



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Vysoká škola báňská – Technická univerzita Ostrava

Fakulta strojní



## TÝMOVÁ SPOLUPRÁCE V NÁVRHU INFORMAČNÍHO SYSTÉMU

Případová studie k předmětu Informační systémy

Jolana Škutová

Ostrava 2012



Tyto studijní materiály vznikly za finanční podpory Evropského sociálního fondu (ESF) a rozpočtu České republiky v rámci řešení projektu OP VK CZ.1.07/2.3.00/09.0147 „Vzdělávání lidských zdrojů pro rozvoj týmů ve vývoji a výzkumu“.

Recenze: <Jméno recenzenta> [Poznámka: případně se tento řádek odstraní]

Název: Týmová spolupráce v návrhu informačního systému

Autor: Jolana Škutová

Vydání: první, 2012

Počet stran: 56

Náklad: 5

Studijní materiály pro stud. obor Automatické řízení a inženýrská informatika Fakulty strojní  
Jazyková korektura: nebyla provedena.



EVROPSKÁ UNIE



MINISTERSTVO ŠKOLSTVÍ,  
MLÁDEŽE A TĚLOVÝCHOVY



OP Vzdělávání  
pro konkurenceschopnost



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

**Tyto studijní materiály vznikly za finanční podpory Evropského sociálního fondu a rozpočtu České republiky v rámci řešení projektu Operačního programu Vzdělávání pro konkurenceschopnost.**



*Název:* Vzdělávání lidských zdrojů pro rozvoj týmů ve vývoji a výzkumu

*Číslo:* CZ.1.07/2.3.00/09.0147

*Realizace:* Vysoká škola báňská – Technická univerzita Ostrava

© Jolana Škutová

© Vysoká škola báňská – Technická univerzita Ostrava

ISBN

Při realizaci případové studie byly jednotlivé etapy rozděleny do následujících bloků:



### Plánování jednotlivých kroků:

Na úvod kapitoly je uveden časový plán nebo rozvrh kapitoly případové studie.



### Cíl:

- ✚ Definovat cíle jednotlivých oblastí
- ✚ Vyřešit následující problémy

Ihned potom jsou uvedeny cíle, které jsou v případové studii objektem zkoumání.



### Výklad a popis případové studie

Následuje vlastní výklad studované látky, zavedení nových pojmů, jejich vysvětlení, vše doprovázeno obrázky, tabulkami, řešenými příklady, grafy či odkazy na animace.



### Rizika a plán na jejich odstranění

Při realizaci plánovaných činností se mohou vyskytnout rizika, které mají vliv na zdárné vykonání cílů. Tento popis by měl navrhnout řešení minimalizace těchto rizik.



### Nápady, výjimečné případy a řešení neplánovaných situací

Protože se v praxi vyskytne celá řada neočekávaných situací, které nejsou přímo rizikem pro finální realizaci, je těmto případům věnována tato kategorie.



### Příklad z praxe

*Příklad z praxe popisuje přímo zkušenosti, které vznikly v době realizace konkrétních činností.*



### Důležité informace

**Důležité informace** jsou pojmy, které rozhodně při realizaci jednotlivých aktivit nemohou být opomenuty, neboť jsou klíčové pro správné dokončení plánovaných akcí.

**Protože různé případové studie se mohou lišit postupem svého zpracování, některé předepsané popisy ikon se tudíž nemusí hodit, naopak je možné vytvořit svůj vlastní popis ikonky, které se bude autorovi případové studie hodit. Proto doporučujeme zachovat šablonu případové studie, popisy ikonky vytvořte dle svého uvážení.**

Nabídka dalších ikonek s možností použití těchto či jiných (vlastních popisků)



**Klíč k řešení**



**Shrnutí pojmů**



**Zajímavost k tématu**



**Další zdroje**



**Průvodce případovou studií**

Text, kterým chceme sdělit formálně či neformálně co bude obsahem následujících kapitol či odstavců – co budeme zkoumat či popisovat.

Následuje základní text stylem Normální.



**„coffee break“ tzv. odbočení od tématu s vysvětlením proč**

Text, kterým se chceme odkázat na jiné studie, případy či vysvětlení souvisejících pojmů.

Následuje základní text stylem Normální.



**Korespondenční úkol**

Zadání úlohy, či definice problému související s případovou studií.



**Příprava na případovou studii**

Sběr materiálu k případové studii.

**Jak můžete vidět, některé ikonky s popisem jsou více řádkové, v takových se do druhého řádku píše ještě doprovodný text. Jednořádkové ikonky s popisem mají spíše roli nadpisu.**



## Charakteristika případové studie

Případová studie je jedním z přístupů kvalitativního výzkumu. Je charakterizovaná obecně jako „detailní studium jednoho případu nebo několika málo případů“. Pedagogický slovník uvádí definici „Výzkumná metoda v empirickém pedagogickém výzkumu, při níž je zkoumání podroben jednotlivý případ (např. žák, malá skupina žáků, jednotlivá třída, škola apod.), detailně popsán a vysvětlován, takže se dochází k takovému typu objasnění, jehož při zkoumání těchto objektů v hromadném souboru nelze dosáhnout. Výhodou metody je možnost hlubokého poznání podstaty případu, nevýhodou omezenost zobecnitelnosti výsledků.“

Je to metoda, která umožňuje zachycení složitosti, detailů, vztahů a procesů probíhajících v daném mikroprostředí. Předpokládá, že podrobný výzkum jednoho případu přispěje k lepšímu porozumění a pochopení jiných, obdobných případů. Tyto případy je ovšem třeba vnímat a chápat v širším kontextu, eventuálně je srovnat s dalšími případy.

Zkoumá, jaké jsou charakteristiky daného případu nebo skupiny porovnávaných případů. Na rozdíl od statistického šetření, které shromažďuje relativně omezené množství dat od mnoha jedinců nebo případů, se snaží o zajištění velkého množství dat od jednoho nebo několika málo jedinců. Jde v ní o zachycení složitosti zkoumaného případu, o popis vztahů.

Případové studie se používají především v lékařství, psychologii, etnografii, sociologii, či v pedagogickém výzkumu.

I když je případová studie považována zpravidla za metodu kvalitativního výzkumu, dva významní autoři, autority v oblasti výzkumu využívajícího případové studie, Stake (1995) a Yin (1994) shodně poukazují na možnost výhradně kvantitativních případových studií, využívajících testy a soubory deskriptivních proměnných a upozorňují na smíšený typ případových studií, jež je frekventovaně využíváný.

Co je podstatné a pro případovou studii příznačné, je odlišnost od laboratorního, izolovaného výzkumu tím, že výzkum využívající případové studie se odehrává v terénu (field research).



Cíl: Případová studie musí splňovat určité podmínky

- ✚ stanovit typy otázek, na něž bude hledat odpovědi odkrýváním zkoumaného případu v terénu Definovat cíle jednotlivých oblastí
- ✚ vymežit roli výzkumníka (výzkumného týmu) Vyřešit následující problémy
- ✚ zvážit, zda bude zkoumat současný stav nebo historii případu.

Každá případová studie má tzv. zakotvenou teorii, tj. svůj vlastní logický rámec, design, akční plán. Vývoj teorie je součástí záměru, zpravidla jsou také stanoveny určité komponenty nebo jednotky analýzy, okruhy nebo typy zdrojů, získávání dat a způsoby jejich záznamu.

Podstatný je výběr případu, který bude zkoumán tak, aby reprezentoval určitý typ nebo skupinu obdobných případů. Různorodost reality daných případů poskytuje více interpretací, na nichž se, jak již bylo uvedeno, podílejí aktéři, role výzkumníka a jeho asertivita je podstatnou formou analytické generalizace.



## Objekty případové studie

Případové studie se mohou značně lišit v předmětu, na něž jsou zaměřeny, v celkovém záměru, použitých technikách a výzkumných nástrojích. Zpravidla se uvádějí jako typy případové studie zkoumající jednotlivé osoby, komunity (malé skupiny), sociální skupiny, organizace nebo instituce, události nebo vztahy.



### Typy případových studií

**Osobní případová studie** je podrobným výzkumem určitého aspektu u jedné osoby. Věnuje se minulosti, kontextovým faktorům a postojům, které zkoumané události předcházely. Zkoumá možné příčiny, determinanty, faktory a procesy, jež s ní měly souvislost. Využívá se velmi často ve zdravotnictví k popisu konkrétních lékařských případů. Jsou zde popisovány symptomy, anamnéza, návrh, průběh a výsledek léčby.

**Studie komunity** zkoumá jednu či více komunit v určité lokalitě, případně lokalitu samotnou. Popisuje a analyzuje vzorce hlavních aspektů života komunity (aspekty politické, práci, volný čas, rodinný život, zvyky, pravidla apod.) Někdy se pro takovéto studie používá označení sociografie.

**Studium sociálních skupin** se zabývá zkoumáním malých přímo komunikujících skupin (například rodin) i větších difúzních skupin (například zaměstnanců jednoho podniku). Popisuje a analyzuje vztahy a aktivity ve skupině.

**Studium organizací a institucí** zkoumá firmy, školy a jiné organizace, implementace programů a intervencí, kulturu organizací, procesy změn a adaptací. Hledá nejlepší vzorce chování, zavedení určitého typu řízení, evaluace.

**Zkoumání událostí, rolí a vztahů** se zaměřuje na určitou událost, částečně se může překrývat se studiem sociálních skupin a organizací. Shrnuje analýzu interakce členů skupiny, konfliktu rolí, stereotypy.

Ve všech výše uvedených typech případových studií se může jednat o různé **výzkumné záměry**.



### Výzkumné záměry případových studií

Objasnění pouze **jednoho daného případu** a proniknutí do hloubky problémů v dané situaci nebo organizaci se záměrem porozumět okolnostem tohoto případu

- objasnění určitého **jevu** vyskytujícího se v realitě na jednom nebo několika případech jev reprezentujících

- zkoumání **více případů** s cílem ověřit určitou koncepci nebo inovaci a získat poznatky prokazující určité společné rysy poskytující vhodnost či nevhodnost této koncepce či inovace.

Při realizaci výzkumu metodou případové studie je podstatné rozhodnout o tom, zda bude zkoumán jeden nebo více případů a které případy budou k výzkumu vybrány. Výběr případů je vždy cílený a zásadní pro celý výzkum. Zvláštním případem jsou tzv. **mnohonásobné případové studie**, které se používají zejména ve srovnávacích výzkumech. Jejich cílem je odhalit určité společné nebo rozdílné znaky či trendy, eventuálně dospět k obecnějším teoretickým závěrům. Takové soubory případových studií vyžadují přesný scénář, vymezení klíčových otázek, základních komponentů a výzkumných nástrojů tak, aby při replikaci ve více případech bylo umožněno srovnání a vyvození závěrů pro celý soubor případů.

Při výzkumu s využitím více případových studií se rozlišují dva přístupy

- **multicase studies**, při němž se realizuje více případových studií, které se srovnávají a vytváří se jedna společná studie

- **single a multiply case studies**, při němž se jednotlivé studie nesrovnávají, každá je unikátním případem.

**OBSAH**

<b>1</b>	<b>OBJEKTOVÝ PŘÍSTUP V NÁVRHU INFORMAČNÍHO SYSTÉMU .....</b>	<b>10</b>
1.1	Proces vývoje informačního systému.....	10
1.2	Definice skupin a tematických okruhů.....	12
1.3	Role týmu a jejich popis.....	14
<b>2</b>	<b>ZPRACOVÁNÍ POŽADAVKŮ INFORMAČNÍHO SYSTÉMU .....</b>	<b>18</b>
2.1	Definice funkčních a nefunkčních požadavků .....	18
2.2	Sběr požadavků pro dílčí tematické celky .....	19
2.2.1	Informační systém sportovních aktivit .....	20
2.2.2	Informační systém pro administraci stravování.....	21
2.2.3	Informační systém osobní agendy ve zdravotnictví.....	23
2.3	Zhodnocení sběru požadavků .....	23
2.3.1	Informační systém sportovních aktivit .....	24
2.3.2	Informační systém pro administraci stravování.....	25
<b>3</b>	<b>REALIZACE NÁVRHU INFORMAČNÍHO SYSTÉMU.....</b>	<b>28</b>
3.1	Modelování případů užití .....	28
3.2	Modelování případů užití pro dílčí tematické celky .....	32
3.2.1	Informační systém sportovních aktivit.....	32
3.2.2	Informační systém pro administraci stravování.....	35
3.2.3	Informační systém osobní agendy ve zdravotnictví.....	37
3.3	Zhodnocení modelování případů užití.....	37
3.3.1	Informační systém sportovních aktivit .....	38
3.3.2	Informační systém pro administraci stravování.....	39
<b>4</b>	<b>IMPLEMENTACE INFORMAČNÍHO SYSTÉMU.....</b>	<b>41</b>
4.1	Diagram tříd .....	41
4.2	Datový model .....	43
4.3	Implementace informačních systémů pro dílčí tematické celky .....	43
4.3.1	Informační systém sportovních aktivit .....	43
4.3.2	Informační systém pro administraci stravování.....	44
4.4	Zhodnocení implementační fáze procesu vývoje IS .....	45
4.4.1	Informační systém sportovních aktivit.....	46
4.4.2	Informační systém pro administraci stravování.....	48



<b>5</b>	<b>STRUKTURA INFORMAČNÍHO SYSTÉMU .....</b>	<b>50</b>
5.1	Informační systém sportovních aktivit.....	50
5.2	Informační systém pro administraci stravování .....	51
5.3	Informační systém osobní agendy ve zdravotnictví .....	52
<b>6</b>	<b>ZHODNOCENÍ TÝMOVÉ SPOLUPRÁCE PŘI NÁVRHU A REALIZACI INFORMAČNÍHO SYSTÉMU .....</b>	<b>54</b>

# 1 OBJEKTOVÝ PŘÍSTUP V NÁVRHU INFORMAČNÍHO SYSTÉMU



## Plánování jednotlivých kroků: 1 hodina

Rozvržení týmů (skupin), rozdělení a definice rolí v týmu a přesné specifikace úkolů týmových specialistů.



### Cíl:

- ✚ Definovat proces vývoje informačního systému
- ✚ Vyčlenit tematické celky řešených projektů
- ✚ Seznámit se s pozicemi v projektovém týmu
- ✚ Rozdělit role projektového týmu



## Výklad a popis případové studie

V procesu tvorby informačního systému je v současné době vhodné postupovat systematicky, objektově a s využitím grafického nástroje UML. Objektový návrh není stále v mnoha směrech podporován, a proto je cílem případové studie poukázat na směr, rozvoj a tvorbu informačního systému v týmové spolupráci různorodých specialistů.

Vývoj informačního systému se navrhuje postupně v několika fázích, na nichž se podílí všichni členové projektového týmu méně či více ve svém zaměření. Vysvětlíme jednotlivé fáze projektu, v nichž probíhá vývoj systému. Bohužel se v projektovém týmu nepostupovalo stejně v čase a nebylo možné postřehnout dílčí fáze zpracování projektu jednotlivými členy týmu.

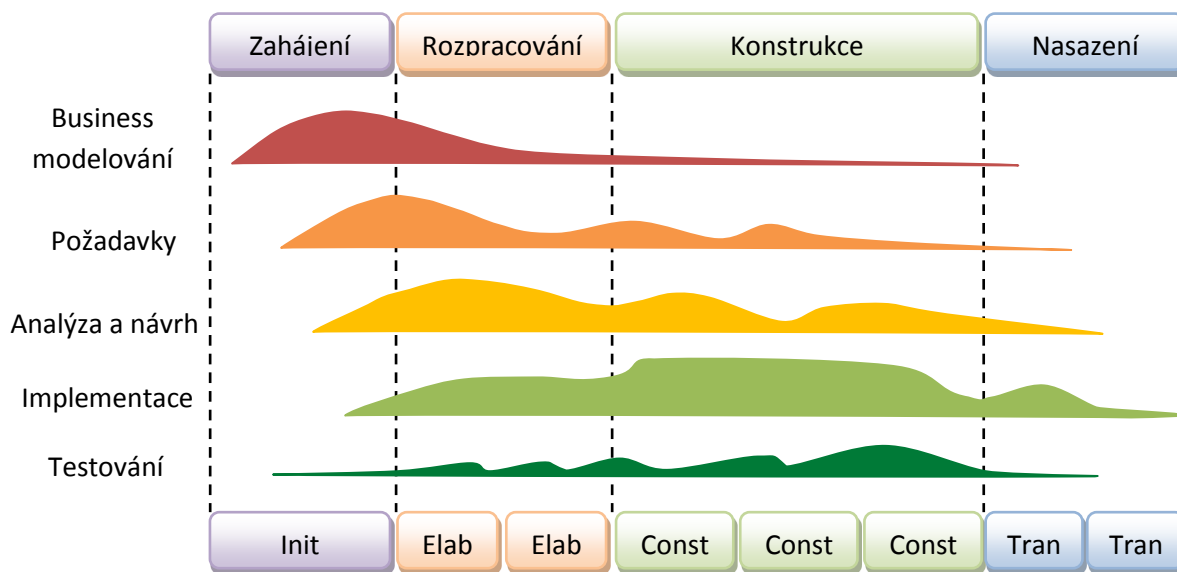
Cílovou skupinu případové studie tvoří samostatně profesní osobnosti projektového týmu, které mají předány role neboli úlohy. Cílem bude sledovat spolupráci v jednom týmu mezi týmovými spolupracovníky (specialisty) a rovněž bude porovnáno mezi různými týmy, jak se stejné role týmu naplňují v týmech s odlišnou tematikou řešeného problému.

Na počátku týmové spolupráce vždy stojí problém definice počtu rolí v týmu, definice dílčích rolí týmu a spolupráce všech účastníků daného týmu. V následujících kapitolách bude definován počet skupin nebo řešitelských týmů a s tím souvisí tedy stejný počet tematicky zaměřených postupů při návrhu informačního systému. Dále bude definována oblast řešení pro každého člena týmu a jejich navazující spolupráce při tvorbě informačního systému.

### 1.1 Proces vývoje informačního systému

Vývoj informačního systému (IS) je založen na objektově orientovaném přístupu, který je založen na přístupu řízeném případy užití (use-case-driven approach) a hodnocení

rizik. Tradiční vývoj IS probíhá v cyklech, což omezuje rizika pro odběratele i projektový tým. Tento přístup umožňuje bez potíží zapracovávat nové požadavky i v průběhu projektu.



Obrázek 1.1 – Životní cyklus unifikovaného procesu

Fáze vývoje informačního systému se člení do čtyř základních linií:

- ✚ **Zahájení** – věcné vymezení projektu, stanovení vize projektu a definice jeho cílů. Na konci fáze probíhá zhodnocení a rozhoduje se, zda se bude v projektu pokračovat s daným záměrem, nebo je potřeba předefinovat cíle a zadání projektu.
- ✚ **Rozpracování** – fáze zahrnuje návrhy a realizace procesu modelování, definice kvality a zpřesnění rizik a požadavků.
- ✚ **Konstrukce** – etapa implementace veškerých funkčních celků, tj. realizace modelovaného systému v daném softwarovém prostředí. Fáze je rozdělena do více iterací, v nichž postupně vznikají verze projektu, které se konzultují s odběratelem. V dalších iteracích se zapracují připomínky, případně nové požadavky do modelovaného systému.
- ✚ **Nasazení** – představuje přechod navrženého software směrem k uživateli, pokud software dosáhl předpokládané úrovně a stability. Další chyby budou opraveny v rámci dalších verzí systému. Primárním cílem přechodné fáze je předat software uživateli, podpořit jeho schopnosti v obsluze a předat dokumentaci.

Jednotlivé fáze se dále rozdělují podle velikosti modelovaného systému na více iterací (Elab #1, Elab #2, Const #1, Const #2, apod.). V těchto iteracích se pak realizují postupně pracovní postupy, kterých se daná fáze dotýká. Např. ve fázi zahájení bude většina času věnována Business modelování, ostatní pracovní postupy se v této fázi podílí na vzniku modelovaného systému mnohem méně.

## 1.2 Definice skupin a tematických okruhů

Při návrhu informačního systému a stanovení rolí mezi dílčí spolupracovníky vždy je vhodné vycházet z časoprostorového hlediska návrhu informačního systému s ohledem na maximálně možný počet osob v týmu.

Informační systém pro naši případovou studii by měl být navržen, implementován a realizován za období tří měsíců podle požadavků a představ zadavatele řešeného problému. Dalším hlediskem při sestavování skupin je maximální možná šíře oblasti specializace neboli množství zpracované části návrhu informačního systému. Čím větší množství zpracovaných problematických částí za předpokladu omezeného času, tím menší hloubka zpracovaného návrhu, které může vést na povrchně zpracovaný návrh informačního systému.



### Definice tematických okruhů

Podle dostupného počtu zúčastněných osob v týmu byly navrženy tři tematické okruhy. Odběratel požaduje navržení projektu a předkládá počáteční nástin řešených problémů pro projektování evidence zápasů, hráčů dané sportovní aktivity, dále projektování systému zajišťující evidenci receptů, jídelníčků a kvalitativního zhodnocení potravin. Poslední projekt je z oblasti zdravotnictví, je určen pro evidenci zdravotnictví, úkonů pro jednotlivce (pacienta) za účelem sledování každoročního finančního čerpání.

### Informační systém sportovních aktivit

Cílem informačního systému je evidence zápasů sportovních aktivit, evidence hráčů, trenéra(ů), zpřístupnění video(foto) záznamů ze zápasů, předávání informací hráčům, evidence tréninků.

Tematický celek „Evidence zápasů“ umožňuje hráčům (ostatním) prohlížet výsledky zápasů, prohlížet video (foto) záznamy z již proběhnutých zápasů. Informace o zápasech, které potřebujeme evidovat, jsou: datum a místo zápasu, seznam nominovaných hráčů z týmu, pokyny k zápasu (textový popis). Trenér má právo zápasy evidovat, rušit, zadávat, nominovat hráče, případně hráče odvolat ze zápasu a nominovat jiného. Pro již proběhnuté zápasy nelze měnit informace o zápasech, ale pouze zobrazovat informace, záznamy. Zápasy (všechny informace, záznamy, data) se automaticky archivují po uplynutí dvou měsíců tak, že již nejsou zobrazovány v hlavní části informačního systému, ale jsou dostupné jako zkomprimované soubory v sekci Archiv. Nominace hráče je možná pouze za předpokladu, že hráč má odehrané minimálně 2 sezóny z 80 procentní účasti na tréninku celé sezóny. Další nominace hráče je omezena jeho účastí na tréninku minimálně 75 procent z předchozího měsíce před zápasem. Hráči mohou stahovat soubory (foto/video) bez omezení. Ostatní mohou soubory pouze prohlížet v rámci zobrazovacího rozhraní informačního systému.

Tematický celek „Evidence hráčů“ je přístupná všem pro prohlížení základních údajů (citlivá data jsou přístupná pouze trenérovi a samotným hráčům pouze jejich vlastní údaje). Hráči mají přístupná data o své osobě, číslo dresu, odehrané sezóny, odehraný počet zápasů, procentuální návštěvnost tréninků. Trenér bude evidovat nábor hráče, přiděluje jeho číslo, eviduje kartu hráče, eviduje neúčasti hráčů na tréninku (zaznamenaná pouze ty, kteří chybí).

Tematický celek „Evidence trenéra“ je přístupná pouze správci systému, který mu po jeho registraci přidělí patřičná oprávnění. V případě odstoupení trenéra z funkce, pak automaticky správce systému zruší jeho oprávnění a zůstávají mu stejná práva jako ostatním osobám (ne hráč, ne trenér).

Tematický celek „Aktuality“ je naplňován pouze trenérem, ostatní a hráči mohou pouze nahlížet. Pokud chtějí ostatní nebo hráči sdělovat informace v rámci tohoto informačního systému, je jim určena část Diskuze (Chat).

Tematický celek „Evidence tréninků“ je průběžně naplňován trenérem, který označí ze seznamu aktuálních hráčů v týmu ty hráče, kteří nebyli přítomni a ti jsou zaznamenáni v systému. Hráčům je pouze přístupná tabulka procentuálních účastí za jednotlivé měsíce v aktuální sezóně. Jednotlivé sezóny jsou editovatelné trenérem dle daného ročního období, svátků apod. Uvažují se dvě sezóny letní (období dubna až konce října vyjma období 6 týdnů dovolené) a halová (začátek listopadu až konec března vyjma vánočních svátků). Vzhledem k výše uvedenému hodnocení je tedy nutné uchovávat informace minimálně o aktuální a dvou předchozích sezónách, pak informace archivovat.

Tematický celek „Diskuze“ je určen k diskusi registrovaných hráčů a dalších registrovaných osob (bez registrace nemůže přispívat do diskuze). Registrace ostatních osob uchovává informace o jméně, příjmení, email, registrační jméno uživatele a heslo, datum registrace. Správce systému, trenér má možnost nevhodné příspěvky z Diskuze odstranit. Mají také možnost u registrované osoby zvolit možnost „nežádoucí“ pro opakované nevhodné vyjadřování. Registrovaná osoba má možnost změnit své údaje a heslo jedině tehdy, pokud nebyla označena jako nežádoucí.

## Informační systém pro administraci stravování

Informační systém je určen k registraci receptů (jidel), jejich položek, cen, množství. Také k administraci vybraných receptů k daným dnům. Statistickému zpracování údajů o množství použitých materiálů za poslední měsíc, statistickému zpracování ceny. Je možné také zahrnout tabulku kalorií pro seznam základních potravin do číselníku a omezit tak výběr potravin pouze z evidovaných názvů.

Tematický celek „Evidence receptů“ je určen k sestavení nového receptu ze seznamu položek, evidovat množství potravin, evidovat počet porcí z připravovaného receptu, případně dobu přípravy.

Název receptu musí být v informačním systému jedinečný.

Tematický celek „Sestavení jídelníčku“ umožňuje výběr jídla ze seznamu receptů, ale omezení výběru receptu je takové, že se nemůže objevit v jídelníčku dříve než za 2 měsíce. Pro sestavení jídelníčku by bylo možné omezit také jídla obsahující maso nebo vyšší částku tak, že nebudou v seznamu nabízených jídel na další plánované období. Pokud recept není v seznamu a uživatel jej chce přidat, pak to řeší dále „Evidence receptů“. Při plánování jídelníčku si osoba zvolí recept a počet porcí, pak se v receptu zobrazí přepočítané množství položek receptu. K danému dni se zpravidla eviduje cena/porci.

Tematický celek „Statistika jídelníčku“ eviduje možnost zobrazení jídelníčku za měsíc (týden) tak, že se zobrazí ceny jídel (1porce)/den, celková cena jídel /měsíc, průměrná cena jídla/měsíc.

Dále se pod statistikou zahrnuje zobrazování statistiky kalorií. Opět za dílčí dny v měsíci, celkově počet kalorií za týden (pro porovnání týdnů mezi sebou navzájem ve vybraném měsíci), za měsíc (pro porovnání kalorií mezi sebou za kalendářní rok).

Informační systém spravuje administrátor systému receptů. Má veškerá oprávnění. Dalším uživatelem bude Strávník (může prohlížet jídelníček, přidávat recepty, zobrazovat statistiky). Uživatelů typu Strávník může být více. Uživatel Plánovač (má stejné oprávnění jako Strávník, dále má oprávnění k sestavování jídelníčku). Předpokládaný počet těchto uživatelů 2. Posledním typem uživatele systému je Zvědavý (chce se inspirovat recepty, takže si může prohlížet pouze je, nemůže zobrazovat žádné informace o jídelníčku ani statistikách). Oprávnění k rozšiřování číselníku potravin má pouze uživatel typu Plánovač a Administrátor. Uživatel Administrátor má oprávnění k evidenci registrace uživatelů (zadat nového uživatele, změnit heslo, odstranit uživatele).

Oprávnění k odstranění receptů má uživatel Plánovač a Strávník a to pouze tehdy, když recept nebyl použit za posledních 6 měsíců v jídelníčku.

## Informační systém osobní agendy ve zdravotnictví

Informační systém (IS) je založen na osobní administraci návštěv zdravotnického zařízení a popisu konkrétních úkonů a plateb v hotovosti. K dané návštěvě také evidujeme získaný recept, lék, množství, cenu, informaci, zda byl recept využit nebo ne (někdy lék bez receptu je levnější než získaný s výdejem na recept).

Informační systém je určen několika klientům, kteří mohou vybírat ze seznamu zdravotnických zařízení, které byly již registrovány. Pokud nejsou registrovány, pak probíhá registrace pouze pověřenými klienty, kteří mají oprávnění k registraci nových zdravotnických zařízení, soukromých zdravotních ošetřujících lékařů apod.

Tematický celek „Návštěva ZZ“ je určena k evidenci pouze vlastních záznamů klienta ze seznamu zdravotnických zařízení. Klient eviduje datum, lékaře (zařízení), informaci o účelu návštěvy (prohlídka, preventivní prohlídka, odběr krve, žádost o recept, úraz, plánovaný pobyt, plánované vyšetření, apod.), čekací doba, doba průběhu vyšetření, informace o receptu, další doplňkové informace.

Posléze k jedné informaci o návštěvě zdravotního zařízení (lékaře) bude po delší době možné doplnit informace z aktuálního ocenění a zhodnocení zdravotní pojišťovny.

Klient průběžně doplňuje další informace a starší údaje se archivují vždy ke zvolenému datu (období) v roce automaticky (po souhlasu klienta IS).

Tematický celek „Registrace klienta“ eviduje osobní údaje klienta. Je určen pouze pro interní nahlížení daného uživatele. Správce systému nemá možnost odstranit data, která nejsou starší 5 let. Archivace dat probíhá se souhlasem uživatele a po delší době než 3 roky. Eviduje také údaje o své zdravotní pojišťovně (název, číslo, sídlo pojišťovny). Sídlo pojišťovny eviduje každý klient sám, pokud dle bydliště má jinou bližší pobočku již evidované pojišťovny, pak přidá pouze adresu své pobočky.

Tematický celek „Evidence lékaře“ je určen k registraci lékaře (zdravotního zařízení). Přidávat nové údaje mohou uživatelé (klienti) sami, ale odstranit údaje může pouze ta osoba, která registraci údajů provedla a to za podmínky, že údaje nevyužívá k administraci jiný klient.

Tematický celek „Správa systému“ zajišťuje standardní správu přihlášení a odhlášení klientů, změnu registračních údajů správcem systému, změnu hesla apod. Správce systému může nahlížet na evidenci přihlašování a odhlašování klientů. Záznamy starší 10 let může kdykoliv ze své aktivity odstranit. Může také odstranit klienta a veškeré jeho záznamy, pokud doba jeho přihlášení překračuje dobu 5 let.

### 1.3 Role týmu a jejich popis



#### Výklad a popis případové studie

Tým je složen z osob, které spolupracují na tvorbě informačního systému. Návrh informačního systému je odborně rozsáhlý. Zasahuje několik oblastí, jejichž odborné zvládnutí na úrovni dobře navrženého systému vyžaduje velké úsilí ke zvládnutí nových odborných termínů, názvů, metodických postupů a zkušeností. Pro návrh skutečně dobře promyšleného a plně funkčního informačního systému je důležité, aby dílčí části a jejich zpracování bylo rozděleno mezi dílčí osoby v týmové spolupráci, tj. jedná se o přidělení rolí v oblasti návrhu a zpracování informačního systému. Dílčím úkolům se pak osoba věnuje intenzivněji a za stejný čas dokáže pochopit a zvládnout svůj úkol lépe, než kdyby se pokusila řešit větší část navrhovaného systému.

Obecně se v procesu vývoje informačního systému dělí činnost jeho návrhu do těchto **základních rolí v projektovém týmu:**

### **Systémový analytik**

Systémový analytik koordinuje požadavky tak, aby splňovaly požadavky na funkčnost systému a vymezení jeho hranic. Z požadavků modeluje systém v podobě aktérů, případů užití a jejich vazeb. Velkým předpokladem pro funkci systémového analytika je jeho nadprůměrná komunikační dovednost, která vede k správně sestavenému celku požadavků na navržený systém. Neméně důležité jsou také jeho analytické schopnosti nově nastupujících řešených problémů, které dosud analyzovány nebyly.

### **Softwarový architekt**

Softwarový architekt mapuje funkční požadavky sestavené systémovým analytikem do případů užití, které dále specifikuje sestavením jejich scénářů, případně sekvenčních/komunikačních diagramů. Softwarový architekt má velmi dobré znalosti o implementačním prostředí (software), včetně znalosti objektově orientované analýzy a podrobné znalosti návrhu informačního systému s využitím technologie UML.

### **Softwarový analytik**

Následnou rolí v návrhu IS je mapování případů užití do modelů tříd nebo datových modelů (dle implementačního prostředí), sestavuje statický pohled na systém diagramem nasazení a diagramem komponent. Softwarový architekt se specializuje na danou oblast software, spolupracuje se softwarovým architektem, upřesňuje návrhy objektů a jejich vazby s využitím technologie UML. Má schopnosti reverzní i dopředné tvorby návrhů IS.

### **Programátor**

Programátor vytváří části softwarové aplikace nebo jejich kombinací staví větší celky, provádí jejich testování a ladění. Jeho úlohou je také tvorba dokumentace, datových a objektových struktur a definice jejich vazeb. Na základě analytické dokumentace programuje v příslušném programovacím jazyce. Podle uživatelských připomínek a návrhu uživatelského rozhraní spolupracuje při vytváření uživatelského rozhraní. Po úspěšném otestování a doladění produktu připravuje podklady pro uživatelskou dokumentaci.

### **Softwarový vývojář**

Softwarový vývojář tvoří softwarové aplikace pro různé účely. Uživatelský software pro hromadné použití, specificky zaměřený software, vestavěný software implementovaný do specializovaných elektronických zařízení. Jeho role spočívá v tvorbě programového kódu, testování jeho funkčnosti, jeho instalaci a integraci, optimalizaci zdrojového kódu, aktualizaci verzí a zajištění technické podpory pro uživatele. Na rozdíl od programátora je vývojář zapojen také do důležité, přípravné fáze při návrhu koncepce a struktury software. Spolupracuje se softwarovým analytikem a programátory.

## Databázový analytik

Databázový analytik navrhuje architekturu databáze a vyvíjí řídicí systém pro správu databáze. Na základě požadavků od uživatelů a konzultací s ostatními IT specialisty, definuje typy informací, jejich organizaci, hierarchii přístupových práv, způsob zobrazování dat. Ve srovnání s databázovým administrátorem, analytik se soustředí na tvorbu struktury databázového systému, administrátor především definuje a zabezpečuje provoz databáze. Podle typu organizace se náplně obou rolí mohou překrývat nebo naopak mohou vytvořit jednu roli. Databázový analytik je speciální rolí softwarového analytika zaměřeného na databázové systémy.

## Databázový specialista

Databázový specialista je všeobecně definovaná pozice se zaměřením na technickou podporu pro databázové systémy. Role databázového specialisty může zahrnovat také činnost databázového administrátora, databázového analytika nebo databázového programátora podle pravomocí jemu svěřených nebo je možné tuto roli rozdělit na další specifické role. Jeho činnost je dána instalací a správou databáze včetně správy uživatelů a jejich práv, analýzou a návrhy datových modelů, jeho optimalizací, tvorbou dotazů, uložených procedur, procedurou testování až po dokumentaci databáze a zaškolení uživatelů.

## Tester

Testování je stejně důležité jako vývoj nových programů. Tester navrhuje, provádí, dokumentuje a vyhodnocuje výsledky testování. Testování je rozsáhlou činností a vzhledem k nezávislosti dílčích činností se rozdělují role testera na nezávislé a neovlivňující se role: manažer testování, analytik testování, návrhář testů a realizátor testů. Testování má za cíl odhalit včas problémy nebo chyby v návrhu informačního systému v průběhu celého procesu vývoje, proto se provádí na konci každé iterace dílčí fáze vývoje.

Testování se tedy provádí po každé iteraci. Testují se nejen funkce navrhovaného systému, ale také design (přehlednost, jednoduchost, rozšiřitelnost, funkčnost). Cílem testování je získat co nejvyšší procento pokrytí a pokud možné s minimální sadou testovacích případů. Kromě funkčních celků IS do testování patří také testování systému z pohledu testování maximálního výkonu IS, jeho zátěže a odolnosti systému, tzv. zátěžové testování a testování výkonu.



### Rozdělení rolí v týmu

Pro danou skupinu 11 studentů byly zadány tři tematické okruhy navrhovaných a realizovaných informačních systémů, které jsou podrobně popsány v předchozí kapitole 1.2. Rozdělení rolí v projektovém týmu je tedy dáno 3 až 4 osobami spolupracujícími na vývoji navrhovaného informačního systému. Vzhledem k požadavkům na konkrétní realizované části a jejich časovou náročnost na zpracování byly role projektového týmu navrženy pro každý tematický celek takto:



- ✚ **Systémový analytik** – zodpovídá za činnost sběru funkčních a nefunkčních požadavků, vymezení hranic navrhovaného systému, návrhu případů užití.
- ✚ **Softwarový architekt** – navrhuje scénáře případů užití, které rozšiřuje o sekvenční diagramy a případně také diagramy aktivit. Spolupracuje se systémovým analytikem.
- ✚ **Softwarový analytik** – analyzuje strukturu objektů v navrhovaném systému a následně navrhuje datový model, z něhož vychází databázový specialista.
- ✚ **Databázový specialista** – realizuje podle datového modelu strukturu objektů v databázovém prostředí, spravuje objekty dle standardů databází, navrhuje výstupní struktury dotazů podle požadavků na webové rozhraní. Návrh webového rozhraní není součástí navrhovaného IS v projektu.

Funkce odběratele a zároveň projektového manažera je určena osobě vedoucí výuku předmětu Informační systémy. Je nutné podotknout, že k diskuzi a spolupráci nad projektem a řízením několika skupin osob není v průběhu zpracování dílčích činností dostatek času. Jsou k dispozici pouze 2 hodiny týdně a to je velmi málo na to, aby projekty byly navrženy dle požadavků odběratele, na potřebnou spolupráci v projektovém týmu a na průběžnou koordinaci činností a průběžnou kontrolu návrhu projektu po celou dobu vývoje procesu.

Cílem zapojení studentů do těchto rolí bude podrobnější seznámení s konkrétní rolí a činností za kterou nese zodpovědnost, diskuze v projektovém týmu za účelem návaznosti návrhu IS a činnost v neznámém prostředí pro analýzu, konkrétně produktu Enterprise Architect a v případě databázového specialisty v prostředí Microsoft SQL Serveru.



## Shrnutí objektového přístupu při návrhu IS

Proces vývoje informačního systému zahrnuje předepsané fáze, které se v průběhu času postupně uplatňují. Fáze vývoje informačního systému, v němž bude probíhat časový vývoj, se nazývají Zahájení, Rozpracování, Vývoj a Nasazení.

Definovali jsme rozdělení osob do tří týmových projektů, které se budou zabývat návrhy IS pro sportovní aktivity, správu receptů a osobní agendu ve zdravotnictví.

Dále byly obecně definovány role v projektovém týmu, který zpracovává návrh informačního systému, z nichž, jsme vybrali čtyři role, z nichž se naše konkrétní tři projektové týmy budou skládat. Jedná se o role systémového analytika, softwarového architekta, softwarového analytika a databázového specialisty.

Poslední role projektového manažera a také odběratele navrhovaného informačního systému byla přidělena osobě vedoucí výuku předmětu Informační systémy, v němž probíhá vývoj informačních systémů.

Dílčí části informačního systému budou zpracovány v produktu Enterprise Architect, případně MS SQL Server pro část návrhu databázového specialisty.

## 2 ZPRACOVÁNÍ POŽADAVKŮ INFORMAČNÍHO SYSTÉMU



### Plánování jednotlivých kroků: 2 hodiny

Vysvětlení sběru požadavků a jejich zařazení do částí informačního systému, tvorba seznamu funkčních a nefunkčních požadavků ve vybraném grafickém nástroji UML.



### Cíl:

- ✚ Definovat pojmy: funkční a nefunkční požadavky
- ✚ Předložit řešení sběru požadavků dílčích projektových týmů
- ✚ Zhodnotit sestavení požadavků vzhledem k představám odběratele a zadání modelovaného systému



### Výklad a popis případové studie

První fáze návrhu informačního systému spočívá v sestavení požadavků na informační systém. Činnost je určena a zpracována systémovým analytikem. Podíváme se na konkrétní zpracování požadavků na systém u všech navrhovaných IS. Zkonzultujeme zadání předané odběratelem a dodržení všech požadavků systémovým analytikem. Mimo to je součástí role systémového analytika zpřesnit původní navržené požadavky odběratele tak, aby byly zajištěny všechny základní funkce navrženého informačního systému. Odběratel nemá předpoklady k tomu, aby byl schopen ve svých požadavcích definovat vše tak, jak vyžaduje realizace objektově řešeného, intuitivně funkčního a plně zabezpečeného informačního systému. Zde je nutná komunikace mezi odběratelem a systémovým analytikem.

### 2.1 Definice funkčních a nefunkčních požadavků

Projekt informačního systému od analýzy až po implementaci musí splňovat základní předpoklady návaznosti dílčích částí projektu, proto je nezbytné správně dokumentovat každou část projektu. V první fázi analýzy a návrhu se jedná o sběr funkčních a nefunkčních požadavků, které definují funkci systému (funkční požadavky) a jeho případná omezení a požadavky na zajištění výkonu, rozšiřitelnosti, integrovanosti a bezporuchovosti realizace projektu (nefunkční požadavky).

Požadavky by měly splňovat základní syntaktické normativy pro jejich jednoznačnost a návaznost na další realizované části projektu. Syntaxe funkčních a nefunkčních požadavků je velmi podobná, významně se odlišují ve smyslu a účelu jejich definice, které jsou vždy stejné uvnitř odkazu výrazem „**bude**“ (funkční požadavek) a „**vyžaduje/vylučuje**“ (nefunkční požadavek). Funkční požadavky jsou zpravidla označeny prioritou jejich plnění v další části návrhu IS. V případě, že všechny požadavky splňují nejvyšší prioritu, tj. požadavek musí být splněn, pak není nutné prioritu v seznamu funkčních požadavků uvádět.

### *Syntaxe funkčních požadavků*

<id> <system> **bude** <funkce> prioritizace požadavků

kde jednotlivé části syntaxe označují:

<id> - číslování funkčního požadavku,

<system> - osoba, věc nebo událost vyvolávající funkci systému,

<funkce> - požadovaná činnost modelovaného systému.

Priorita požadavků v syntaxi funkčního požadavku je dána mírou nezbytnosti splnění funkce daného požadavku a je označována takto:

M – (must), požadavek musí být splněn,

S – (should), požadavek musí být splněn, pokud je to vůbec možné,

C – (could), požadavek může být splněn, pokud to nemá vliv na další požadavky,

W – (won't), požadavek nyní splněn nebude, ale v budoucím čase bude požadován.

### *Syntaxe nefunkčních požadavků*

<id> <předmět > **vyžaduje / vylučuje** <podmínka >

kde jednotlivé části syntaxe označují:

<id> - číslování nefunkčního požadavku,

<předmět> - osoba, věc nebo událost vyžadující nebo vylučující okolnosti, za nichž by měl být IS provozován,

<podmínka> - definuje okolnosti provozu IS, které by měly být splněny.

Sběr funkčních a nefunkčních požadavků zajišťuje podklady ke konfrontaci plnění úkolů při realizaci IS mezi odběratelem a projektovým týmem. Další rozšíření možností požadavků na IS je pak otázkou navýšení ceny za objem vykonané práce. Sběr požadavků je činnost systémového analytika a je podkladem k vyjednávání projektového manažera a odběratele, mimo jiné také podklad pro další navazující činnost, např. softwarového architekta.

## **2.2 Sběr požadavků pro dílčí tematické celky**

Výsledky sběru požadavků pro jednotlivé informační systémy byly zpracovány v produktu Enterprise Architect, verze 8 firmy Sparx Systems Pty Ltd., který byl vybrán z několika možných softwarových produktů pro podporu modelování z důvodu ceny, rozsahu nabízených služeb, podpory implementačních softwarových produktů a jednoduchého a přehledného grafického prostředí.

V rámci sběru požadavků navrhnul systémový analytik také rozdělení modelovaného systému do souvisejících funkčních celků z důvodu přehlednosti a eventuálně i rozdělení činnosti na více rolí stejného typu.

V následujících podkapitolách jsou stručně uvedeny výsledky sběru požadavků pro jednotlivé informační systémy v podobě zpracování produktem Enterprise Architect.

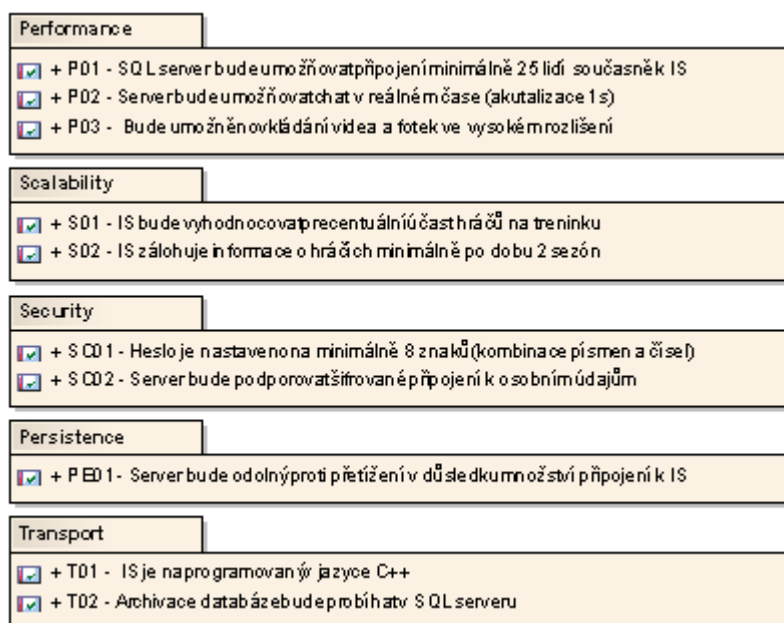
## 2.2.1 Informační systém sportovních aktivit

<b>Aktuality</b>
<input checked="" type="checkbox"/> + A01 - Trenér bude přidávat aktuality <input checked="" type="checkbox"/> + A02 - Trenér bude mazat aktuality
<b>Diskuze</b>
<input checked="" type="checkbox"/> + D01 - Správce systému a trenér bude odstraňovat příspěvky <input checked="" type="checkbox"/> + D02 - Všichni budou přidávat příspěvky <input checked="" type="checkbox"/> + D03 - Správce systému, trenér a hráči budou zakládat nové diskuze <input checked="" type="checkbox"/> + D04 - Správce systému a trenér budou mazat diskuze
<b>Evidence hráčů</b>
<input checked="" type="checkbox"/> + EH01 - Všichni budou prohlížet základní údaje <input checked="" type="checkbox"/> + EH02 - Trenér a hráči budou prohlížet osobní (citlivé) údaje <input checked="" type="checkbox"/> + EH03 - Trenér bude přidělovat čísla hráčů <input checked="" type="checkbox"/> + EH04 - Trenér bude evidovat karty hráče <input checked="" type="checkbox"/> + EH05 - Trenér bude evidovat neúčasti hráčů na tréninku (pouze ti co chybí) <input checked="" type="checkbox"/> + EH06 - Trenér bude evidovat nábor hráčů
<b>Evidence trenéra</b>
<input checked="" type="checkbox"/> + ET01 - Správce systému bude udělovat trenérské oprávnění <input checked="" type="checkbox"/> + ET02 - Správce systému bude odebírat trenérské oprávnění
<b>Evidence tréninků</b>
<input checked="" type="checkbox"/> + ET01 - Trenér bude evidovat nepřítomnosti na tréninku <input checked="" type="checkbox"/> + ET02 - Letní sezóna <input checked="" type="checkbox"/> + ET03 - Zimní (halová) sezóna <input checked="" type="checkbox"/> + ET04 - Systém bude archivovat informace o sezonách <input checked="" type="checkbox"/> + ET05 - Systém bude vypočítávat procentuální účast hráčů z tréninků <input checked="" type="checkbox"/> + ET06 - Hráči můžou prohlížet procentuální účast
<b>Evidence zápasů</b>
<input checked="" type="checkbox"/> + EZ01 - Všichni budou prohlížet výsledky zápasů <input checked="" type="checkbox"/> + EZ02 - Všichni budou prohlížet video/foto <input checked="" type="checkbox"/> + EZ03 - Trenér bude vkládat informace o budoucích zápasech <input checked="" type="checkbox"/> + EZ04 - Všichni budou číst v archivech <input checked="" type="checkbox"/> + EZ05 - Trenér bude editovat výsledků zápasů <input checked="" type="checkbox"/> + EZ06 - Trenér bude editovat video/foto <input checked="" type="checkbox"/> + EZ07 - Trenér bude nominovat hráče
<b>Správa systému</b>
<input checked="" type="checkbox"/> + SS01 - Všichni se budou moct registrovat <input checked="" type="checkbox"/> + SS02 - Registrovaní uživatelé se budou přihlašovat k IS <input checked="" type="checkbox"/> + SS03 - Registrovaní uživatelé se budou odhlašovat z IS <input checked="" type="checkbox"/> + SS04 - Registrovaní uživatelé budou moct měnit své údaje <input checked="" type="checkbox"/> + SS05 - Správce systému a trenér bude označovat "nežádoucí" uživatele <input checked="" type="checkbox"/> + SS06 - Registrovaní uživatelé budou moct měnit své heslo

Obrázek 2.1 – Funkční požadavky IS sportovních aktivit

Funkční požadavky sportovních aktivit (Obrázek 2.1) jsou rozděleny do balíčků: Aktuality, Diskuze, Evidence hráčů, Evidence trenéra, Evidence zápasů, Správa systémů. Syntaxe a význam požadavků označených ET02 a ET03 není správně definována a tudíž není jasný význam požadavku. Nahrazení výrazu „můžou“ striktně předepsaným výrazem „bude“ ve funkčním požadavku ET06 není přesně dán význam činnosti. Pokud existuje nějaká

podmínka, kdy „můžou“ a kdy „nemůžou“ prohlížet procentuální účast, pak není definována. Požadavek není přesně formulován. V několika požadavcích se v části <system> několikrát objevuje výraz „všichni“, přičemž je to neurčitý výraz, není nikde blíže určen. Takové požadavky nelze dále přesně analyzovat v dalších částech projektu.

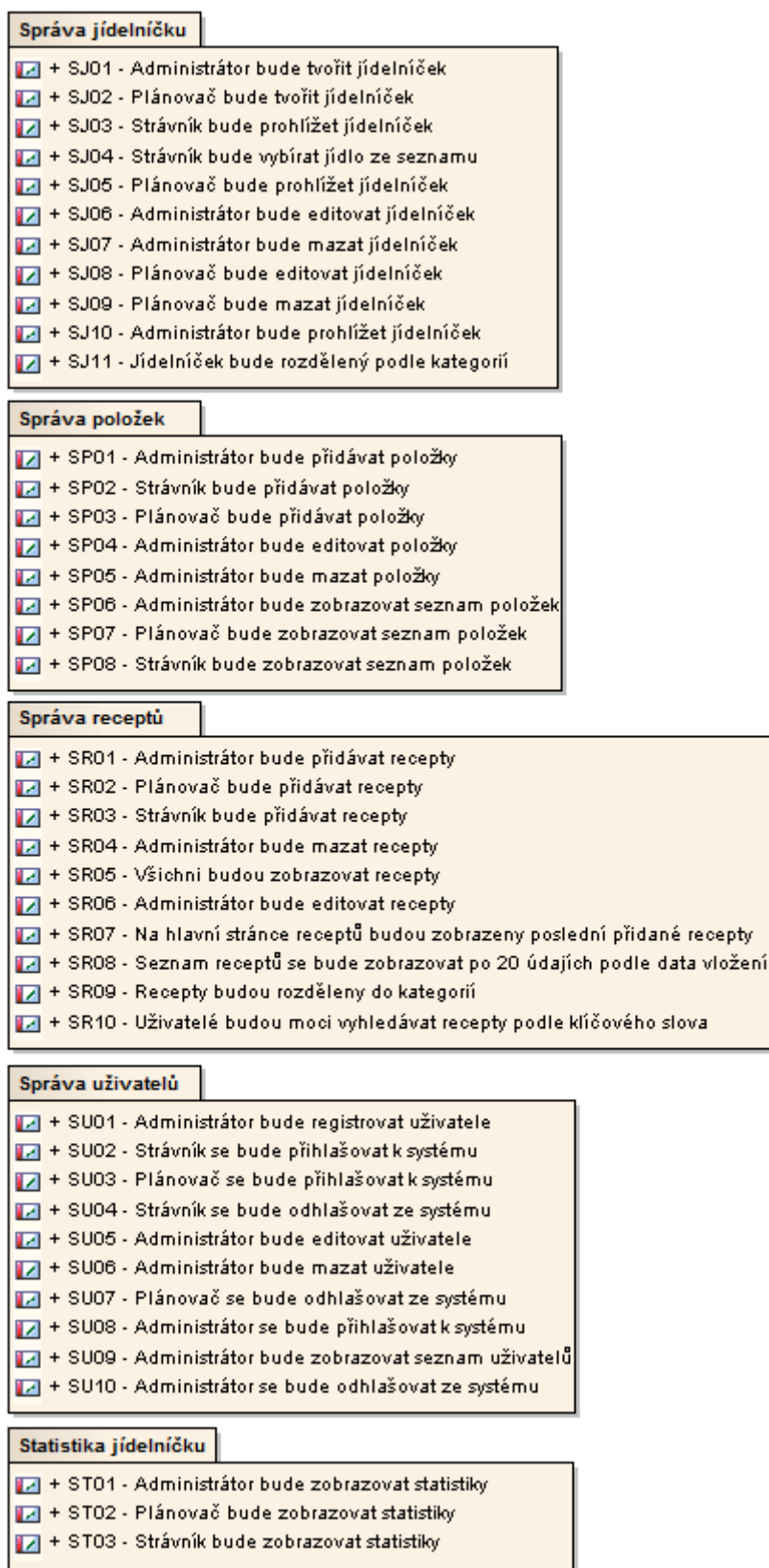


Obrázek 2.2 – Nefunkční požadavky IS sportovních aktivit

Nefunkční požadavky sportovních aktivit (Obrázek 2.2) jsou rozděleny do standardních částí nabízených produktem Enterprise Architect a jejich syntaxe striktně nesplňuje formální řazení částí a výrazů v syntaxi dle standardů UML. Nefunkční požadavek P03 nemá vyznačen <předmět>, tudíž není přesně definován jeho význam. Požadavek T01 a T02 je zařazen do části nefunkčních požadavků s označením Transport, která je určena k definování nefunkčních požadavků týkajících se přenosu informací po síti, patří zde požadavky na kvalitu přenosu informací mezi uzly, informace o počítačových sítích a jejich protokolech.

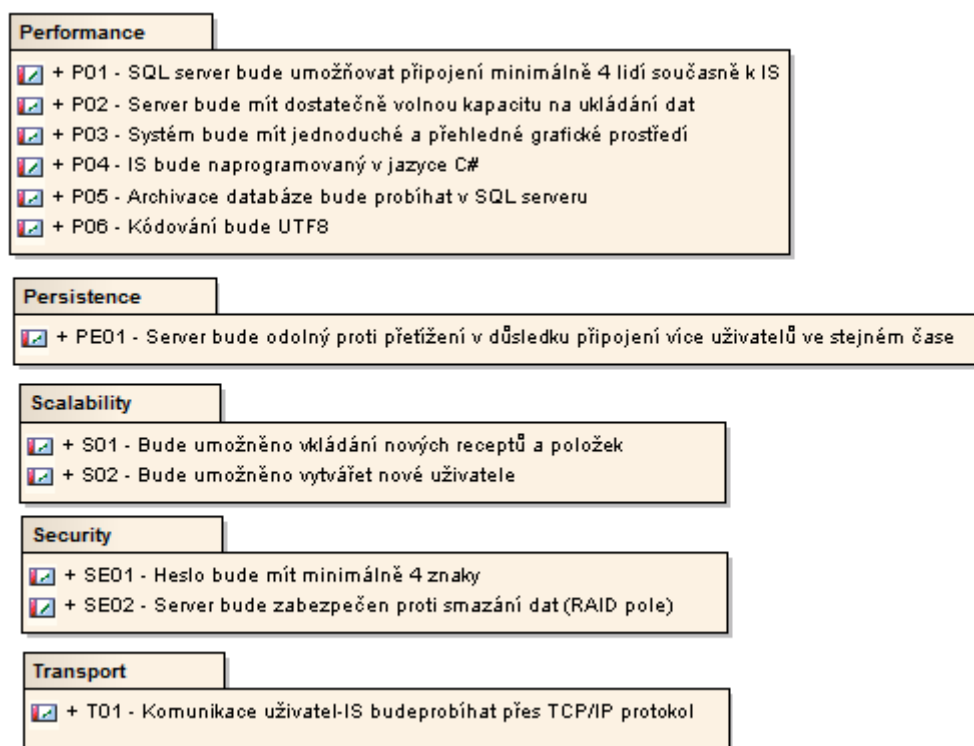
### 2.2.2 Informační systém pro administraci stravování

Funkční požadavky modelovaného systému pro administraci stravování (Obrázek 2.3) rozděluje své činnosti do balíčků: Správa jídelníčku, Správa položek, Správa uživatelů a Statistika jídelníčku. Syntaxe všech funkčních požadavků tohoto informačního systému je definována správně, z popisu jednotlivých funkčních požadavků nejsou zřejmé žádné nepřesnosti a nejasnosti, které by mohly způsobit chyby v následné analýze. Další podrobné zhodnocení obsahu činností systému daných těmito požadavky vůči požadavkům odběratele bude zhodnoceno v následující kapitole.



Obrázek 2.3 – Funkční požadavky IS administrace stravování

Nefunkční požadavky IS pro administraci stravování (Obrázek 2.4) dodržuje strukturu nefunkčních požadavků navržených programem Enterprise Architect. Požadavky S01 a S02 nepatří do části označené Scalability, která je určena nefunkčním požadavkům definujícím předpokládanou velikost paměti pro navrhovaný systém, maximální počet uživatelů systému, kapacitě dat pro uložení např. receptů, položek, fotografií apod. Nefunkční požadavek P06 není zcela přesně popsán, v požadavcích je možné doplnit detaily, které nejsou viditelné v názvu požadavku, a tím není jeho popis tak dlouhý. Systémový analytik nevyužil možnosti vložení dalších poznámek či omezení k definovaným požadavkům na modelovaný systém, proto není jeho analýza striktně definující své požadavky. Mohly by vzniknout při dalším návrhu systému chyby.



Obrázek 2.4 – Funkční požadavky IS administrace stravování

### 2.2.3 Informační systém osobní agendy ve zdravotnictví

Funkční a nefunkční požadavky pro informační systém osobní agendy ve zdravotnictví nebyly sestaveny. Systémový analytik odstoupil z osobních důvodů z tvorby projektu a náhradní osoby na funkce v projektovém týmu nebyly, proto se informační systém osobní agendy navrhoval bez seznamu požadavků. Fáze sběru požadavků nebude hodnocena v případové studii pro tento informační systém.

## 2.3 Zhodnocení sběru požadavků

Návrhy funkčních a nefunkčních požadavků a týmová spolupráce, v tomto případě spolupráce odběratele a systémového analytika bude zhodnoceno v této kapitole pro jednotlivé informační systémy. Sběr funkčních požadavků systémovým analytikem je založeno převážně na dobré komunikativnosti s odběratelem a jeho analytickým schopnostem,

což v případě správného postupu zajistí zahrnutí všech představ odběratele na modelovaný systém. Mimo jiné je systémový analytik také zodpovědný za analýzu takového informačního systému, který splňuje předpoklady snadno rozšiřitelného a udržitelného systému.

### 2.3.1 Informační systém sportovních aktivit

Seznam funkčních požadavků pro IS sportovních aktivit (Obrázek 2.1) je podkladem k dalšímu rozvoji modelovaného systému. Bohužel analýza požadavků nezahrnuje všechny požadované činnosti nebo omezení zadané odběratelem (kapitola 1.2).

Shrneme chyby a nedostatky sběru požadavků systémového analytika neboli nedostatečnou komunikativnost s odběratelem systému, která vážně ohrožuje funkčnost realizovaného informačního systému již ve fázi analýzy a návrhu IS:

- ✚ Chybí uživatel systému, který nepatří mezi navrhované aktéry: hráč, trenér, správce systému ani rodičovský aktér s označením „Ostatní“, který má v určitých situacích práva všech jeho potomků (hráč, trenér, správce systému). Dle zadání odběratele má existovat takový uživatel systému, který nemá oprávnění ani jednoho z uživatelů hráč, trenér, správce systému, ale je pouhým hostem IS, který má nejnižší úroveň oprávnění.
- ✚ V části Evidence zápasů nejsou zahrnuty požadavky:
  - Trenér bude měnit informace o budoucích zápasech.
  - Trenér bude odstraňovat informace o budoucích zápasech.
  - Trenér bude nominovat hráče do zápasu (dle podmínek o procentuální účasti na tréninku).
  - Trenér bude odvolávat hráče ze zápasu (porušení dohodnutých pravidel o procentuální účasti na tréninku).
  - Hráči budou stahovat soubory video/foto bez omezení.
  - Trenér bude vkládat video/foto.
  - Trenér bude odstraňovat video/foto.
- ✚ V části Evidence hráčů nejsou zahrnuty požadavky:
  - Hráč bude prohlížet své osobní údaje.
  - Trenér bude přidávat kartu hráče.
  - Trenér bude odstraňovat kartu hráče.
- ✚ V části Evidence trenéra nejsou zahrnuty požadavky:
  - Správce systému bude vkládat informace o trenérovi.
  - Správce systému bude editovat informace o trenérovi.
  - Správce systému bude odstraňovat informace o trenérovi.
- ✚ V části Aktuality nejsou zahrnuty požadavky:
  - Trenér bude editovat aktuality.
  - Ostatní uživatelé budou prohlížet aktuality.
  - Hráči budou prohlížet aktuality.



- ✚ V části Evidence tréninků nejsou zahrnuty požadavky:
  - Trenér vkládá informace o hrací sezóně.
  - Trenér edituje informace o hrací sezóně.
  - Trenér odstraňuje informace o hrací sezóně.
- ✚ V části Správa systému nejsou zahrnuty požadavky:
  - Systém bude vkládat informace o registrovaných osobách.

Procentuální úspěšnost systémového analytika při sběru funkčních požadavků:

- ✚ Aktuality – 40 %,
- ✚ Diskuze – 100 %,
- ✚ Evidence hráčů – 66 %,
- ✚ Evidence trenéra – 40 %,
- ✚ Evidence tréninků – 66 %,
- ✚ Evidence zápasů – 50 %,
- ✚ Správa systému – 86 %.

**Celkově je práce systémového analytika** v oblasti funkčních požadavků hodnocena **64 %**. Takové hodnocení je pro modelovaný systém nepostačující, v projektovém týmu by měla být větší komunikace mezi systémovým analytikem a odběratelem a také řízená a kontrolní činnost projektovým manažerem.

Nefunkční požadavky jsou sestaveny dle potřeb modelovaného informačního systému. Zde nebyly shledány žádné závažné nedostatky, mimo již výše komentovaných syntaktických chyb a nesprávného zařazení (kapitola 2.2.1).

### 2.3.2 Informační systém pro administraci stravování

Systémový analytik, který zpracoval sběr požadavků informačního systému pro administraci stravování, sestavil funkční požadavky s velkou shodou se všemi požadovanými činnostmi nebo omezením zadané odběratelem (kapitola 1.2). Kromě jediného funkčního chybějícího požadavku:

- ✚ Zvědavý bude prohlížet recepty.

Celkový přístup analytika je v sestavování požadavků je systematický, protože rozlišil standardní činnosti, které jsou běžné v modelech informačních systémů, tj. tvorbu, editaci, odstraňování dílčích částí IS, činnost prohlížení částí IS a management uživatelů a jejich práv k IS. Požadavky nejsou konkrétní, a tudíž neobsahují informace konkretizující a podmiňující některé z funkčních požadavků na IS. Např. odběratel požaduje podmínku, aby se recept v sestavovaném jídelníčku neobjevil dříve než za 2 měsíce. Systémový analytik ale při sběru požadavků o jídelníčku sestavil pouze tyto požadavky:

- Plánovač bude tvořit jídelníček.
- Plánovač bude prohlížet jídelníček.
- Plánovač bude editovat jídelníček.

- Plánovač bude odstraňovat jídelníček.

V těchto funkčních požadavcích nejsou konkretizovány žádné podrobnější informace, tudíž další část návrhu modelu informačního systému by nebyla správná. Celkově tedy můžeme zhodnotit úspěšnost sestavení požadavků, pokud se týká celkového obsahu činností, na 97 %. Ale pokud bude na sběr požadavků pohlíženo celkově, zda zahrnují všechny informace, které předat odběratel, tj.:

- Omezení výběru receptu do jídelníčku ne dřív než za 2 měsíce.
- Konkretizace zobrazování jídel bezmasých, sladkých, ... a v jiných kategoriích.
- Plánovač jídelníčku volí počet porcí.
- Systém zobrazí přepočítané množství položek receptu podle počtu porcí.
- Systém nabízí zobrazení jídelníčku s výběrem období (měsíc/14 dní/týden/den).
- Systém vyhodnotí počet kalorií za vybrané období na jednu porci.
- Administrátor bude evidovat maximálně 2 osoby s oprávněním Plánovač.
- Plánovač bude odstraňovat recept pouze tehdy, pokud nebyl použit za posledních 6 měsíců v jídelníčku.

Vzhledem k výše uvedeným osmi požadavkům, které nejsou zahrnuty do analýzy, se pak hodnocení obsažnosti požadavků snižuje na 80 %.

**Celkově je tedy možné hodnotit práci systémového analytika (Obrázek 2.3) vzhledem k požadavkům odběratele na činnost nebo omezení modelovaného systému (kapitola 1.2) na 78 %.**

Nefunkční požadavky jsou sestaveny dle potřeb modelovaného informačního systému. Zde nebyly shledány žádné závažné nedostatky, mimo již výše komentovaných syntaktických chyb a nesprávného zařazení (kapitola 2.2.2).



## Shrnutí zpracování požadavků informačních systémů

Práce systémového analytika informačního systému sportovních aktivit dosáhla celkem hodnocení přínosu 64 %.

Komunikativnost v týmu s odběratelem a projektovým manažerem byla malá, z čehož vyplývají jeho chyby při sběru požadavků. Taktéž kvalita práce systémového analytika z pohledu jeho kvalifikace na danou činnost není vysoká, systémový analytik není příliš systematický ve své práci, standardní činnosti IS nejsou zahrnuty do sběru požadavků, vyjadřování není přesné, tj. stejnou funkční činnost systému označuje různými názvy, rozhodně by takový systémový analytik nebyl přínosem pro projektový tým.

Práce systémového analytika informačního systému pro administraci stravování dosáhla celkem hodnocení přínosu 78 %.

Sběr požadavků zahrnoval téměř všechny požadované činnosti na IS podobného zaměření, ale systémový analytik do požadavků nedefinoval konkrétní omezení podrobně, takže by dál systém nebyl navržen bez chyb. Komunikace s odběratelem a ostatními členy projektového týmu byly dobré, ale projektový manažer by měl více kontrolovat konkrétní výstupy tak, aby bylo z požadavků striktně dáno chování IS. Zde je možné konstatovat chybnou komunikaci se softwarovým architektem, který by měl požadovat konkrétnější funkční požadavky, než které mu byly předány.

### 3 REALIZACE NÁVRHU INFORMAČNÍHO SYSTÉMU



#### Plánování jednotlivých kroků: 3 hodiny

Vysvětlení a popis důležité části návrhu informačního systému, kterou je modelování případů užití. Představení výsledků analýzy pro jednotlivé informační systémy, která využívá grafického nástroje Enterprise Architect a jejich zhodnocení.



#### Cíl:

- ✚ Definovat základní pojmy z oblasti modelování případů užití
- ✚ Předložit řešení případů užití dílčích projektových týmů
- ✚ Zhodnotit analýzu realizovanou diagramem případů užití vzhledem k práci systémového analytika



#### Výklad a popis případové studie

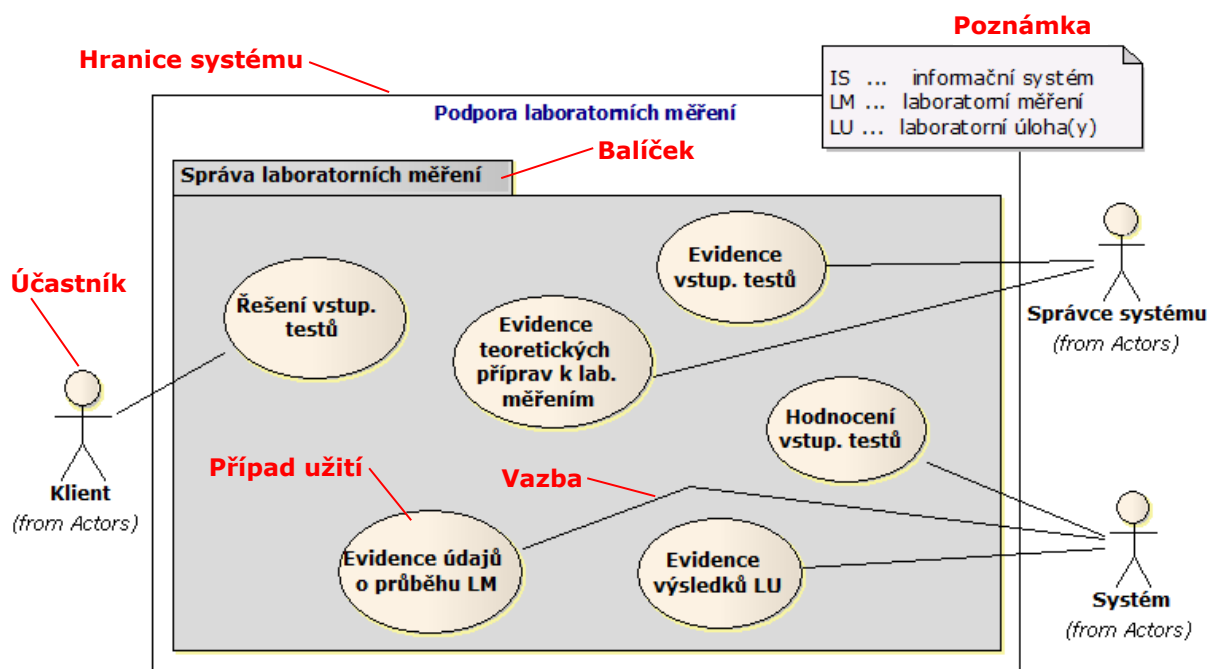
Součástí první fáze návrhu a analýzy informačního systému je také zpracování případů užití pro daný informační systém. Analýza je zpracována softwarovým architektem. Podíváme se na konkrétní zpracování případů užití pro navrhované a modelované IS. Výsledky analýzy budou konzultovány s předchozí činností systémového analytika (za předpokladu, že postupoval správně) za účelem zahrnutí všech navržených funkčních požadavků do logického pohledu na systém, který je diagram případů užití. Ve fázi analýzy je důležitá komunikace mezi systémovým analytikem a softwarovým architektem v projektovém týmu za přispění podpory projektového manažera.

#### 3.1 Modelování případů užití

Podrobnější formou inženýrství požadavků je oblast **modelování případů užití**, které rozšiřují informace o modelovaném systému tím, že se uskuteční:

- ✚ definice hranic modelovaného systému,
- ✚ analýza účastníků,
- ✚ analýza případů užití,
- ✚ logické uspořádání případů užití,

tvorba scénářů (doplňující informace o přesném postupu případu užití).



Obrázek 3.1 – Modelování případů užití v programu Enterprise Architect

### Definice hranic modelovaného systému

**Hranice modelovaného systému** se vizuálně zobrazují obdélníkem, kde uvnitř jsou dílčí případy užití a vně jsou účastníci (obrázek 3.1). Ve skutečnosti však hranice systému představují souhrn požadavků, které má modelovaný systém splňovat. Zde je nutné podotknout, že někdy může nevhodná definice hranic systému vyvolat konflikt mezi zadavatelem a analytikem a to tehdy, pokud hranice systému nezahrnují vše, co si zadavatel představoval.

### Analýza účastníků

Modelovaný systém má splnit požadavky, jejichž činnost se zpravidla aktivuje zvnějšku (za hranicemi systému) osobou nebo předmětem souhrnně nazývaným **účastník** (Actors) daného modelu případů užití. Účastník je role, kterou má uživatel ve vztahu k systému. Účastníkem může být člověk nebo stroj, ale také další systém nebo subsystém modelu. V jazyce UML je účastník označován symbolem kreslené figurky včetně jejího označení umístěného pod ní (obrázek 3.1).

Při modelování situace v systému je možné, že konkrétní činnost je aktivována nikoliv osobou nebo systémem, ale konkrétním časovým údajem, ve kterém se tato činnost provede. Pak je vhodné zavést abstraktního účastníka s názvem Čas.

### Analýza případů užití

**Případ užití** (Use Case) definuje posloupnost činností, které systém, podsystém nebo třída může vykonat prostřednictvím jejich aktivace účastníkem modelovaného systému. Symbolicky se případ užití obrazuje elipsou, která uvnitř obsahuje stručný popis daného případu užití (obrázek 3.1).

## Logické uspořádání případů užití

Rozsáhlé modelované systémy je vhodné uspořádat do logických celků, tzv. **balíčků** (Package). Analýzu systému lze distribuovat ke zpracování dílčích částí se specifickým názvem odpovídající dané problematice. Existence jednoho balíčku v diagramu případu užití nemá význam, viz obrázek 3.1, který je ukázkou prvků diagramu případu užití.

## Tvorba scénářů

Scénáře jsou nezbytnou součástí diagramů případů užití, zejména pro rozsáhlejší činnosti, kdy název případu užití nedefinuje plně cíl, postup a konkrétní komunikaci mezi dalšími účastníky. Sekvenční diagramy jsou v podstatě scénáře v grafické podobě zahrnující navíc vizualizaci následnosti jednotlivých kroků, podíl konkrétních tříd a instancí na dané činnosti případu užití, vznik nebo zánik konkrétních objektů apod. Pro jednoduché scénáře se nedoporučuje zbytečná tvorba sekvenčních diagramů, která by způsobila navýšení počtu diagramů v dané analýze.

Scénář případu užití specifikuje všechny navržené případy užití tím, že podrobně definuje:

- ✚ stav systému před spuštěním,
- ✚ konkrétní postup událostí po aktivaci daného případu užití účastníkem,
- ✚ stav systému po ukončení případu užití.

Syntaxe scénáře není pevně definovaná standardem UML, ale v podstatě ve všech publikacích jsou uvedeny scénáře ve formě seznamu (tabulky) viz tab. 3.1. Obecnou strukturu základního scénáře je vhodné sestavovat ve formě číslovaného seznamu.

Identifikátor jednoznačně určuje daný případ užití. Nijak číslování případů užití nenavazuje na číslování požadavků, protože jejich počet není stejný. Jeden funkční požadavek může být realizován jedním nebo i více případy užití a naopak.

Pokud nenastane chyba, která naruší postup činnosti případu užití, pak se provádí konkrétní případ užití podle základního scénáře. Analýza musí být řádně provedena také pro nepředvídané situace, které mohou ovlivnit činnost modelovaného systému. Těm jsou určeny alternativní scénáře. Počet alternativních scénářů pro jeden případ užití může být 0, 1 nebo také více. Opět to závisí na konkrétním případě užití. Alternativní scénáře se označují názvem, který vystihuje nestandardní průběh případu užití nebo konkrétní chybovou situaci.

Základní a alternativní scénář je formátován v podobě jednoúrovňového nebo víceúrovňového seznamu. Řízení toku činnosti případů užití, obdobně jako v programování, se označuje klíčovými slovy, která označují typ řízení činnosti.

Podmíněné kroky postupu činnosti jsou vyjádřeny klíčovým slovem KDYŽ (+ JINAK). Opakování kroků činnosti se vyjadřuje klíčovým slovem PRO, za nímž následuje upřesnění počtu opakování, např. „PRO každou šestou úlohu laboratorního měření“. Pokud opakování kroků činnosti není přesně stanoveno na daný počet opakování a je závislé na splnění (nesplnění) konkrétní podmínky, pak se využívá klíčové slovo DOKUD. Všechny tyto

zmíněné postupy, které se obecně využívají k řízení toku činnosti, jsou ve scénáři odlišeny podřízenou úrovní číslovaného seznamu pro všechny jeho dílčí kroky (tab. 3.1).

Tab. 3.1 – Formální vzhled scénáře případu užití.

Název případu užití	<b>Přihlášení k IS</b>
Identifikátor	UC01
Cíl	Zpřístupnění činnosti informačního systému.
Primární účastník	Uživatel systému
Sekundární účastník	Systém, Správce systému
Vstupní podmínky	IS je v daném období (semestru) přístupný.
Výstupní podmínky	Uživatel je přihlášen k IS.
Základní scénář	<ol style="list-style-type: none"> <li>1. <i>Uživatel systému</i> zadá požadavek na přihlášení.</li> <li>2. Systém umožní přístup k vyplnění přihlašovacích údajů.</li> <li>3. DOKUD <i>Uživatel systému</i> trvá na požadavku přihlášení k IS, pak: <ol style="list-style-type: none"> <li>3.1. <i>Uživatel systému</i> zadává přihlašovací údaje.</li> <li>3.2. <i>Uživatel systému</i> požaduje přihlášení k IS.</li> <li>3.3. Systém ověřuje zadané údaje pro přihlášení.</li> <li>3.4. KDYŽ jsou údaje pro přihlášení správné, PAK: <ol style="list-style-type: none"> <li>3.4.1. <i>Uživatel systému</i> je přihlášen k IS.</li> </ol> </li> <li>3.5. JINAK: <ol style="list-style-type: none"> <li>3.5.1. Systém otevírá opět formulář pro změnu přihlašovacích údajů.</li> </ol> </li> </ol> </li> </ol>
Alternativní scénář(e)	<p><b>Překročený limit možností přihlášení k IS</b> (spustí se v kroku 3.3., když je překročen limit možných přihlášení k IS)</p> <ol style="list-style-type: none"> <li>3.3.1. DOKUD <i>Uživatel systému</i> požaduje aktivaci přístupu k IS, pak: <ol style="list-style-type: none"> <li>3.3.1.1. <i>Systém</i> požaduje kontaktní e-mail pro zaslání informace o aktivaci přístupu k IS.</li> <li>3.3.1.2. <i>Uživatel systému</i> zasílá žádost o aktivaci přístupu k IS <i>Správci systému</i>.</li> </ol> </li> <li>3.3.2. KDYŽ <i>Uživatel systému</i> požaduje změnu hesla <ol style="list-style-type: none"> <li>3.3.2.1. Rozšíření UC03 <b>Změna hesla</b>.</li> </ol> </li> <li>3.3.3. JINAK <i>Uživatel systému</i> požaduje ukončení činnosti IS.</li> </ol>

Syntaxe zápisu scénářů v této publikaci odlišuje názvy případů užití **zvýrazněním textu tučně**. Klíčová slova vyjadřující tok řízení činnosti se vyznačují VELKÝMI PÍSMENY. Účastníci, kteří se podílejí na činnosti ve scénáři se zvýrazní *Textem se skloněným písmem*, jehož počáteční znak je zpravidla formátován velkým písmenem.

Je nutné definovat u alternativních scénářů úroveň kroku, kdy se spustí tato alternativní činnost. Pokud je alternativních scénářů více, pak jsou definovány následně za sebou podle pořadí úrovní, kdy jsou prováděny. Jsou uváděny v tabulce za sebou a vizuálně se oddělují tučně zvýrazněným názvem alternativního scénáře.

### 3.2 Modelování případů užití pro dílčí tematické celky

Výsledky modelování případů užití pro jednotlivé informační systémy byly zpracovány v produktu Enterprise Architect, verze 8 firmy Sparx Systems Pty Ltd. V rámci modelování případů užití navrhnul softwarový architekt také scénáře dílčích případů užití, které tady z hlediska objemu práce není možné zahrnout, proto se bude v této případové studii hodnotit celkový návrh logického modelu zahrnujícího diagram případů užití a sestavení dílčích scénářů. Např. objem práce systémového architekta při návrhu scénářů je příliš velký pro jednu osobu, proto se v projektovém týmu mohou objevit dvě osoby stejné role (softwarového analytika), které zpracovávají paralelně každý svou dílčí část návrhu scénářů. Zde se předpokládá velmi blízká komunikativnost při jejich práci, případně sdílení podkladových souborů.

V následujících podkapitolách jsou stručně uvedeny výsledky sběru požadavků pro jednotlivé informační systémy v podobě zpracování produktem Enterprise Architect.

#### 3.2.1 Informační systém sportovních aktivit

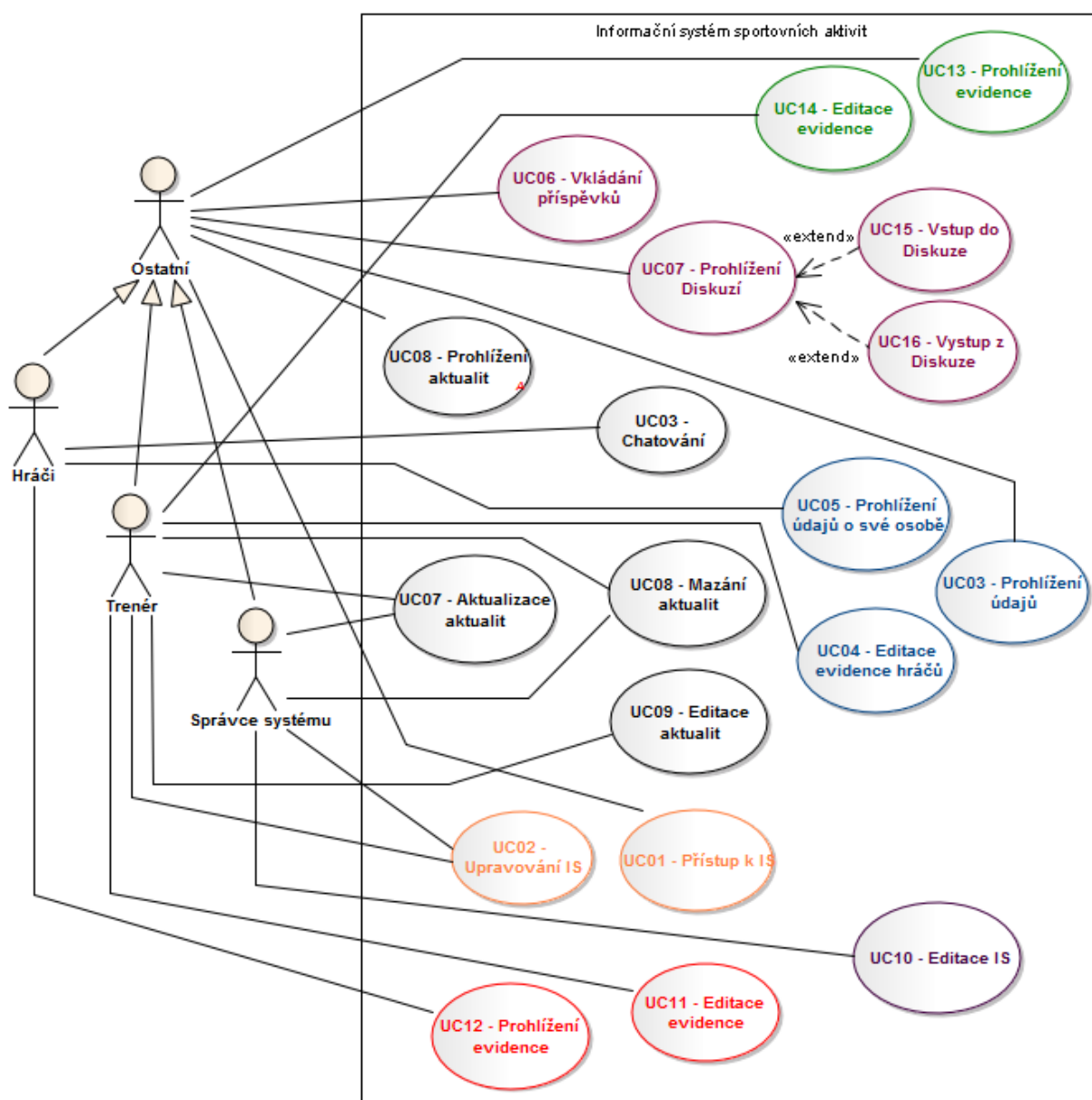
Případy užití sportovních aktivit (Obrázek 3.2) zahrnují účastníky Hráči, Trenér, Správce systému, kteří jsou z důvodu zobecnění modelování případů užití a zjednodušení celkového modelu potomky účastníka Ostatní.

Případy užití byly rozděleny do balíčků:

<b>Aktuality</b>	(černá)	UC03, UC07, UC08, UC08, UC09.
<b>Diskuze</b>	(tmavě červená)	UC06, UC07, UC15, UC16.
<b>Evidence hráčů</b>	(modrá)	UC03, UC04, UC05.
<b>Evidence trenéra</b>	(fialová)	UC10.
<b>Evidence tréninků</b>	(červená)	UC11, UC12.
<b>Evidence zápasů</b>	(zelená)	UC13, UC14.
<b>Správa systémů</b>	(oranžová)	UC01, UC02.

Případy užití byly původně softwarovým architektem sestaveny do dílčích diagramů pro jednotlivé balíčky. Náhled na případy užití byl zbytečně rozdělen do malých celků, pak není možné nahlížet na celkový koncept logického pohledu – případů užití. Vzhledem k danému počtu případů užití to bylo zbytečně rozděleno. I když jsou případy užití označeny pro jednoznačnost určením číslem dle syntaxe, není vhodné využívat stejných názvů (UC11 versus UC14 a UC12 versus UC13). Při komunikaci v projektovém týmu může dojít k nesrovnalostem při diskuzi.





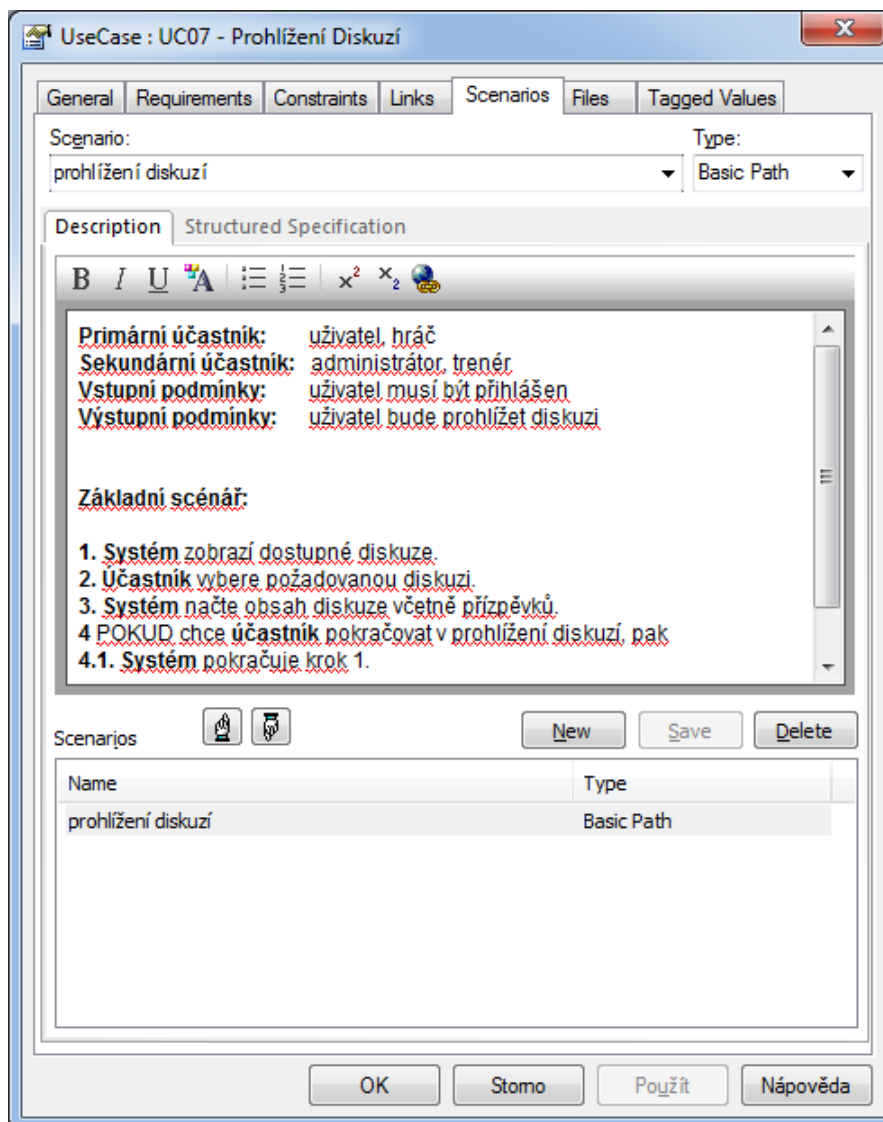
Obrázek 3.2 – Případy užití IS sportovních aktivit

Model případů užití zahrnuje 3 účastníky a 19 případů užití. Syntaxe zápisů případů užití je správná, Model případů užití obsahuje vazby mezi účastníky a navazující případy užití. Rovněž zahrnuje dva případy užití s vazbou <<extend>> na případ užití UC07. Rozšiřující vazba je dána stereotypem s označením „extend“, ale nezahrnuje podmínku rozšíření, což v modelu chybí pro určení, za jakých podmínek se rozšiřující případ užití aktivuje.

Obsahové zhodnocení modelu případů užití bude provedeno v následující kapitole, kde proběhne kontrola návaznosti na sběr požadavků systémového analytika.

Každý případ užití obsahuje pouze základní scénář, přičemž u některých případů užití by bylo vhodné doplnit i alternativní scénáře pro neočekávané události při zpracování standardní činnosti případu užití.

Většina základních scénářů obsahuje na konci scénáře kroky pro návrat na první krok základního scénáře. Takový postup není správný, protože pro znovu aktivování stejného případu užití účastník opět stejnou činností v rozhraní volá opět stejný případ užití. Tyto kroky by znamenaly, že se po každém skončení činnosti (případu užití) se systém dotazuje účastníka, zda chce spustit stejnou činnost ještě jednou. Jednak je taková činnost v podobě Dialogového okna zbytečně obtěžující uživatele systému a také se v případě opakování stejné činnosti nepostupuje, pokud se opakují všechny kroky scénáře znovu.



Obrázek 3.3 – Scénář případu užití UC07 – Prohlížení diskuzí

Další nesrovnalostí při modelování scénářů případů užití je nedodržení stejných názvů v označení základního scénáře a názvu případu užití. Ve většině scénářů případů užití se jedná o chyby formální, které nezajistí jednotnost práce softwarového architekta. Struktura všech scénářů dle doporučení UML byla dodržena. Velkou chybou softwarového architekta bylo sestavení scénářů rozšířeného UC07 a rozšiřujících UC15 a UC16 případů užití tak, jak je to nezbytné pro vložení kroků rozšiřujících případů užití. Chybí informace, v kterém kroku se scénář rozšiřuje a za jaké podmínky.

### 3.2.2 Informační systém pro administraci stravování

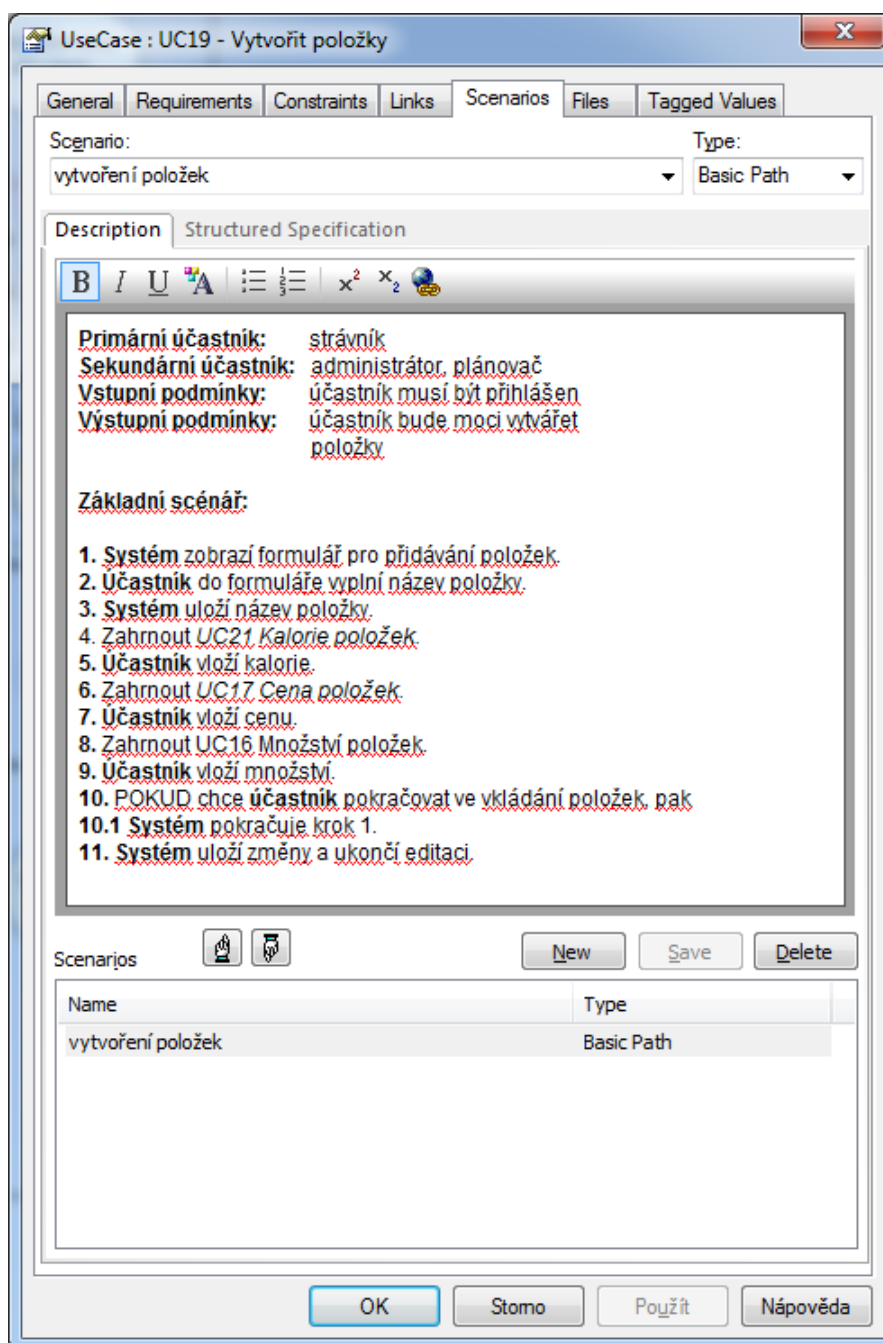
Případy užití IS pro administraci stravování (Obrázek 3.4) zahrnují účastníky Host, Strávník, Plánovač a Administrátor. Účastníci Strávník, Plánovač a Administrátor jsou zobecněni rodičovským účastníkem Registrovaný uživatel.



Obrázek 3.4 – Případy užití IS administrace stravování

Případy užití byly rozděleny do balíčků:

<b>Správa jídelníčku</b>	(zelená)	UC13, UC14, UC15, UC16, UC17.
<b>Správa položek</b>	(černá)	UC18, UC19, UC20, UC21, UC22, UC23.
<b>Správa receptů</b>	(červená)	UC09, UC10, UC11, UC12.
<b>Správa uživatelů</b>	(modrá)	UC01, UC02, UC03, UC04, UC05, UC06, UC07.
<b>Statistika jídelníčku</b>	(fialová)	UC08.



Obrázek 3.5 – Scénář případu užití UC07 – Prohlížení diskuzí

Případy užití byly softwarovým architektem sestaveny do dílčích diagramů pro jednotlivé balíčky, které odpovídají rozvržení požadavků do stejných balíčků. Náhled na případy užití byl rozdělen do malých celků, což nepřispělo k pohledu na celkový koncept logického pohledu – případů užití. Názvy jednotlivých případů užití se odlišovaly, byly syntakticky správně navrženy a očíslovány. Systematický přístup k modelování případů užití byl výborný, takže práci softwarového architekta můžeme hodnotit jako práci velmi přínosnou pro projektový tým.

Logický model IS pro administraci stravování pro stránce obsahové bude zhodnocen v následující kapitole včetně kontroly shody s funkčními požadavky představenými v kapitole 2.2.2.

Každý případ užití obsahuje pouze základní scénář, přičemž u některých případech užití by bylo vhodné doplnit i alternativní scénáře pro neočekávané události při zpracování standardní činnosti případu užití. Stejná situace jako v předchozím modelovaném systému sportovních aktivit.

Rovněž stejný přístup návratu k prvnímu kroku základního scénáře jako u předchozího modelovaného IS sportovních aktivit. Na druhou stranu je nutné podotknout, že scénáře případů užití UC20, UC21 a US22 byly správně zakomponovány do scénářů UC19 nebo UC23 dle příslušné vazby <<include>> nebo <<extend>> (Obrázek 3.5).

### 3.2.3 Informační systém osobní agendy ve zdravotnictví

Logický model pro informační systém osobní agendy ve zdravotnictví nebyl sestaven obdobně jako sběr požadavků, neboť z projektového týmu kromě systémového analytika odstoupil také softwarový architekt z tvorby projektu. Fáze modelování případů užití pro tento informační systém nebude v případové studii hodnocena.

## 3.3 Zhodnocení modelování případů užití

Návrhy logických modelů a týmová spolupráce, v tomto případě spolupráce systémového analytika a softwarového architekta, bude zhodnoceno v této kapitole pro jednotlivé informační systémy.

Výsledný model případů užití sestavený softwarovým architektem je založen založeno převážně na bezchybném sběru funkčních a nefunkčních požadavků systémovým analytikem, což dle hodnocení v předchozí kapitole nebylo dosaženo. Přesto se hodnocení z důvodu objektivity práce softwarového architekta zakládá na tom, že sběr požadavků byl bezchybný. Celkový proces návrhu modelu případů užití probíhá v několika iteračních krocích a měl by být navržen bez chyb, neboť je základem modelování celkového informačního systému, z něhož vychází následné fáze vývoje projektu.

Je nutné, aby byla splněna 100 % návaznost všech případů užití na funkční požadavky. Počet a číslování případů užití nesouhlasí zpravidla s počtem a číslováním funkčních požadavků. Jednak je to dáno tím, že se při modelování softwarový architekt snaží o zjednodušení pohledu na modelovaný systém tím, že zavádí zobecnění případů užití nebo zobecnění účastníků. K dispozici je nástroj **Relationship Matrix**, kdy přehledová tabulka ukazuje *vzájemné vztahy* mezi (v této situaci konkrétně) případy užití a funkčními požadavky

(obecně lze definovat libovolný zdroj a cíl zkoumání vzájemné návaznosti prvků). Přehledová tabulka je pro zobrazení v tomto dokumentu příliš obsáhlá, proto je zhodnocení provedeno v mnohem jednodušších a dalšími parametry hodnocení doplněných tabulkách v následujících podkapitolách.

### 3.3.1 Informační systém sportovních aktivit

Model případů užití pro IS sportovních aktivit (Obrázek 3.2) vychází z **33 funkčních požadavků (FP)** a obsahuje **19 případů užití (PU)**.

Tab. 3.2 – Zhodnocení modelování případů užití pro IS sportovních aktivit

Balíček	Případ užití	Funkční požadavek	Správné přiřazení účastníka	Správnost obsahu případu užití
Správa systémů	UC01	SS01, SS02, SS03, SS04	×	×
Správa systémů	UC02	SS05, SS06	✓	×
Aktuality	UC03	–	×	×
Evidence hráčů	UC03	EH01, EH02	×	✓
Evidence hráčů	UC04	EH03, EH04, EH05, EH06	✓	×
Evidence hráčů	UC05	–	×	×
Diskuze	UC06	D02	✓	✓
Diskuze	UC07	SS05	×	×
Aktuality	UC07	–	×	×
Aktuality	UC08	–	×	×
Aktuality	UC08	–	×	×
Aktuality	UC09	A01, A02	✓	×
Evidence trenéra	UC10	ET01, ET02	✓	✓
Evidence tréninků	UC11	ET01, ET02, ET03, ET04	×	×
Evidence tréninků	UC12	ET05	×	×
Evidence zápasů	UC13	EZ01, EZ02, EZ03	×	×
Evidence zápasů	UC14	EZ03, EZ05, EZ06, EZ07	✓	✓
Diskuze	UC15	D01, D03, D04	×	×
Diskuze	UC16	–	×	×

Hodnocení je dáno procentuálním vyjádřením zpracované analýzy z hlediska následujících kritérií:

- ✚ zajištění všech FP – **94 %**.
- ✚ návrh PU, které splňují minimálně jeden FP – **68 %**.
- ✚ hodnocení jednoznačnosti PU (číslování) – **68 %**.
- ✚ správnost přiřazení účastníka – **32 %**.
- ✚ správnost obsahu PU (v návaznosti na text FP) – **21 %**.

Za předpokladu stejné váhy výše uvedených procentuálních hodnocení je celkové průměrné zhodnocení činnosti softwarového architekta a jeho návaznosti na práci systémového analytika hodnocena **57 %**.

### 3.3.2 Informační systém pro administraci stravování

Model případů užití pro IS administrace stravování (Obrázek 3.4) vychází z **42 funkčních požadavků (FP)** a obsahuje **23 případů užití (PU)**.

Tab. 3.3 – Zhodnocení modelování případů užití pro IS administrace stravování

Balíček	Případ užití	Funkční požadavek	Správné přiřazení účastníka	Správnost obsahu případu užití
Správa uživatelů	UC01	SU02, SU03, SU08	✓	✓
Správa uživatelů	UC02	SU04, SU07, SU10	✓	✓
Správa uživatelů	UC03	–	✗	✗
Správa uživatelů	UC04	SU01	✓	✓
Správa uživatelů	UC05	SU05	✓	✓
Správa uživatelů	UC06	SU06	✓	✓
Správa uživatelů	UC07	SU09	✓	✓
Statistika jídelníčku	UC08	ST01, ST02, ST03	✓	✓
Správa receptů	UC09	SR01, SR02, SR03	✓	✓
Správa receptů	UC10	SR05	✓	✓
Správa receptů	UC11	SR06	✓	✓
Správa receptů	UC12	SR04	✓	✓
Správa jídelníčku	UC13	SJ03, SJ05, SJ10	✓	✓
Správa jídelníčku	UC14	SJ04	✗	✓
Správa jídelníčku	UC15	SJ01, SJ02	✓	✓
Správa jídelníčku	UC16	SJ06, SJ08	✓	✓
Správa jídelníčku	UC17	SJ07, SJ09	✓	✓

Balíček	Případ užití	Funkční požadavek	Správné přiřazení účastníka	Správnost obsahu případu užití
Správa položek	UC18	SP06, SP07, SP08	✓	✓
Správa položek	UC19	SP01, SP02, SP03	✓	✓
Správa položek	UC20	–	✗	✗
Správa položek	UC21	SP05	✗	✗
Správa položek	UC22	–	✗	✗
Správa položek	UC23	SP04	✓	✓

Hodnocení je dáno procentuálním vyjádřením zpracované analýzy z hlediska následujících kritérií:

- ✚ zajištění všech FP – **88 %**.
- ✚ návrh PU, které splňují minimálně jeden FP – **87 %**.
- ✚ hodnocení jednoznačnosti PU (číslování) – **100 %**.
- ✚ správnost přiřazení účastníka – **88 %**.
- ✚ správnost obsahu PU (v návaznosti na text FP) – **90 %**.

Za předpokladu stejné váhy výše uvedených procentuálních hodnocení je celkové průměrné zhodnocení činnosti softwarového architekta a jeho návaznosti na práci systémového analytika hodnocena **91 %**.



## Shrnutí realizace návrhu informačního systému

Práce softwarového architekta informačního systému sportovních aktivit dosáhla celkem hodnocení přínosu 57 %.

Softwarový architekt nevěnoval pozornost syntaxi případů užití, jejich návaznosti na funkční požadavky. Přesto je práce originální a účelně popisná v názvech případů užití. Při větší pozornosti při systematické práci a větší spolupráci se systémovým analytikem by model případů užití byl hodnocen výrazně lépe než IS administrace receptů.

Práce systémového analytika informačního systému pro administraci stravování dosáhla celkem hodnocení přínosu 91 %.

Model případů užití pro IS administrace receptů byl celkově hodnocen na výborné úrovni, ale popisy a názvy případů užití jsou příliš obecné. Je to dobré pro systémy, které lze dále přizpůsobit pro jiné podmínky nového IS.

Hodnocení náplně scénářů, jejich obsahu a potřebného počtu alternativních scénářů dle předpokládaných a nečekaných situací při nasazení IS do provozu by bylo příliš obsáhlé pro tuto případovou studii, proto zde není uvedeno.



## 4 IMPLEMENTACE INFORMAČNÍHO SYSTÉMU



### Plánování jednotlivých kroků: 2 hodiny

Popis fáze implementace a základních pojmů z objektového modelování. Ukázat řešení a návrhy diagramů tříd pro jednotlivé projektové týmy, které byly realizovány v prostředí Enterprise Architect. Zhodnotit návrhy a práci softwarového analytika.



#### Cíl:

- ✚ Definovat implementační fázi modelování informačních systémů.
- ✚ Předložit řešení implementace IS dílčích projektových týmů
- ✚ Zhodnotit fázi implementace, konkrétně návrhu datového modelu pro jednotlivé projektové týmy.



### Výklad a popis případové studie

Vývojová fáze implementace nabývá zpravidla konkrétní podoby fyzických objektů a jejich vazeb. Do této části vývoje informačního systému patří konkrétní výstupy v podobě podkladů pro tvorbu programových kódů ve vybraném programovém prostředí. Návrhy diagramů tříd, respektive datových modelů pro databáze, se zabývá softwarový analytik, jehož činnost vychází ze základního pohledu na modelovaný systém, tj. modelu případů užití. Výsledky implementace jsou navrženy s ohledem na nefunkční požadavky a větší měrou pak přispívají k následné tvorbě programových kódů, které jsou realizovány programátory nebo databázovými specialisty.

V následujících podkapitolách si stručně objasníme základní pojmy diagramů tříd a datových modelů a představíme výsledky dílčích projektových týmů. V závěru se pokusíme objektivně zhodnotit práci softwarových analytiků dílčích projektových týmů.

### 4.1 Diagram tříd

Pojmem objekt z problémové domény je označován **business objekt** a odpovídá libovolnému předmětu, osobě v reálném prostředí modelované oblasti. V procesu modelování jsou business objekty transformovány postupně na **softwarové objekty**.

Softwarové objekty se v oblasti objektového modelování označují třídy a jsou určeny pro implementaci v prostředí C#, C++, Java, Visual Basic, Python, .NET a dalších. Pokud je implementace určena pro databázové prostředí, pak se softwarové objekty v oblasti databázového modelování nazývají relační tabulky.

*Diagram tříd* představuje primární nástroj pro modelování business objektů, jejich vlastností a vztahů v prostředí modelovacího nástroje UML, je základním diagramem pro

generování kódu. Všechny ostatní diagramy čerpají informace z diagramu tříd. Diagram tříd se skládá ze dvou základních entit: **třída** a **relace**.

## Třída

Pojem třída je objekt, který vzniká přechodem z business objektu na konceptuální objekt a jeho dalším upřesněním. Někdy je možné se také setkat s pojmem softwarový objekt, což představuje v softwarovém prostředí entitu s názvem **třída**. V diagramu tříd je symbol třídy definován pomocí tří základních částí:

- název (definice) třídy,
- seznam atributů dané třídy,
- seznam operací dané třídy.

Třídy jsou mezi sebou navzájem zpravidla vázány vhodným typem relace, kde ve vlastnosti relace je pak možné definovat jejich vztah slovy.

## Relace diagramu tříd

Asociace představují obecný vztah mezi dvěma třídami softwarových objektů. Definuje, jak navzájem k sobě mohou přistupovat a co je přípustné. Stejně jako je důležité správně označit názvy tříd, atributů a operací, je také důležité správně definovat typ a název asociace. Základní typy asociací:

- asociace (jednosměrná/obousměrná),
- agregace,
- kompozice,
- generalizace.

Relace diagramu tříd jsou označeny příslušnou násobností asociace. Návrhem dané relace mezi třídami jsou zpřístupněny nebo zneprístupněny informace mezi třídami v souvislosti také s navrženým oprávněním přístupu atributů.

### *Násobnost asociací*

Pojem násobnost asociace se určuje podle definované role k dané asociaci. Pak podle počtu objektů dané třídy, které mohou navazovat na konkrétní objekt jiné třídy spojené s ní asociací, je definován typ násobnosti, např.:

- 1      právě jeden,
- 0..1    žádný nebo jeden,
- 0..\*    žádný nebo jeden nebo více,
- 1..\*    jeden nebo více.

Ve většině případů se využívají výše uvedené označení pro násobnost asociací, ale pro konkrétní rozsah hodnot je možné uvést dolní a horní mez násobnosti asociace, které vyřeší problém s další omezující specifikací dané asociace.

## 4.2 Datový model

Datový model vzniká mapováním objektového modelu (modelu tříd) do prostředí relačních databází. Datový model je odlišný ve vlastnostech typických pro databázové systémy a nezbytné integritě dat v databázových systémech. Musí být splněny předpoklady struktury databázového systému, která je zajištěna těmito dílčími postupy:

- ✚ Mapováním tříd do tabulek.
- ✚ Mapováním atributů do sloupců – je nutné zajistit primární klíče tabulek.
- ✚ Mapováním operací do objektů modelujících chování entity – činnosti v diagramech tříd zastupují operace, v databázových systémech jsou operace nahrazeny dotazy, uloženými procedurami, spouštěmi, indexy apod.
- ✚ Mapováním asociací mezi objekty – nastavit vazby mezi objekty, které mohou obsahovat pouze následující typy násobností asociací 1:1, 1:N (N:1), M:N a s tím souvisí tvorba cizích klíčů.
- ✚ Mapováním dědičnosti/zobecnění tříd do jednoho nebo více datových objektů.
- ✚ Mapováním asociačních tříd vázaných na asociaci mezi třídami na tabulku s vazbou na okolní třídy, které jsou vázány asociační vazbou.

Struktura diagramu tříd se transformuje tak, aby bylo možné aplikovat datový model s jeho objekty (tabulky) a vazbami do prostředí databáze.

## 4.3 Implementace informačních systémů pro dílčí tematické celky

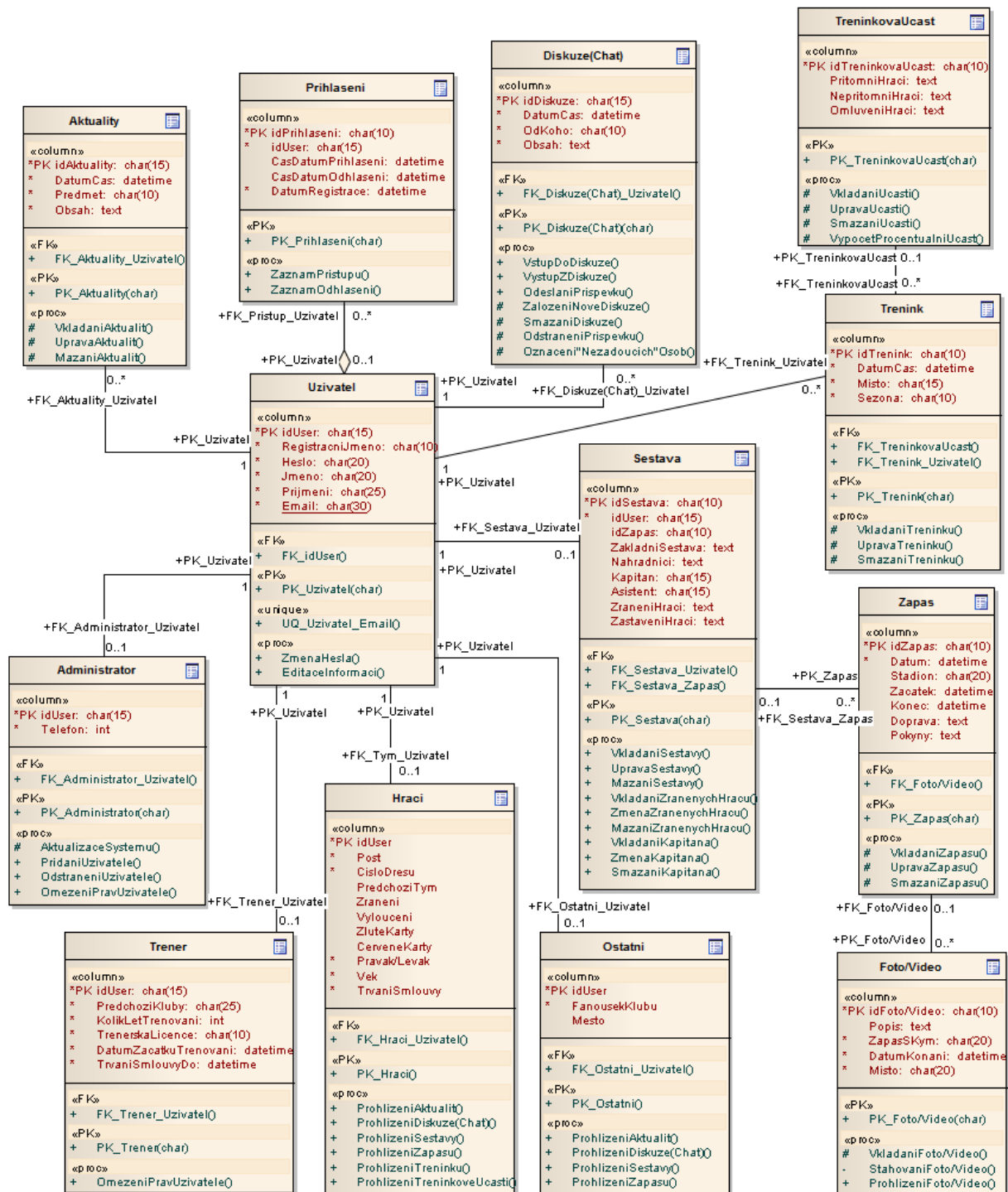
Datové modely pro jednotlivé informační systémy byly navrženy s využitím nástrojů a podpory produktu Enterprise Architect. V rámci modelování případů užití navrhnul softwarový architekt také scénáře dílčích případů užití, které tady z hlediska objemu práce není možné zahrnout, proto se bude v této případové studii hodnotit celkový návrh logického modelu zahrnujícího diagram případů užití a sestavení dílčích scénářů. Např. objem práce systémového architekta při návrhu scénářů je příliš velký pro jednu osobu, proto se v projektovém týmu mohou objevit dvě osoby stejné role (softwarového analytika), které zpracovávají paralelně každý svou dílčí část návrhu scénářů. Zde se předpokládá velmi blízká komunikativnost při jejich práci, případně sdílení podkladových souborů.

V následujících podkapitolách jsou stručně uvedeny výsledky sběru požadavků pro jednotlivé informační systémy v podobě zpracování produktem Enterprise Architect.

### 4.3.1 Informační systém sportovních aktivit

V návrhu diagramu tříd pro IS sportovních aktivit je navrženo **13** datových objektů (tabulek). Mapování **zobecnění účastníka**, který byl v modelu případů užití označen názvem **Ostatní** (rodič), je v datovém modelu přiřazen k názvu **Uživatel**, což je matoucí. Navíc je potomek rodičovského účastníka **Uživatel** označen jako **Ostatní**.

Z případů užití daného systému pak softwarový analytik navrhnul datové objekty s názvy: **Aktuality**, **Diskuze (Chat)**, **Foto/Video**, **Prihlášení**, **Sestava**, **Trenink**, **Treninkova Ucast** a **Zapas**. Navržené objekty navazují na předmětné informace z případů užití, případně označení balíčků.

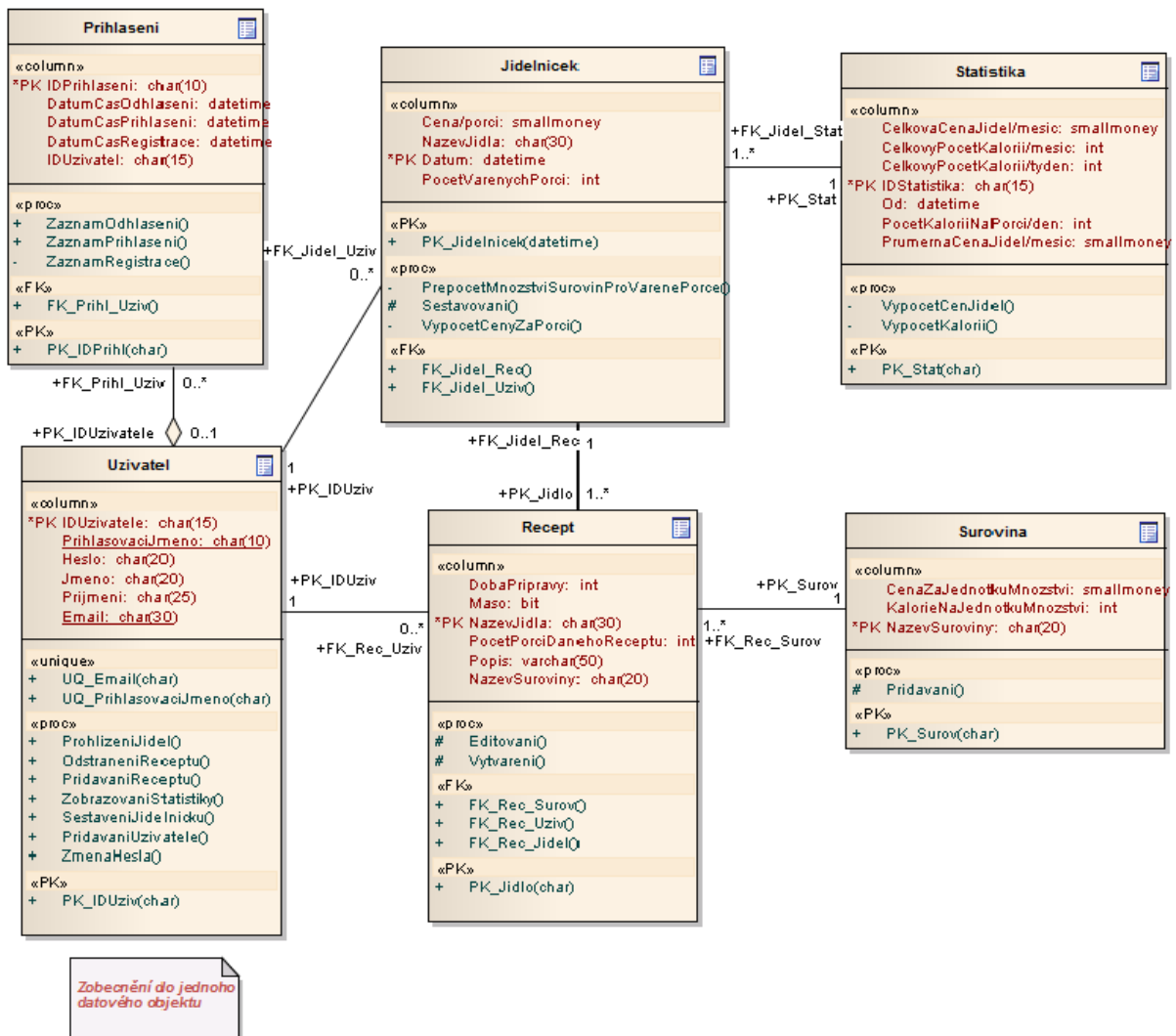


Obrázek 4.1 – Datový model IS sportovních aktivit

#### 4.3.2 Informační systém pro administraci stravování

V návrhu diagramu tříd informačního systému pro administraci stravování je vytvořeno **6** datových objektů (tabulek). Mapování **zobecnění účastníka**, který byl v modelu případů užití označen názvem **Registrovaný uživatel** (rodič) s potomky Strávník, Plánovač a Administrátor, je implementováno pouze jedním datovým objektem **Uživatel**. Mapování

vazby zobecnění do jednoho datového objektu je správné pouze za podmínky vytvoření nového atributu rozlišujícího potomky.



Obrázek 4.2 – Datový model IS administrace receptů

Další datové objekty s názvy: **Prihlášení**, **Jidelnicek**, **Recept**, **Statistika**, **Surovina** zahrnují předmětné výrazy z modelu případů užití.

#### 4.4 Zhodnocení implementační fáze procesu vývoje IS

Navržené datové modely a jejich konkrétní implementace v podobě softwarových objektů představují přechod logického modelu do modelu fyzického. Jestliže byla analýza navržená softwarovým analytikem zpracována efektivně a bezchybně, pak činnost databázového specialisty nemůže mít po stránce obsahové žádné chyby, neboť striktně vychází z navrženého datového modelu. Softwarový analytik by měl mít také schopnost zvážit v návrhu, zda bude datový model schopen fyzické existence a bezchybného provozu. V případě pochybností by své návrhy měl konzultovat ve výjimečných případech s databázovým specialistou. A naopak veškeré změny ve fyzickém datovém modelu neboli

konkrétní databázi by měly být zaznamenány v návrhu datového modelu (v logickém pohledu).

Hodnocení implementační fáze rozdělíme do několika pohledů na dílčí části, které by měly naplňovat návrh datového modelu:

- ✚ Návaznost případů užití na datové objekty.
- ✚ Mapování zobecnění účastníků.
- ✚ Mapování tříd a jejich atributů do tabulek.
- ✚ Mapování asociací mezi objekty.
- ✚ Efektivnost návrhu datového modelu.

Při hodnocení nebude použito procentuální zhodnocení, ale individuálním popisem chyb, návrhu ke zjednodušení datového modelu či doplnění datového modelu o nerealizované požadavky na systém.

#### 4.4.1 Informační systém sportovních aktivit

Pro hodnocení informačního systému sportovních aktivit v části **návaznosti případů užití** na datové objekty byla využita tabulka (Obrázek 4.2) jako výsledek již zmiňovaného nástroje **Relationship Matrix** programu Enterprise Architect.

	UC01 - Přístup k IS	UC02 - Upravování IS	UC03 - Chatování	UC03 - Prohlížení údajů	UC04 - Editace evidence hráčů	UC05 - Prohlížení údajů o své osobě	UC06 - Vkládání příspěvků	UC07 - Aktualizace aktualit	UC07 - Prohlížení Diskuzí	UC08 - Mazání aktualit	UC08 - Prohlížení aktualit	UC09 - Editace aktualit	UC10 - Editace IS	UC11 - Editace evidence	UC12 - Prohlížení evidence	UC13 - Prohlížení evidence	UC14 - Editace evidence	UC15 - Vstup do Diskuze	UC16 - Výstup z Diskuze	
Administrator																				
Aktuality								↑		↑		↑								
Diskuze(Chat)				↑															↑	↑
Foto/Video																				
Hraci																				
Ostatni																				
Prihlaseni	↑	↑																		
Sestava																				
Trener																				
Trenink														↑						
TreninkovaUcast																				
Uzivatel																				
Zapas																			↑	

Obrázek 4.3 – Implementace případů užití v datovém modelu (IS sportovních aktivit)

Ze vztahu závislosti datových objektů na případy užití je zřejmé, že 10 případů užití bylo realizováno ve formě uložené procedury pro konkrétní datový objekt, ale 9 případů užití není implementováno. Samozřejmě je zde chyba v komunikace softwarového architekta a softwarového analytika, kde každá činnost byla zpracována odděleně a evidentně na sebe nenavazuje. Chyba je vždy na obou stranách osob projektového týmu, neboť softwarový architekt by zpětně měl provést kontrolu implementace navržených případů užití a zároveň nové činnosti, které byly doplněny z hlediska udržitelnosti fyzického modelu IS, je nutné změnit zpětně v návrhu případů užití, protože tvoří výchozí model IS.

### Mapování zobecnění účastníků

Vazba zobecnění byla mapována do stejného počtu datových objektů z důvodu různorodosti většího množství atributů jednotlivých potomků a jejich činností (uložených procedur). Tento přístup je pro udržování informací nepříliš vhodný, ale pro zajištění bezpečnosti dat a rychlosti přístupu k datům je efektivní.

Negativně lze hodnotit pouze nevhodnost a nesprávnost ve volbě názvů potomků a rodiče zobecněné vazby ve vztahu k označení potomků a rodičů v modelu případů užití. Z toho by vyplývalo, že z rodiče Ostatní se při implementaci stal potomek. A to je nepřijatelné.

### Mapování tříd a jejich atributů do tabulek

Rozsah navržených tříd a jejich atributů odpovídá potřebným informacím uloženým v informačním systému. Datové typy jednotlivých atributů odpovídají potřebnému rozsahu a datovému typu informace. Všechny datové objekty obsahují primární klíče.

### Mapování asociací mezi objekty

Zde je situace kritická. Integrita dat by nebyla zajištěna, protože v tabulce **Aktuality**, **Diskuze** a **Trenink** chybí atribut **idUser**. Taktéž v tabulce **Prihlaseni** chybí objekt cizího klíče, čímž není implementována vazba mezi tabulkami **Prihlaseni** a **Uzivatel**.

Navíc je struktura návaznosti tabulek **Uzivatel-Trenink-TreninkovaUcast** svázána chybně a informace jsou naprosto nevhodným způsobem mapovány do tabulky **TreninkovaUcast**, kdy informace o nepřítomnosti a omluvení hráčů na tréninku jsou vkládány jako text bez vazby na **idUser** (hráče). Zde se jedná o závažnou chybu v návrhu struktury datového modelu.

### Efektivnost návrhu datového modelu

Datový model v této navržené podobě obsahuje řadu nedostatků a chyb, není pro správu databáze udržitelný a snadno přístupný. Kromě výše uvedených chyb je zde ještě jedna nevhodná chyba v návrhu datového objektu Sestava, kde jsou vloženy informace o členech sestavy jako text vkládaný do tabulek, čímž mohou vznikat duplikace dat ve jménech hráčů týmů. Zde by měly být atributy: **Nahradnici**, **Kapitan**, **Asistent**, **ZraneniHraci**, **ZastaveniHraci** nahrazení funkcí nebo typem člena sestavy, které by mohl definovat dále další datový objekt. Zde by byly jednoduše uloženy pouze úspornější informace v podobě **idUser** a **idZapas**.

#### 4.4.2 Informační systém pro administraci stravování

Pro hodnocení informačního systému sportovních aktivit v části **návaznosti případů užití** na datové objekty byla využita tabulka (Obrázek 4.4) jako výsledek již zmiňovaného nástroje **Relationship Matrix** programu Enterprise Architect.

	UC01 - Přihlásit uživatele	UC02 - Odhlásit uživatele	UC03 - Nastavit roli uživatele	UC04 - Vytvořit uživatele	UC05 - Změnit uživatele	UC06 - Odstranit uživatele	UC07 - Zobrazení uživatele	UC08 - Zobrazit statistiku	UC09 - Vytvořit recept	UC10 - Zobrazit recept	UC11 - Změnit recept	UC12 - Odstranit recept	UC13 - Zobrazit jídelniček	UC14 - Vyběr jídla z jídelniček	UC15 - Vytvořit jídelniček	UC16 - Změnit jídelniček	UC17 - Odstranit jídelniček	UC18 - Zobrazit položky	UC19 - Vytvořit položky	UC20 - Zadat množství	UC21 - Zadat cenu	UC22 - Zadat kalorie	UC23 - Změnit položky	UC24 - Odstranit položky	
Administrator																									
Jidelniček															↑										
Planovac																									
Prihlaseni	↑	↑																							
Recept									↑	↑															
Statistika								↑																	
Stravnik																									
Surovina																			↑	↑	↑	↑			
Tablel																									
Uzivatel			↑	↑	↑					↑		↑	↑												
Zvedavy																									

Obrázek 4.4 – Implementace případů užití v datovém modelu (IS pro administraci stravování)

Ze vztahu závislosti datových objektů na případy užití je zřejmé, že 16 případů užití bylo realizováno ve formě uložené procedury pro konkrétní datový objekt, ale 8 případů užití není implementováno. Opět chyba v komunikaci a kontrole práci softwarovým architektem, který navrhnul model případů užití a také softwarového analytika, který měl konzultovat změny v návaznosti na model případů užití podle potřeby databáze se softwarovým architektem.

#### Mapování zobecnění účastníků

Vazba zobecnění byla mapována do jediného (společného) datového objektu z důvodu stejných atributů, pouze rozdílných práv k přístupu do databáze. Tento přístup je pro udržování informací přínosný.

Chybou tohoto mapování je, že naprosto zaniká vazba zobecnění, protože nejsou žádným argumentem rozlišení daní potomci vazby zobecnění. Všichni účastníci jsou takto postaveni na stejnou úroveň. Zabezpečení přístupu k informacím nebude splněno.

#### Mapování tříd a jejich atributů do tabulek

Rozsah navržených tříd a jejich atributů odpovídá potřebným informacím uchovaným v informačním systému. Datové typy jednotlivých atributů odpovídají potřebnému rozsahu a datovému typu informace. Všechny datové objekty obsahují primární klíče. Některé primární klíče jsou navrženy s nevhodným (příliš rozsáhlým) datovým typem (char(20)). Vzhledem k typu informací, která není citlivá pro přístup veřejnosti, je datový typ primárního klíče chybně navržen. A naprosto nevhodný typ primárního klíče v tabulce **Jidelniček**, konkrétně



atribut **Datum**. Datový typ „datetime“ je rozsahem naprosto nevhodný pro volbu primárního klíče. Vhodným řešením by bylo vytvořit nový atribut pro primární klíč s menší alokací prostoru dat.

### Mapování asociací mezi objekty

Zde je situace rovněž kritická. Nebyla zajištěna integrita dat. V tabulkách nejsou atributy, které by zajistily návaznost datových objektů neboli propojení dat ve smyslu asociace. Přičemž počet objektů reprezentujících asociace prostřednictvím cizích klíčů je naplněn bezchybně. Zde se jedná výhradně o neznalost ze strany softwarového analytika a jeho neznalostem, což by mohl v projektovém týmu odhalit a upravit databázový specialista. Ale je to předávání činnosti a zodpovědnosti za návrh systému v rámci projektového týmu.

Rozhodně kruhová vazba ve struktuře vazeb datového modelu není vhodná a správně navržená, konkrétně kruhová asociace mezi datovými objekty **Uživatel-Jidelnicek-Recept**.

### Efektivnost návrhu datového modelu

Datový model v této navržené podobě obsahuje velké množství nedostatků a chyb, není pro správu databáze udržovatelný a snadno přístupný. Práci softwarového architekta, který provedl svůj návrh implementace chybně, může zachránit jedině svou prací, která není v jeho kompetenci, databázový specialista.



## Shrnutí implementace informačního systému

Vzhledem k předchozím vývojovým fázím projektu je fáze implementace zřejmě nejhorší ve vztahu k funkčnosti a realizovatelnosti navrhovaného systému. Projektový tým při implementaci navrhovaného modelu není funkční, a tudíž jsou výsledky špatné.

Minimálně měla být zajištěna kontinuita s modelem případů užití, což bylo snadno dohledatelné a bez problémů realizované. Rozhodně se zde týmová spolupráce neprojevila a softwaroví architekti začali navrhovat datový model tzv. „na zelené louce“.

Rozhodně nejsou ani schopnosti a znalosti z oblasti objektového modelování a implementace informačních systémů na vynikající úrovni. To se pak projevuje realizací informačního systému s chybami a zároveň mimo představy odběratele. Za výsledky své práce a celkový honorář však odpovídá celý tým jako celek. Určitě by pak projektový manažer takového týmu měl zvážit, zda bude chtít ve svém týmu takové neprospěšné týmové spolupracovníky.

Celkově lze shrnout výsledky implementací informačních systémů jako nesystematicky zpracované, bez možného testování návaznosti a spolupráce s ostatními členy týmu a rovněž z odborného hlediska softwarových analytiků na velmi nízké úrovni.

## 5 STRUKTURA INFORMAČNÍHO SYSTÉMU



### Plánování jednotlivých kroků: 1 hodina

Popis oblasti návrhu informačního systému, vysvětlení problematiky a rozsahu zpracovaných tematických celků.



#### Cíl:

- ✚ Definovat implementační fázi modelování informačních systémů.
- ✚ Předložit výslednou strukturu IS dílčích projektových týmů
- ✚ Zhodnotit fázi realizace datového modelu pro jednotlivé projektové týmy.



### Výklad a popis případové studie

Realizační fáze projektu je vytvoření fyzického modelu informačního systému v prostředí databázového serveru MS SQL Server 2008. Náplní této fáze projektu a činností databázového specialisty je vytvořit datové objekty, zajistit referenční integritu s využitím primárních, cizích klíčů, indexů a naplnit datové objekty sadou ukázkových dat pro následné testování vytvořených dotazů nebo uložených procedur.

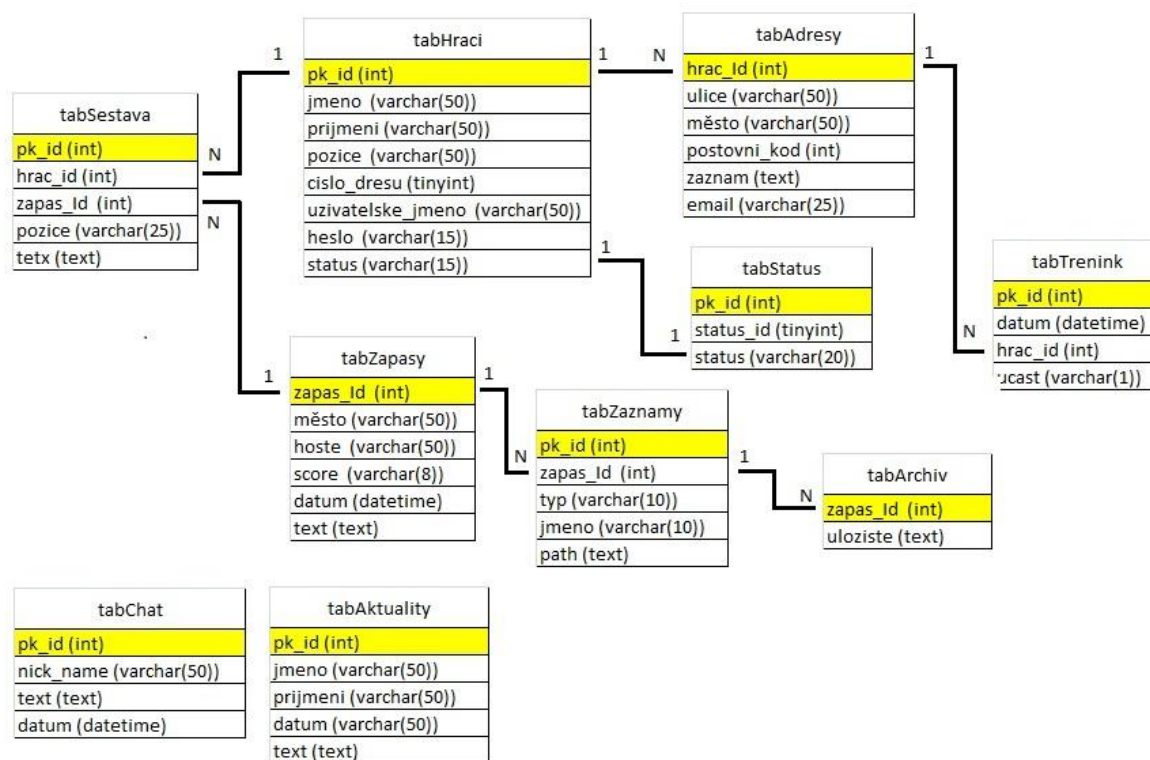
Vzhledem k velkému rozsahu činnosti databázového specialisty v daném čase byla omezena tvorba informačního systému zejména na databázovou strukturu s daty a několika objekty realizujícími vybrané požadované činnosti modelovaného informačního systému.

V následujících podkapitolách budou představeny výsledné struktury datových objektů (tabulek) v prostředí databáze MS SQL Server a provedeno zhodnocení návaznosti na logický model (datový model) navržený softwarovým analytikem.

Fyzický soubor s databází není možné z nepředvídané chyby načíst, proto se podíváme pouze na vzhled struktury databáze předaný dokumentací projektového týmu.

### 5.1 Informační systém sportovních aktivit

Datový model IS sportovních aktivit obsahuje **13** datových objektů, struktura fyzického modelu databáze obsahuje **10** datových objektů. Datový objekt zajišťující uchování informací o přihlašování uživatelů do systémů ve fyzickém modelu chybí. Takže požadavky na splnění evidence přihlašování nebudou splněny. Již v této chvíli je zřejmé, že logický a fyzický model informačního systému nejsou shodné a nenavazuje realizace fyzického modelu na model logický. Co se týká vazeb mezi datovými objekty tak logický model obsahoval naprosto jiné uspořádání struktury vazeb mezi tabulkami než model fyzický.



Obrázek 5.1 – Struktura IS sportovních aktivit

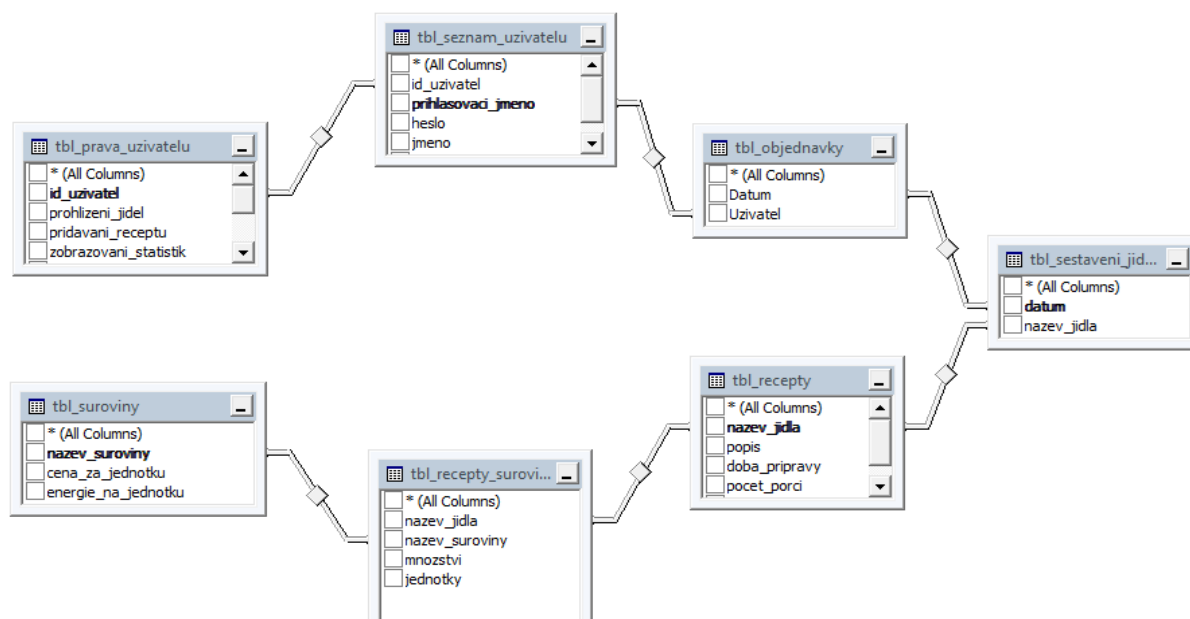
Shoda logického a fyzického modelu je důležitá pro další rozšiřitelnost systému a jeho dokumentaci. Je tedy možné zpětným inženýrstvím získat aktualizovaný logický model navazující na fyzický model a dále upravit a transformovat základní model požadavků na systém.

## 5.2 Informační systém pro administraci stravování

Datový model IS pro administraci stravování obsahuje 6 datových objektů, struktura fyzického modelu zahrnuje 7 datových objektů, ani v tomto IS není dodržen stejný počet datových objektů v logickém i fyzickém modelu.

Nelze vždy pohlížet pouze na počet a vazby mezi datovými objekty, neboť datový model zahrnoval objekt určený pro uložení informací o přihlašování uživatelů, zatímco ve fyzickém modelu se evidence přihlašování uživatelů neprovádí. Rozhodně je z hlediska realizace upravena vazba mezi datovým objektem **Recept** a **Surovina**, která byla nesprávně navržena a nyní je realizována s využitím odkazu na přídatnou tabulku **tbl\_recepty\_suroviny**.

Další změnou ve struktuře fyzického modelu v porovnání s logickým modelem je datový objekt s názvem **tbl\_objednavky**. Požadavek na evidenci objednávek jídel nebyl nikdy definován.

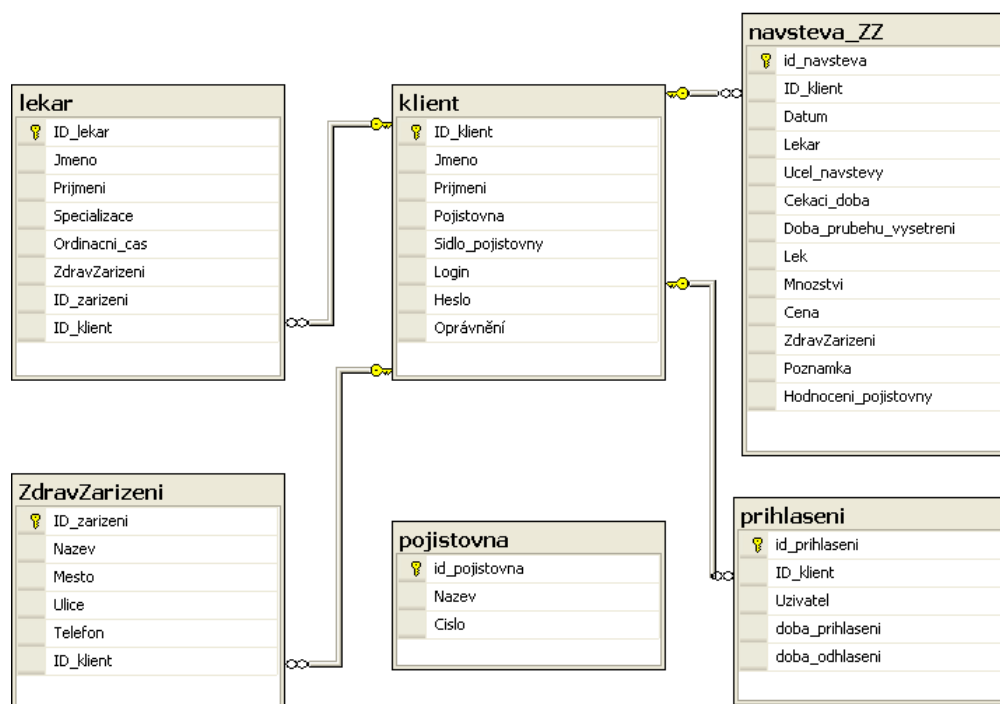


Obrázek 5.2 – Struktura IS pro administraci stravování

Stejně jako v předchozím IS sportovních aktivit i zde dochází k velkým nesrovnalostem ve fyzickém a logickém modelu, který je opět nepochopitelně pozměněn a realizován dle jiných než na počátku ve fázi návrhu sestavených požadavků.

### 5.3 Informační systém osobní agendy ve zdravotnictví

Realizace fyzického modelu sestavená databázovým specialistou jako jediným členem posledního projektového týmu s tematikou osobní agendy ve zdravotnictví je dána návrhem a tvorbou 6 datových objektů a vazeb mezi nimi (kromě datového objektu „pojistovna“).



Obrázek 5.3 – Struktura IS osobní agendy ve zdravotnictví

Struktura splňuje požadavky na integritu systému z pohledu primárních a cizích klíčů. Návaznost této struktury na předchozí logický model neexistuje. Přesto lze zpětným inženýrstvím vytvořit logický model v prostředí grafického nástroje Enterprise Architect.



### Shrnutí návrhu struktury informačního systému

Hodnocení práce databázových specialistů a komunikace v rámci projektového týmu je z pohledu spolupráce v týmu na velmi nízké úrovni. Jediným odůvodněním je zřejmě časové rozvržení vývoje informačního systému, kdy realizace IS zpravidla začíná v poslední fázi vývoje systému (když opomineme fázi testování). Postup není možné hodnotit kladně a proto je z toho zřejmé, že týmová spolupráce nesplnila očekávání.

Navržené struktury datových objektů naplnily pouze představy databázového specialisty, který rozhodně nekonzultoval vše podrobně s projektovým týmem, neboť by musely být naplněny požadavky na systém definované na začátku.

## 6 ZHODNOCENÍ TÝMOVÉ SPOLUPRÁCE PŘI NÁVRHU A REALIZACI INFORMAČNÍHO SYSTÉMU

V případové studii je posuzován přístup týmu a spolupráce mezi členy týmu. Vzhledem k dostupnému počtu zúčastněných osob na týmové spolupráci a výukových cílech byly rozděleny osoby do tří tematických okruhů neboli tří realizovaných informačních systémů: informační systém sportovních aktivit, informační systém pro administraci stravování a informační systém osobní agendy ve zdravotnictví.

Role projektového týmu byly v každém tematickém okruhu rozděleny na tyto členy projektu: systémový analytik, softwarový architekt, softwarový analytik a databázový specialista. Vzhledem k rozdělení nesourodého počtu 11 osob do tří tematických okruhů a jednotlivých rolí se poslední tým omezil o softwarového analytika a částečně jeho práci přebírá databázový specialista. Návaznost činnosti posledního týmu tak nebyla optimální. Rovněž z osobních důvodů odstoupily dvě osoby během vývoje informačních systémů a byly to role systémového analytika a softwarového architekta v posledním tematickém celku, který navrhoval informační systém osobní agendy ve zdravotnictví. V průběhu vývoje informačních systémů tyto osoby nebylo možné nahradit jinými osobami, neboť se jedná o proces výuky nikoliv o firemní vývoj IS. Proto posouzení posledního tematického okruhu nebylo možné plně porovnat s procesem vývoje a výsledky při návrhu a realizaci ostatních dvou informačních systémů.

Informační systém pro naši případovou studii by měl být navržen, implementován a realizován za období tří měsíců podle požadavků a představ zadavatele řešeného problému.

Zhodnocení celkového přístupu ke svým rolím v rámci projektového týmu je podloženo výsledky práce dílčích osob projektového týmu a je v rámci dílčích hodnocení dokumentováno v závěru jednotlivých kapitol. Činnost projektového týmu nebyla dobrá. Návaznost jednotlivých úkolů projektu je špatná. Pro lepší spolupráci odhaduji nezbytnost většího společného času nad projektem a zejména větší zájem ze strany osob projektového týmu, které nebyly patřičně motivovány k maximálnímu úsilí při návrhu IS.

Jako navrhovatel projektu nemohu soudit chybu nebo popis zadaných projektů, zda nevznikají problémy při chybách v návrhu IS ve vztahu k danému zadání. Tento fakt však vyvažuje poskytnutý čas a prostor ke komunikaci osob projektového týmu k diskuzi s odběratelem projektu, který nebyl zcela využit členy týmu. Nemalou měrou k tomu přispívá také nezájem o teoretickou přípravu znalostí práce jednotlivých členů týmu v rámci přednášek k předmětu, kde nebyla zpravidla plná účast všech osob. Jestliže role v týmu nebyly podloženy dobrými znalostmi o zpracovávané tématice daného specialisty, pak také nelze očekávat podnětnou komunikaci nad zpracovávaným návrhem IS v průběhu jeho vývoje.

Celkově je práce velmi individuální při zpracování projektu návrhu IS a zde je potřeba velkého zájmu ze strany dílčích rolí (osob, studentů) ve smyslu potřebné analýzy a zpracování všech dílčích částí v takovém rozsahu, aby byl takto navržený informační systém schopen převedení do reálného provozu. Tento zájem z jejich strany na zpracování funkčního systému v průběhu vývoje informačního systému nebyl zjevný. Rozhodně zpracování částí návrhu IS

pouze v poslední části vývoje IS je špatné. K tomu přispívaly částečné kontroly nad rozpracovanými návrhy a analýzami projektu, které byly často chybné a nesprávné.

Celkově je činnost návrhu IS v daném časovém období velmi složitou činností, která vyžaduje patřičné úsilí, dobrou teoretickou znalost vlastní specializace a samozřejmě také povrchní znalost ostatních specializací, na něž navazuje.

Rozhodně je v závěru nutné poukázat na nezbytné kroky, které je vhodné dodržovat v rámci spolupráce projektového týmu:

- ✚ Neustále podněcovat diskuzi členů jednotlivých projektových týmů v rámci možného časového prostoru, který mají k dispozici v rámci výuky.
- ✚ Provádět průběžné testování dílčích částí IS důrazněji než bylo uskutečněno.
- ✚ Zdůrazňovat hranice a kompetence dílčích rolí projektového týmu a povinnost návaznosti na předchozí část návrhu IS, jak dopřednou, tak i zpětnou.
- ✚ Podněcovat osoby projektového týmu k všeobecnému pohledu na návrh IS, který zajistí nejen zadané úkoly, ale celkovou funkčnost systému. Analytici IS musí mít schopnost rozboru všech možných i nemožných vlivů, které mohou na navrhovaný informační systém působit.
- ✚ Požadovat jednoduchost a průhlednost myšlenek návrhu IS, neboť věci složité jsou i složité na údržbu a další rozšiřování systému.

Závěr ze zhodnocení týmové spolupráce v rámci magisterského studia na vysoké škole je taková, že osoby v projektovém týmu jsou na různorodé úrovni zájmu k vykonání požadované práce, na různé rovině schopnosti k vykonávání takové činnosti, zejména samostatného myšlení a neméně důležitého úsilí k vykonané práci, za kterou jsou hodnoceni nikoliv finančně, ale pouze splněním dané povinnosti v rámci studia. Určitě je přístup velmi individuální a v posouzení práce týmů v minulých letech a v současnosti má pozvolna klesající tendenci ve výkonu a výsledcích jejich práce.



## Další zdroje

- FOWLER, M. 2009. *Destilované UML : knihovna programátora*. Praha: Grada Publishing, 2009. 173 s. ISBN 978-80-247-2062-3.
- KANISOVÁ, H. & MÜLLER, M. 2006. *UML srozumitelně*. Brno: Computer Press, 2006. 176 s. ISBN 80-251-1083-4.
- KISZKA, B. 2003. *UML a unifikovaný proces vývoje aplikací*. Brno: Computer Press, 2003. 387 s. ISBN 80-7226-947-X.
- PENDER, T. 2003. *UML bible*. Indianapolis: Wiley, 2003. 1120 s. ISBN 0-7645-2604-9.
- REJNKOVÁ, P. *Příklady použití diagramů UML 2.0* [online]. 2009 vyd. [cit. 2012-05-30]. Dostupné z: <http://uml.czweb.org/>.
- Richta, K. 2003. Unifikovaný modelovací jazyk UML. In: *System Integration 2003*. Praha: Česká společnost pro systémovou integraci, 2003. pp. 386-393. ISBN 80-245-0522-3.
- SCHMULLER, J. 2001. *Myslíme v jazyku UML*. Praha: Grada Publishing, 2001. 359 s. ISBN 80-247-0029-8.
- SPARX SYSTEMS. *Enterprise Architect* [online]. 2000-2012. [cit. 2012-05-30]. Dostupné z: <http://www.sparxsystems.com/>.
- UML Resource Page* [online]. 1997-2012. [cit. 2012-05-30]. Dostupné z: <http://www.uml.org>.