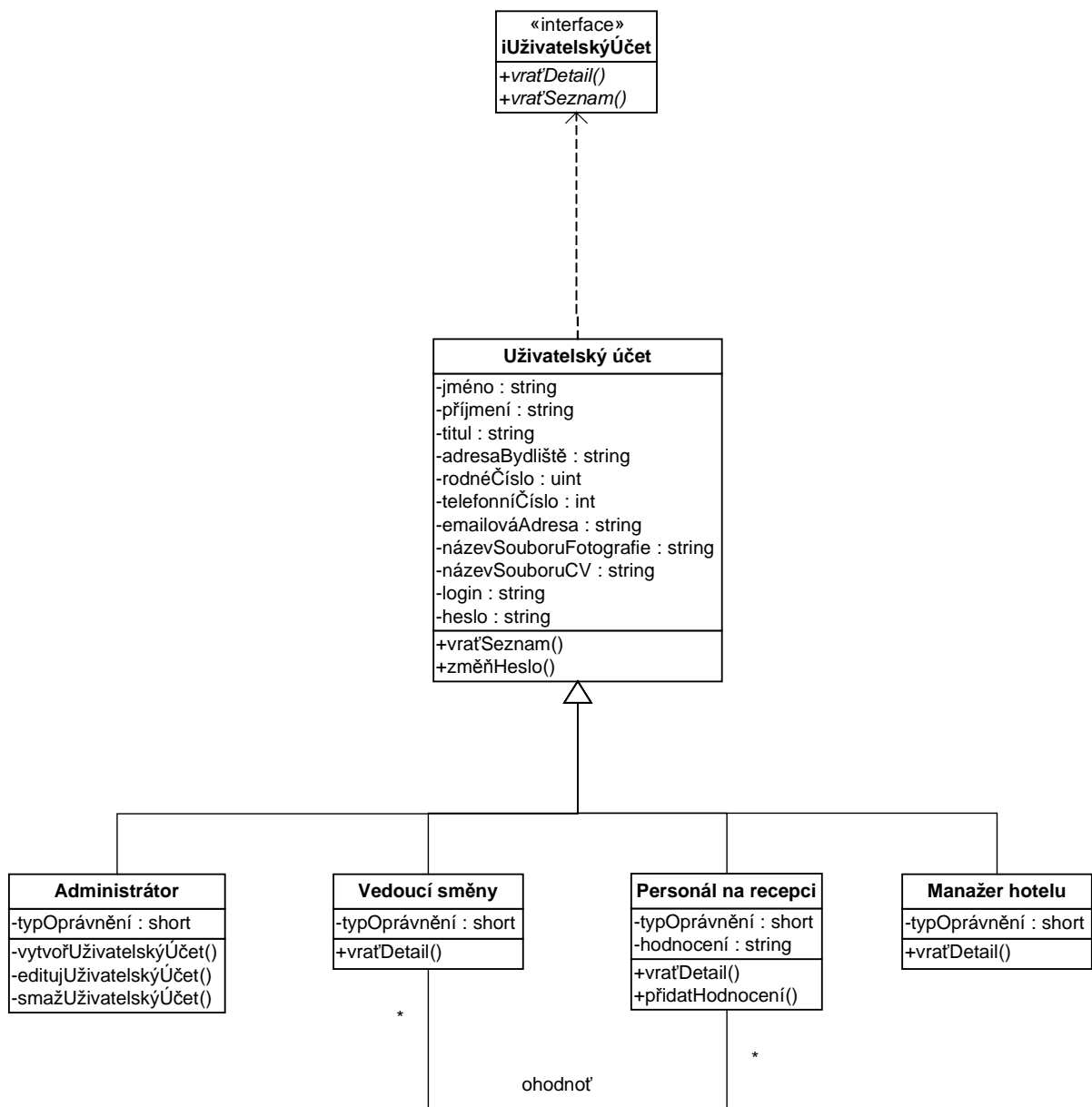
Logický pohled znázorňuje logickou strukturu systému z hlediska výsledné funkcionality. Jde o abstraktní model návrhu, který popisuje hlavní moduly, subsystémy a třídy.

Hlavním prostředkem je zde diagram tříd znázorňující objektové třídy a vztahy mezi nimi (asociace, dědičnost). Pro popis komunikace mezi objekty je použito sekvenčních diagramů.

4.2.1.1 Návrhový model problémové domény (diagramy tříd)

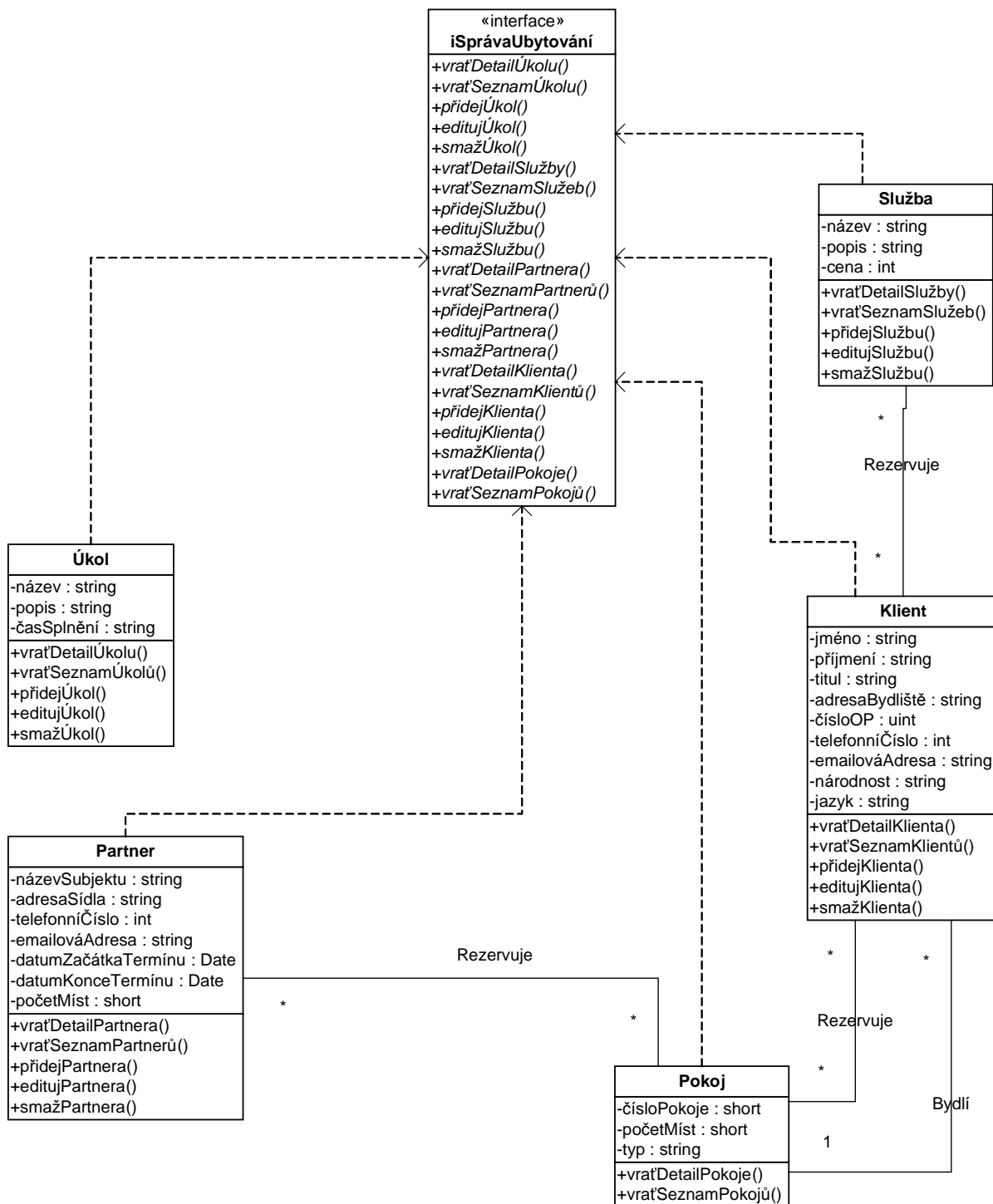
Problémová doména (informační systém hotelu) je rozdělena do dvou hlavních balíčků. A to konkrétně Uživatelský účet a Správa ubytování. Každý balíček obsahuje rozhraní (interface), pomocí kterého spolu oba komunikují.

Balíček Uživatelský účet (Obr. 21) zobrazuje strukturu uživatelských účtů. Abstraktní třída Uživatelský účet generalizuje jednotlivé aktéry Administrátor, Vedoucí směny, Personál na recepci a Manažer hotelu. Pro odlišení aktérů během komunikace s balíčkem Správa ubytování je použit atribut typOprávnění.



Obr. 21: Balíček Uživatelský účet

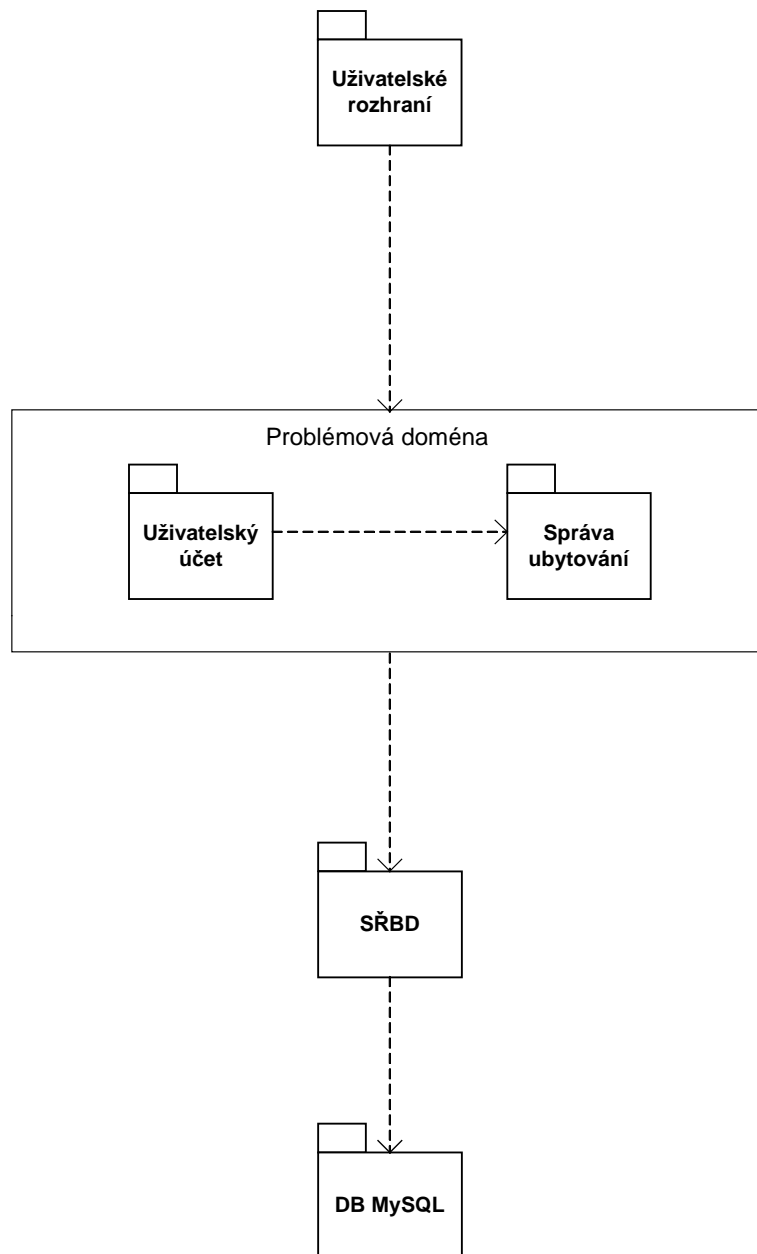
Balíček Správa ubytování (Obr. 22) zobrazuje strukturu tříd Úkol, Služba, Partner, Klient a Pokoj. Pomocí těchto tříd se pak v databázi realizují rezervace hotelových pokojů a služeb a rozdělují se úkoly zaměstnancům.



Obr. 22: Balíček Správa ubytování

4.2.1.2 Diagram balíčků

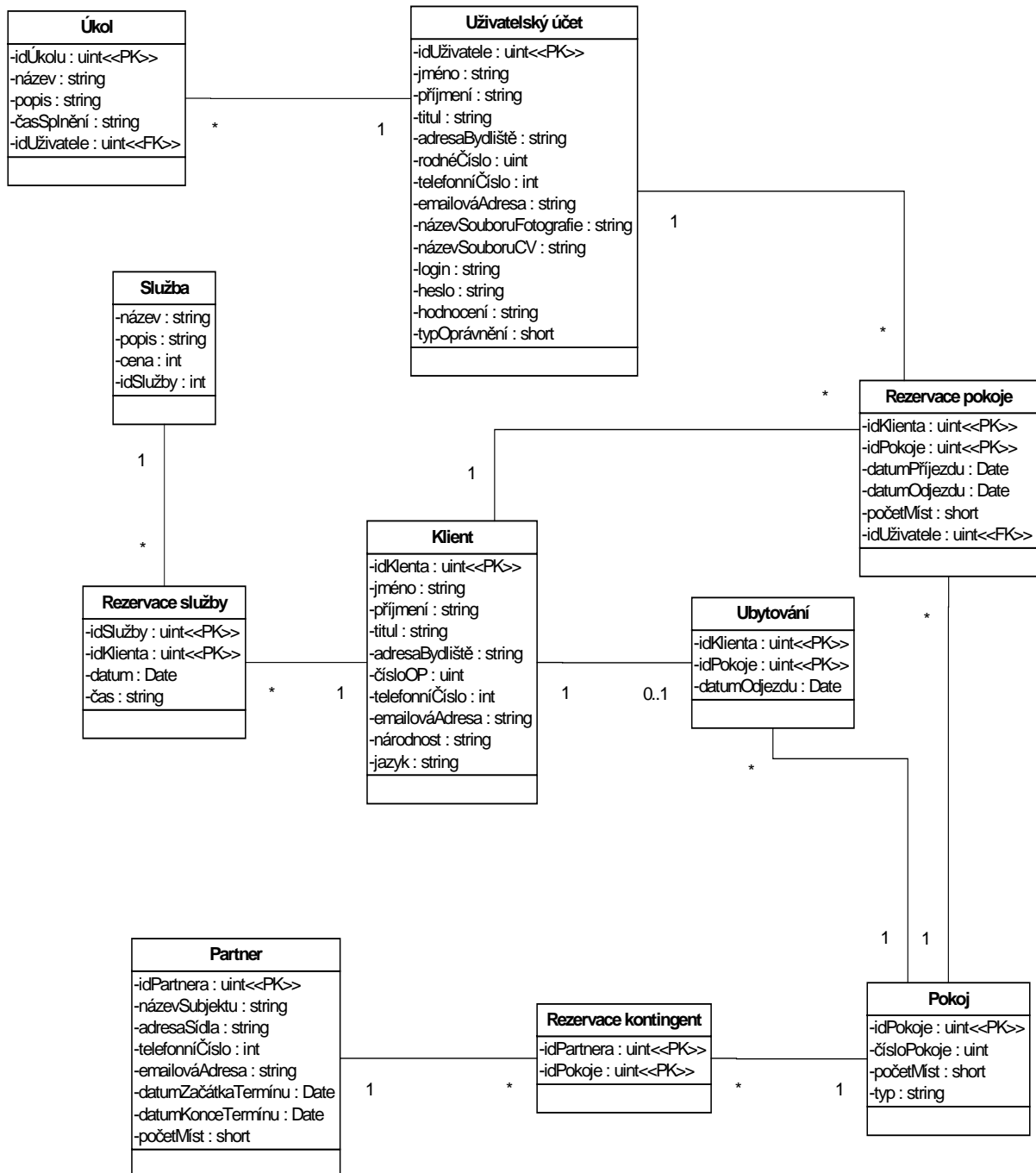
V tomto diagramu (Obr. 23) je názorně zachyceno propojení jednotlivých částí systému. Analytický model zpravidla specifikuje pouze balíčky problémové domény, proto nejsou ostatní balíčky popsány pomocí diagramů tříd.



Obr. 23: Diagram celkového pohledu na balíčky systému

4.2.1.3 Datový model problémové domény (ER diagram)

ER diagram specifikuje návrh databáze. Pro různé databázové platformy mohou vzniknout různé datové modely. V tomto případě se je systém navržen pro databázi MySQL.



Obr. 24: ER diagram (návrh databáze)

4.2.2 Pohled nasazení

Pro potřeby vyvíjeného systému bude použito tzv. třívrstvé architektury. Model třívrstvé architektury se skládá ze třech základních vrstev (Obr. 25) a je pokračovatelem, z dnešního pohledu již zastaralé, dvouvrstvé architektury. Každá vrstva poskytuje navenek určité rozhraní, přes které s ní může druhá vrstva komunikovat. Na rozdíl od dvouvrstvé architektury, klientovi není dovoleno přímo komunikovat s datovou vrstvou. Použití třívrstvé architektury neznámá, že pro každou vrstvu musí být vyhrazen samostatný počítač. [14].

Datová vrstva

Tato vrstva slouží jako datová základna a lhostejno jestli je v pozadí relační databáze, souborový systém, webová služba či jiná aplikace. [13]

Aplikační vrstva

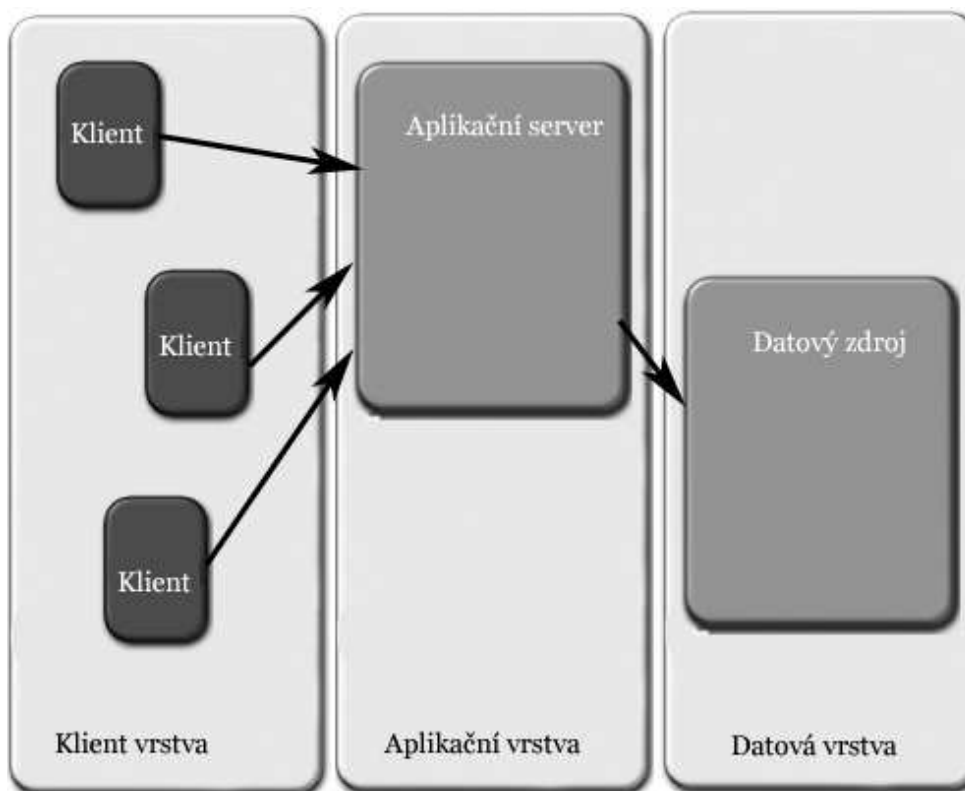
Je to vrstva tvořící skořápku pro aplikační logiku. Zajišťuje přístup k datům, práci s daty a z jejich vystavení ve vhodném formátu (XML, HTML) pro klient vrstvu. Jako nejjednodušší realizaci aplikační vrstvy můžeme považovat webový server s CGI rozhraním, jehož pomocí se volají výkonné rutiny, prezentované spustitelnými soubory a později serverovými skripty. [13]

Aplikační vrstvou je v našem případě webový server Apache, který spracovává skripty v jazyce PHP.

Klientská vrstva

Poslední vrstva má za úkol pouze prezentovat data uživateli. Klientskou vrstvou v případě vytvářeného informačního systému hotelu je zde tzv. tenký klient a to v podobě internetového prohlížeče.

Tenký klient je takový klient, který neobsahuje žádnou aplikační logiku. Aplikační logiku mu zprostředkuje aplikační server, ke kterému tenký klient přistupuje.[13]



Obr. 25: Model třívrstvé architektury (převzato z [13])

4.2.3 Implementační pohled

Zobrazuje organizaci statických softwarových komponent (exe, dll, html...). Zachycuje také rozdělení systému do jednotlivých modulů. Diagram komponent představuje hlavní prostředek znázorňující tento pohled. Implementační pohled je zpravidla řešen v konstrukční fázi, proto zde není uveden.

4.3 Upravený seznam rizik.

V elaborační fázi se také upravuje počáteční seznam rizik, uvedený na konci inceptní fáze. A to zpravidla po návrhu architektury systému. Jednotlivé pohledy totiž poskytují přesnější náhled na problém a mohou tak zapříčinit přehodnocení priorit a vážností jednotlivých rizik.

Následující tabulka (Tab. 5) ukazuje upravený seznam rizik. Ohodnocení vážností a priorit rizik je realizováno stejně jako v tabulce počáteční identifikace a ohodnocení rizik.


| Upravený seznam rizik | | | |
|--|--|------------------|--------------|
| Riziko | Popis změn hodnot | Vážnost | Priorita |
| Přihlášení | Hodnoty tohoto rizika se nezměnily. | 1,0 | 1 |
| Odhlášení | Hodnoty se zmenšily, neboť se došlo k závěru, že riziko nepředstavuje takovou hrozbu. | 0,6 (0,9) | 3 (1) |
| Vícenásobná rezervace stejného pokoje | Mírně se zvýšila vážnost rizika, protože je tato funkce z hlediska celého systému klíčová. | 0,8 (0,7) | 2 |
| Kontrola dat ve formuláři | Hodnoty tohoto rizika se nezměnily. | 0,5 | 6 |
| Oddělení nabídky vzhledem k typu oprávnění | Hodnoty tohoto rizika se nezměnily. | 0,7 | 3 |
| Uživatelské rozhraní | Hodnoty byly zvýšeny, protože v případě špatně navrženého uživatelského rozhraní by mohly nastat nečekané chyby. | 0,4 (0,2) | 7 (8) |

Tab. 5: Upravený seznam rizik


4.4 Plán vývoje softwaru

Pro grafické zobrazení takového plánu na časové stupnici se používá tzv. „ganttův diagram“. Na základě faktu, že je systém vyvíjen a implementován jedinou osobou, je zde zbytečné ganttův diagram uvádět, neboť všechny dílčí kroky v plánu na sebe navazují postupně a žádné dva nemohou být prováděny paralelně.


Níže uvedené obrázky (Obr.26 – Obr. 31) znázorňují celkový plán implementace projektu. Ten byl sestaven na základě standardních omezení: pracovní doba je 8 hodin, práce neprobíhají o víkendech (sobota, neděle). Plán byl vytvořen v aplikaci MS Project 2003 a na přiloženém CD je soubor vytvořený touto aplikací, který obsahuje kompletní ganttův diagram.

| |  Task Name | Duration | Start | Finish | Predecessors |
|----|---|------------------|--------------------|--------------------|--------------|
| 1 | <input type="checkbox"/> Etapa 1 | 3,72 days | Thu 10.5.07 | Tue 15.5.07 | |
| 2 | <input type="checkbox"/> Realizace základní kostry systému | 2,38 days | Thu 10.5.07 | Mon 14.5.07 | |
| 3 | Realizace návrhu grafického uživatelského rozhraní | 8 hrs | Thu 10.5.07 | Fri 11.5.07 | |
| 4 | Realizace HTML šablony | 5 hrs | Fri 11.5.07 | Fri 11.5.07 | 3 |
| 5 | Realizace základních CSS stylů | 4 hrs | Fri 11.5.07 | Mon 14.5.07 | 4 |
| 6 | Realizace skriptu pro připojení k databázi | 2 hrs | Mon 14.5.07 | Mon 14.5.07 | 5 |
| 7 | <input type="checkbox"/> Realizace tabulek databáze | 0,09 days | Mon 14.5.07 | Mon 14.5.07 | |
| 8 | Realizace tabulky uživatelský účet | 45 mins | Mon 14.5.07 | Mon 14.5.07 | 6 |
| 9 | <input type="checkbox"/> Realizace formuláře pro autentizaci uživatelů | 0,63 days | Mon 14.5.07 | Tue 15.5.07 | |
| 10 | Realizace jednotlivých prvků formuláře | 1 hr | Mon 14.5.07 | Mon 14.5.07 | 8 |
| 11 | Realizace skriptu pro zpracování dat z formuláře | 3 hrs | Mon 14.5.07 | Tue 15.5.07 | 10 |
| 12 | Realizace funkce pro ošetření dat z formuláře | 1 hr | Tue 15.5.07 | Tue 15.5.07 | 11 |
| 13 | <input type="checkbox"/> Realizace formuláře pro změnu uživatelského hesla | 0,63 days | Tue 15.5.07 | Tue 15.5.07 | |
| 14 | Realizace jednotlivých prvků formuláře | 1 hr | Tue 15.5.07 | Tue 15.5.07 | 12 |
| 15 | Realizace skriptu pro zpracování dat z formuláře | 3 hrs | Tue 15.5.07 | Tue 15.5.07 | 14 |
| 16 | Realizace funkce pro ošetření dat z formuláře | 1 hr | Tue 15.5.07 | Tue 15.5.07 | 15 |

Obr. 26: Plán etapy 1

| |  Task Name | Duration | Start | Finish | Predecessors |
|----|---|------------------|--------------------|--------------------|--------------|
| 17 | <input type="checkbox"/> Etapa 2 | 2,13 days | Tue 15.5.07 | Thu 17.5.07 | |
| 18 | <input type="checkbox"/> Realizace uživatelského rozhraní pro administrátora | 0,25 days | Tue 15.5.07 | Wed 16.5.07 | |
| 19 | Realizace tlačítek v menu | 2 hrs | Tue 15.5.07 | Wed 16.5.07 | 16 |
| 20 | <input type="checkbox"/> Realizace nabídky výpisu seznamu uživatelských účtů | 0,63 days | Wed 16.5.07 | Wed 16.5.07 | |
| 21 | Realizace skriptu pro výpis všech uživatelských účtů | 5 hrs | Wed 16.5.07 | Wed 16.5.07 | 19 |
| 22 | <input type="checkbox"/> Realizace formuláře pro správu uživatelských účtů | 1,25 days | Wed 16.5.07 | Thu 17.5.07 | |
| 23 | Realizace jednotlivých prvků formuláře | 2 hrs | Wed 16.5.07 | Wed 16.5.07 | 21 |
| 24 | Realizace skriptu pro zpracování dat z formuláře | 6 hrs | Wed 16.5.07 | Thu 17.5.07 | 23 |
| 25 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Thu 17.5.07 | Thu 17.5.07 | 24 |

Obr. 27: Plán etapy 2

| |  Task Name | Duration | Start | Finish | Predecessors |
|----|---|------------------|--------------------|--------------------|--------------|
| 26 | <input type="checkbox"/> Etapa 3 | 5,06 days | Thu 17.5.07 | Fri 25.5.07 | |
| 27 | <input type="checkbox"/> Realizace tabulek databáze | 0,19 days | Thu 17.5.07 | Fri 18.5.07 | |
| 28 | Realizace tabulky klient | 45 mins | Thu 17.5.07 | Fri 18.5.07 | 25 |
| 29 | Realizace tabulky partner | 45 mins | Fri 18.5.07 | Fri 18.5.07 | 28 |
| 30 | <input type="checkbox"/> Realizace uživatelského rozhraní pro manažera hotelu | 0,63 days | Fri 18.5.07 | Fri 18.5.07 | |
| 31 | Realizace tlačítek v menu | 5 hrs | Fri 18.5.07 | Fri 18.5.07 | 29 |
| 32 | <input type="checkbox"/> Realizace nabídky výpisu seznamu všech partnerů | 0,5 days | Fri 18.5.07 | Mon 21.5.07 | |
| 33 | Realizace skriptu pro výpis všech partnerů | 4 hrs | Fri 18.5.07 | Mon 21.5.07 | 31 |
| 34 | <input type="checkbox"/> Realizace formuláře pro správu partnerů | 1,25 days | Mon 21.5.07 | Tue 22.5.07 | |
| 35 | Realizace jednotlivých prvků formuláře | 2 hrs | Mon 21.5.07 | Mon 21.5.07 | 33 |
| 36 | Realizace skriptu pro zpracování dat z formuláře | 6 hrs | Mon 21.5.07 | Tue 22.5.07 | 35 |
| 37 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Tue 22.5.07 | Tue 22.5.07 | 36 |
| 38 | <input type="checkbox"/> Realizace uživatelského rozhraní pro personál na recepci | 0,75 days | Tue 22.5.07 | Wed 23.5.07 | |
| 39 | Realizace tlačítek v menu | 6 hrs | Tue 22.5.07 | Wed 23.5.07 | 37 |
| 40 | <input type="checkbox"/> Realizace nabídky výpisu seznamu všech klientů | 0,5 days | Wed 23.5.07 | Wed 23.5.07 | |
| 41 | Realizace skriptu pro výpis všech klientů | 4 hrs | Wed 23.5.07 | Wed 23.5.07 | 39 |
| 42 | <input type="checkbox"/> Realizace formuláře pro správu klientů | 1,25 days | Wed 23.5.07 | Fri 25.5.07 | |
| 43 | Realizace jednotlivých prvků formuláře | 2 hrs | Wed 23.5.07 | Thu 24.5.07 | 41 |
| 44 | Realizace skriptu pro zpracování dat z formuláře | 6 hrs | Thu 24.5.07 | Thu 24.5.07 | 43 |
| 45 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Thu 24.5.07 | Fri 25.5.07 | 44 |


Obr. 28: Plán etapy 3

| | i Task Name | Duration | Start | Finish | Predecessors |
|----|--|------------------|--------------------|--------------------|--------------|
| 46 | Etapa 4 | 4,75 days | Fri 25.5.07 | Thu 31.5.07 | |
| 47 | Realizace tabulek databáze | 0,38 days | Fri 25.5.07 | Fri 25.5.07 | |
| 48 | Realizace tabulky pokoj | 45 mins | Fri 25.5.07 | Fri 25.5.07 | 45 |
| 49 | Realizace tabulky rezervace pokoje | 45 mins | Fri 25.5.07 | Fri 25.5.07 | 48 |
| 50 | Realizace tabulky rezervace kontingent | 45 mins | Fri 25.5.07 | Fri 25.5.07 | 49 |
| 51 | Realizace tabulky ubytování | 45 mins | Fri 25.5.07 | Fri 25.5.07 | 50 |
| 52 | Realizace nabídky výpisu seznamu rezervací | 0,88 days | Fri 25.5.07 | Mon 28.5.07 | |
| 53 | Realizace skriptu pro výpis rezervací | 7 hrs | Fri 25.5.07 | Mon 28.5.07 | 51 |
| 54 | Realizace formuláře pro správu rezervací pokojů | 1,88 days | Mon 28.5.07 | Wed 30.5.07 | |
| 55 | Realizace jednotlivých prvků formuláře | 4 hrs | Mon 28.5.07 | Mon 28.5.07 | 53 |
| 56 | Realizace skriptu pro zpracování dat z formuláře | 8 hrs | Mon 28.5.07 | Tue 29.5.07 | 55 |
| 57 | Realizace funkce pro ošetření dat z formuláře | 3 hrs | Tue 29.5.07 | Wed 30.5.07 | 56 |
| 58 | Realizace nabídky výpisu seznamu pokojů | 0,5 days | Wed 30.5.07 | Wed 30.5.07 | |
| 59 | Realizace skriptu pro výpis pokojů | 4 hrs | Wed 30.5.07 | Wed 30.5.07 | 57 |
| 60 | Realizace formuláře pro správu ubytování | 1,13 days | Wed 30.5.07 | Thu 31.5.07 | |
| 61 | Realizace jednotlivých prvků formuláře | 3 hrs | Wed 30.5.07 | Thu 31.5.07 | 59 |
| 62 | Realizace skriptu pro zpracování dat z formuláře | 4 hrs | Thu 31.5.07 | Thu 31.5.07 | 61 |
| 63 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Thu 31.5.07 | Thu 31.5.07 | 62 |

Obr. 29: Plán etapy 4

| | i Task Name | Duration | Start | Finish | Predecessors |
|----|--|------------------|--------------------|--------------------|--------------|
| 64 | Etapa 5 | 7,28 days | Thu 31.5.07 | Tue 12.6.07 | |
| 65 | Realizace tabulek databáze | 0,28 days | Thu 31.5.07 | Fri 1.6.07 | |
| 66 | Realizace tabulky úkol | 45 mins | Thu 31.5.07 | Thu 31.5.07 | 63 |
| 67 | Realizace tabulky služba | 45 mins | Thu 31.5.07 | Thu 31.5.07 | 66 |
| 68 | Realizace tabulky rezervace služby | 45 mins | Thu 31.5.07 | Fri 1.6.07 | 67 |
| 69 | Realizace uživatelského rozhraní pro vedoucího směny | 0,5 days | Fri 1.6.07 | Fri 1.6.07 | |
| 70 | Realizace tlačítek v menu | 4 hrs | Fri 1.6.07 | Fri 1.6.07 | 68 |
| 71 | Realizace nabídky výpisu seznamu všech úkolů | 0,38 days | Fri 1.6.07 | Fri 1.6.07 | |
| 72 | Realizace skriptu pro výpis všech úkolů | 3 hrs | Fri 1.6.07 | Fri 1.6.07 | 70 |
| 73 | Realizace formuláře pro správu úkolů | 1,13 days | Fri 1.6.07 | Tue 5.6.07 | |
| 74 | Realizace jednotlivých prvků formuláře | 3 hrs | Fri 1.6.07 | Mon 4.6.07 | 72 |
| 75 | Realizace skriptu pro zpracování dat z formuláře | 4 hrs | Mon 4.6.07 | Mon 4.6.07 | 74 |
| 76 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Mon 4.6.07 | Tue 5.6.07 | 75 |
| 77 | Realizace nabídky výpisu seznamu zaměstnanců | 0,75 days | Tue 5.6.07 | Tue 5.6.07 | |
| 78 | Realizace skriptu pro výpis zaměstnanců | 3 hrs | Tue 5.6.07 | Tue 5.6.07 | 76 |
| 79 | Realizace tlačítek pro správu úkolů jednotlivých zaměstnanců | 3 hrs | Tue 5.6.07 | Tue 5.6.07 | 78 |
| 80 | Realizace nabídky výpisu seznamu všech služeb | 0,38 days | Tue 5.6.07 | Wed 6.6.07 | |
| 81 | Realizace skriptu pro výpis všech služeb | 3 hrs | Tue 5.6.07 | Wed 6.6.07 | 79 |
| 82 | Realizace formuláře pro správu služeb | 1,13 days | Wed 6.6.07 | Thu 7.6.07 | |
| 83 | Realizace jednotlivých prvků formuláře | 3 hrs | Wed 6.6.07 | Wed 6.6.07 | 81 |
| 84 | Realizace skriptu pro zpracování dat z formuláře | 4 hrs | Wed 6.6.07 | Thu 7.6.07 | 83 |
| 85 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Thu 7.6.07 | Thu 7.6.07 | 84 |
| 86 | Realizace nabídky výpisu detailu služby | 0,75 days | Thu 7.6.07 | Fri 8.6.07 | |
| 87 | Realizace skriptu pro výpis všech informací o službě | 6 hrs | Thu 7.6.07 | Fri 8.6.07 | 85 |
| 88 | Realizace formuláře pro rezervaci služeb | 1,5 days | Fri 8.6.07 | Mon 11.6.07 | |
| 89 | Realizace jednotlivých prvků formuláře | 3 hrs | Fri 8.6.07 | Fri 8.6.07 | 87 |
| 90 | Realizace skriptu pro zpracování dat z formuláře | 6 hrs | Fri 8.6.07 | Mon 11.6.07 | 89 |
| 91 | Realizace funkce pro ošetření dat z formuláře | 3 hrs | Mon 11.6.07 | Mon 11.6.07 | 90 |
| 92 | Realizace formuláře pro zadávání hodnocení zaměstnanců | 0,5 days | Mon 11.6.07 | Tue 12.6.07 | |
| 93 | Realizace jednotlivých prvků formuláře | 1 hr | Mon 11.6.07 | Mon 11.6.07 | 91 |
| 94 | Realizace skriptu pro zpracování dat z formuláře | 2 hrs | Mon 11.6.07 | Mon 11.6.07 | 93 |
| 95 | Realizace funkce pro ošetření dat z formuláře | 1 hr | Mon 11.6.07 | Tue 12.6.07 | 94 |

Obr. 30: Plán etapy 5

| |  Task Name | Duration | Start | Finish | Predecessors |
|-----|---|------------------|--------------------|--------------------|--------------|
| 96 | [-] Etapa 6 | 2,88 days | Tue 12.6.07 | Thu 14.6.07 | |
| 97 | [-] Realizace nabídky výpisu získaných dat | 1 day | Tue 12.6.07 | Wed 13.6.07 | |
| 98 | Realizace skriptu pro výpis získaných dat | 8 hrs | Tue 12.6.07 | Wed 13.6.07 | 95 |
| 99 | [-] Realizace formuláře pro editaci výpisu získaných dat | 1,38 days | Wed 13.6.07 | Thu 14.6.07 | |
| 100 | Realizace jednotlivých prvků formuláře | 3 hrs | Wed 13.6.07 | Wed 13.6.07 | 98 |
| 101 | Realizace skriptu pro zpracování dat z formuláře | 6 hrs | Wed 13.6.07 | Thu 14.6.07 | 100 |
| 102 | Realizace funkce pro ošetření dat z formuláře | 2 hrs | Thu 14.6.07 | Thu 14.6.07 | 101 |
| 103 | [-] Realizace nabídky výpisu hodnocení zaměstnanců | 0,5 days | Thu 14.6.07 | Thu 14.6.07 | |
| 104 | Realizace skriptu pro výpis hodnocení zaměstnanců | 2 hrs | Thu 14.6.07 | Thu 14.6.07 | 102 |
| 105 | Realizace tlačítka pro smazání hodnocení zaměstnance | 2 hrs | Thu 14.6.07 | Thu 14.6.07 | 104 |

Obr. 31: Plán etapy 6

4.5 Prototyp systému

Jako jeden z výsledků elaborační fáze by měl být funkční prototyp, na kterém lze testovat navrženou architekturu systému. Tento prototyp by měl implementovat přibližně 10% případů užití specifikovaných v inepční fázi. V případě úspěšných testů architektury a uživatelského rozhraní, do kterých se může zapojit i zadavatel zakázky, se pokračuje následující fází, fází implementační.

Funkční prototyp modelované problémové domény (informační systém hotelu) je dostupný na přiloženém CD.

5 Závěr

Cílem diplomové práce bylo seznámit se s metodikou IBM Rational Unified Process (RUP) a to především podrobněji prostudovat inepční a elaborační fázi. Dále pak provést inepční a elaborační fázi informačního systému pokrývající zvolenou business problematiku..

Náklady spojené s vývojem a nákupem software dnes ve většině případů převyšují pořizovací náklady související s hardware. Je obecně známo, že vysoké procento softwarových projektů (přes 90%) je neúspěšných. Tento fakt je především způsoben nedokonalou správou požadavků (popř. nesrovnalostí požadavků), nejasnou komunikací, křehkou architekturou systému, podceněním složitosti řešeného projektu, návrhem a implementací, neadekvátním testováním, subjektivním odhadem stavu projektu, nepředcházením rizik, nekontrolovaným prováděním změn, nedostatečnou automatizací apod. Použitím obecné metodiky pro vývoj software zajišťuje úspěch celého procesu vývoje. Takovou metodikou je právě IBM Rational Unified Process. Přestože neobsahuje obecný návod vhodný pro každý softwarový vývoj, vyhovuje jak malým vývojovým týmům, tak velkým vývojovým organizacím. Jak jsem popsal v mé práci, jde o soubor dohodnutých pravidel, principů a postupů ovlivňujících efektivnost a organizaci práce, což vede k úspěšnému vyvinutí software.

Pro účely diplomové práce, jsem si zvolil business doménu, na které jsem se pokusil demonstrovat postup vývoje projektu pomocí metodiky RUP. A to konkrétně informační systém hotelu. Volbu jsem konzultoval s inženýrem Markem Beránkem a s vedoucí diplomové práce doktorkou Jitkou Kreslíkovou.

V teoretické části této práce jsem důkladně popsal aspekty metodiky RUP a to především její základní filosofii, základní stavební kameny, jednotlivé fáze a disciplíny životního cyklu vývoje projektu v RUP a architekturu systému.

V praktické části jsem se věnoval vypracování výstupů inepční a elaborační fáze pro informační systém hotelu. Nejrozsáhlejší částí práce je detailní popis případů užití a to proto, že metodika RUP se soustředí právě na tuto část vývoje systému nejvíce.

Závěrem bych chtěl podotknout, že výhody přístupu k vývoji projektu pomocí RUP jsou nesporné a byla o nich již řeč. Objevil jsem však i nevýhody. Ty vyplývají paradoxně z výhod metodiky. Především univerzálnost přístupu má za následek tvorbu někdy zbytečně obsáhlé projektové dokumentace a to v případě vytváření méně rozsáhlých projektů. Navíc osvojení metodiky je relativně časově náročné a pro menší projekty to vede nakonec k navýšení času potřebného k jejich tvorbě.

Literatura

- [1] Maciaszek, L.A., Liong, B.L.: Practical Software Engineering: A Case Study Approach. Harlow, Addison-Wesley, 864 p., 2005, ISBN 0-321-20465-4
- [2] Arlow, J., Neustadt, I.: UML a unifikovaný proces vývoje aplikací, Computer Press 2003, ISBN 80-7226-947-X
- [3] Kadlec, V.: Rational Unified Process: základní pojmy, <http://www.zive.cz>, 05. 08. 2003
- [4] doc. Ing. Zendulka J., aj.: Analýza a návrh informačních systémů [studijní opora], verze 31.10.2006, VUT v Brně
- [5] Aldorf, F.: Metodika RUP [diplomová práce], <http://objekty.vse.cz/Objekty/Rup1>
- [6] Štork, R., Vitouš, O.: Rational Unified Process stručný průvodce, UNICORN Multimedia 2000, ISBN 80-238-6358-4
- [7] Wikipedia, the free encyclopedia (Redirected from Rational Unified Process), IBM Rational Unified Process: http://en.wikipedia.org/wiki/Rational_Unified_Process
- [8] Internetové stránky: <http://mpavus.wz.cz/uml>
- [9] Ing. Rábová, I.: Diagramy UML a Rational Rose, Provozně ekonomická fakulta MZLU v Brně 2000
- [10] Internetové stránky: <http://www.unicorn.cz/cz/press/>, článek „Co je Rational Unified Process?“
- [11] Beránek, M., Slabý, A.: Řízení projektu dle RUP využívající princip KKTR, Fakulta informatiky a managementu UHK Hradec Králové
- [12] Beránek, M.: Systém pro evidenci uchazečů v zaměstnání [diplomová práce], Fakulta informatiky a managementu UHK Hradec Králové
- [13] Internetové stránky: <http://www.sweb.cz/pichlik/>
- [14] Internetové stránky: <http://www.itexpert.cz/trivrstva-architektura/>

Seznam příloh

Příloha 1. CD

Příložený datový nosič se zdrojovými kódy prototypu, instalační příručkou, diplomovou prací ve formátu „doc“ (Microsoft Word), diplomovou prací ve formátu „pdf“ a plánem projektu ve formátu „mpp“ (Microsoft Project 2003).