



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA PODNIKATELSKÁ  
ÚSTAV INFORMATIKY

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUTE OF INFORMATICS

## ANALÝZA A NÁVRH INFORMAČNÍHO SYSTÉMU PRO FIRMU ESPORTS.CZ, S.R.O.

THE DESIGN OF INFORMATION SYSTEM FOR COMPANY ESPORTS, S.R.O.

DIPLOMOVÁ PRÁCE

MASTERS'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. MICHAL KOBELKA

VEDOUCÍ PRÁCE  
SUPERVISOR

Ing. HANA KLČOVÁ, Ph.D.

BRNO 2015

## ZADÁNÍ DIPLOMOVÉ PRÁCE

**Kobelka Michal, Bc.**

---

Informační management (6209T015)

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů zadává diplomovou práci s názvem:

**Analýza a návrh informačního systému pro firmu eSports.cz, s.r.o.**

v anglickém jazyce:

**The Design of Information System for Company eSport.cz, s.r.o.**

Pokyny pro vypracování:

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrh řešení a jeho přínosy

Závěr

Seznam použité literatury

Seznam odborné literatury:

- BRUCKNER, Tomáš. Tvorba informačních systémů: principy, metodiky, architektury. 1. vyd. Praha: Grada, 2012, 357 s. Management v informační společnosti. ISBN 978-80-247-4153-6.
- CONNOLLY, Thomas. Database Systems: A Practical Approach to Design, Implementation, and Management. 4th ed. Harlow: Addison-Wesley, 2005, 1374 s. ISBN 03-212-1025-5.
- GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. Podniková informatika. 2., přeprac. a aktualiz. vyd. Praha: Grada, 2009, 496 s. Expert (Grada). ISBN 978-80-247-2615-1.
- MARAKAS, George M a James A O'BRIEN. Introduction to information systems. 16th ed. New York, NY: McGraw-Hill/Irwin, c2013, xxxii, 732 p. ISBN 00-775-0643-X.
- SODOMKA, Petr a Hana KLČOVÁ. Informační systémy v podnikové praxi. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 2010, 501 s. ISBN 978-80-251-2878-7.

Vedoucí diplomové práce: Ing. Hana Klčová, Ph.D.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2014/2015.

L.S.

---

doc. RNDr. Bedřich Půža, CSc.  
Ředitel ústavu

---

doc. Ing. et Ing. Stanislav Škapa, Ph.D.  
Děkan fakulty

V Brně, dne 28.2.2015

## **ABSTRAKT**

Diplomová práce se zabývá analýzou a návrhem informačního systému pro firmu eSports, s.r.o. První kapitola předkládá teoretická východiska nutná k pochopení celé problematiky. Druhá kapitola posuzuje současný stav společnosti eSports, s.r.o. a analyzuje hlavní firemní procesy. V poslední kapitole jsou předloženy návrhy datového modelu nového informačního systému.

## **ABSTRACT**

The master's thesis deals with analysis and design of information system for company eSports, s.r.o. First chapter presents theoretical basis which is necessary for understanding the problem. Second chapter assesses current situation of eSports, s.r.o. and analyzes main processes of the company. Last chapter provides proposals for data model of new information system.

## **KLÍČOVÁ SLOVA**

Datový model, informační systém, návrh databáze, MySQL, procesní analýza.

## **KEYWORDS**

Data model, information system, database design, MySQL, process analysis.

## **BIBLIOGRAFICKÁ CITACE**

KOBELKA, M. Analýza a návrh informačního systému pro firmu eSports.cz, s.r.o. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2015. 92 s. Vedoucí diplomové práce Ing. Hana Klčová, Ph.D.

## **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že předložená diplomová práce je původní a zpracoval jsem ji samostatně.

Prohlašuji, že citace použitých pramenů je úplná, že jsem ve své práci neporušil

Autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském a o právech souvisejících s právem autorským).

V Brně dne 12. května 2015

.....

Michal Kobelka

## **PODĚKOVÁNÍ**

Na tomto místě bych rád poděkoval vedoucí své diplomové práce, Ing. Haně Klčové, Ph.D., za odborný dohled a pomoc při její tvorbě. Rovněž děkuji firmě eSports, s.r.o. za umožnění pracovat na diplomové práci a za poskytnuté informace.

# Obsah

Úvod .....	11
Cíl a metodika práce .....	12
1 Teoretická východiska .....	13
1.1 Data, informace, znalosti a vědění.....	13
1.1.1 Data .....	13
1.1.2 Informace.....	14
1.1.3 Znalosti.....	14
1.1.4 Vědění .....	15
1.2 Systém a informační systém.....	15
1.2.1 Systém.....	15
1.2.2 Informační systém.....	16
1.3 Typy informačních systémů .....	17
1.3.1 Informační systémy pro podporu provozu .....	19
1.3.2 Informační systémy pro podporu řízení .....	20
1.4 Databázový systém .....	21
1.4.1 Databáze .....	22
1.4.2 Systém řízení báze dat.....	23
1.4.3 Databázové aplikační programy .....	24
1.4.4 Výhody a nevýhody SŘBD .....	24
1.5 Datové modelování.....	26
1.5.1 Datový model.....	26
1.6 Analytické metody použité v práci.....	31
1.6.1 PEST analýza.....	31
1.6.2 Porterův model pěti konkurenčních sil.....	31
1.6.3 Analýza 7S.....	32
1.6.4 SWOT analýza.....	32
2 Analýza současného stavu.....	33
2.1 Základní údaje o společnosti.....	33
2.1.1 Hlavní činnosti firmy .....	33



2.1.2	Projekt Onlajny .....	34
2.2	Organizační struktura společnosti .....	34
2.3	Analýza společnosti .....	36
2.3.1	PEST analýza .....	36
2.3.2	Porterův model pěti konkurenčních sil .....	38
2.3.3	Analýza 7S .....	40
2.3.4	SWOT analýza .....	41
2.4	Hlavní firemní procesy .....	42
2.4.1	Hodnototvorné procesy .....	43
2.4.2	Podpůrné procesy .....	47
2.5	Firemní informační systémy a aplikace .....	55
2.5.1	Interní firemní informační systém .....	55
2.5.2	Trello .....	55
2.5.3	Interní tiketovací systém .....	57
2.5.4	Informační systém pro projekt Onlajny .....	57
2.6	Shrnutí analytické části .....	57
3	Návrh řešení .....	58
3.1	Modul uživatelů .....	59
3.1.1	Tabulka user .....	59
3.1.2	Tabulka user_address .....	60
3.1.3	Tabulka user_bank_account .....	60
3.1.4	Tabulky user_positions a user_to_position .....	61
3.1.5	Tabulky user_role, user_role_rights, user_roles_to_right a user_to_role ...	62
3.1.6	Shrnutí modulu uživatelů .....	64
3.2	Modul klientů .....	65
3.2.1	Tabulka client .....	65
3.2.2	Tabulka client_address .....	65
3.2.3	Tabulky client_group a client_to_group .....	66
3.2.4	Tabulka client person .....	67
3.2.5	Shrnutí modulu klientů .....	69
3.3	Modul projektů .....	70
3.3.1	Tabulka project a user_to_project .....	70

3.3.2	Tabulka project_status.....	71
3.3.3	Tabulky project_group a project_to_group .....	71
3.3.4	Tabulky project_label a label_to_project .....	72
3.3.5	Tabulky project_discussion a project_comment.....	73
3.3.6	Shrnutí modulu projektů.....	75
3.4	Modul tiketů .....	76
3.4.1	Tabulka ticket a user_to_ticket .....	76
3.4.2	Tabulka ticket_category a ticket_status .....	77
3.4.3	Tabulka ticket_comment .....	78
3.4.4	Shrnutí modulu tiketů.....	78
3.5	Modul diskusí.....	80
3.5.1	Tabulka discussion .....	80
3.5.2	Tabulka discussion_comment.....	80
3.5.3	Tabulky discussion_category a discussion_to_category .....	81
3.5.4	Shrnutí modulu diskusí.....	82
3.6	Ekonomické zhodnocení .....	83
4	Závěr .....	85
	Seznam použitých zdrojů.....	86
	Seznamy.....	90
	Seznam tabulek .....	90
	Seznam obrázků .....	91
	Seznam použitých zkratk .....	92
	Seznam příloh.....	92
	Přílohy .....	I

## Úvod

Informační systém je dnes nepostradatelnou součástí mnoha podniků. Informační systémy jsou dnes důležité téměř na všech úrovních podniku, ať už se jedná o informační systém pro podporu operativních činností či manažerský informační systém podporující rozhodování managementu. Informace dnes hrají velmi důležitou roli a jsou nástrojem konkurenční výhody. Informační systémy pomáhají tyto informace uchovávat a dále s nimi pracovat.

Společnost eSports, s.r.o. používá několik informačních systémů a podpůrných programů pro svoji činnost. Byly vyvinuty téměř před 10 lety a dnes již nevyhovují technologickým standardům. Společnost se proto rozhodla nahradit tyto informační systémy a programy novým informačním systémem. Nový informační systém bude integrovat funkce ze stávajících informačních systémů, takže společnost nebude muset používat více zastaralých programů, vystačí si pouze s jedním informačním systémem.

Tato diplomová práce se zabývá vývojem informačního systému pro společnost eSports, s.r.o., konkrétně návrhem datového modelu pro nový informační systém. Je rozdělena na tři části. V první části jsou předloženy teoretické poznatky nutné k pochopení celé problematiky, ať už vývoje informačního systému či návrhu datového modelu. Ve druhé části jsem analyzoval hodnototvorné a podpůrné firemní procesy a analyzoval jsem informační systémy a programy, které firma aktuálně používá při své činnosti. Ve třetí části jsem navrhl datový model nového informačního systému s přispěním poznatků z předchozích dvou částí.

## **Cíl a metodika práce**

Cílem této diplomové práce je analýza firemních procesů a požadavků společnosti eSports, s.r.o. a návrh datového modelu nového informačního systému, který bude splňovat zadané požadavky.

Společnost eSports, s.r.o. využívá k podpoře své činnosti několik jednotlivých informačních systémů a podpůrných programů, které nejsou nijak integrovány a pracují tedy zcela odděleně. Cílem diplomové práce je navrhnout datový model informačního systému, který bude integrovat funkce těchto programů a informačních systémů tak, aby společnost mohla využívat pouze jediný informační systém.

Díličím cílem je analýza jednotlivých programů a informačních systémů, které firma používá a společně s požadavky navrhnout takové funkce, které budou obsaženy v navrženém informačním systému.

V případě firemních procesů budou identifikovány a analyzovány hlavní a podpůrné procesy, a to pomocí EPC diagramů (Event-driven Process Chain) a matic odpovědnosti RACI (Responsible, Accountable, Consulted, and Informed).

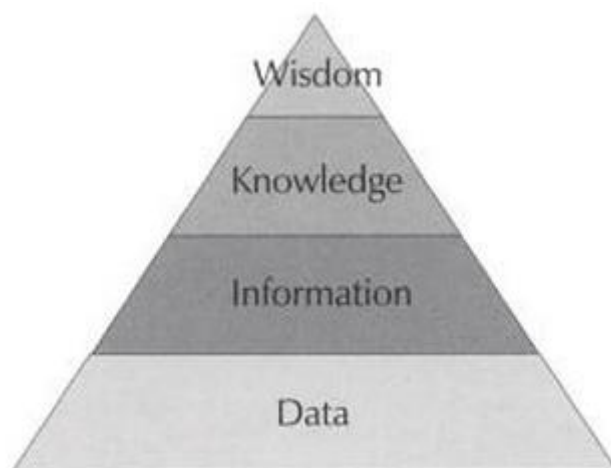
Datový model informačního systému bude navržen pomocí relačního systému řízení báze dat MySQL, a to v podobě tabulek a vazeb mezi těmito tabulkami. Výsledkem bude funkční datový model informačního systému, který bude dále použit při vývoji informačního systému.

# 1 Teoretická východiska

V kapitole teoretických východisek popíšu teoretické základy problematiky řešené v této diplomové práci. Jedná se zejména o poznatky z oblasti informačních systémů, databázového návrhu informačních systémů a metodik analýzy informačních systémů.

## 1.1 Data, informace, znalosti a vědění

Jsou-li pro informační systém základním prvkem informace a práce s nimi, je nutné nejdříve pochopit pojmy data, informace, znalosti a vědění. Ačkoliv se může na první pohled zdát, že všechny tyto pojmy vyjadřují shodné věci, je nutné mezi nimi rozlišovat. Například Wilson (1, s. 113) popisuje vztah mezi těmito pojmy jako hierarchický, což je ilustrováno na obrázku č. 1:



Obr. č. 1: Data, informace, znalosti, vědění  
(Zdroj: 1, s. 113)

Hierarchické uspořádání v podobě pyramidy pak naznačuje, že se postupně z dat stávají informace, z informací znalosti a ze znalostí v poslední fázi vědění. Podobný pohled na uspořádání nabízí Clark (2), který však danou problematiku znázornil formou řetězce.

### 1.1.1 Data

V literatuře se objevuje hned několik významů pojmu data. Například Awad popisuje data jako „*neorganizovaná a nezpracovaná fakta*“ (3, s. 60). Wilson se domnívá, že data se skládají z „*obrazů, zvuků, digitálních přenosů apod., jsou bezkontextová a je těžké jim*

porozumět“ (1, s. 112). Wang data chápe jako „*soubor nezávislých, oddělených faktů, měření, znaků, čísel a symbolů*“ (4, s. 38).

Ačkoliv každý autor popisuje data trochu jinak, základ zůstává stejný: data nemají pro člověka hodnotu, dokud se nedají do kontextu pro jejich správné pochopení. Tedy dokud se data nestanou informacemi.

### **1.1.2 Informace**

Wilson tedy přeměnu dat na informaci vysvětluje následovně: „*data mají malou hodnotu, dokud nejsou smysluplně interpretována, například členěním, filtrováním nebo sumarizací*“ (1, s. 112). Jinými slovy se tímto procesem data stávají informacemi.

Awad (3, s. 60) poukazuje na to, že informace je odvozena od slova *inform*, které znamená „dát něčemu tvar“. Informace je podle něj tedy formování dat tak, aby v očích příjemce tato data nabyla významu.

Wang tvrdí, že „*informace jsou výsledek, který lidé obdrží poté, co shromáždí, zorganizují, upraví a zanalyzují surová data*“ (4, s. 40). Dodává, že smyslem tohoto procesu je ze surových dat vytvořit strukturovaný, uspořádaný a související produkt.

### **1.1.3 Znalosti**

Zatímco transformace dat na informace je pro většinu autorů snadno popsatelná a výklad o tomto procesu je víceméně shodný, přechod z informací ke znalostem již tak jednoznačný není. Awad říká, že znalosti mají různý význam podle toho, v jaké disciplíně se jich užívá (3, s. 61). Dodává, že jsou podle něj znalosti „*lidské chápání specializované oblasti zájmu, které bylo získáno studováním a zkušenostmi*“. Hlavním způsobem, jak získat znalosti z některé oblasti, pak přisuzuje zejména učení, myšlení a důkladnému seznámení se s danou problematikou.

Wang souhlasí s tím, že vysvětlit rozdíl mezi informacemi a znalostmi je obtížné (4, s. 41). Poukazuje na to, že znalosti nejsou jen složitější verzí informací, ale liší se i ve způsobu použití. Dodává také, že znalosti jsou použitelné opakovaně, zatímco data a informace se týkají konkrétního problému.

### 1.1.4 Vědění

Vědění v pomyslné pyramidě stojí nejvýše, tedy nad daty, informacemi i znalostmi. Awad chápe vědění jako nejvyšší stupeň abstrakce, která zahrnuje vizi, prozíravost a schopnost „vidět za horizont“ ve smyslu překonávání vlastních limitů (3, s. 64). Wang říká, že vědění jako takové nemůže být tvořeno stejným způsobem, jako data a informace, a nelze jej ani sdílet s ostatními lidmi jako znalosti (4, s. 43). Dodává také, že vědění lze u ostatních ocenit, ale vytvářet jej musíme hlavně pro sebe; nevzniká totiž přirozeně či náhodně, ale úmyslně a cílevědomě (4, s. 43). Wilson říká, že věděním často označujeme znalosti zkombinované se zkušenostmi a intuicí (1, s. 113).

V následující tabulce (tabulka č. 1) jsou shrnuty informace o úrovni, rozlišení, procesu učení a výsledku předchozích kapitol data, informace, znalosti a vědění:

**Tabulka č. 1: Data, informace, znalosti, vědění**  
(Zdroj: vlastní zpracování dle 5)

Úroveň	Rozlišení	Proces učení	Výsledek
Data	Surová fakta	Sběr dat	Ukládání dat
Informace	Smysluplná, užitečná data	Přiřazení formy a funkčnosti	Porozumění
Znalosti	Jasně pochopení informací	Analýza a syntéza	Pochopení
Vědění	Využití znalostí k vytvoření a dosažení cílů	Posouzení a přijímání odpovídajících opatření	Lepší živobytí, úspěchy

## 1.2 Systém a informační systém

Existuje mnoho definic pojmu *systém* a *informační systém*. Je důležité pochopit, že ačkoliv moderní informační systémy začaly postupně vznikat až v druhé polovině dvacátého století, systém jako takový existoval mnohem dříve, než o něm lidé vůbec začali přemýšlet a pojmenovali ho slovem *systém*.

### 1.2.1 Systém

Podle Gály je systém „úcelově definovaná neprázdná množina prvků a množina vazeb mezi nimi, přičemž vlastnosti prvků a vazeb mezi nimi určují vlastnosti (chování) celku“ (6 s. 23). Brukner dodává, že systém je definován jako celek, který je tvořený svojí

celistvostí a souhrnem částí a jejich vzájemných vztahů, které jsou velmi často dynamické (7, s. 14).

Podle Hronka je systém definován jako „*organizovaná množina myšlenek, principů, doktrín, seskupená za účelem vysvětlení vnitřního uspořádání nebo činnosti celku*“ (8, s. 9). Dodává také, že je to „*množina komponent, která interaguje, aby splnila nějaký cíl*“ (8, s. 9).

Oz systém definuje jako „*řadu komponent, které spolupracují k dosažení společného cíle (nebo více cílů) tím, že přijímají vstup, zpracují ho a vyprodukují výstup organizovaným způsobem*“ (9, s. 11).

Obecně lze tedy říci, že systém je soubor nějakých prvků spolu s vazbami mezi těmito prvky. Navenek pak systém vystupuje jako celek.

Gála dále identifikuje jednotlivé prvky systému:

- **účel systému** – cílové chování systému,
- **strukturu systému** – prvky systému a vazby mezi nimi,
- **vlastnosti prvků** – významné pro celkové chování systému,
- **vlastnosti vazeb** – mezi jednotlivými prvky systému,
- **okolí systému** – vymezení prvků, které již nepatří do systému, ale které tento systém významně ovlivňují,
- **subsystémy** – pokud zkoumání systému jako celku je složité (6, s. 23).

### 1.2.2 Informační systém

Brukner definuje informační systém takto: „*účelem informačního systému je zajištění správných informací na správném místě a ve správný čas*“ (7, s. 15). Podle Hirschheima (10, s. 11) existují na informační systémy dva základní způsoby náhledu: strukturální hledisko a funkční hledisko.

- **Strukturální hledisko:** informační systém sestává z lidí, procesů, dat, modelů, technologií a částečně formálního jazyka. Formuje soudržnou strukturu, která slouží jistému účelu nebo funkci organizace (10, s. 11).



- **Funkční hledisko:** Informační systém je technologicky implementované médium pro zaznamenávání, uchovávání a šíření informací (10, s. 11).

S těmito hledisky souhlasí i Marakas. Podle něj může být informačním systémem „*každá organizovaná kombinace lidí, hardwaru, softwaru, komunikačních sítí, datových zdrojů, zásad a procedur, které ukládají, získávají, transformují a šíří informace v organizaci*“ (11, s. 6).

Marakas se také přiklání k názoru, že ačkoliv jsou dnešní informační systémy spojované nejvíce s počítači, v jiné formě je lidstvo používalo již od svých počátků a používá je dodnes. Jako příklad uvádí **kouřové signály** (pro komunikaci s lidmi, kteří nebyli v doslechu a byli příliš daleko, aby viděli odesílatele) nebo **knihovnické systémy** založené na bázi karetních katalogů, které uchovávají data o knihách v organizované podobě tak, aby je návštěvník mohl vyhledat co nejjednodušeji (11, s. 6).

Důvody, proč se informační systémy staly pro podniky tak důležité a proč se velmi rozšířily, předkládá Oz:

- **Výpočetní výkon počítačů** se mnohonásobně zvýšil, zatímco ceny se neustále snižují.
- **Kapacita datových úložišť** se postupně zvyšuje, zatímco ceny se snižují.
- Zvýšila se **variabilita úkonů**, ke kterým můžeme počítače použít.
- **Rychlá a spolehlivá komunikace** přes internet je cenově i geograficky dostupnější.
- **Rapidní růst internetu** otevřel nové možnosti a podpořil soutěž na globálních trzích.
- Stále větší část pracovníků je **počítačově gramotných**. (9, s. 15)

### 1.3 Typy informačních systémů

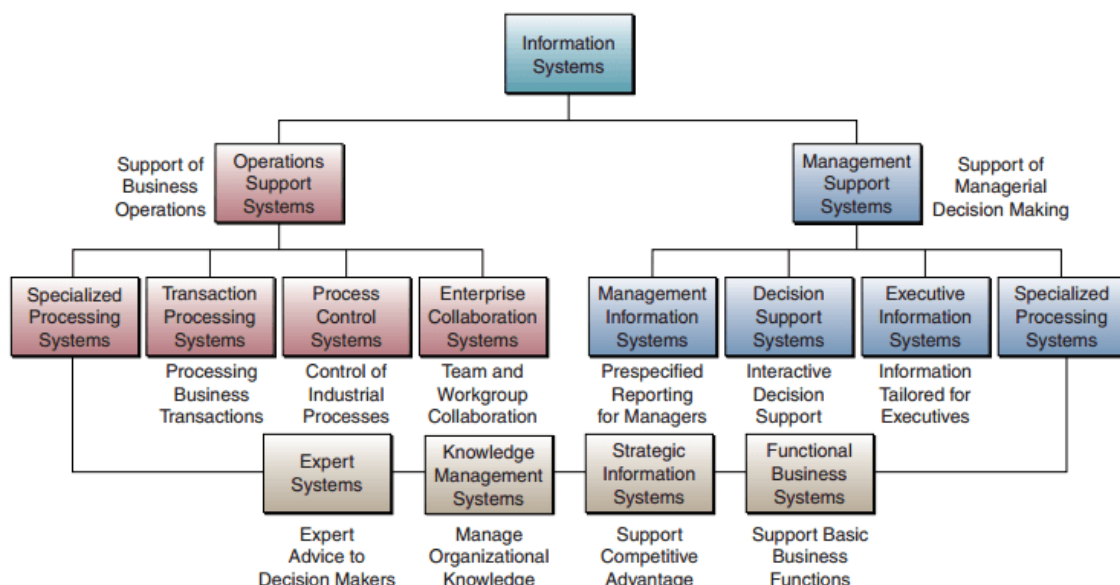
Různé informační systémy slouží různým účelům. Jiné potřeby mají odlišné typy podniků či odlišná oddělení v rámci jednoho podniku. Požadavky na informační systém se liší podle obchodní potřeby nebo podle úrovně řízení. Jiné obchodní společnosti mají jiné

úkoly, strukturu, zájmy či přístupy. Informační systémy se však dají obecně rozdělit podle úrovně systémové složitosti a typu funkcí, které zastává (9).

Existují systémy, jejichž úkolem je pouze jediná činnost, ale také ty, které slouží hned několika účelům zároveň. Informační systémy vytvořené pro jediný účel sloužily hlavně v minulém století, při jejich postupném vzniku. Neexistovala dostupná řešení, která by podniky mohly přijmout a začít používat při řešení svých problémů. Proto se velmi často vyvíjely informační systémy tak, aby sloužily pouze jedné společnosti a řešily jeden konkrétní problém. Postupem času společnosti zjišťovaly, že vyvíjené informační systémy mají některé společné charakteristiky. To byl první impuls k tomu, aby v budoucnu vznikl jeden informační systém, který by řešil podobné a opakující se problémy. Bylo však nutné definovat, jak a kde tento informační systém používat a k čemu by vlastně sloužil. Proto vznikly snahy o klasifikaci informačních systémů, aby společnosti měly jasnou představu o tom, co potřebují (12).

Jak upozorňuje sám Marakas (11), a přidává se k němu i Kimble (12), neexistuje jediná klasifikace informačních systémů. Informační systémy lze dělit podle různých hledisek či kritérií a neexistuje jediné správné řešení tohoto rozdělení.

Marakas tedy uvažuje o následujícím rozdělení informačních systémů:



**Obr. č. 2: Rozdělení informačních systémů**  
(Zdroj: 11, str. 13)

Informační systémy Marakas rozděluje do dvou větví: **Operations Support Systems** (informační systémy pro podporu provozu) a **Management Support Systems** (informační systém pro podporu řízení). Logicky tedy rozdělil informační systémy podle toho, zda slouží spíše výrobnímu procesu, nebo zda je ke svému účelu využívá management podniku.

### **1.3.1 Informační systémy pro podporu provozu**

Tyto systémy slouží zejména k podpoře obchodních operací a generují velké množství informací pro interní i externí použití. Nejsou to však informace, které by mohli použít přímo manažeři společnosti – k tomu je vyžadováno další zpracování v manažerském informačním systému (11, s. 13).

Hlavními úkoly informačních systémů pro podporu provozu jsou podle Marakase *„zpracovávat obchodní transakce, řídit průmyslové procesy, podporovat podnikovou komunikaci a spolupráci a zajišťovat efektivní aktualizaci firemních databází“* (11, s. 13).

Ke každé z těchto činností slouží jiný druh informačního systému, Marakas je tedy rozděluje na transakční systémy, systémy řízení procesů a systém pro podnikovou spolupráci (11).

#### **1.3.1.1 Transakční systém**

Obchodní transakci definuje Bernstein takto: *„Obchodní transakce je interakce v reálném světě, obvykle mezi podnikem a člověkem nebo podnikem a jiným podnikem, přičemž dochází k určité výměně.“* (13, s. 1) Výměnou pak rozumí např. výměnu peněz, produktů či informací.

Transakční systémy (TPS – Transaction processing systems) zaznamenávají a zpracovávají data, která vznikají obchodními operacemi. Pracují dvěma způsoby – dávkovým zpracováním a v reálném čase. Dávkové zpracování nashromáždí data za určitý časový úsek a zpracovává je v intervalech. Druhý způsob, tedy zpracování v reálném čase, probíhá ihned poté, co nastala obchodní operace (11).

### **1.3.1.2      Systém řízení procesů**

Systém řízení procesů (PCS – Process control system) sleduje a řídí fyzikální procesy (11, s. 13). Podle Patricka se systémy řízení procesů používají hlavně v průmyslové výrobě, kde řídí různé proměnné či veličiny, které mají jistý vliv na kvalitu výrobku, který v podniku vzniká. Těmito veličinami může být například tlak, teplota či hladina a průtok systémem (14, s. 21).

### **1.3.1.3      Systém pro podnikovou spolupráci (groupware)**

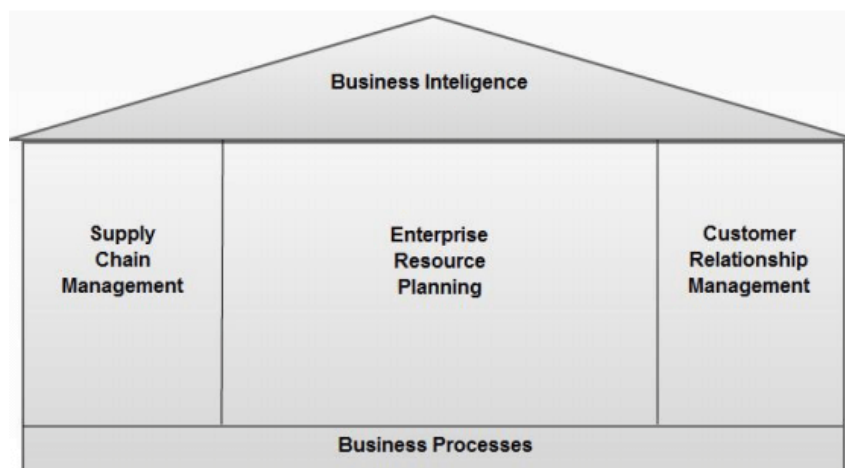
Bagad definuje systém pro podnikovou spolupráci (ECS – Enterprise collaboration system) jako „*typ informačního systému, který využívá různé informační technologie k tomu, aby lidé mohli pracovat spolu*“ (15, s. 76). V obchodních organizacích se využívá těchto informačních technologií ke zlepšení produktivity a kreativity u pracovníků (15). Dnes se pro tyto informační systémy používá častěji název *groupware*.

## **1.3.2      Informační systémy pro podporu řízení**

Druhá větev informačních systémů dle Marakase podporuje řízení a rozhodování. Hlavním úkolem tedy již není získávání dat, ale jejich smysluplné využití při rozhodování nebo při řízení podniku. Mezi tyto informační systémy Marakas řadí manažerské informační systémy, systémy pro podporu rozhodování a exekutivní informační systémy (11, s. 13).

Jiný pohled na rozdělení informačních systémů mají Sodomka a Klčová (16). Podle nich se podnikový informační systém dělí podle holisticko-procesní klasifikace na:

- ERP jádro, zaměřené na řízení interních podnikových procesů,
- CRM systém obsluhující procesy směřované k zákazníkům,
- SCM systém řídicí dodavatelský řetězec; jeho součástí bývá systém APS sloužící k pokročilému plánování,
- MIS – manažerský informační systém, který sbírá data z ERP, CRM a APS/SCM. (16, s. 77).



**Obr. č. 3: Holisticko-procesní pohled na podnikové IS**  
(Zdroj: 16, s. 78)

S tímto rozdělením souhlasí i Basl, který ve své knize píše: „ERP představuje jádro podnikového informačního systému, které spolu s aplikacemi SCM, CRM a BI tvoří rozšířené ERP, resp. ERP II.“ (17, s. 66)

### **1.3.2.1 Manažerský informační systém**

Brukner definuje manažerský informační systém takto: „MIS: Management Information System je blok orientovaný na řízení podniku na taktické úrovni, která zahrnuje ekonomická, organizační a obchodní hlediska.“ (7, s. 258)

Podle Laudona jsou manažerské informační systémy specifickou kategorií, které slouží především střednímu managementu. Říká, že tyto systémy poskytují manažerům různá hlášení a informace o aktuálním výkonu podniku; tyto informace pak manažeri využívají pro monitorování a řízení podniku a k předpovídání budoucích výkonů. (18, s. 47)

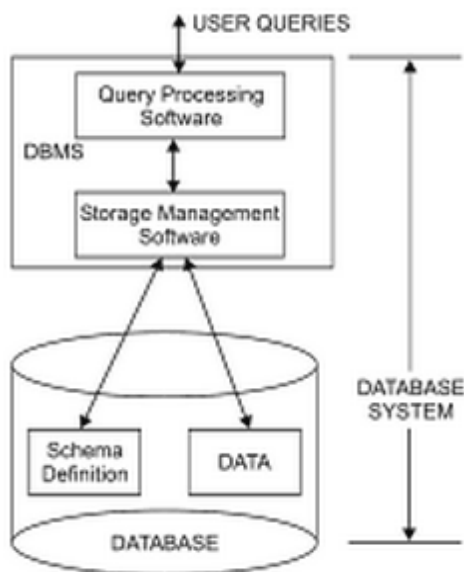
Součástí MIS bývají i systémy pro podporu rozhodování (DSS: Decision support system). Ty dříve plnily samostatnou úlohu, nyní jsou však integrovanou součástí MIS. DSS poskytuje informační podporu pro manažery při rozhodovacím procesu (11).

## **1.4 Databázový systém**

Databázové systémy a databáze jsou dnes, stejně jako informační systémy, pro podniky jedním z nezbytných faktorů. Ačkoliv to pro podnik nemusí být na první pohled patrné a

pro spoustu podniků je otázka databází zbytečná díky jejich integraci v informačních systémech, většina dat a informací je uložených právě v nejrůznějších typech databází.

Databázový systém se skládá ze dvou hlavních částí – databáze a speciálních programů, které řídí přístup k této databázi. Programy, které zajišťují řídicí činnost, se obecně nazývají *systém řízení báze dat* (SŘBD, anglicky DBMS: database management system) a poskytují zejména spojení uživatele a uložených dat (19, s. 1).



**Obr. č. 4: Databázový systém**  
(Zdroj: 19, s. 1)

Na obrázku č. 4 je znázorněn databázový systém tak, jak ho chápe Gill. Systém řízení báze dat (DBMS) reaguje na dotazy od uživatele (user queries) pomocí speciálního softwaru, který zpracovává uživatelské dotazy (Query Processing Software). Ten pak dále komunikuje s programem, který zajišťuje správu ukládání dat (Storage Management Software). Na základě uživatelského dotazu provede tento program potřebné operace s daty, které jsou uloženy v databázi, a ke kterým má přístup (19, s. 1).

#### 1.4.1 Databáze

Connolly databázi definuje jako „sdílenou kolekci logicky propojených dat a popis těchto dat, navržených tak, aby splňovaly informační potřeby organizace“ (20, s. 15).

Garcia-Molina chápe databázi jako kolekci informací, které existují po dlouhé časové období, často i několik let (23, s. 1). Podle Connollyho jsou dnes databáze tak běžnou součástí lidských životů, že si lidé ani neuvědomují, když nějakou databázi používají (20, s. 4).

Gill definuje databázi jako „*kolekci logicky propojených dat, které je možno zaznamenat*“ (19, s. 1). Dodává také podmínky, která platí pro databázi a data v ní uložená:

- Databáze musí reprezentovat hledisko reálného světa. Jinými slovy – data v databázi by se měla týkat např. reálné společnosti, podniku apod.
- Databáze musí být logicky koherentní a data v ní obsažená musí mít dobře pochopitelný vnitřní význam (musí být sémantická).
- Úložiště dat musí být navrženo a vyvinuto pro konkrétní účel. (21, s. 2)

#### 1.4.2 Systém řízení báze dat

Gill popisuje systém řízení báze dat jako „*sadu programů, které slouží k definování vytváření, údržbě a manipulaci s databází*“ (19, s. 2). Dále také definuje funkce, které by měla systém řízení báze dat zprostředkovávat:

- **Definice schéma databáze** – zejména definování datových typů, vztahů mezi daty a integritních omezení.
- **Manipulace s databází** – zejména vkládání nových dat a úprava, mazání a čtení stávajících dat.
- **Sdílení databáze** – systém řízení báze dat by měl zajistit současný přístup k databázi více než jedním uživatelem, aniž by to ovlivnilo konzistenci databáze.
- **Ochrana databáze** – zejména ochrana proti neautorizovanému či škodlivému přístupu.
- **Obnovení databáze** – pokud došlo k závažným problémům, pádům apod. (19, s. 2)

Connolly dodává, že systém řízení báze dat je software, který komunikuje s aplikačním programem uživatele a databází. Definuje trojici funkcí, které by měl systém řízení báze dat zprostředkovávat:

- **Definice databáze** pomocí jazyka pro definici dat a jejich vazeb (DDL: Data Definition Language).
- **Operace nad daty** (vkládání, upravování, získávání, mazání) pomocí jazyka pro manipulaci s daty (DML: Data Manipulation Language).
- **Řízený přístup k databázi** – zabezpečovací systém, integritní systém, systém řízení přístupu více uživatelů. (20, s. 16)

### 1.4.3 Databázové aplikační programy

Connolly definuje databázový aplikační program jako „*počítačový program, který interaguje s databází tím, že zadává příslušné požadavky (typicky SQL příkaz) systému řízení báze dat*“ (20, s. 17). Dodává, že uživatel pracuje s databází pomocí několika aplikačních programů, pomocí kterých databázi vytváří, spravuje a získává z ní informace.

### 1.4.4 Výhody a nevýhody SŘBD

Rob i Connolly se shodují, že použití databáze s sebou přináší mnohé výhody. Díky systému řízení báze dat je totiž zajištěno hned několik podstatných výhod oproti jiným systémům uložení informací.

- **Vylepšené sdílení dat** – uživatelé databáze mají lepší přístup k datům, která jsou lépe uspořádána. Díky tomuto přístupu může uživatel mnohem rychleji reagovat na změny (22, s. 8). Connolly dodává, že v podniku většinou data neslouží pouze jednomu účelu či jedinému uživateli, ale že jsou tato data sdílena napříč organizací (20, s. 27).
- **Lepší datová integrita** – je zajištěna validita a konzistence dat uložených v databázi. Tato integrita je většinou vyjádřena formou různých omezení, což je sada pravidel, která uživatel při práci s databází nemůže porušit. Tato omezení se můžou vztahovat buď na jediný záznam, nebo na vztah mezi dvěma záznamy (20, s. 27).
- **Větší bezpečnost** – databáze je chráněna před neautorizovaným přístupem. Uživatel je povinen prokázat se uživatelským jménem a heslem a jeho práva jsou velmi často omezena pouze na některé operace (20, s. 27).



- **Prosazování standardů** – databázové systémy integrují různé standardy a doporučení, od standardizovaných datových formátů, pojmenování, dokumentace či přístupových práv (20, s. 27).
- **Úspory z rozsahu** – sloučení všech podnikových dat do jedné databáze může být pro podnik levnější a výhodnější (20, s. 28).
- **Lepší přístupnost a schopnost reagovat** – díky integraci dat z celého podniku jsou tato data lépe přístupná pro uživatele. Většinu informací je uživatel schopný získat vlastním přičiněním, aniž by vyžadoval zásah programátora či jiného experta (20, s. 28).
- **Zvýšená produktivita** – s přihlédnutím na všechny předchozí výhody je zřejmé, že uživatel dosáhne vyšší produktivity, pokud je mu k datům usnadněn přístup a dokáže snáze tato data používat.

Kromě velké řady výhod však Connolly poukazuje i na nevýhody systému řízení báze dat a databázového systému obecně:

- **Složitost** – databázové systémy jsou velmi složitý software. Návrháři, vývojáři, administrátoři i koncoví uživatelé musí být seznámeni s funkcionalitou systému, aby mohli naplno využít jejich potenciál. Špatné pochopení systému může vést ke špatným rozhodnutím, což může mít vážné následky pro celý podnik (20, s. 29).
- **Velikost** – komplexnost a velký záběr funkcionality dělá z databázového systému poměrně velký software, který může zabírat velké množství místa na pevném disku. Toto byl problém zejména v minulosti; dnes se díky mnohonásobně větším úložištím naštěstí nejedná o velký problém (20, s. 29).
- **Cena databázového systému** – s rostoucími požadavky na velikost a obsáhlost systému úměrně roste i cena systému. Zatímco jednoduchý software pro domácí použití se dá dnes získat i zdarma, popř. za velmi nízký poplatek, velké databázové systémy mohou podle Connollyho stát i více než milion dolarů. Upozorňuje i na fakt, že cenu zvyšuje i každoroční poplatek za správu databáze (20, s. 30)

## 1.5 Datové modelování

Datové modelování je podle Gao způsob, jakým se strukturují a organizují data a je velmi často aplikováno ve velkém množství různých oborů (23). Ponniah dodává, že „*datové modelování je nedílnou součástí procesu návrhu a vývoje datového systému*“ (24, s. 4).

Podle Johnsona je datové modelování „*proces zjišťování, jak ukládat digitalizované informace v logicky uspořádané počítačové databázi*“ (25, s. 3). Doplňuje, že se jedná o technický proces, který zahrnuje porozumění, identifikaci a transformaci podnikových informací na logické objekty, které lze ukládat v databázi.

Witt dodává, že „*zatímco datové modelování je návrhový obor, datový model musí splnit jisté obchodní požadavky. Zjednodušeně můžeme datové modelování považovat za spojení dvou disciplín: analýzy (obchodních požadavků) a návrhu (které reagují na tyto požadavky.*“ (26, s. 7).

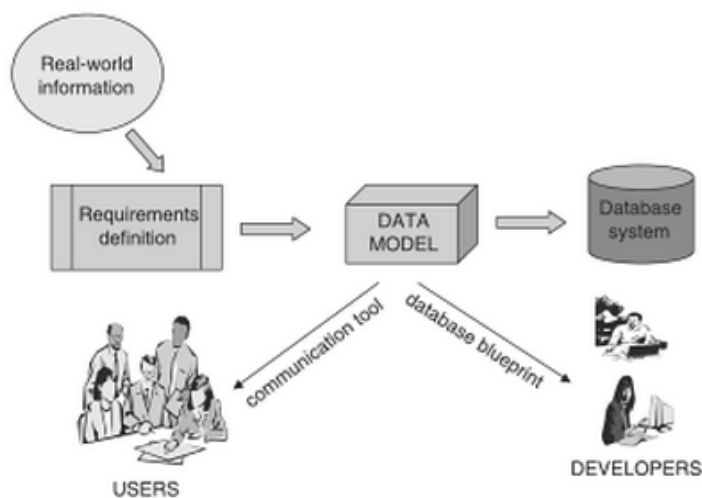
Ponniah dále tvrdí, že „*datové modelování poskytuje metody a prostředky pro požadavky na informace reálného světa tak, aby je pochopily zainteresované strany v organizaci. To znamená, že datové modelování umožňuje databázovým expertům použít tyto požadavky na informace a implementovat je do počítačového databázového systému, který bude podporovat podnikání organizace.*“ (24, s. 5).

### 1.5.1 Datový model

Garcia-Molina poznamenává, že datový model je jedním z nejzásadnějších pojmů v oboru databázových systémů (27, s. 17). Dodává, že datový model je způsob, jakým se popisují data a informace. Tento popis se pak podle něj skládá ze tří částí:

1. **Struktura dat** – datovým strukturám, které slouží k ukládání dat, se v počítačové terminologii občas říká *fyzický datový model*.
2. **Operace nad daty** – v konceptu datových modelů jsou tyto operace omezené (jedná se o omezenou sadu operací).
3. **Omezení dat** – databázové modely většinou definují různá pravidla, kterým data podléhají. (27, s. 18)

Na obrázku č. 5 Ponniah naznačil, proč je datový model důležitý minimálně pro dvě zainteresované skupiny.



**Obr. č. 5: Reálný svět, datový model a databázový systém**  
(Zdroj: 24, s. 5)

První skupinou jsou uživatelé, kteří definují určité požadavky (requirements definition) na konečný databázový systém. Tyto požadavky vycházejí z informací z reálného světa. Vývojáři (developers) tyto požadavky přetvářejí do datového modelu, který v pozdější fázi slouží jako předloha databázového systému (24, s. 5).

Existuje několik typů datových modelů. Některé z nich se dnes používají spíše sporadicky, a to hlavně z toho důvodu, že nově vyvinuté datové modely jsou v mnoha ohledech pro datové modelování vhodnější.

### 1.5.1.1 Hierarchický datový model

Podle Pujari hierarchický model prezentuje data v hierarchii podobné obrácenému stromu (28). Jak upozorňuje Behl, jedná se o nejstarší typ datového modelu, byl vyvinut v 60. letech minulého století a rozšířil se v 70. letech minulého století. Zobrazení ve formě stromové struktury umožňovalo vytvořit vztah mezi dvěma záznamy tehdy, pokud měly přímou spojitost. Dodržuje se pravidlo rodičů a potomků, kdy jeden záznam je rodičovským prvkem a může mít několik potomků, ale jeden potomek může mít vždy pouze jeden rodičovský prvek (29, s 228).

### 1.5.1.2 Síťový datový model

Síťový model oproti hierarchickému modelu dovolil potomkům mít více než jednoho rodiče, což je také největší rozdíl oproti hierarchickému modelu (29, s. 228). Jak upozorňuje Pujari, síťové modely jsou však pomalé a složité na implementaci (28).

Connolly zařazuje hierarchický datový model a síťový datový model do první generace datových modelů. Upozorňuje, že oba modely měly několik nevýhod:

- Pro získání i těch nejjednodušších dat bylo nutné psát komplexní programy.
- Datová nezávislost byla minimální.
- Neexistoval široce přijímaný teoretický základ pro tyto modely. (20, s. 828)

### 1.5.1.3 Relační datový model

V relačním datovém modelu je podle Marstona základním stavebním prvkem tzv. *relace* (32). Pujari tvrdí, že relační model byl vytvořen proto, aby se vypořádal s problémy předchozích datových modelů, zejména se složitostí a nízkou flexibilitou. Relační datový model je podle ní nejen velmi jednoduchý, ale také výkonný (28).

Podle Garcii-Moliny nám relační datový model poskytuje jedinou možnost, jak zobrazovat data: dvoudimenzionální tabulkou zvanou *relace* (27, s. 21). Podle Connollyho je relační datový model založen na konceptu matematických relací. Dodává, že v relačním datovém modelu, který navrhnul E. F. Codd v roce 1970, jsou data logicky uspořádána v relacích neboli tabulkách (20, s. 69). Každá relace obsahuje sloupce a řádky. Sloupce se nazývají atributy, řádky se nazývají n-tice.

#### Atribut

Atribut je *sloupec* v relaci. Každý sloupec musí mít unikátní jméno v rámci své relace, nemůže se tedy stát, že by jedna relace obsahovala více sloupců se stejným pojmenováním. Sloupec musí být homogenní – záznamy ve stejném sloupci musí být vždy stejného typu. Každý sloupec má svoji doménu, což je souhrn všech možných hodnot, které se ve sloupci mohou objevit (30).

#### Schéma relace

Název relace společně se všemi svými atributy se nazývá *schéma relace* (27, s. 22).

## **N-tice**

Jednotlivé prvky relace se nazývají n-tice (jinak také záznamy) a jedná se o *řádky* v relaci. Každá n-tice se sestává právě z tolika položek, kolik má relace atributů. Pokud tedy relace obsahuje např. 5 atributů, bud mít každý řádek této relace právě 5 hodnot (20, s. 73).

## **Stupeň a kardinalita**

Počtu atributů v relaci se obecně říká *stupeň relace*. Relace pouze s jedním atributem se nazývá unární relace, relace se dvěma atributy se nazývá binární relace a relace se třemi atributy se nazývá ternární relace. Kardinalita pak udává, kolik záznamů je v relaci obsaženo. Pokud je tedy v relaci 5 záznamů, je kardinalita této relace právě 5 (20, s. 74).

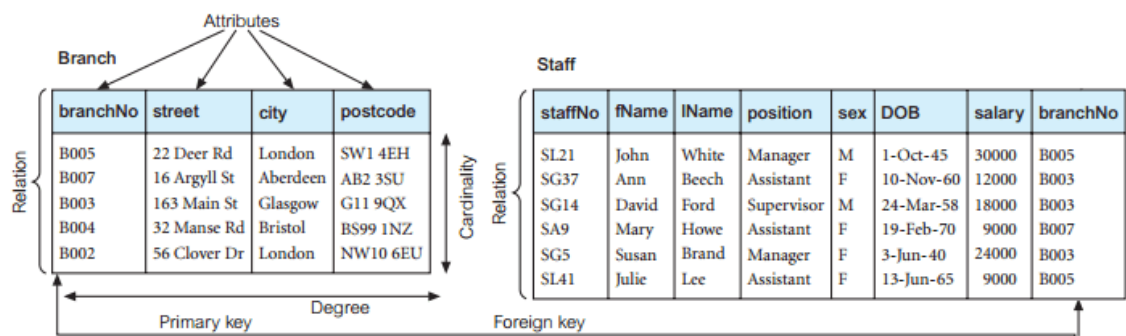
## **Relační klíče**

Každý záznam v tabulce musí být jednoznačně identifikovatelný – nemůže se stát, že by v jedné relaci byly duplicitní záznamy. Každý záznam musí být identifikovaný relačním klíčem (20, s. 78).

- **Jednoduchý klíč** – obsahuje pouze jeden atribut.
- **Kompozitní klíč** – obsahuje více než jeden atribut.
- **Kandidátní klíč** – je atribut nebo sada atributů, které unikátně identifikují záznam. Kandidátní klíč musí být unikátní (stejný klíč se nesmí v relaci objevit u dvou záznamů) a nezredukovatelný (žádný atribut v kandidátním klíči nesmí být odstraněn, aniž by klíč ztratil svoji unikátnost).
- **Primární klíč** – je kandidátní klíč, který být zvolen pro jednoznačnou identifikovatelnost záznamů v relaci; každá relace musí obsahovat primární klíč.
- **Superklíč** – je jakýkoliv soubor atributů, který unikátně identifikuje záznam v relaci; od kandidátního klíče se liší tím, že nevyžaduje neredukovatelnost.
- **Cizí klíč** – je atribut (nebo sada atributů), který obvykle vytváří vazbu s jinou relací tak, že je propojen s primárním klíčem v této relaci. (30)

Na obrázku č. 6 je znázorněna část relačního datového modelu. Jsou na něm znázorněny dvě relace (tabulky): *Pobočka* (Branch) a *Zaměstnanci* (Staff). Relace *Pobočka* má stupeň rovný čtyřem (4 atributy: číslo pobočky, ulice, město, PSČ) a kardinalitu rovnou pěti (5 položek v relaci – 5 jednotlivých poboček). Stejně tak relace *Zaměstnanci* obsahuje 8

atributů, je tedy osmého stupně. Kardinalita této relace je rovna šesti, neboť v relaci je právě 6 záznamů.



**Obr. č. 6: Relační datový model**  
(Zdroj: vlastní, upraveno podle 20)

## 1.6 Analytické metody použité v práci

### 1.6.1 PEST analýza

PEST analýza se využívá zejména při strategickém plánování. Vypovídá zejména o externích vlivech, které na firmu působí – jedná se o analýzu externího okolí (33).

Název analýzy PEST je odvozen od čtyřech hlavních oblastí, které jsou analyzovány:

- **Politické faktory** – zejména politická a legislativní situace v zemi či oblasti, kde společnost podniká. Např. daňová politika, státu, regulace vlády, obchodní zákony (32).
- **Ekonomické faktory** – sem patří zejména ekonomické podmínky na trhu – podpora zaměstnanosti, mzdové náklady, podpora exportu (31).
- **Sociální faktory** – zde se zkoumají zejména otázky týkající se demografického vývoje, věkového profilu, vzdělání (31).
- **Technologické faktory** – zde jsou analyzovány především technologická omezení či vyspělost v dané oblasti, náklady na výzkum a vývoj, vybavenost konkurence (31).

### 1.6.2 Porterův model pěti konkurenčních sil

Porterova analýza se zabývá především otázkou vnějších vlivů na podnikání firem. Michael Eugene Porter v roce 1979 nadeřinoval celkem 5 sil, které ovlivňují podnikání firem v daném odvětví (33):

- **Konkurenční rivalita** – zde se zkoumají konkurenční tlaky v daném odvětví, zda je možné v daném odvětví s konkurencí udržet krok.
- **Hrozba vstupu nových konkurentů** – zde se zkoumá, jak složité či jednoduché je pro nového konkurenta vstoupit na daný trh.
- **Hrozba vzniku nových substitutů** – v této oblasti se zkoumá schopnost konkurence přinést na trh takový výrobek, který by ohrozil naše vlastní portfolio.
- **Smluvní síla odběratelů** – jde zejména o schopnost odběratelů vyjednávat o ceně či lepších podmínkách.

- **Smluvní síla dodavatelů** – zde se zkoumá závislost naší společnosti na dodavatelích. Především se pak zajímáme o schopnost vyjednat o ceně či složení výrobků s našimi dodavateli.

### 1.6.3 Analýza 7S

Analýza 7S, kterou sepsal v 80. letech 20. století Tom Peters a Robert Waterman, patří k analýze vnitřního prostředí firmy. Zabývá se analýzou sedmi faktorů (34):

- **Struktura** – způsob, jakým je organizace strukturována.
- **Strategie** – hlavní firemní strategie.
- **Systém** – denní pracovní náplň a systém, jakým je dosahováno cílů.
- **Spolupracovníci** – zaměstnanci a jejich dovednosti.
- **Schopnosti** – konkrétní schopnosti a dovednosti zaměstnanců.
- **Styl** – styl řízení podniku.
- **Sdílené hodnoty a cíle** – hodnoty a cíle podniku, firemní kultura. (34)

### 1.6.4 SWOT analýza

Pomocí SWOT analýzy může společnost jednoduše analyzovat silné a slabé stránky podniku, příležitosti a hrozby. SWOT analýza je analýzou jak vnitřního prostředí (silné a slabé stránky), tak i vnějšího (příležitosti a hrozby). Díky SWOT analýze je firma schopna odhalit slabiny a problematické oblasti, zároveň by se měla snažit minimalizovat či eliminovat slabé stránky a maximalizovat silné stránky (35).



## 2 Analýza současného stavu

### 2.1 Základní údaje o společnosti

Společnost eSports, s.r.o. byla založena v roce 2001 se sídlem v Plzni, na trhu tedy působí již 14 let. Vlastnická struktura se během existence firmy několikrát změnila, v současné chvíli je vlastněna třemi společníky. Zaměstnává téměř 50 kmenových zaměstnanců, kteří jsou rozmístěni v kancelářích v Praze a ve Vsetíně, a spolupracuje s dalším více než tisícem externích spolupracovníků z České republiky i ze zahraničí.

**Tabulka č. 2: Základní informace o společnosti**  
(Zdroj: vlastní zpracování)

Název společnosti	eSports, s.r.o.
IČO	26340933
DIČ	CZ 26340933
Sídlo společnosti	Jeřabinová 836/30, 326 00 Plzeň
Počet zaměstnanců	50
Obrat za rok 2013	26 866 000 Kč
Hlavní činnost firmy	výroba, obchod a služby neuvedené v přílohách 1 až 3 živnostenského zákona
Webové stránky	<a href="http://www.esports.cz">http://www.esports.cz</a>



Obr. č. 7: Logo společnosti eSports, s.r.o.

#### 2.1.1 Hlavní činnosti firmy

##### 2.1.1.1 Tvorba informačních systémů a webových stránek

Společnost se zabývá vývojem informačních systémů, webových stránek a mobilních aplikací. Zaměřuje se na oblast sportu – její produkty většinou využívají sportovní kluby, svazové organizace či jiné subjekty, které působí ve sportovním odvětví.

### **2.1.1.2 Redaktorská a novinářská činnost**

Pro většinou sportovních webů společnost zajišťuje tvorbu obsahu, např. redaktorskou činností. Tento obsah vytvářejí zejména externisté, kteří s firmou dlouhodobě spolupracují.

### **2.1.1.3 Poskytování sportovních výsledků**

Společnost nabízí sportovní výsledky a živé komentáře z různých zápasů, akcí či turnajů ze všech sportovních odvětví. Tyto výsledky přenáší živě na svých internetových stránkách v několika jazykových mutacích a prodává je jiným společnostem.

### **2.1.1.4 Marketing a reklama**

Společnost nabízí svým klientům provoz marketingových kampaní, reklamu na internetu (optimalizace pro vyhledávače, správa PPC reklam, kontextová reklama) i v tiskové podobě či PR služby na míru.

### **2.1.1.5 Služby pro pořadatele sportovních turnajů**

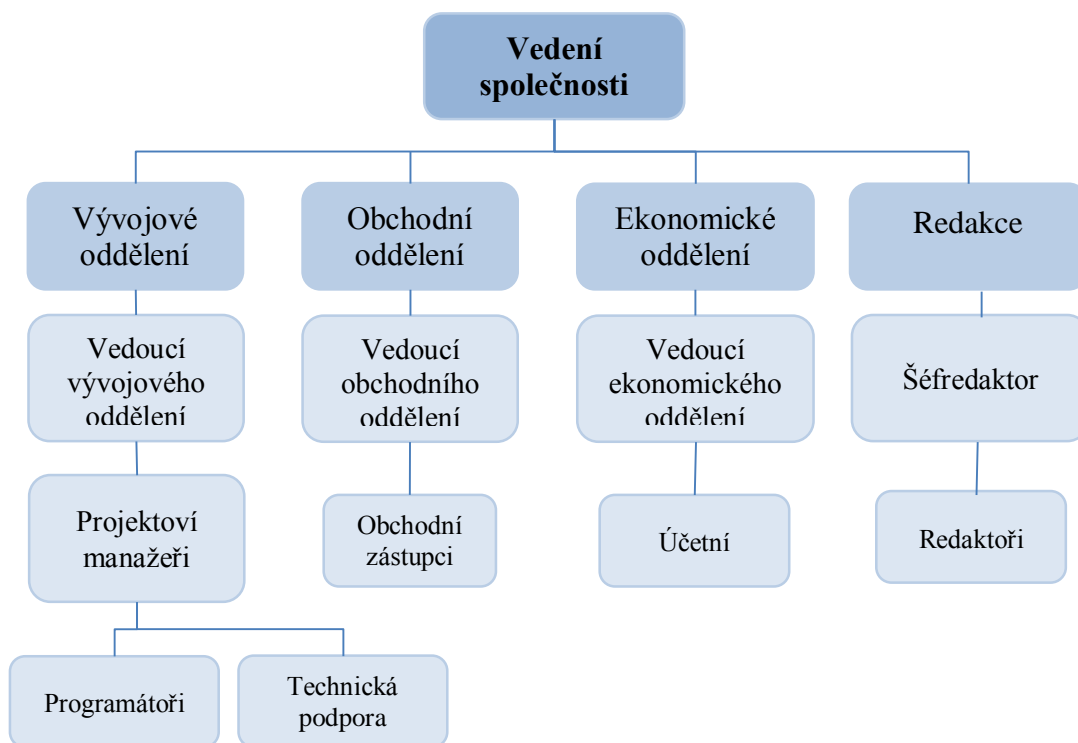
Pro pořadatele sportovních turnajů či zápasů společnost zajišťuje lidské zdroje na velké množství pracovních pozic – redaktory, novináře, dobrovolníky apod.

## **2.1.2 Projekt Onlajny**

Projekt Onlajny je vlastním projektem společnosti eSports, s.r.o. Jedná se o interně vyvíjený sportovní portál, který slouží k poskytování sportovních výsledků. Dostupný je v šesti jazykových verzích – kromě české je to slovenská, polská, ruská, anglická a maďarská verze. Každý den je na stránkách dostupných několik desítek sportovních zápasů, u kterých jednotliví komentátoři píšou textové přenosy.

## **2.2 Organizační struktura společnosti**

Společnost je rozdělena na tři oddělení: ekonomické oddělení, obchodní oddělení a vývojové oddělení. Každé z těchto oddělení má svého vedoucího pracovníka, který za dané oddělení zodpovídá.



**Obr. č. 8: Organizační struktura společnosti**  
(Zdroj: vlastní zpracování)

## **2.3 Analýza společnosti**

V této kapitole postupně provedu PEST analýzu, Porterovu analýzu pěti konkurenčních sil a SWOT analýzu společnosti.

### **2.3.1 PEST analýza**

#### **Politické faktory**

Politický vývoj v České republice v posledním roce je spíše pozitivní pro podnikatele a firmy. Vládní koalice v čele se stranou ANO před volbami jasně avizovala lepší prostředí pro podnikatele. Současná koalice je silná a nezdá se, že by v tomto volebním období došlo k nenadálým změnám, které by pro podnikatele a firmy znamenaly větší překážky v podnikání.

Legislativní prostředí je v České republice nakloněno podnikání. Ačkoliv je česká legislativa vnímána jako přílišně zatěžována byrokracií, podnikání nejsou kladeny žádné větší překážky. Firma se samozřejmě musí řídit platnými zákony České republiky. Na její oblast podnikání se nevztahují žádné speciální regulace či zákazy, není tedy omezována.

#### **Ekonomické faktory**

Do ekonomických faktorů započítáváme např. úrokovou míru, se kterou jsou poskytovány úvěry společností. Jelikož firma řeší veškeré financování vlastními prostředky, nemá výše úrokové míry zásadní vliv na její fungování.

Problémem může být vývoj sazby daně z příjmů právnických osob. Sazba se sice od roku 2009 nezměnila, politické debaty o jejím zvýšení či snížení jsou téměř na denním pořádku, což může vyvolat jisté pochybnosti. V projednávání je i jiné rozdělení daňových pásem a kategorií, což může pro firmy přinést faktor nejistoty.

Z hlediska ratingové společnosti Moody's je stav české ekonomiky dlouhodobě ohodnocen známkou A1, což je středně dobrá známka (38).

## **Sociální faktory**

Firma se zaměřuje na informační technologie a tento obor je dlouhodobě v České republice na vzestupu. Vysoké školy každoročně opouští velké množství kvalifikovaných absolventů v oboru, kteří snadno hledají uplatnění. Za plus se dá považovat i vysoký stupeň počítačové gramotnosti v České republice. Velkým kladem je bezesporu fakt, že firma zaměstnává lidi i na tzv. „home office“ pozice, kdy pracovník nemusí pracovat z kanceláří firmy, ale z vlastního domu. K tomuto způsobu práce se přiklání čím dál větší množství mladých lidí, pro které je možnost práce z domu velkým benefitem.

## **Technologické faktory**

V oboru, ve kterém firma podniká, dochází k neustálým změnám a technologickému vývoji. Firma si tedy osvojuje nové postupy a techniky, díky kterým je ve svém oboru konkurenceschopná. Na rozdíl od jiných oborů není technologický vývoj v tomto oboru nijak závislý na podpoře vlády či vlastních výdajů na výzkum.

## **2.3.2 Porterův model pěti konkurenčních sil**

### **Riziko vstupu potenciálních konkurentů**

V oblasti internetových technologií vznikají noví konkurenti téměř na denním pořádku. Vzhledem k tomu, že pro vstup do tohoto odvětví stačí minimální náklady (založení firmy, počítač), vznikají nové firmy velmi rychle. Největší přírůstky nových konkurentů má odvětví již za sebou (přelom tisíciletí), na nové konkurenty je však neustále potřeba brát ohledy.

Na druhou stranu není lehké získat zákazníka, jelikož ten většinou preferuje dlouhodobější spolupráci s již nalezeným dodavatelem.

### **Rivalita mezi konkurenty**

Obecně dnes platí, že většina firem či podnikatelů se chce zviditelnit na internetu formou nějaké internetové prezentace. To snižuje rivalitu mezi konkurenty v tomto odvětví, protože pokud nezískáme jednoho klienta, existuje celá další řada potenciálních klientů. Na rozdíl od jiných odvětví není ani nutné se s klientem setkat osobně – velké množství zakázek v tomto odvětví bylo a je získáváno pouze pomocí domluvy s klientem pomocí e-mailu či telefonu. To opět snižuje rivalitu, jelikož firma nemusí nutně usilovat o zákazníky ze svého nejbližšího okolí či města.

### **Smluvní síla odběratelů**

V tomto odvětví mají odběratelé velkou vyjednávací sílu. Internetových studií existuje mnoho a odběratel si tedy může vybírat svého smluvního partnera celkem snadno. Přejít k jinému dodavateli také není žádný problém, nejedná se o žádnou specifickou výrobu či montáž.

### **Smluvní síla dodavatelů**

Síla dodavatelů je v tomto odvětví nízká – firma si většinou prakticky vystačí s počítačem a připojením k internetu. Neodebírá žádné výrobky či polotovary, se kterými by nadále pracovala. Služby, které odebírá, poskytuje velké množství dodavatelů, je tedy možné mezi nimi snadno vybírat.

## **Hrozba substitučních výrobků**

Ještě před několika lety pro webovou prezentaci neexistoval substituční výrobek (když pomineme reklamu v některém z tehdejších médií, která je však jednoúčelová a většinou nesplňuje takové požadavky, které jsou kladeny na produkt typu webové stránky), dnes však můžeme hovořit například o mobilních aplikacích. Ty jsou však spíše doplňkem než substitutem.

Dalším substitutem může být profil firmy na sociálních sítích (např. stránka na sociální síti Facebook) – některé firmy se rozhodly jít cestou bez internetových stránek a vystačí si pouze s takovým profilem. K substitutu můžeme řadit i webové služby typu „udělej si sám“, ve kterých je možné jednoduchou internetovou prezentaci vytvořit vlastními silami. Zde jsou však možnosti omezené a složitější věci nelze tímto způsobem substituovat. Samostatnou kapitolou jsou specializované informační systémy, které nelze substituovat.

### **2.3.3 Analýza 7S**

#### **Struktura**

Firma má jasně danou organizační struktura, kterou se řídí. Každé oddělení má vlastního vedoucího pracovníka, celá společnost má jasně dané vedení.

#### **Strategie**

Firma má vypracované dokumenty k firemním cílům i k firemní strategii. Těmito dokumenty se snaží řídit.

#### **Systémy**

Komunikační kanály ve firmě jsou přizpůsobeny zejména k velmi rychlé komunikaci mezi pracovníky, kteří si potřebují předávat důležité informace. Pro komunikaci jsou použity hromadné e-maily, skupinové diskuse, chat a informační systém, do kterého jsou zaznamenávána hlášení.

#### **Spolupracovníci**

Pro vedení společnosti je důležité, aby pracovníci byli firmě loajální. Zaměstnanci jsou motivováni především finančně a možnostmi podílet se svými nápady na chodu firmy. Kterýkoliv zaměstnanec může přispět k lepšímu chodu firmy svými nápady. Společnost dbá i na socializaci pracovníků. Mnozí z nich pracují z domova a nesetkají se tak během pracovního procesu s ostatními spolupracovníky, společnost proto pořádá různé srazy či pracovní meetingy tak, aby se lidé mohli lépe poznat i osobně.

#### **Schopnosti**

Schopnosti pracovníků jsou zajištěny jednak vzájemným učením se a sdílením pracovních postupů s novými zaměstnanci, a jednak také školeními, které firma pro své zaměstnance zajišťuje.



## Styl

Vedení společnosti se zaměstnanci komunikuje jako se sobě rovnými. Ačkoliv jsou vedoucí jednotlivých oddělení jasně daní, jde spíše o formální rozdělení a jasné určení odpovědnosti. Mezi spolupracovníky a vedením je udržován přátelský neformální vztah.

## Sdílené hodnoty a cíle

Spolupracovníci sdílí stejné hodnoty, firma vyžaduje jistou firemní kulturu. Zaměstnanci chápou cíle společnosti a společnými silami se jich snaží dosáhnout.

### 2.3.4 SWOT analýza

Tabulka č. 3: SWOT analýza

(Zdroj: vlastní zpracování)

Silné stránky	Příležitosti
<ul style="list-style-type: none"><li>• Silná pozice na trhu se sportovními stránkami</li><li>• Předávání schopností a znalostí v pracovním kolektivu</li><li>• Výborný vztah se zákazníky</li><li>• Financování podniku vlastními zdroji</li></ul>	<ul style="list-style-type: none"><li>• Pokračování v započaté expanzi do zahraničí</li><li>• Vytvoření informačního systému pro malé sportovní kluby</li><li>• Větší propagace mobilních aplikací a mobilních verzí stránek</li></ul>
Slabé stránky	Hrozby
<ul style="list-style-type: none"><li>• Absence integrujícího informačního systému</li><li>• Zaměstnanci pracující z domu nejsou v každodenním kontaktu s vedením firmy</li><li>• Neschopnost rychle nahradit starší verzi softwaru novější verzí</li></ul>	<ul style="list-style-type: none"><li>• Zakázky odvislé od stavu sportovního odvětví v ČR</li><li>• Příchod zahraniční konkurence se službou sportovních výsledků</li><li>• Rozšíření sportovních videopřenosů do mobilních zařízení by přineslo odliv návštěvníků stránek</li></ul>

## 2.4 Hlavní firemní procesy

Ve firmě existují jednak hodnototvorné procesy a jednak podpůrné procesy. Hodnototvornými procesy rozumíme ty, které vytvářejí hodnotu pro zákazníka. Podpůrnými procesy rozumíme ty, jejichž cílem je zajištění fungování hodnototvorných procesů a samy o sobě nevytvářejí hodnotu pro zákazníka.

Pro každý proces bude sestrojen EPC diagram a RACI matice odpovědnosti. Pro lepší přehlednost je výčet jednotlivých profesních rolí převeden na následující číselník:

- |                                  |                              |
|----------------------------------|------------------------------|
| 1. Vedoucí vývojového oddělení   | 9. Klient                    |
| 2. Vedoucí obchodního oddělení   | 10. Vedoucí projektu Onlajny |
| 3. Vedoucí ekonomického oddělení | 11. Redaktor                 |
| 4. Vývojový tým                  | 12. Jednatel                 |
| 5. Obchodní zástupce             | 13. Tester                   |
| 6. Projektový manažer            | 14. Šéfredaktor              |
| 7. Pracovník technické podpory   | 15. Externí komentátor       |
| 8. Účetní                        |                              |

## 2.4.1 Hodnototvorné procesy

### 2.4.1.1 Zpracování zakázky

Vstupem v tomto procesu je objednávka nového projektu od klienta. Tímto projektem může být tvorba webových stránek, tvorba informačního systému nebo grafické práce. Činnosti v těchto procesech se výrazně neliší, lze je tedy obsáhnout do jednoho procesu.

Projektový manažer se setká s potenciálním klientem a sepíšíou klientovy požadavky. Projektový manažer vyhotoví kalkulaci ceny za projekt. Pokud klient cenovou nabídku odmítne, projekt se neuskuteční. Pokud klient cenovou nabídku přijme, sestaví projektový manažer časový harmonogram a milníky, které musí být v projektu dodrženy. Následně je podepsána smlouva.

Po podepsání smlouvy projektový manažer vybere vývojový tým, který na vybraném projektu bude pracovat. Vývojový tým postupně na základě časového harmonogramu zpracuje projekt. Po zpracování je tento projekt řádně otestován. Po otestování je projekt předán klientovi a podepsán protokol o předání.

**Tabulka č. 4: RACI matice procesu zpracování zakázky**  
(Zdroj: vlastní zpracování)

Činnost	Profesní role													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Setkání s klientem						AR			C					
Vytvoření kalkulace						AR								
Seznámení klienta s kalkulací						AR			I					
Vytvoření harmonogr.						AR			C					
Vytvoření milníků						AR			C					
Podepsání smlouvy									I			AR		
Výběr vývojového týmu	C					AR								
Zpracování projektu	A			R					I					
Otestování projektu	A					I							R	
Předání klientovi						AR			I					
Podepsání předávacího protokolu						AR			I					



Obr. č. 9: EPC diagram procesu zpracování zakázky  
(Zdroj: vlastní zpracování)

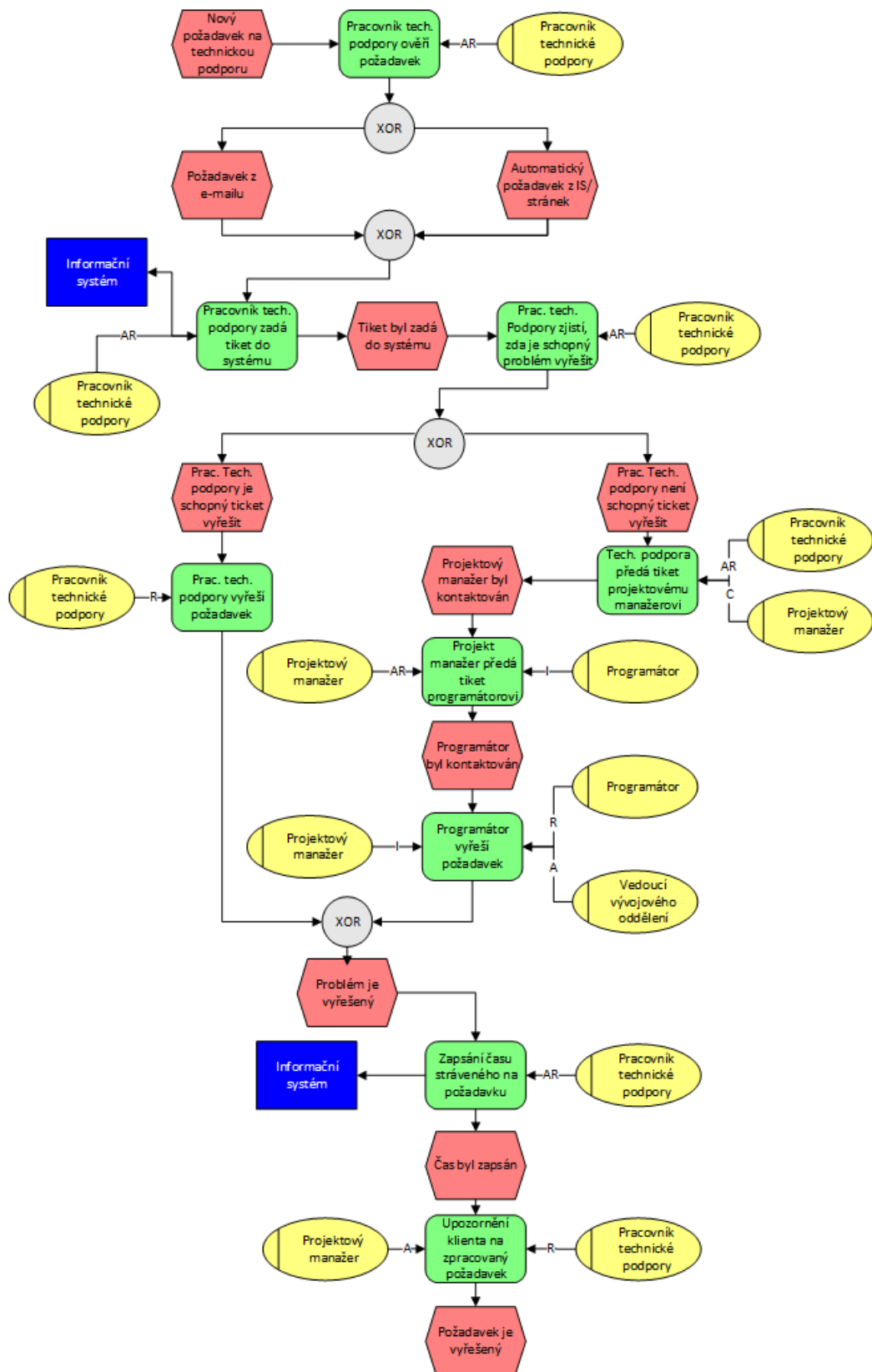
### 2.4.1.2 Řešení zákaznických požadavků na technické podpoře

Firma eSports, s.r.o. za dobu své působnosti vytvořila již velké množství webových prezentací a informačních systémů. Pokud by se o vytvořené a již běžící projekty nikdo nestaral, poškozovalo by to jméno firmy a zákazníci by mohli upřednostnit firmu jinou – takovou, která technickou podporu provozuje.

Vstupem jsou požadavky od klientů na změnu funkčnosti v některém projektu, popřípadě i stížnosti na nefunkční část projektu, a také automaticky generovaná chybová hlášení přímo z projektů samotných. Tato hlášení většinou indikují nějakou chybu či závadu a automaticky je jim přiřazena priorita podle toho, o jak závažnou chybu se jedná (což neplatí u požadavků klientů, u kterých není možné automaticky z emailu detekovat závažnost požadavku či chyby). Pracovník technické podpory zavede daný požadavek do systému jako tiket a zjistí, zda je schopný daný problém vyřešit vlastními silami. Pokud ano, následuje oprava nebo přidání nové funkce. Pokud není schopen vlastními silami problém vyřešit, kontaktuje projektového manažera, který má projekt na starosti. Ten předá požadavek programátorovi, který jej zpracuje. Posledním krokem je zapsání informace o času stráveném na projektu a upozornění klienta na vyřešení požadavku.

**Tabulka č. 5: RACI matice procesu řešení zákaznických požadavků**  
(Zdroj: vlastní zpracování)

Činnost	Profesní role													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Ověření požadavku							AR							
Zadání tiketu do systému							AR							
Zjištění, zda je tech. podpora schopná problém vyřešit							AR							
Vyřešení požadavku tech. podporou							AR							
Předání projektovému manažerovi						C	AR							
Předání programátorovi				I		AR								
Vyřešení požadavku	A			R		I								
Zapsání času							AR							
Upozornění klienta						AR			I					



Obr. č. 10: EPC diagram procesu řešení požadavků na technické podpoře  
(Zdroj: vlastní zpracování)

## 2.4.2 Podpůrné procesy

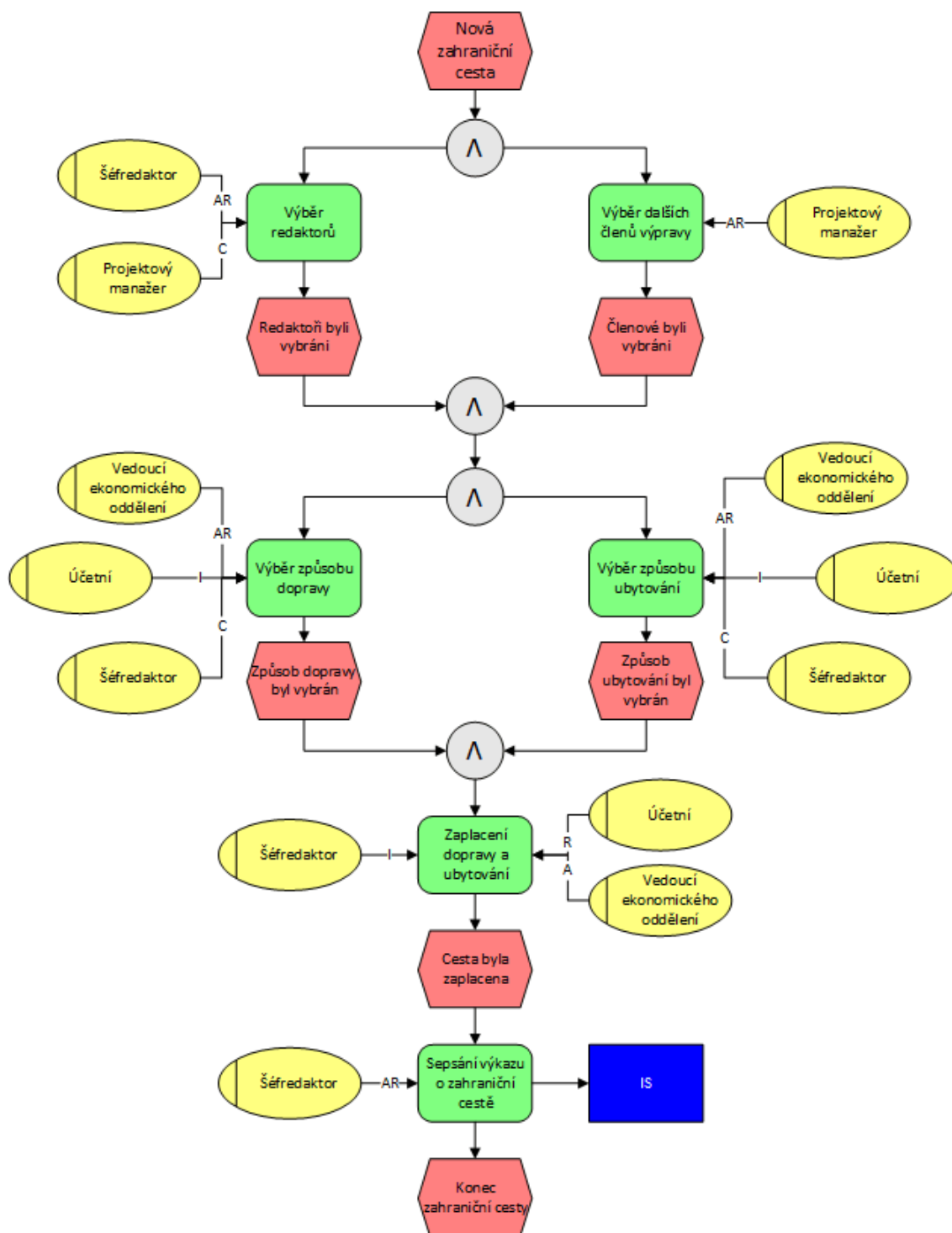
### 2.4.2.1 Plánování zahraničních cest

Zaměstnanci firmy provozují redaktorskou činnost, což mimo jiné zahrnuje i psaní reportáží z různých turnajů či událostí. Tyto události se velmi často odehrávají v zahraničí a je proto nutné zahraniční cesty dopředu naplánovat.

Pokud se tedy objeví zahraniční akce, na kterou firma vyšle vlastní delegaci, je nutné vybrat redaktory, kteří se dané akce zúčastní. Výběr redaktorů provede šéfredaktor, který je zároveň vedoucím výpravy. Pokud se má cesty zúčastnit i jiný zaměstnanec, rozhodne o tom projektový manažer. Následně se podle destinace vybere způsob dopravy a způsob ubytování během celé akce. Po skončení zahraniční cesty je sepsán výkaz o zahraniční cestě.

**Tabulka č. 6: RACI matice procesu plánování zahraničních cest**  
(Zdroj: vlastní zpracování)

Činnost	Profesní role													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Vybrání redaktorů												C		AR
Vybrání ostatních členů výpravy						AR								
Vybrání způsobu dopravy			AR					I						C
Vybrání způsobu ubytování			AR					I						C
Zaplacení dopravy a ubytování			A					R						I
Sepsání výkazu o zahraniční cestě														AR



Obr. č. 11: EPC diagram procesu plánování zahraničních cest  
(Zdroj: vlastní zpracování)



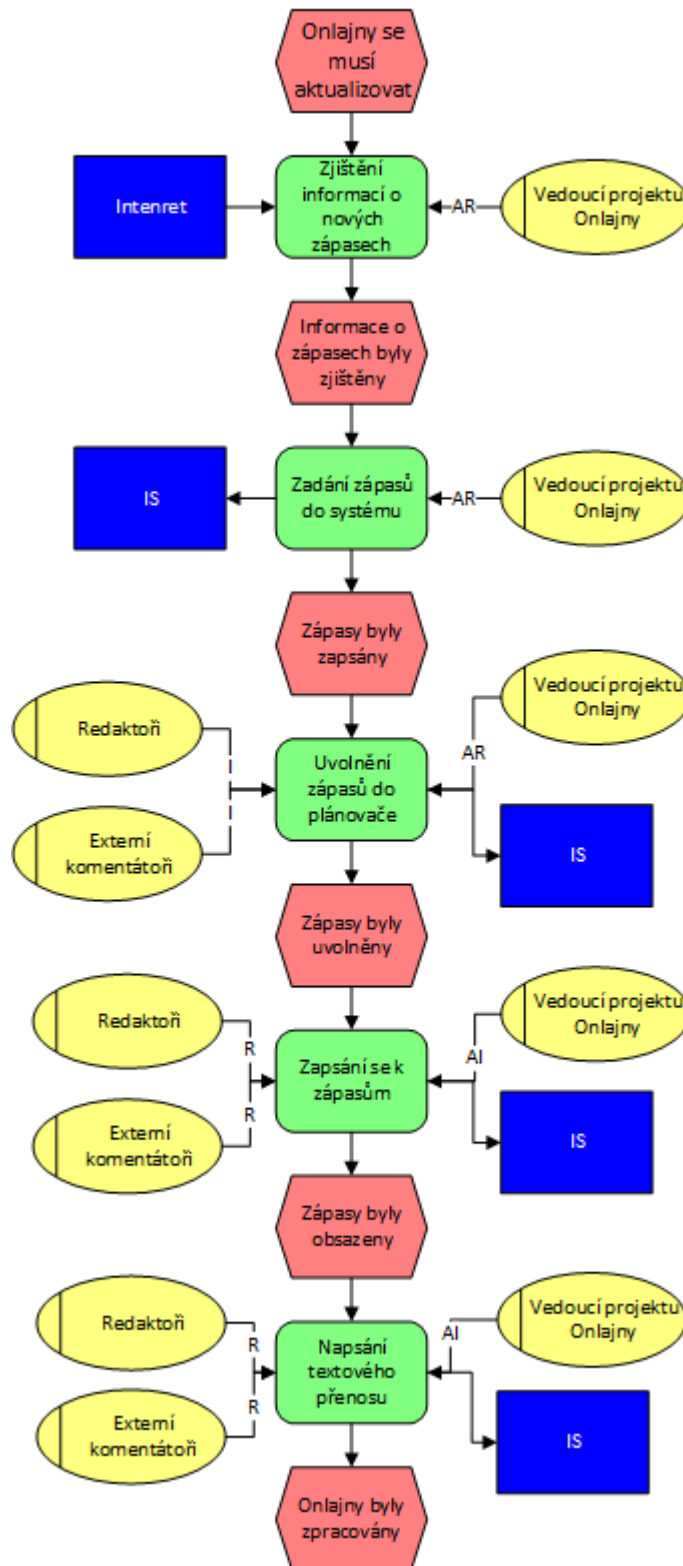
### 2.4.2.2 Obsluha projektu Onlajny

Projekt Onlajny je vlastní projekt firmy, který vytvořila a provozuje. Jedná se o webový portál se sportovními výsledky a sportovní zpravodajstvím.

Vedoucím projektu zjistí informace o nových zápasech, které se v daný den či kdykoliv v budoucnu budou hrát. Tyto zápasy zadá do informačního systému a nastaví jím všechny nutné parametry (čas začátku, sport apod.). Zápasy jsou uvolněny do tzv. plánovače a redaktoři nebo externí komentátoři se přiřadí k jednotlivým zápasům, což znamená, že k danému zápasu budou psát podrobný textový přenos. Poté proběhnou samotné zápasy a komentátoři napíší textový přenos.

**Tabulka č. 7: RACI matice procesu obsluha projektu Onlajny**  
(Zdroj: vlastní zpracování)

Činnost	Profesní role														
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Zjištění informací o nových zápasech									AR						
Zapsání zápasů do systému									AR						
Uvolnění zápasů do plánovače									AR	I				I	
Zapsání se k zápasům									AI	R				R	
Napsání textového přenosu									AI	R				R	



Obr. č. 12: EPC diagram procesu obsluha projektu Onlajny  
(Zdroj: vlastní zpracování)

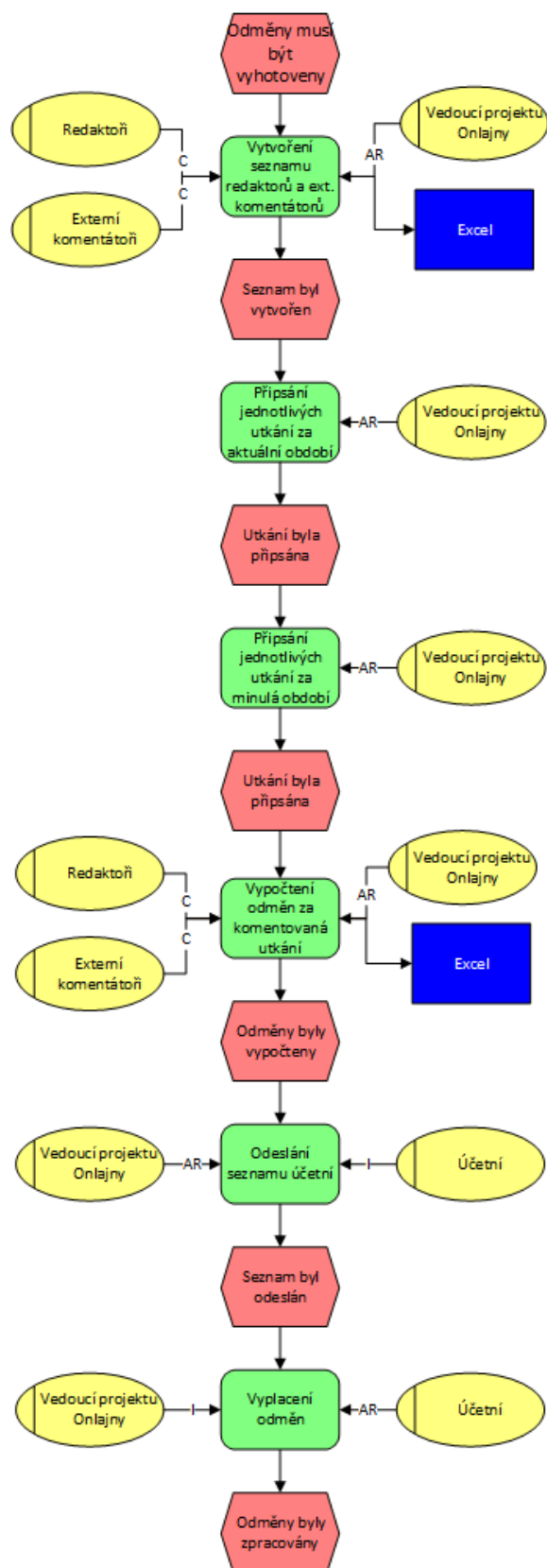
### 2.4.2.3 Vyhotovení odměn za předchozí měsíc

Redaktoři a externí komentátoři jsou za svoje služby placeni na základě jednotlivých utkání, která komentovali. Výše odměny za jednotlivá utkání se liší podle důležitosti daného utkání i doby, kdy se utkání odehrálo (velké množství zápasů se odehrává i v noci našeho času).

Vedoucí projektu Onlajny vytvoří seznam všech redaktorů a externích komentátorů, kteří v předchozím měsíci komentovali alespoň jeden zápas. K jednotlivým redaktorům zjistí utkání, která komentovali, a vypočítá výši odměn za tato utkání. Zjistí také, zda u některých komentátorů nedošlo k odložení vyplacení odměn z minulého období. Pokud některé odměny byly odloženy do následujícího období, připíše je k současnému seznamu. Celý soupis následně odešle mzdové účetní, která dané odměny vyplatí.

**Tabulka č. 8: RACI matice procesu vyhotovení odměn**  
(Zdroj: vlastní zpracování)

Činnost	Profesní role														
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
Vytvoření seznamu redaktorů									AR	C				C	
Připsání utkání za aktuální období									AR						
Připsání utkání za minulá období									AR						
Vypočtení odměn za komentovaná utkání									AR	C				C	
Odeslání aktuálního seznamu účetní							I		AR						
Vyplacení odměn za aktuální období							AR		I						



Obr. č. 13: EPC diagram procesu vyhotovení odměn  
(Zdroj: vlastní zpracování)

#### 2.4.2.4 Fakturace

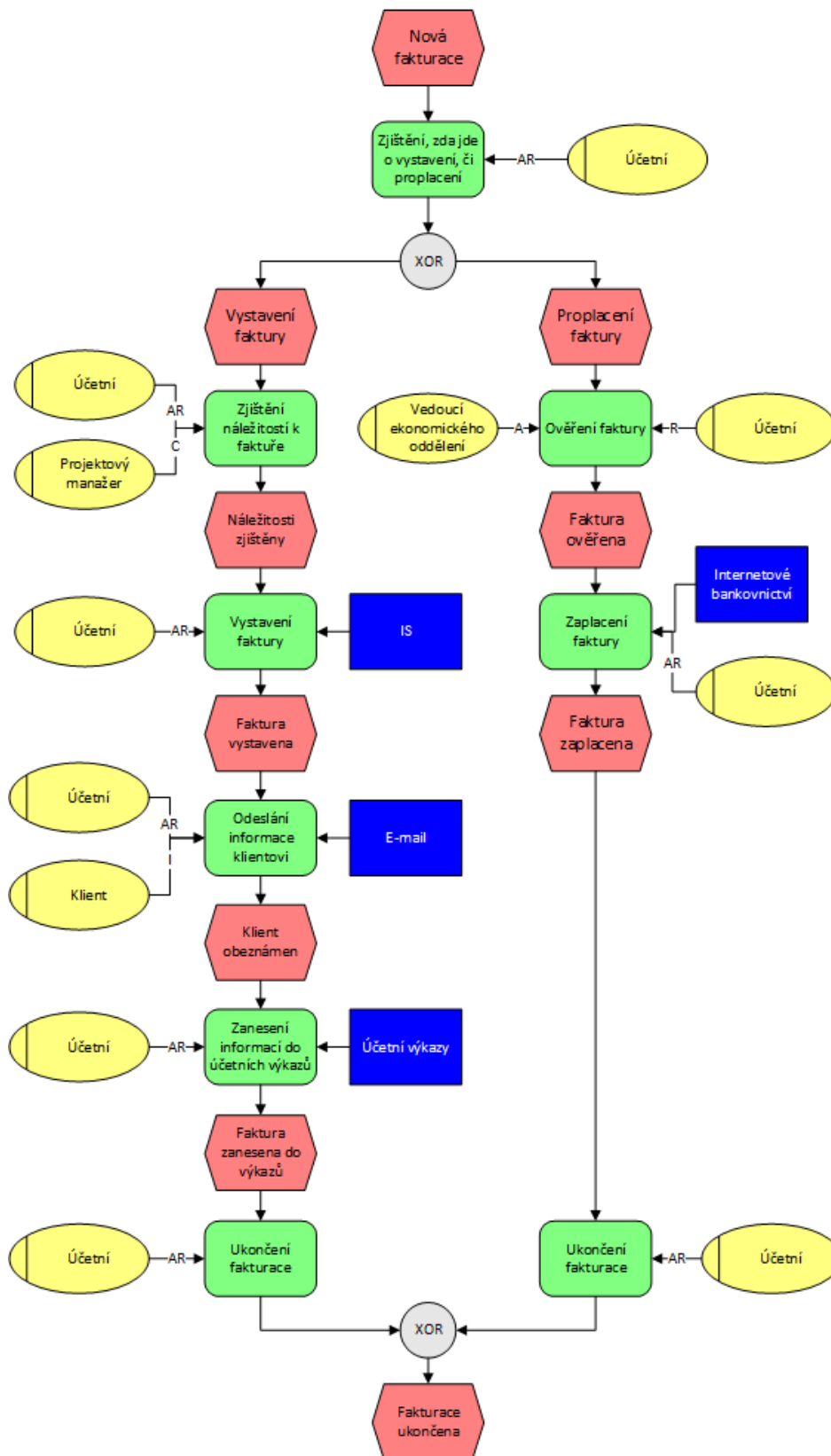
Fakturace patří k běžným procesům, které se ve firmách řeší. Firma faktury buď vystavuje, nebo je obdrží od svých dodavatelů.

Pokud fakturu vystavuje, obdrží účetní od projektového manažera nutné náležitosti k faktuře (odběratel, částka, splatnost), fakturu vystaví a odešle klientovi.

Pokud se jedná o proplacení faktury dodavatelům, musí účetní nejprve danou fakturu ověřit, zda údaje na faktuře souhlasí. Pokud údaje souhlasí, danou fakturu zaplatí. Nakonec je faktura zanesena do interních účetních výkazů.

**Tabulka č. 9: RACI matice procesu fakturace**  
(Zdroj: vlastní zpracování)

Činnost	Profesní role													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Zjištění, zda jde o vystavení či proplacení								AR						
Zjištění náležitostí k faktuře						C		AR						
Vystavení faktury								AR						
Ověření faktury			A					R				C		
Zaplacení faktury								AR						
Odeslání informace klientovi							I	AR						
Zanesení informací do účetních výkazů								AR						



Obr. č. 14: EPC diagram procesu fakturace  
(Zdroj: vlastní zpracování)

## 2.5 Firemní informační systémy a aplikace

Společnost pro svoji činnost využívá několik informačních systémů a programů. Ty však již nevyhovují požadavkům a potřebám, které společnost pro svoji činnost vyžaduje. Chybí zejména propojení jednotlivých systémů tak, aby vzájemně sdílely data.

### 2.5.1 Interní firemní informační systém

Společnost eSports, s.r.o. pro svoji činnost vyvinula interní informační systém v roce 2005, tedy čtyři roky po svém vzniku. Informační systém se používá dodnes a to i přes značnou zastaralost technického řešení. Technická zastaralost tohoto systému byla jedním z hlavních impulzů k vytvoření nového informačního systému.

Informační systém byl naprogramován pomocí několika technologií:

- **ASP** – hlavní dynamický programovací jazyk na straně serveru. V současné době se společnost od tohoto programovacího jazyku snaží přejít k jinému dynamickému jazyku, kterým je PHP. Sjednocení všech řešení na jednotný jazyk PHP byl jedním z dalších impulzů k vytvoření nového informačního systému.
- **HTML** – značkovací jazyk používaný pro tvorbu webových stránek.
- **Javascript** – programovací jazyk určený pro vytváření dynamických efektů na straně klienta, nikoliv serveru.

V interním informačním systému se nacházejí tyto moduly:

- **Modul uživatelů** – jedná se o modul pro správu zaměstnanců, redaktorů i externích komentátorů.
- **Modul projektů** – jedná se o správu projektů, které má společnost eSports, s.r.o. na starosti.
- **Modul diskusního fóra** – jedná se o diskusní fórum, do kterého mohou přispívat všichni zaměstnanci, redaktoři i externí komentátoři.

### 2.5.2 Trello

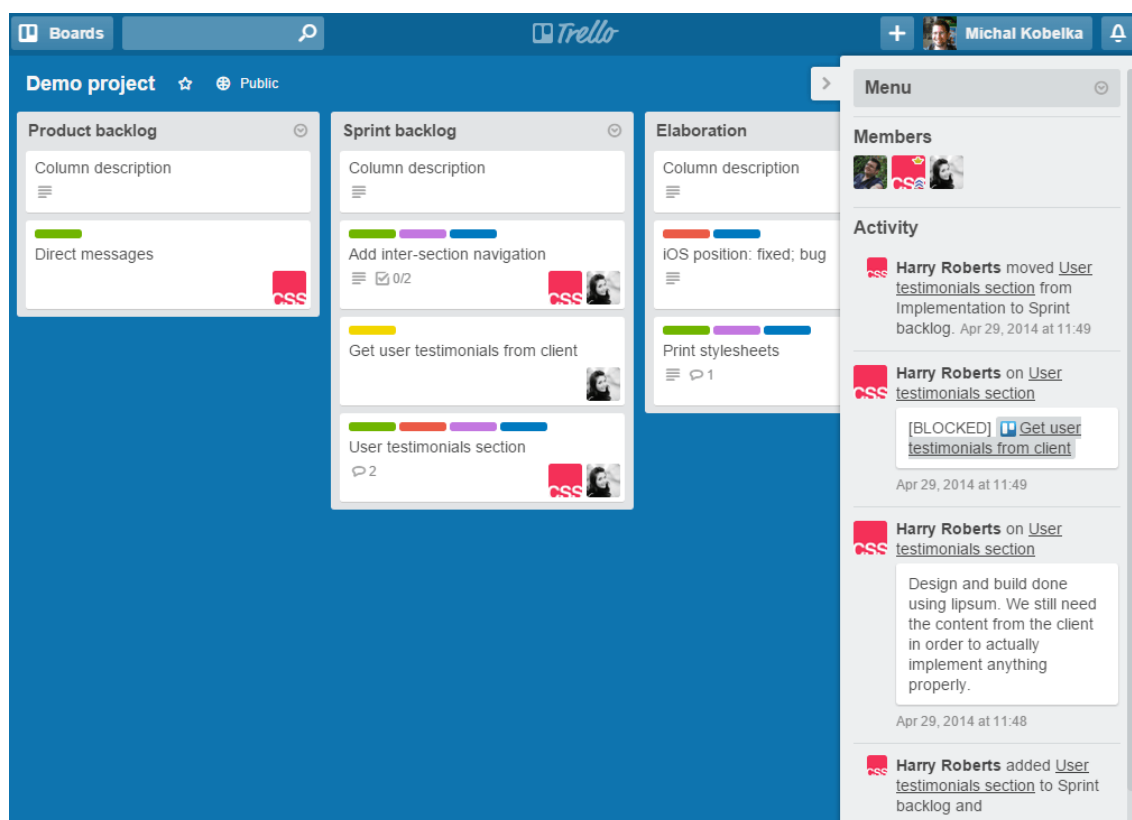
Trello je webová aplikace pro správu projektů vytvořená firmou Fog Creek Software v roce 2011. Jedná se o tzv. cloudovou službu, což znamená, že aplikace není fyzicky

umístěna na serveru společnosti eSports, s.r.o. Aplikace je umístěna na serveru poskytovatele a je k ní přístupováno vzdáleně.

Trello využívá tzv. systému kanban (výraz z japonštiny, který doslova znamená „karta“). Trello je systém nástěnek a karet, které lze mezi nástěnkami jednoduše přesouvat, přiřazovat k nim pracovníky či štítky. Pomáhá k lepší organizaci projektů.

Tento systém by společnost ráda používala i po vzniku nového informačního systému. Chtěla by však využít možnosti napojit nový informační systém pomocí API rozhraní a získávat tak informace o projektech ze systému Trello a zobrazovat je v novém informačním systému.

Tímto krokem se jednak zpřístupní informace o projektech i těm pracovníkům, kteří běžně Trello v organizaci nevyužívají, ale bude možné v budoucnu v případě nutnosti tento systém libovolně rozšířit.



**Obr. č. 15: Trello**  
(Zdroj: vlastní zpracování dle 39)



### **2.5.3 Interní tiketovací systém**

Firma od roku 2010 používá interní tiketovací systém. Jedná se o jednoduchou aplikaci na vytváření tiketů, které slouží k přiřazování pracovních úkolů jednotlivým zaměstnancům.

Funkcionalita tohoto systému je dostačující i dnes, společnost by však ráda integrovala tiketovací modul do nového informačního systému a tento samostatný informační systém by vyřadila z provozu.

### **2.5.4 Informační systém pro projekt Onlajny**

Pro obsluhu projektu Onlajny má firma vyvinutý vlastní informační systém z roku 2010. Tento systém je však natolik komplexní, že integrace do nově vzniklého informačního systému by přinesla spíše problémy než užitek.

## **2.6 Shrnutí analytické části**

Nejdůležitějším procesem pro společnost je zpracování zakázky. Téměř veškeré činnosti, které společnosti přinášejí zisk, pramení z vývoje informačních systémů, webový stránek, grafických návrhů či mobilních aplikací. V novém informačním systému je tedy nutné mít zejména správu uživatelů a klientů a na ně navázanou správu projektů. Výborná organizace projektů je pro firmu nutná. Je také důležité, aby u projektů bylo možné zadávat tzv. tikety, díky kterým je možné přiřazovat práci pracovníkům.

V informačním systému by také neměly chybět obecné diskuse, které společnost používá pro vnitropodnikovou komunikaci.

### 3 Návrh řešení

V návrhové části navrhnu datový model nového informačního systému. Tento datový model bude použit společností eSports, s.r.o. jako základ nového informačního systému. V průběhu roku 2015 bude k tomuto databázovému modelu naprogramována aplikace v programu PHP jak pro serverovou část, tak pro klientskou část. Nový informační systém tedy bude zprovozněn v průběhu roku 2015.

Se společností eSports, s.r.o. bylo dohodnuto, že navrhovaný informační systém bude pro svoji databázovou část využívat databázový systém MySQL. Pro návrh databázového modelu informačního systému tedy použiji jazyk SQL, konkrétně nadstavbu MySQL.

Informační systém bude obsahovat celkem pět modulů. **Modul uživatelů** bude sloužit k ukládání informací o uživateli včetně adres, bankovních spojení, pozic ve firmě či rolí v informačním systému.

**Modul klientů** bude sloužit k ukládání informací o klientech společnosti. Klienty lze třídit do různých skupin, což napomáhá jejich organizaci. K jednotlivým klientům lze také přiřazovat osoby, které mají ke společnosti eSports, s.r.o. nějaký vztah.

**Modul projektů** bude sloužit k organizaci firemních projektů. K projektům lze přiřazovat uživatele a klienty a třídit je do skupin pro lepší orientaci v projektech. Projektům lze přiřazovat status tak, aby bylo možné je odlišit v různé fázi vývoje. K projektům lze také přiřazovat štítky, které napomáhají lepší organizaci projektů. K jednotlivým projektům lze vytvářet diskusní vlákna a vytvářet komentáře v těchto vláknech.

**Modul tiketů** bude sloužit k zadávání chybových hlášení či pracovních pokynů k jednotlivým projektům. Tikety lze organizovat pomocí kategorií a statusů. K tiketům lze přidávat komentáře, což slouží např. k upřesnění úkolů ze zadání tiketu mezi zadavatelem a osobou odpovědnou za provedení úkolu

**Modul diskusí** bude sloužit k interním diskusím zaměstnanců eSports. Obecné diskuse nejsou na rozdíl od projektových diskusí navázány na žádný konkrétní projekt.

### 3.1 Modul uživatelů

Modul uživatelů bude sloužit k uchování informací o uživateli. Uživatel je středobodem celého informačního systému a jsou na něj navázány všechny ostatní moduly. Modul uživatelů se bude skládat celkem z devíti tabulek.

#### 3.1.1 Tabulka user

Tabulka user je hlavní tabulka modulu user, ve které jsou uloženy primární informace o jednotlivých uživateli informačního systému. Kromě obligátních informací o uživateli (jméno, příjmení, pohlaví apod.) jsou v tabulce obsaženy ještě speciální atributy:

- **Is\_active** – tento atribut označuje, zda je uživatel aktivní, nebo zda jeho působnost v rámci informačního systému byla ukončena. V takovém případě má uživatel zakázané přihlašování do informačního systému.
- **Is\_blocked** – tento atribut označuje, zda byl uživatel zablokovaný. V takovém případě má uživatel zakázané přihlašování do informačního systému.
- **Position** – tento atribut označuje pozici uživatele ve firmě (např. programátor, redaktor apod.). Pozice jsou předvolené ve formě číselníku v tabulce *user\_position*.

**Tabulka č. 10: Tabulka user**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(20)	Jméno	
surname	varchar(20)	Příjmení	
title	varchar(10)	Titul	
email	varchar(45)	E-mailová adresa	
password	varchar(40)	Heslo (zaheslované pomocí funkce SHA-1)	
phone	varchar(20)	Telefon	
sex	varchar(1)	Pohlaví	
is_active	tinyint(1)	Indikátor, zda je uživatel aktivní	
is_blocked	tinyint(1)	Indikátor, zda je uživatel blokový	

### 3.1.2 Tabulka `user_address`

V tabulce `user_address` jsou uloženy adresy uživatelů. Každý uživatel může mít přiděleno více než jednu adresu (např. korespondenční, adresu trvalého bydliště) s tím, že si tyto adresy může pojmenovat pro lepší orientaci.

V tabulce figuruje jediný cizí klíč, a tím je atribut `user_id`. Přes tento atribut jsou propojeny tabulky `user_address` a `user`.

**Tabulka č. 11: Tabulka `user_address`**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
<code>id</code>	<code>int(11) AI</code>	Jednoznačný identifikátor	Primární klíč
<code>name</code>	<code>varchar(20)</code>	Jméno	
<code>surname</code>	<code>varchar(20)</code>	Příjmení	
<code>phone</code>	<code>varchar(20)</code>	Telefon	
<code>street</code>	<code>varchar(25)</code>	Ulice	
<code>number</code>	<code>varchar(10)</code>	Číslo	
<code>city</code>	<code>varchar(40)</code>	Město	
<code>zip</code>	<code>varchar(8)</code>	PSC	
<code>country</code>	<code>varchar(20)</code>	Země	
<code>address_name</code>	<code>varchar(45)</code>	Název adresy	
<code>user_id</code>	<code>int(11)</code>	ID uživatele	Cizí klíč

### 3.1.3 Tabulka `user_bank_account`

V tabulce `user_bank_account` jsou uložena bankovní spojení na uživatele. Tyto informace slouží k uhrazování mezd, zasílání odměn apod. Každý uživatel může mít přiřazené více než jedno bankovní spojení. Spojení na tabulku `user` je realizováno pomocí atributu `user_id`, což je cizí klíč odkazující na tabulku `user`.

Tabulka se skládá z 5 atributů. Kromě jednoznačného identifikátoru (`id`) a cizího klíče (`user_id`) se jedná o atribut `number`, což je číslo účtu. V České republice se používá šestnáctimístné číslo účtu, ve kterém se vyskytuje pomlčka. Datový typ jsem tedy zvolil jako `varchar`, aby tato pomlčka mohla být vyplněna.

Dalším atributem je *bank\_code*, tedy kód banky. Zde jsem opět zvolil typ *varchar* s délkou čtyř znaků – kódy bank jsou čtyřmístné a vyskytují se i takové kódy, které mají na první pozici nulu.

Atribut *primary* indikuje, zda je účet primární. V případě, že je k uživateli přiřazen více než jeden účet, je nutné vědět, který z nich je primárním.

**Tabulka č. 12: Tabulka *user\_bank\_account***  
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
number	varchar(16)	Číslo účtu	
bank_code	varchar(4)	Kód banky	
primary	bool	Indikátor, zda je spojení primární	
user_id	int(11)	ID uživatele	Cizí klíč

### 3.1.4 Tabulky *user\_positions* a *user\_to\_position*

V tabulkách *user\_positions* a *user\_to\_position* se řeší vztah uživatele k firemní hierarchii. Jedná se tedy o přiřazení pracovní pozice k pracovníkovi.

**Tabulka č. 13: Tabulka *user\_positions***  
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(40)	Název pozice	
superior	int(11)	Nadřazená pozice	Cizí klíč

V tabulce se nachází pouze primární klíč, název pozice (atribut *name*) a cizí klíč *superior*, který odkazuje na nadřazenou pracovní pozici ve stejné tabulce.

Tabulka *user\_to\_position* je spojovací tabulka, která zajišťuje vztah m:n mezi tabulkami *user* a *user\_positions*. Spojovací tabulka je nutná téměř vždy, když do databázového modelu chceme zakomponovat vztah m:n. Ten se používá tehdy, pokud k více záznamům z jedné tabulky chceme přiřadit více záznamů z tabulky jiné.

Tabulka se skládá pouze ze tří atributů: jednoznačného identifikátoru (*id*) a dvou cizích klíčů, které směřují do dvou různých tabulek: *user\_positions* a *user*.

**Tabulka č. 14: Tabulka *user\_to\_position***  
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
user_id	int(11)	ID uživatele	Cizí klíč
position_id	int(11)	ID pozice	Cizí klíč

### 3.1.5 Tabulky *user\_role*, *user\_role\_rights*, *user\_roles\_to\_right* a *user\_to\_role*

Uživatelská práva jednotlivých uživatelů informačního systému jsou vyřešena v těchto čtyřech tabulkách.

Tabulka *user\_role* obsahuje jednotlivé role, které lze v systému přiřadit uživateli (např. super administrátor, administrátor, běžný uživatel apod.). Skládá se tří atributů – jednoznačného identifikátoru (*id*), názvu role (*name*) a atributu *superuser*, který označuje, zda se jedná o instanci nejvyšší možné role v systému, která má práva ke všem částem informačního systému a je nadřazená všem ostatním rolím.

**Tabulka č. 15: Tabulka *user\_role***  
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(20)	Název role	
superuser	bool	Indikátor superuživatele	

Jelikož jeden uživatel může mít více rolí a jedna role může být přiřazena více uživatelům, je opět nutné vztah m:n vybudovat pomocí pomocné tabulky. Ta má název *user\_to\_role* a je velmi podobná tabulce *user\_to\_position*. Skládá se opět z jednoho primárního klíče a dvou cizích klíčů, které spojí obě tabulky *user* a *user\_roles*.

**Tabulka č. 16: Tabulka user\_to\_role**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
user_id	int(11)	ID uživatele	Cizí klíč
user_role_id	int(11)	ID role	Cizí klíč

Přirazením rolí uživatelům ale ještě nedosáhneme omezení přístupu a práv těchto uživatelů. Abychom tak učinili, musíme jednotlivým rolím přiřadit práva. Těmito právy se rozumí možnost uskutečnit v informačním systému nějakou akci, prohlédnout si zdroj informací apod.

Tabulka jednotlivých práv se nazývá *user\_role\_rights* a obsahuje tři atributy: jednoznačný identifikátor (*id*), název (*name*) a hodnotu (*value*). Jednotlivých práv bude velké množství a každé z těchto práv bude jedním řádkem v tabulce.

**Tabulka č. 17: Tabulka user\_role\_rights**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(20)	Název práva	
value	varchar(20)	Hodnota práva	

Pro napojení práv k rolím slouží spojovací tabulka *user\_roles\_to\_right*. Opět se jedná o pomocnou tabulku pro vztah m:n – jedna role může obsahovat více práv, jedno právo může být obsaženo ve více rolích.

**Tabulka č. 18: Tabulka user\_roles\_to\_right**

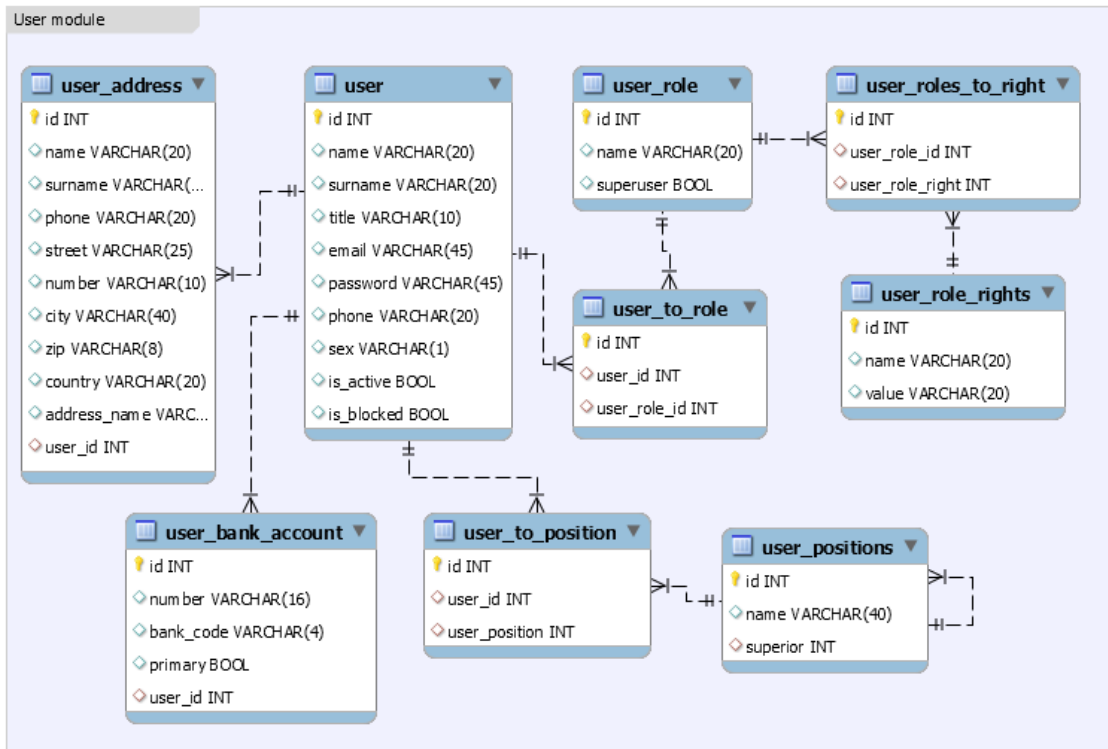
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
user_role_id	int(11)	ID uživatele	Cizí klíč
user_role_right	int(11)	ID práva	Cizí klíč

### 3.1.6 Shrnutí modulu uživatelů

Modul uživatelů se skládá celkem z devíti tabulek. V tomto modulu je vyřešeno ukládání uživatelů, adres uživatelů a bankovních účtů. Dále jsou uživatelům přiřazovány pozice ve firmě a role v informačním systému spolu s právy, které tyto role obsahují.

Na následujícím obrázku je navržené schéma modulu uživatelů se všemi jeho tabulkami a vazbami.



Obr. č. 16: Modul uživatelů  
(Zdroj: vlastní zpracování)



## 3.2 Modul klientů

Modul klientů slouží ke správě a organizaci klientů, pro které společnost vytváří projekty. Jedná se o jednoduchou databázi klientů, které lze přiřazovat do skupin, což napomáhá lepší organizaci klientů. Ke každému klientovi lze přiřadit jednu či více adres a tyto adresy pojmenovat (např. fakturační adresa, dodací adresa apod.).

Je běžné, že společnost komunikuje s více než jednou osobu z klientské firmy. Proto je v modulu klientů vytvořena i databáze osob, které lze přiřazovat ke klientovi.

### 3.2.1 Tabulka client

V tabulce *client* jsou uloženi jednotliví klienti společnost eSports, s.r.o. Skládá se ze tří atributů – jednoznačného identifikátoru (*id*), názvu společnosti (*name*) a identifikačního čísla (*company\_id*). V České republice se používá osmimístné identifikační číslo osoby, v zahraničí je však možné narazit i na delší identifikátory. Proto je délka atributu zvolena s rezervou. Pokud by společnost obchodovala pouze s českými společnostmi, mohl by datový typ této položky být omezen pouze na osm znaků.

#### Tabulka č. 19: Tabulka client

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název společnosti	
company_id	varchar(20)	IČO	

### 3.2.2 Tabulka client\_address

Ke klientům je možné přiřazovat jednu či více adres (fakturační, dodací apod.). K tomu slouží tabulka *client\_address*. Adresy je možné pojmenovat pro lepší orientaci v adresách.

Tabulka *client\_address* má shodnou strukturu jako tabulka *user\_address*, která je použita v modulu uživatelů. Jediným rozdílem je název cizího klíče (u *user\_address* je cizím klíčem *user\_id*, zde *client\_id*) a napojení přes cizí klíč na jinou tabulku (u *user\_address* napojeno na tabulku *user*, zde na tabulku *client*).

**Tabulka č. 20: Tabulka client\_address**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(20)	Jméno	
surname	varchar(20)	Příjmení	
phone	varchar(20)	Telefon	
street	varchar(25)	Ulice	
number	varchar(10)	Číslo	
city	varchar(40)	Město	
zip	varchar(8)	PSČ	
country	varchar(20)	Země	
address_name	varchar(45)	Název adresy	
client_id	int(11)	ID klienta	Cizí klíč

### 3.2.3 Tabulky client\_group a client\_to\_group

Klienty je možné přiřazovat do různých skupin tak, aby bylo dosaženo lepší organizace klientů. K tomu slouží tabulky *client\_group* a *client\_to\_group*. Jeden klient může náležet do více skupin a každá skupina může obsahovat více klientů, je tedy nutné vztah m:n zajistit pomocnou tabulkou, v tomto případě *client\_to\_group*.

V tabulce *client\_group* jsou uloženy jednotlivé skupiny, do kterých lze klienty přiřazovat. Obsahuje čtyři atributy: jednoznačný identifikátor (*id*), název a popis skupiny (*name*, *surname*) a cizí klíč *superior\_group\_id*. Cizí klíč v tomto případě slouží k identifikaci nadřazené skupiny. Díky možnosti zvolit nadřazenou skupinu lze dosáhnout hierarchie mezi skupinami, a tedy k výsledné lepší organizaci klientů.

**Tabulka č. 21: Tabulka client\_group**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název skupiny	
description	varchar(255)	Popis skupiny	
superior_group_id	int(11)	ID nadřazené skupiny	Cizí klíč

Tabulka *client\_to\_group* je spojovací tabulkou. Obsahuje pouze tři atributy: jednoznačný identifikátor (*id*) a dva cizí klíče – ID klienta (*client\_id*) a ID skupiny (*client\_group\_id*).

**Tabulka č. 22: Tabulka *client\_to\_group***

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
client_id	int(11)	ID klienta	Cizí klíč
client_group_id	int(11)	ID skupiny	Cizí klíč

### 3.2.4 Tabulka *client person*

Ke každému klientovi lze přiřadit osoby. Společnost většinou nekomunikuje pouze s jedním kontaktem z klientské firmy, proto lze osob přiřadit více. Tabulka se skládá z jednoznačného identifikátoru (*id*), titulu, jména a příjmení kontaktní osoby (*name*, *surname*). Dále se ukládají kontaktní informace – telefonní číslo (*phone*) a e-mailová adresa (*email*) a pohlaví osoby (*sex*). V tabulce se nachází jeden cizí klíč, a tím je atribut *client\_id*. Ten ukazuje do tabulky *client* a přiřazuje osobu ke klientovi.

**Tabulka č. 23: Tabulka *client\_to\_group***

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(20)	Jméno	
surname	varchar(20)	Příjmení	
title	varchar(10)	Titul	
phone	varchar(45)	Telefon	
email	varchar(45)	E-mail	
sex	varchar(1)	Pohlaví	
client_id	int(11)	ID klienta	Cizí klíč

Každá osoba, kterou ke klientovi přiřadíme, má k naší společnosti nějaký vztah, nebo v klientské společnosti zastává určitou pozici. Aby společnost eSports, s.r.o. měla přehled o těchto osobách, je možné osobám přiřadit pozici.

Pozice osob se ukládají do tabulky *client\_person\_position*. Jelikož jedna osoba může zastávat i více pozic a jednu pozici může zastávat více osob, je nutné vztah m:n zajistit pomocnou tabulkou *client\_person\_to\_position*.

**Tabulka č. 24: Tabulka *client\_person\_position***

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název pozice	

Spojovací tabulka *client\_person\_to\_position* obsahuje pouze tři atributy:

**Tabulka č. 25: Tabulka *client\_person\_to\_position***

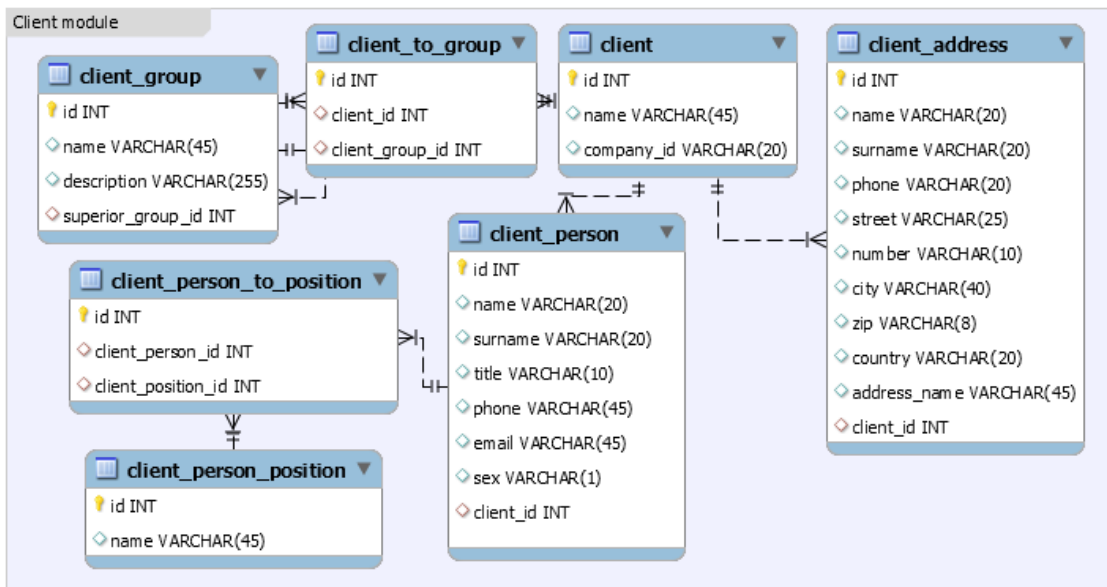
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
client_person_id	int(11)	ID osoby	Cizí klíč
client_position_id	int(11)	ID pozice	Cizí klíč

### 3.2.5 Shrnutí modulu klientů

Modul klientů slouží k ukládání klientů společnosti eSports, s.r.o. Ke každému klientovi je možné přiřadit osoby, které mají z dané klientské společnosti ke společnosti eSports, s.r.o. nějaký vztah. Ke každé osobě je možné přiřadit pozici nebo vztah, který ke společnosti eSports s.r.o. mají.

Klienty lze přiřazovat do skupin tak, aby byli klienti lépe organizováni. Ke každému z klientů je možné přiřadit jednu či více adres a tyto adresy si pojmenovat.



**Obr. č. 17: Modul klientů**  
(Zdroj: vlastní zpracování)

### 3.3 Modul projektů

Modul projektů slouží ke správě nejrůznějších projektů, na kterých společnost pracuje. Požadavkem byla jednoduchá správa projektů, jejich kategorizace, přiřazování pracovníků k projektům a možnost ke každému z projektů vytvářet diskuse. Modul se skládá z devíti tabulek.

#### 3.3.1 Tabulka *project* a *user\_to\_project*

V tabulce *project* jsou uloženy hlavní informace o projektech. K těm patří název (atribut *name*) a popis projektu (*description*). Dále se u projektu vede datum začátku a datum konce jeho platnosti (*date\_start*, *date\_end*) a indikace toho, zda je projekt stále aktivní (*is\_active*). Ke každému projektu lze také přiřadit tzv. Trello API klíč (*trello\_api\_key*), Trello API heslo (*trello\_api\_secret*) a Trello board ID (*trello\_board\_id*), které jsou nutné pro spárování projektu se systémem Trello, který je popsán v kapitole 2.5.2. Díky spárování s programem Trello je možné získávat informace z tohoto systému a zobrazovat je v navrženém informačním systému. Ke každému projektu je také přiřazen klient z modulu klientů, a to pomocí cizího klíče, který obsahuje ID klienta.

**Tabulka č. 26: Tabulka *project***  
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(40)	Název projektu	
description	text	Popis projektu	
date_start	date	Datum začátku projektu	
date_end	date	Datum konce projektu	
is_active	tinyint(1)	Indikátor, zda je projekt aktivní	
status	int(11)	Status projektu	Cizí klíč
trello_api_key	varchar(32)	API klíč k systému Trello	
trello_api_secret	varchar(40)	API heslo k systému Trello	
trello_board_id	varchar(24)	API identifikátor projektu v Trello	
client_id	int(11)	ID klienta	Cizí klíč

Každý projekt má přiřazené pracovníky a klienty. K přiřazení osob slouží spojovací tabulky *user\_to\_project*.

**Tabulka č. 27: Tabulka user\_to\_project**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	
user_id	int(11)	ID uživatele	
project_id	int(11)	ID projektu	
function	varchar(45)	Funkce v projektu	

### 3.3.2 Tabulka project\_status

Ke každému projektu se také přiřazuje status (atribut *status*) z množiny předem navolených statusů (např. v přípravě, ukončeno, odevzdáno apod.). K tomu slouží tabulka *status*, která je s tabulkou *project* propojena cizím klíčem *status*.

**Tabulka č. 28: Tabulka project\_status**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(40)	Název statusu	

### 3.3.3 Tabulky project\_group a project\_to\_group

Projekty lze přiřazovat do skupin, které slouží k lepší kategorizaci projektů. Projekty lze pak snadno třídit do kategorií a mít v nich pořádek. Projektové skupiny jsou uloženy v tabulce *project\_group*. Každá skupina může být podřazená jiné skupině. Díky tomu lze vytvořit hierarchii skupin, což opět přispívá k lepší organizaci projektů.

**Tabulka č. 29: Tabulka project\_group**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(40)	Název statusu	
description	varchar(255)	Popis skupiny	
superior_group	int(11)	Nadřazená skupiny	Cizí klíč

Každý projekt může být přiřazen do více než jedné skupiny, vztah projekt – skupina se tedy musí realizovat přes pomocnou tabulku *project\_to\_group*. Ta se, podobně jako další spojovací tabulky, skládá pouze ze tří atributů: jednoznačného identifikátoru a dvou cizích klíčů, které ukazují do tabulek *project* a *project\_group*.

**Tabulka č. 30: Tabulka *project\_to\_group***

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
project_id	int(11)	ID projektu	Cizí klíč
project_group_id	int(11)	ID skupiny	Cizí klíč

### 3.3.4 Tabulky *project\_label* a *label\_to\_project*

Přiřazení do skupin zásadně vylepšuje organizaci projektů. Další způsob, kterým lze organizaci zlepšit, je možnost přiřadit k projektům štítky. K jednomu projektu lze opět přiřadit více štítků, proto jsou štítky řešeny pomocí hlavní a spojovací tabulky: *project\_labels* a *label\_to\_project*.

Tabulka *project\_labels* bude obsahovat uložené štítky. Ty se skládají z názvu (*name*), popisu (*description*) a barvy štítku (*color*). Barva štítku bude zadávána v hexadecimální podobě, proto je pro tento atribut zvolen datový typ *varchar* s délkou šesti znaků, jelikož hexadecimální zápis barev je právě šestimístný.

**Tabulka č. 31: Tabulka *project\_labels***

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název štítku	
color	varchar(6)	Barva štítku	
description	varchar(255)	Popis štítku	

Štítky jsou s projekty propojeny přes tabulku *label\_to\_project*.



**Tabulka č. 32: Tabulka label\_to\_project**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
project_id	int(11)	ID projektu	Cizí klíč
label_id	int(11)	ID štítku	Cizí klíč

### 3.3.5 Tabulky project\_discussion a project\_comment

Posledním prvkem v modulu projektů jsou diskusní fóra k projektům. Ta jsou uložena v tabulce *project\_discussion*. K projektu může vzniknout více diskusních vláken. Není však možné, aby jedna diskuse patřila k více projektům. Z toho důvodu je vztah řešen vazbou 1:N a není nutné jej řešit pomocnou tabulkou.

Tabulka *project\_discussion* obsahuje jednoznačný identifikátor (*id*), název (*name*) a popis diskuse (*description*), dále časový údaj o otevření, popř. zavření diskuse (*created*, *closed*) a indikátor o zavření diskuse (*is\_closed*). Dále se v tabulce nacházejí dva cizí klíče – *project\_id* pro spojení s konkrétním projektem a *author\_id*. V tomto atributu je uloženo ID uživatele, který diskusi vytvořil.

**Tabulka č. 33: Tabulka project\_discussion**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
project_id	int(11)	ID projektu	Cizí klíč
name	varchar(45)	Název	
description	varchar(255)	Popis diskuse	
is_closed	tinyint(1)	Indikátor, zda je diskuse zavřená	
author_id	int(11)	ID autora	Cizí klíč
created	datetime	Datum vytvoření	
closed	datetime	Datum zavření	

Jednotlivé komentáře v diskusi jsou uloženy v tabulce *project\_comment*. Kromě jednoznačného identifikátoru (*id*), názvu a obsahu příspěvku (*name*, *value*) a data vytvoření a editace příspěvku (*created*, *edited*) obsahuje tabulka tři cizí klíče. Prvním

z klíčů je ID diskuse (*project\_discussion\_id*), který spojuje tabulku komentářů s tabulkou diskusí. Druhým klíčem je *author\_id*, který spojuje tabulku komentářů s tabulkou user z modulu uživatelů. Posledním klíčem je atribut *reply\_to*, který ukazuje do stejné tabulky, tedy *project\_comment*. Pokud je komentář odpovědí na jiný komentář, je v tomto atributu uloženo ID komentáře.

**Tabulka č. 34: Tabulka *project\_comment***

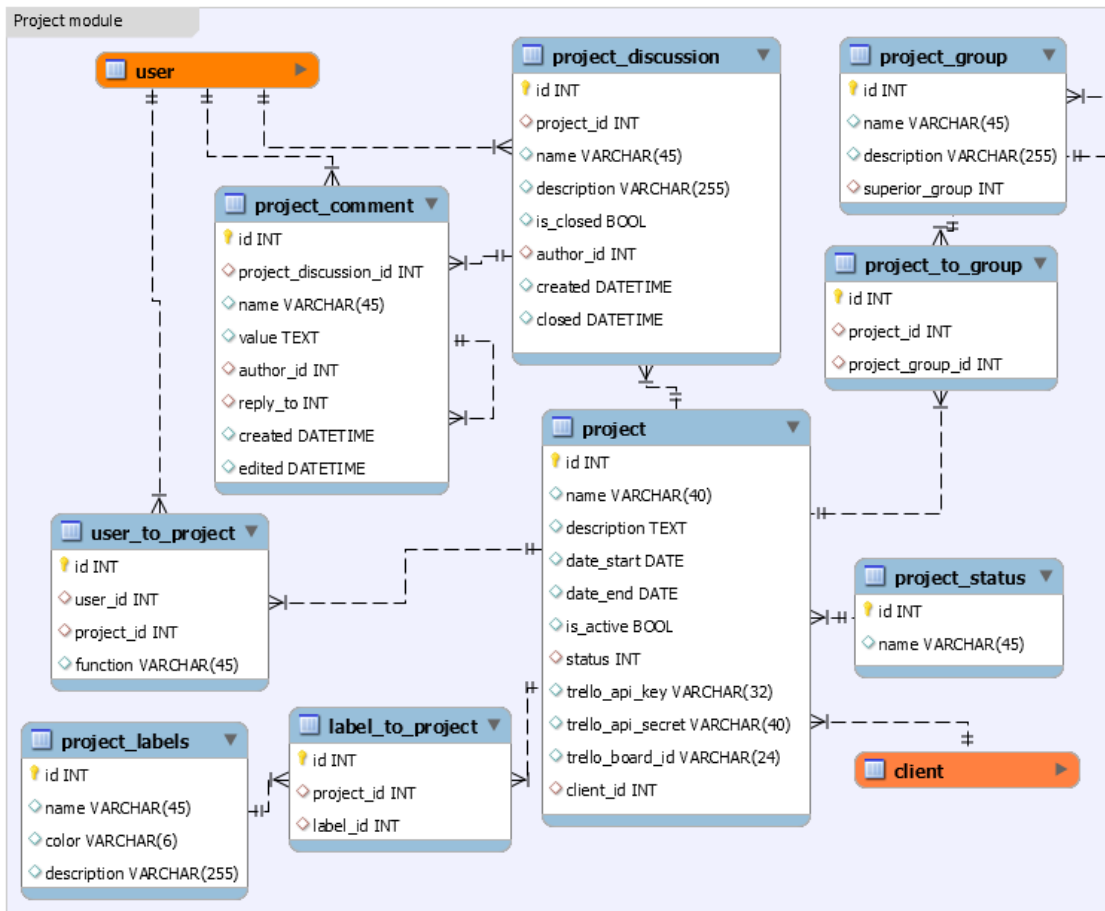
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
project_discussion_id	int(11)	ID diskuse	Cizí klíč
name	varchar(45)	Název příspěvku	
value	text	Obsah příspěvku	
author	int(11)	ID autora	Cizí klíč
reply_to	int(11)	ID komentáře, na který je odpovídáno	Cizí klíč
created	datetime	Datum vytvoření	
edited	datetime	Datum úpravy	

### 3.3.6 Shrnutí modulu projektů

Modul projektů se skládá z devíti tabulek. Slouží k organizaci projektů (zakládání projektů, kategorizace projektů, přiřazování štítků), přiřazování lidských zdrojů k projektům a vedení diskusí u projektů.

Na následujícím obrázku je navržené schéma modulu projektů se všemi jeho tabulkami a vazbami. Oranžově jsou vyznačeny tabulky z jiných modulů.



Obr. č. 18: Modul projektů  
(Zdroj: vlastní zpracování)

### 3.4 Modul tiketů

Modul tiketů slouží k zadávání chybových hlášení či pracovních pokynů k jednotlivým projektům. Pokud se vyskytne problém či požadavek klienta na novou funkcionalitu, je tento požadavek zadán ve formě tzv. tiketu do informačního systému. K tiketu jsou přiřazeni pracovníci, kteří jej mají za úkol zpracovat. Každému tiketu se také přiřazuje status a kategorie tak, aby bylo možné tikety lépe třídít a kategorizovat.

#### 3.4.1 Tabulka ticket a user\_to\_ticket

V tabulce *ticket* jsou uloženy jednotlivé tikety. Jednotlivé tikety se skládají z jednoznačného identifikátoru (*id*), názvu (*name*) a popisu, ve kterém je vysvětlen problém či požadavek (*description*). U každého tiketu je také zaznamenáno datum vzniku požadavku (*created*) a datum uzavření tiketu (*closed*).

Tabulka dále obsahuje čtyři cizí klíče, které propojují tiket s jinými tabulkami. Atribut *author\_id* spojuje tabulku *ticket* s tabulkou *user* a slouží k označení autora tiketu. Atribut *project\_id* spojuje tabulku *ticket* s tabulkou *project* (tiket vždy patří k nějakému projektu). Atribut *status* spojuje tabulku *ticket* s tabulkou *ticket\_status*, kde jsou uloženy předvolené statusy pro tikety. Atribut *category\_id* spojuje tabulku *ticket* s tabulkou *ticket\_category*, kde jsou uloženy předvolené kategorie tiketů.

#### Tabulka č. 35: Tabulka ticket

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název tiketu	
description	text	Popis tiketu	
created	datetime	Datum vytvoření	
closed	datetime	Datum uzavření tiketu	
author_id	int(11)	ID autora	Cizí klíč
project_id	int(11)	ID projektu	Cizí klíč
status	int(11)	ID statusu	Cizí klíč
category_id	int(11)	ID kategorie	Cizí klíč

K jednotlivým tiketům jsou přiřazováni pracovníci, kteří mají daný tiket na starost. K tomu slouží tabulka *user\_to\_ticket*, která slouží jako propojovací tabulka mezi tabulkami *user* a *ticket*.

**Tabulka č. 36: Tabulka user\_to\_ticket**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
user_id	int(11)	ID uživatele	Cizí klíč
ticket_id	int(11)	ID tiketu	Cizí klíč

### 3.4.2 Tabulka ticket\_category a ticket\_status

V tabulce *ticket\_category* jsou uloženy kategorie, které lze tiketu přiřadit. Kategorií může být např. oprava systému, vylepšení funkce apod. Díky přiřazení kategorie k tiketu jsou tikety lépe organizovány a je snazší se v nich orientovat.

**Tabulka č. 37: Tabulka ticket\_category**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název kategorie	

V tabulce *ticket\_status* jsou uloženy statusy, které lze tiketu přiřadit. Statusem může být např. vyřešeno, odloženo, řeší se apod. Díky přiřazeným statusům je možné filtrovat tikety a lépe se v nich orientovat.

**Tabulka č. 38: Tabulka ticket\_status**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název statusu	

### 3.4.3 Tabulka `ticket_comment`

K jednotlivým tiketům lze přiřazovat komentáře, což slouží např. k upřesnění úkolů ze zadání tiketu mezi zadavatelem a osobou odpovědnou za provedení úkolu. K těmto účelům slouží komentáře k tiketu, které se ukládají do tabulky `ticket_comment`.

Tabulka obsahuje jednoznačný identifikátor (`id`), název komentáře (`name`), obsah komentáře (`value`) a také datum vytvoření a datum editace komentáře (`created`, `edited`). Dalšími třemi atributy jsou cizí klíče. Atribut `author_id` ukazuje do tabulky `user` a identifikuje autora daného komentáře. Atribut `ticket_id` ukazuje do tabulky `ticket` a slouží k přiřazení komentáře k jednotlivému tiketu. Poslední atribut, `reply_to`, ukazuje zpět do tabulky `ticket_comment` a identifikuje komentář, na který bylo odpovězeno (pokud na některý komentář bylo odpovězeno).

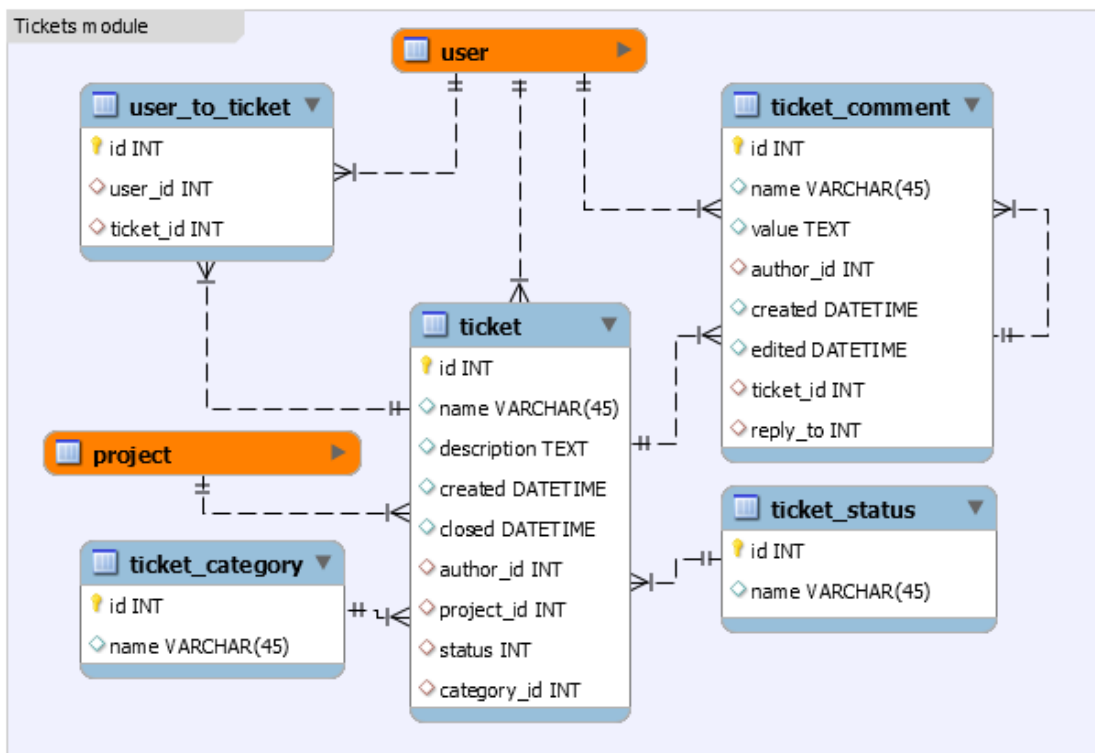
**Tabulka č. 39: Tabulka `ticket_comment`**  
(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
<code>id</code>	<code>int(11)</code> AI	Jednoznačný identifikátor	Primární klíč
<code>name</code>	<code>varchar(45)</code>	Název komentáře	
<code>value</code>	<code>text</code>	Obsah komentáře	
<code>author_id</code>	<code>int(11)</code>	ID autora	Cizí klíč
<code>created</code>	<code>datetime</code>	Datum a čas vytvoření	
<code>edited</code>	<code>datetime</code>	Datum a čas úpravy	
<code>ticket_id</code>	<code>int(11)</code>	ID tiketu	Cizí klíč
<code>reply_to</code>	<code>int(11)</code>	ID komentáře, na který je odpovídáno	Cizí klíč

### 3.4.4 Shrnutí modulu tiketů

Modul tiketů slouží k práci s tzv. tikety. Ty slouží k zadávání práce na projektech či k hlášení chyb na existujících projektech. V modulu tiketů je celkem pět tabulek, slouží k ukládání tiketů, přiřazování kategorií a statusů jednotlivým tiketům a k vytváření komentářů k tiketům.

Na následujícím obrázku je navržené schéma modulu projektů se všemi jeho tabulkami a vazbami.



**Obr. č. 19: Modul tiketů**  
(Zdroj: vlastní zpracování)

## 3.5 Modul diskusí

Kromě diskusí u jednotlivých projektů společnost eSports, s.r.o. využívá diskusní vlákna pro řešení interních záležitostí. K tomu v novém informačním systému bude sloužit modul diskusí.

### 3.5.1 Tabulka *discussion*

Tabulka *discussion* má stejnou strukturu jako *project\_discussion*. Obsahuje tedy jednoznačný identifikátor (*id*), název (*name*) a popis diskuse (*description*), dále časový údaj o otevření, popř. zavření diskuse (*created*, *closed*) a indikátor o zavření diskuse (*is\_closed*). Oproti tabulce *project\_discussion* chybí cizí klíč *project\_id*, jelikož obecné diskuse nejsou navázány na konkrétní projekt. V atributu *author\_id* je uloženo ID uživatele, který diskusi vytvořil.

#### Tabulka č. 40: Tabulka *discussion*

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název	
description	varchar(255)	Popis diskuse	
is_closed	tinyint(1)	Indikátor, zda je diskuse zavřená	
author_id	int(11)	ID autora	Cizí klíč
created	datetime	Datum vytvoření	
closed	datetime	Datum zavření	

### 3.5.2 Tabulka *discussion\_comment*

Jednotlivé komentáře v diskusi jsou uloženy v tabulce *discussion\_comment*. Kromě jednoznačného identifikátoru (*id*), názvu a obsahu příspěvku (*name*, *value*) a data vytvoření a editace příspěvku (*created*, *edited*) obsahuje tabulka tři cizí klíče. Prvním z klíčů je ID diskuse (*discussion\_id*), který spojuje tabulku komentářů s tabulkou diskusí. Druhým klíčem je *author\_id*, který spojuje tabulku komentářů s tabulkou user z modulu uživatelů. Posledním klíčem je atribut *reply\_to*, který ukazuje do stejné tabulky, tedy *discussion\_comment*. Pokud je komentář odpovědí na jiný komentář, je v tomto atributu uloženo ID komentáře.



**Tabulka č. 41: Tabulka discussion\_comment**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
discussion_id	int(11)	ID diskuse	Cizí klíč
name	varchar(45)	Název příspěvku	
value	text	Obsah příspěvku	
author	int(11)	ID autora	Cizí klíč
reply_to	int(11)	ID komentáře, na který je odpovídáno	Cizí klíč
created	datetime	Datum vytvoření	
edited	datetime	Datum úpravy	

### 3.5.3 Tabulky discussion\_category a discussion\_to\_category

Diskuse je možné přiřadit do kategorií, a to i do více než jedné zároveň. Jednotlivé kategorie jsou uloženy v tabulce *discussion\_category*, která má dva atributy: jednoznačný identifikátor (*id*) a název skupiny (*name*).

**Tabulka č. 42: Tabulka discussion\_category**

(Zdroj: vlastní zpracování)

Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
name	varchar(45)	Název skupiny	

Pomocná tabulka, která řeší vztah m:n diskusí a kategorií, se nazývá *discussion\_to\_category* a skládá se z jednoznačného identifikátoru (*id*) a dvou cizích klíčů. Klíč *discussion\_id* míří do tabulky *discussion*, klíč *category\_id* míří do tabulky *discussion\_category*.

**Tabulka č. 43: Tabulka discussion\_to\_category**

(Zdroj: vlastní zpracování)

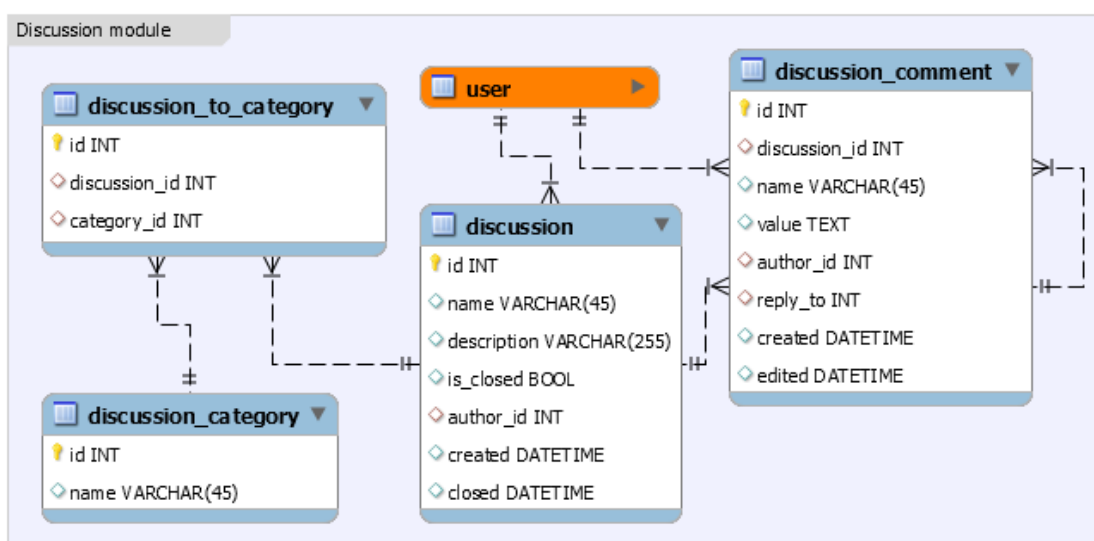
Název	Datový typ	Komentář	Klíče
id	int(11) AI	Jednoznačný identifikátor	Primární klíč
discussion_id	int(11)	ID diskuse	Cizí klíč
category_id	int(11)	ID kategorie	Cizí klíč

### 3.5.4 Shrnutí modulu diskusí

Modul diskusí slouží k vytváření diskusních vláken a přidávání komentářů do těchto vláken. Oproti projektovým diskusím tyto obecné diskuse nejsou navázané na konkrétní projekt a slouží spíše k interním diskusím.

Každou diskusi lze přiřadit do jedné či více skupin tak, aby bylo možné diskuse lépe organizovat a bylo snadnější se v nich orientovat.

Na následujícím obrázku je navržené schéma modulu diskusí se všemi jeho tabulkami a vazbami.



Obr. č. 20: Modul diskusí  
(Zdroj: vlastní zpracování)

### 3.6 Ekonomické zhodnocení

Vytvoření nového informačního systému pro společnost přináší jisté náklady spojené s vývojem. Informační systém bude vytvořen ve třech fázích.

V **první fázi** jsem analyzoval firemní procesy a firemní informační systémy a navrhl datový model informačního systému. Náklady na tuto činnost lze vyjádřit zcela přesně, jelikož činnost již byla provedena. Celkový čas, který jsem strávil v první fázi, je 60 hodin.

Ve **druhé fázi** bude naprogramována serverová i klientská část. Tato fáze bude realizována v průběhu roku 2015 pracovníky společnosti eSports, s.r.o. Odhad času nutného pro realizaci druhé fáze je přibližně 250 člověkohodin. Pokud by v druhé fázi pracovali dva programátoři, každý s platem 500 korun na hodinu, byla by celková cena za druhou fázi 125000.

Ve **třetí fázi** se informační systém bude testovat a po řádném otestování bude uveden do běžného provozu.

**Tabulka č. 44: Ekonomické zhodnocení**  
(Zdroj: vlastní zpracování)

Činnost	Počet hodin	Hod. mzda (Kč)	Cena (Kč s DPH)
<b>První fáze</b>			
Analýza firemních procesů a IS	30	400	12 000
Návrh databázového modelu	30	400	12 000
<b>Celkem za první fázi</b>			<b>24 000</b>
<b>Druhá fáze</b>			
Programování – serverová část	150	500	75 000
Programování – klientská část	100	500	50 000
<b>Celkem za druhou fázi</b>			<b>125 000</b>
<b>Třetí fáze</b>			
Testování	50	250	12 500
Uvedení do provozu	10	500	5 000
<b>Celkem za třetí fázi</b>			<b>17 500</b>
<b>Výsledná částka</b>			<b>166 500</b>

Vytvořením nového informačního systému, který bude integrovat stávající informační systémy, dojde ke zjednodušení práce a vyšší efektivnosti činností. Pracovníci nebudou muset používat oddělené informační systémy, bude jim stačit jeden. Nový informační systém bude obsahovat správu uživatelů, klientů, projektů, tiketů a interních diskusí.

## 4 Závěr

V této diplomové práci jsem řešil vývoj informačního systému pro společnost eSports s.r.o. Mým cílem bylo vytvoření datového modelu pro informační systém.

Diplomovou práci jsem rozdělil do tří částí. V **první části** jsem předložil teoretické poznatky nutné k pochopení celé problematiky, ať už vývoje informačního systému či návrhu datového modelu. Ve **druhé části** jsem analyzoval hodnototvorné a podpůrné firemní procesy a analyzoval jsem informační systémy a programy, které firma aktuálně používá při své činnosti.

Ve **třetí části** jsem navrhl datový model nového informačního systému s přispěním poznatků z předchozích dvou částí. Datový model se skládá celkem z pěti modulů, které obsahují 34 tabulek.

**Modul uživatelů** obsahuje informace o uživateli společnosti. Dovoluje ukládat informace o uživateli včetně pozic, které zastávají ve společnosti, a rolí, které jsou jim přiřazené v informačním systému. **Modul klientů** obsahuje informace o klientech společnosti. Ke každému klientovi je možné přiřadit osoby, které mají z dané společnosti nějaký vztah ke společnosti eSports s.r.o. **Modul projektů** slouží k organizaci projektů (zakládání projektů, kategorizace projektů, přiřazování štítků), přiřazování lidských zdrojů k projektům a vedení diskusí u projektů. **Modul tiketů** slouží k práci s tzv. tikety. Pomocí nich se zadávají nové požadavky na úpravu projektů nebo oznámení o chybách v existujících projektech. **Modul diskusí** slouží k vytváření diskusních vláken a přidávání komentářů do těchto vláken. Oproti projektovým diskusím tyto obecné diskuse nejsou navázané na konkrétní projekt a slouží spíše k interním diskusím.

Navrhnutý datový model bude pro společnost eSports, s.r.o. sloužit při dalším vývoji informačního systému. K navrhnutému datovému modelu bude naprogramována serverová a klientská část. Dokončení informačního systému se plánuje do konce roku 2015. Celkové náklady na vývoj informačního systému (navržení datového modelu, programování, testování a implementace) byly spočítány na 166 500 Kč s DPH.

## Seznam použitých zdrojů

- (1) WILSON, John P. *Human resource development: learning*. 2nd ed. Sterling, VA: Kogan Page Limited, 2005, xxvi, 564 p. ISBN 07-494-4352-9.
- (2) Talent & Performance. CLARK, Donald. *Big Dog and Little Dog's Performance Juxtaposition* [online]. 2004 [cit. 2015-01-16]. Dostupné z: <http://www.nwlink.com/~donclark/performance/understanding.html>
- (3) ELIAS M. AWAD, Elias M.Hassan M. *Knowledge management*. 1st ed. Delhi: Dorling Kindersley, licensees of Pearson Education in South Asia, 2007. ISBN 978-813-1714-034.
- (4) WANG, Kesheng, Ove Rustung HJELMERVIK a Bernt BREMDAL. *Introduction to knowledge management: principles and practice*. Trondheim, Norway: Tapir Academic Press, c2001, xiii, 329 p. ISBN 82-519-1660-7.
- (5) BIERLY III, Paul E., Eric H. KESSLER a Edward W CHRISTENSEN. Organizational learning, knowledge and wisdom. *Journal of organizational change management*, 2000, 13.6: 595-618.
- (6) GÁLA, Libor, Jan POUR a Zuzana ŠEDIVÁ. *Podniková informatika. 2.*, přeprac. a aktualiz. vyd. Praha: Grada, 2009, 496 s. Expert (Grada). ISBN 978-80-247-2615-1.
- (7) BRUCKNER, Tomáš. *Tvorba informačních systémů: principy, metodiky, architektury*. 1. vyd. Praha: Grada, 2012, 357 s. Management v informační společnosti. ISBN 978-80-247-4153-6.
- (8) HRONEK, Jiří. *Informační systémy*. Olomouc, 2007. Dostupné z: <http://phoenix.inf.upol.cz/esf/ucebni/infoSys.pdf>. Učební text. Univerzita Palackého.
- (9) OZ, Effy. *Management information systems*. 6th ed. Boston, Mass.: Thomson/Course Technology, c2009, xxvii, 564 p. ISBN 14-239-0178-9.
- (10) HIRSCHHEIM, R, Heinz-Karl KLEIN a Kalle LYYTINEN. *Information systems development and data modeling: conceptual and philosophical foundations*. New York: Cambridge University Press, 1995, xiv, 289 p. ISBN 05-213-7369-7.
- (11) MARAKAS, George M a James A O'BRIEN. *Introduction to information systems*. 16th ed. New York, NY: McGraw-Hill/Irwin, c2013, xxxii, 732 p. ISBN 00-775-0643-X.

- (12) KIMBLE, Chris. Different Types of Information System and the Pyramid Model. *Chris-Kimble.Com for books, papers and courses from Chris Kimble* [online]. 2010 [cit. 2015-01-21]. Dostupné z: [http://www.chris-kimble.com/Courses/World\\_Med\\_MBA/Types-of-Information-System.html](http://www.chris-kimble.com/Courses/World_Med_MBA/Types-of-Information-System.html)
- (13) BERNSTEIN, Philip A. a Eric NEWCOMER. *Principles of transaction processing*. 2nd ed. Burlington, MA: Morgan Kaufmann Publishers, 2009. ISBN 978-008-0948-416.
- (14) PATRICK, Dale R a Stephen W FARDO. *Industrial process control systems*. 2nd ed. Atlanta, GA: Fairmont Press, Inc., 2009, x, 449 p. ISBN 978-143-9815-762.
- (15) BAGAD, V. S. *Management Information Systems*. Pune: Technical Publications, 2009. ISBN 9788184315769.
- (16) SODOMKA, Petr a Hana KLČOVÁ. 2010. *Informační systémy v podnikové praxi*. 2. aktualiz. a rozš. vyd. Brno: Computer Press, 501 s. ISBN 978-80-251-2878-7.
- (17) BASL, Josef. 2008. *Podnikové informační systémy: podnik v informační společnosti*. 2., výrazně přeprac. a rozš. vyd. Praha: Grada, 283 s. Management v informační společnosti. ISBN 978-80-247-2279-5.
- (18) LAUDON, Kenneth C a Jane Price LAUDON. *Management information systems: managing the digital firm*. 12th ed. Boston: Prentice Hall, c2012, 1 v. (various pagings). ISBN 01-321-4285-6.
- (19) GILL, P.S. *Database management systems*. New Delhi: I.K. International, 2008. ISBN 978-818-9866-839.
- (20) CONNOLLY, Thomas. *Database Systems: A Practical Approach to Design, Implementation, and Management*. 4th ed. Harlow: Addison-Wesley, 2005, 1374 s. ISBN 03-212-1025-5.
- (21) GARCIA-MOLINA, Hector, Jeffrey D ULLMAN a Jennifer WIDOM. *Database systems: the complete book*. 2nd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2009, xxxvi, 1203 p. ISBN 01-318-7325-3.
- (22) PETER ROB, Carlos Coronel. *Database systems: design, implementation*. International ed. London: Cengage Learning, 2008. ISBN 978-184-4807-321.

- (23) Data Modeling in System Analysis. GAO, Shuang. *University of Missouri-St. Louis* [online]. 2013 [cit. 2015-01-28]. Dostupné z:  
<http://www.umsl.edu/~sgfb7/#a6>
- (24) PONNIAH, Paulraj. *Data Modeling Fundamentals a Practical Guide for IT Professionals*. Hoboken: John Wiley, 2007. ISBN 978-047-0141-014.
- (25) JOHNSON, Eric a Joshua JONES. *A developer's guide to data modeling for SQL server: covering SQL server 2005 and 2008*. Upper Saddle River, NJ: Addison-Wesley, c2008, xix, 277 p. ISBN 978-032-1497-642.
- (26) WITT, Graeme C. Simson and Graham C. *Data Modeling Essentials*. 3rd ed. Burlington: Elsevier, 2004. ISBN 978-008-0488-677.
- (27) GARCIA-MOLINA, Hector, Jeffrey D ULLMAN a Jennifer WIDOM. *Database systems: the complete book*. 2nd ed. Upper Saddle River, N.J.: Pearson Prentice Hall, c2009, xxxvi, 1203 p. ISBN 01-318-7325-3.
- (28) Models of Database Architecture: Hierarchical, Network and Relational Models. PUJARI, Saritha. *YourArticleLibrary.com: The Next Generation Library* [online]. 2015 [cit. 2015-01-29]. Dostupné z:  
<http://www.yourarticlelibrary.com/database/models-of-database-architecture-hierarchical-network-and-relational-models/10389/>
- (29) BEHL, Ramesh. *Information technology for management*. New Delhi, India: Tata McGraw-Hill, 2009. ISBN 978-007-0144-927.
- (30) MARSTON, Tony. The Relational Data Model, Normalisation and effective Database Design. *Tony Marston's Page on PHP/MySQL stuff* [online]. 30. 9. 2004, 12. 8. 2005 [cit. 2015-01-29]. Dostupné z: <http://www.tonymarston.net/php-mysql/database-design.html>
- (31) PEST analýza. *EDOLO s.r.o.* [online]. 2014 [cit. 2015-04-16]. Dostupné z:  
<http://www.edolo.cz/sluzby-pro-expanzi/pest-analyza/>
- (32) Kde se vzala a k čemu je PEST analýza. ZIKMUND, Martin. *BusinessVize.cz: Informace pro vaše podnikání* [online]. 29. 11. 2010 [cit. 2015-04-16]. Dostupné z:  
<http://www.businessvize.cz/planovani/kde-se-vzala-a-k-cemu-je-pest-analyza>
- (33) Porterova analýza 5 sil vám prozradí, co ovlivní váš business. ZIKMUND, Martin. *BusinessVize.cz: Informace pro vaše podnikání* [online]. 6. 1. 2011 [cit.



- 2015-04-16]. Dostupné z: <http://www.businessvize.cz/planovani/porterova-analyza-5-sil-vam-prozradi-co-ovlivni-vas-business>
- (34) The McKinsey 7-S Framework. *Mind Tools: Management Training and Leadership Training* [online]. 2015 [cit. 2015-04-16]. Dostupné z: [http://www.mindtools.com/pages/article/newSTR\\_91.htm](http://www.mindtools.com/pages/article/newSTR_91.htm)
- (35) SWOT analýza odhalí pravdivou tvář vaší firmy a pomůže vám nahlédnout do budoucnosti. *IPodnikatel.cz* [online]. 26. 10. 2011 [cit. 2015-04-16]. Dostupné z: <http://www.ipodnikatel.cz/Marketing/swot-analyza-odhali-pravdivou-tvar-vasi-firmy-a-pomuze-vam-nahlednout-do-budoucnosti.html>
- (36) Moody's: Czech Republic benefits from rebounding economy, though political instability and weak fiscal framework pose challenges. *Moody's: credit ratings, research, tools and analysis for the global capital markets* [online]. 10. 12. 2014 [cit. 2015-04-16]. Dostupné z: [https://www.moody.com/research/Moodys-Czech-Republic-benefits-from-rebounding-economy-though-political-instability--PR\\_314822](https://www.moody.com/research/Moodys-Czech-Republic-benefits-from-rebounding-economy-though-political-instability--PR_314822)
- (37) Trello: Demo project. *Trello* [online]. 2011 [cit. 2015-05-01]. Dostupné z: <https://trello.com/b/KaRKUPfD/demo-project>

# Seznamy

## Seznam tabulek

DATA, INFORMACE, ZNALOSTI, VĚDĚNÍ.....	15
ZÁKLADNÍ INFORMACE O SPOLEČNOSTI.....	33
SWOT ANALÝZA.....	41
RACI MATICE PROCESU ZPRACOVÁNÍ ZAKÁZKY .....	43
RACI MATICE PROCESU ŘEŠENÍ ZÁKAZNICKÝCH POŽADAVKŮ .....	45
RACI MATICE PROCESU PLÁNOVÁNÍ ZAHRANIČNÍCH CEST.....	47
RACI MATICE PROCESU OBSLUHA PROJEKTU ONLAJNY.....	49
RACI MATICE PROCESU VYHOTOVENÍ ODMĚN .....	51
RACI MATICE PROCESU FAKTURACE .....	53
TABULKA USER.....	59
TABULKA USER_ADDRESS.....	60
TABULKA USER_BANK_ACCOUNT.....	61
TABULKA USER_POSITIONS.....	61
TABULKA USER_TO_POSITION.....	62
TABULKA USER_ROLE .....	62
TABULKA USER_TO_ROLE.....	63
TABULKA USER_ROLE_RIGHTS .....	63
TABULKA USER_ROLES_TO_RIGHT.....	63
TABULKA CLIENT.....	65
TABULKA CLIENT_ADDRESS.....	66
TABULKA CLIENT_GROUP.....	66
TABULKA CLIENT_TO_GROUP .....	67
TABULKA CLIENT_TO_GROUP .....	67
TABULKA CLIENT_PERSON_POSITION.....	68
TABULKA CLIENT_PERSON_TO_POSITION .....	68
TABULKA PROJECT.....	70
TABULKA USER_TO_PROJECT.....	71
TABULKA PROJECT_STATUS.....	71
TABULKA PROJECT_GROUP.....	71
TABULKA PROJECT_TO_GROUP .....	72
TABULKA PROJECT_LABELS .....	72
TABULKA LABEL_TO_PROJECT .....	73
TABULKA PROJECT_DISCUSSION .....	73
TABULKA PROJECT_COMMENT .....	74

TABULKA TICKET.....	76
TABULKA USER_TO_TICKET.....	77
TABULKA TICKET_CATEGORY.....	77
TABULKA TICKET_STATUS.....	77
TABULKA TICKET_COMMENT.....	78
TABULKA DISCUSSION.....	80
TABULKA DISCUSSION_COMMENT.....	81
TABULKA DISCUSSION_CATEGORY.....	81
TABULKA DISCUSSION_TO_CATEGORY.....	81
EKONOMICKÉ ZHODNOCENÍ.....	83

## Seznam obrázků

OBR. Č. 1: DATA, INFORMACE, ZNALOSTI, VĚDĚNÍ.....	13
OBR. Č. 2: ROZDĚLENÍ INFORMAČNÍCH SYSTÉMŮ.....	18
OBR. Č. 3: HOLISTICKO-PROCESNÍ POHLED NA PODNIKOVÉ IS.....	21
OBR. Č. 4: DATABÁZOVÝ SYSTÉM.....	22
OBR. Č. 5: REÁLNÝ SVĚT, DATOVÝ MODEL A DATABÁZOVÝ SYSTÉM.....	27
OBR. Č. 6: RELAČNÍ DATOVÝ MODEL.....	30
OBR. Č. 7: LOGO SPOLEČNOSTI ESPORTS, S.R.O.....	33
OBR. Č. 8: ORGANIZAČNÍ STRUKTURA SPOLEČNOSTI.....	35
OBR. Č. 9: EPC DIAGRAM PROCESU ZPRACOVÁNÍ ZAKÁZKY.....	44
OBR. Č. 10: EPC DIAGRAM PROCESU ŘEŠENÍ POŽADAVKŮ NA TECHNICKÉ PODPOŘE.....	46
OBR. Č. 11: EPC DIAGRAM PROCESU PLÁNOVÁNÍ ZAHRANIČNÍCH CEST.....	48
OBR. Č. 12: EPC DIAGRAM PROCESU OBSLUHA PROJEKTU ONLAJNY.....	50
OBR. Č. 13: EPC DIAGRAM PROCESU VYHOTOVENÍ ODMĚN.....	52
OBR. Č. 14: EPC DIAGRAM PROCESU FAKTURACE.....	54
OBR. Č. 15: TRELLO.....	56
OBR. Č. 16: MODUL UŽIVATELŮ.....	64
OBR. Č. 17: MODUL KLIENTŮ.....	69
OBR. Č. 18: MODUL PROJEKTŮ.....	75
OBR. Č. 19: MODUL TIKETŮ.....	79
OBR. Č. 20: MODUL DISKUSÍ.....	82

## Seznam použitých zkratk

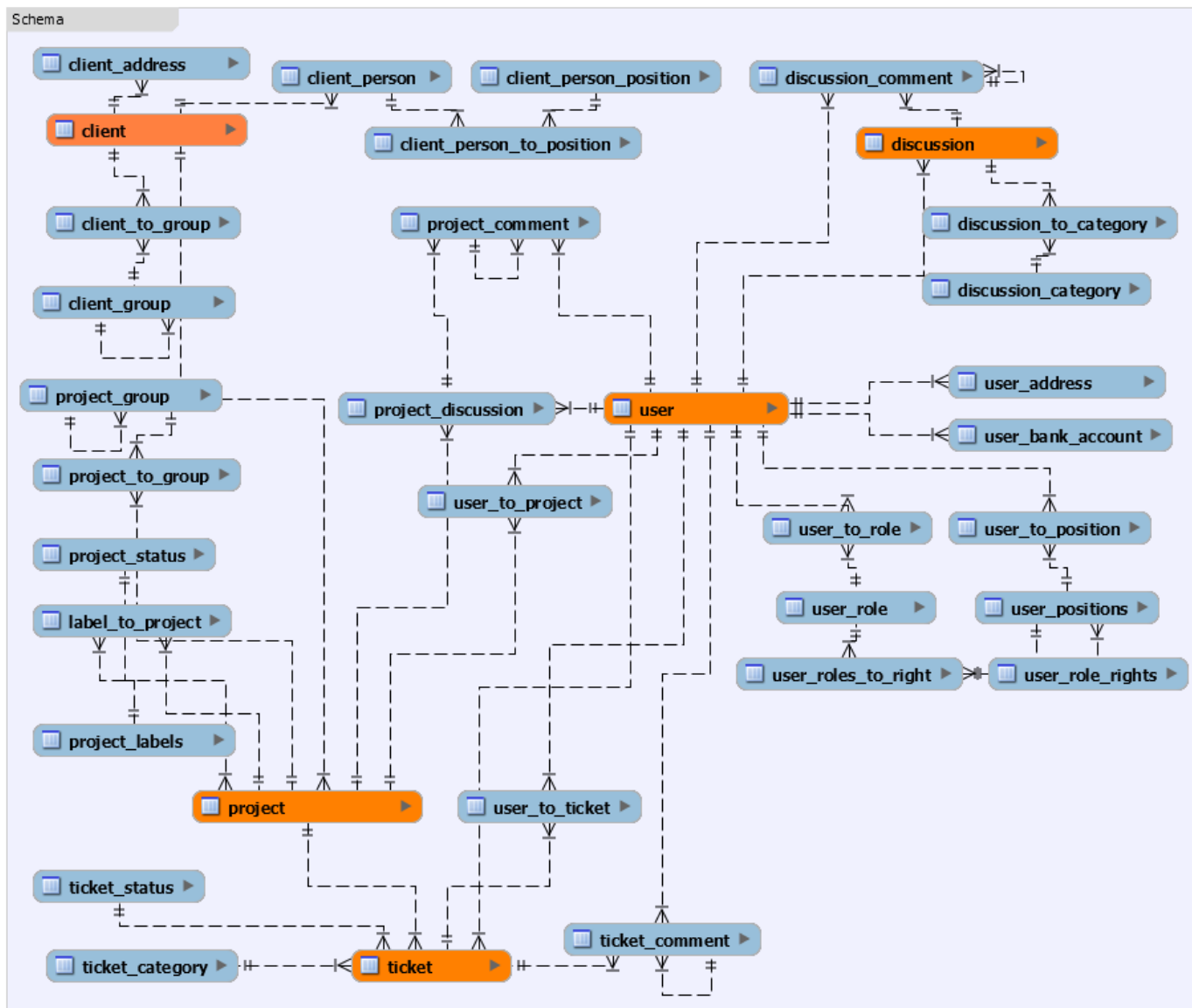
EPC	Event-driven Process Chain
RACI	Responsible, Accountable, Consulted, and Informed
OSS	Operations Support System
MSS	Management Support Systems
TPS	Transaction Processing Systems
PCS	Process Control System
ECS	Enterprise Collaboration System
ERP	Enterprise Resource Planning
CRM	Customer Relationship Management
SCM	Supply Chain Management
MIS	Management Information System
APS	Advanced Planning System
DSS	Decision Support System
SŘBD	System Řízení Báže Dat
DBMS	Database Management System
DML	Data Manipulation Language
ASP	Active Server Pages
HTML	HyperText Markup Language
PHP	PHP: Hypertext Preprocessor

## Seznam příloh

PŘÍLOHA 1: DATABÁZOVÉ SCHÉMA .....	I
PŘÍLOHA 2: ZDROJOVÉ KÓDY MYSQL.....	II

# Přílohy

## Příloha 1: Databázové schéma



## Příloha 2: Zdrojové kódy MySQL

```
CREATE SCHEMA IF NOT EXISTS `is` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `is` ;

CREATE TABLE IF NOT EXISTS `is`.`user` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NULL,
  `surname` VARCHAR(20) NULL,
  `title` VARCHAR(10) NULL,
  `email` VARCHAR(45) NULL,
  `password` VARCHAR(40) NULL,
  `phone` VARCHAR(20) NULL,
  `sex` VARCHAR(1) NULL,
  `is_active` TINYINT(1) NULL DEFAULT 1,
  `is_blocked` TINYINT(1) NULL DEFAULT 0,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_role` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NULL,
  `superuser` TINYINT(1) NULL DEFAULT 0,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_address` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NULL,
  `surname` VARCHAR(20) NULL,
  `phone` VARCHAR(20) NULL,
  `street` VARCHAR(25) NULL,
  `number` VARCHAR(10) NULL,
  `city` VARCHAR(40) NULL,
  `zip` VARCHAR(8) NULL,
  `country` VARCHAR(20) NULL,
  `address_name` VARCHAR(45) NULL,
  `user_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `user_id_idx` (`user_id` ASC),
  CONSTRAINT `user_id_6`
    FOREIGN KEY (`user_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_to_role` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NULL,
  `user_role_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `user_id_idx` (`user_id` ASC),
  INDEX `user_role_id_idx` (`user_role_id` ASC),
  CONSTRAINT `user_id_2`
    FOREIGN KEY (`user_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `user_role_id`
    FOREIGN KEY (`user_role_id`)
    REFERENCES `is`.`user_role` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_positions` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(40) NULL,
  `superior` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `superior_position_id_idx` (`superior` ASC),
  CONSTRAINT `superior_position_id`
    FOREIGN KEY (`superior`)
    REFERENCES `is`.`user_positions` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;
COMMENT = 'Pozice ve společnosti';

CREATE TABLE IF NOT EXISTS `is`.`user_bank_account` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `number` VARCHAR(16) NULL,
  `bank_code` VARCHAR(4) NULL,
  `primary` TINYINT(1) NULL DEFAULT 0,
  `user_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `user_id_idx` (`user_id` ASC),
  CONSTRAINT `user_id_3`
    FOREIGN KEY (`user_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_role_rights` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NULL,
  `value` VARCHAR(20) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_roles_to_right` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_role_id` INT NULL,
  `user_role_right` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `user_role_id_idx` (`user_role_id` ASC),
  INDEX `user_role_right_id_idx` (`user_role_right` ASC),
  CONSTRAINT `user_role_id_2`
    FOREIGN KEY (`user_role_id`)
    REFERENCES `is`.`user_role` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `user_role_right_id`
    FOREIGN KEY (`user_role_right`)
    REFERENCES `is`.`user_role_rights` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_to_position` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NULL,
  `user_position` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `user_id_3_idx` (`user_id` ASC),
  INDEX `user_position_idx` (`user_position` ASC),
  CONSTRAINT `user_id_4`
    FOREIGN KEY (`user_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `user_position`
    FOREIGN KEY (`user_position`)
    REFERENCES `is`.`user_positions` (`id`)
    ON DELETE NO ACTION)
```

```

ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`project_status` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`client` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `company_id` VARCHAR(20) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`project` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(40) NULL,
  `description` TEXT NULL,
  `date_start` DATE NULL,
  `date_end` DATE NULL,
  `is_active` TINYINT(1) NULL,
  `status` INT NULL,
  `trello_api_key` VARCHAR(32) NULL,
  `trello_api_secret` VARCHAR(40) NULL,
  `trello_board_id` VARCHAR(24) NULL,
  `client_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `project_status_idx` (`status` ASC),
  INDEX `client_id_3_idx` (`client_id` ASC),
  CONSTRAINT `project_status`
    FOREIGN KEY (`status`)
      REFERENCES `is`.`project_status` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `client_id_3`
    FOREIGN KEY (`client_id`)
      REFERENCES `is`.`client` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`project_discussion` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `project_id` INT NULL,
  `name` VARCHAR(45) NULL,
  `description` VARCHAR(255) NULL,
  `is_closed` TINYINT(1) NULL,
  `author_id` INT NULL,
  `created` DATETIME NULL,
  `closed` DATETIME NULL,
  PRIMARY KEY (`id`),
  INDEX `project_id_idx` (`project_id` ASC),
  INDEX `discussion_author_idx` (`author_id` ASC),
  CONSTRAINT `project_id`
    FOREIGN KEY (`project_id`)
      REFERENCES `is`.`project` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `discussion_author`
    FOREIGN KEY (`author_id`)
      REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`project_comment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `project_discussion_id` INT NULL,
  `name` VARCHAR(45) NULL,
  `value` TEXT NULL,
  `author_id` INT NULL,
  `reply_to` INT NULL,
  `created` DATETIME NULL,
  `edited` DATETIME NULL,
  PRIMARY KEY (`id`),
  INDEX `project_discussion_idx` (`project_discussion_id`
  ASC),
  INDEX `comment_author_idx` (`author_id` ASC),
  INDEX `reply_to_idx` (`reply_to` ASC),
  CONSTRAINT `project_discussion`
    FOREIGN KEY (`project_discussion_id`)
      REFERENCES `is`.`project_discussion` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `comment_author`
    FOREIGN KEY (`author_id`)
      REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `reply_to`
    FOREIGN KEY (`reply_to`)
      REFERENCES `is`.`project_comment` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`user_to_project` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NULL,
  `project_id` INT NULL,
  `function` VARCHAR(45) NULL,
  PRIMARY KEY (`id`),
  INDEX `project_user_id_idx` (`user_id` ASC),
  INDEX `project_id_2_idx` (`project_id` ASC),
  CONSTRAINT `project_user_id`
    FOREIGN KEY (`user_id`)
      REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `project_id_2`
    FOREIGN KEY (`project_id`)
      REFERENCES `is`.`project` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`project_group` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `description` VARCHAR(255) NULL,
  `superior_group` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `superior_group_idx` (`superior_group` ASC),
  CONSTRAINT `superior_group`
    FOREIGN KEY (`superior_group`)
      REFERENCES `is`.`project_group` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`project_to_group` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `project_id` INT NULL,
  `project_group_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `project_id_3_idx` (`project_id` ASC),
  INDEX `project_group_id_2_idx` (`project_group_id` ASC),
  CONSTRAINT `project_id_3`
    FOREIGN KEY (`project_id`)
      REFERENCES `is`.`project` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `project_group_id_2`
    FOREIGN KEY (`project_group_id`)
      REFERENCES `is`.`project_group` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `is`.`project_labels` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `color` VARCHAR(6) NULL,
  `description` VARCHAR(255) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`label_to_project` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `project_id` INT NULL,
  `label_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `project_id_4_idx` (`project_id` ASC),
  INDEX `label_id_idx` (`label_id` ASC),
  CONSTRAINT `project_id_4`
    FOREIGN KEY (`project_id`)
    REFERENCES `is`.`project` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `label_id`
    FOREIGN KEY (`label_id`)
    REFERENCES `is`.`project_labels` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`ticket_status` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`ticket_category` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`ticket` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `description` TEXT NULL,
  `created` DATETIME NULL,
  `closed` DATETIME NULL,
  `author_id` INT NULL,
  `project_id` INT NULL,
  `status` INT NULL,
  `category_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `ticket_status_idx` (`status` ASC),
  INDEX `ticket_author_idx` (`author_id` ASC),
  INDEX `ticket_project_idx` (`project_id` ASC),
  INDEX `ticket_category_idx` (`category_id` ASC),
  CONSTRAINT `ticket_status`
    FOREIGN KEY (`status`)
    REFERENCES `is`.`ticket_status` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ticket_author`
    FOREIGN KEY (`author_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ticket_project`
    FOREIGN KEY (`project_id`)
    REFERENCES `is`.`project` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ticket_category`
    FOREIGN KEY (`category_id`)
    REFERENCES `is`.`ticket_category` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

```

```
ENGINE = InnoDB;
```

```

CREATE TABLE IF NOT EXISTS `is`.`user_to_ticket` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `user_id` INT NULL,
  `ticket_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `ticket_user_idx` (`user_id` ASC),
  INDEX `ticket_id_idx` (`ticket_id` ASC),
  CONSTRAINT `ticket_user`
    FOREIGN KEY (`user_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ticket_id`
    FOREIGN KEY (`ticket_id`)
    REFERENCES `is`.`ticket` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `is`.`ticket_comment` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,
  `value` TEXT NULL,
  `author_id` INT NULL,
  `created` DATETIME NULL,
  `edited` DATETIME NULL,
  `ticket_id` INT NULL,
  `reply_to` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `ticket_id_2_idx` (`ticket_id` ASC),
  INDEX `ticket_ator_id_2_idx` (`author_id` ASC),
  INDEX `ticket_reply_idx` (`reply_to` ASC),
  CONSTRAINT `ticket_id_2`
    FOREIGN KEY (`ticket_id`)
    REFERENCES `is`.`ticket` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ticket_ator_id_2`
    FOREIGN KEY (`author_id`)
    REFERENCES `is`.`user` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT `ticket_reply`
    FOREIGN KEY (`reply_to`)
    REFERENCES `is`.`ticket_comment` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `is`.`client_address` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(20) NULL,
  `surname` VARCHAR(20) NULL,
  `phone` VARCHAR(20) NULL,
  `street` VARCHAR(25) NULL,
  `number` VARCHAR(10) NULL,
  `city` VARCHAR(40) NULL,
  `zip` VARCHAR(8) NULL,
  `country` VARCHAR(20) NULL,
  `address_name` VARCHAR(45) NULL,
  `client_id` INT NULL,
  PRIMARY KEY (`id`),
  INDEX `client_id_idx` (`client_id` ASC),
  CONSTRAINT `client_id`
    FOREIGN KEY (`client_id`)
    REFERENCES `is`.`client` (`id`)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB;

```

```

CREATE TABLE IF NOT EXISTS `is`.`client_group` (
  `id` INT NOT NULL AUTO_INCREMENT,
  `name` VARCHAR(45) NULL,

```



```

`description` VARCHAR(255) NULL,
`superior_group_id` INT NULL,
PRIMARY KEY (`id`),
INDEX `superior_client_group_idx` (`superior_group_id`
ASC),
CONSTRAINT `superior_client_group`
FOREIGN KEY (`superior_group_id`)
REFERENCES `is`.`client_group` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`client_to_group` (
`id` INT NOT NULL AUTO_INCREMENT,
`client_id` INT NULL,
`client_group_id` INT NULL,
PRIMARY KEY (`id`),
INDEX `client_id_2_idx` (`client_id` ASC),
INDEX `client_group_id_idx` (`client_group_id` ASC),
CONSTRAINT `client_id_2`
FOREIGN KEY (`client_id`)
REFERENCES `is`.`client` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `client_group_id`
FOREIGN KEY (`client_group_id`)
REFERENCES `is`.`client_group` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`client_person` (
`id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(20) NULL,
`surname` VARCHAR(20) NULL,
`title` VARCHAR(10) NULL,
`phone` VARCHAR(45) NULL,
`email` VARCHAR(45) NULL,
`sex` VARCHAR(1) NULL,
`client_id` INT NULL,
PRIMARY KEY (`id`),
INDEX `client_id_2_idx` (`client_id` ASC),
CONSTRAINT `client_id_4`
FOREIGN KEY (`client_id`)
REFERENCES `is`.`client` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`client_person_position` (
`id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(45) NULL,
PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`client_person_to_position`
(
`id` INT NOT NULL AUTO_INCREMENT,
`client_person_id` INT NULL,
`client_position_id` INT NULL,
PRIMARY KEY (`id`),
INDEX `client_person_id_idx` (`client_person_id` ASC),
INDEX `client_person_position_id_idx`
(`client_position_id` ASC),
CONSTRAINT `client_person_id`
FOREIGN KEY (`client_person_id`)
REFERENCES `is`.`client_person` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `client_person_position_id`
FOREIGN KEY (`client_position_id`)
REFERENCES `is`.`client_person_position` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`discussion` (
`id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(45) NULL,
`description` VARCHAR(255) NULL,
`is_closed` TINYINT(1) NULL,
`author_id` INT NULL,
`created` DATETIME NULL,
`closed` DATETIME NULL,
PRIMARY KEY (`id`),
INDEX `discussion_author_idx` (`author_id` ASC),
CONSTRAINT `discussion_author_2`
FOREIGN KEY (`author_id`)
REFERENCES `is`.`user` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`discussion_comment` (
`id` INT NOT NULL AUTO_INCREMENT,
`discussion_id` INT NULL,
`name` VARCHAR(45) NULL,
`value` TEXT NULL,
`author_id` INT NULL,
`reply_to` INT NULL,
`created` DATETIME NULL,
`edited` DATETIME NULL,
PRIMARY KEY (`id`),
INDEX `comment_author_idx` (`author_id` ASC),
INDEX `reply_to_idx` (`reply_to` ASC),
INDEX `discussion_id_idx` (`discussion_id` ASC),
CONSTRAINT `discussion_id_2`
FOREIGN KEY (`discussion_id`)
REFERENCES `is`.`discussion` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `comment_author_2`
FOREIGN KEY (`author_id`)
REFERENCES `is`.`user` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `reply_to_2`
FOREIGN KEY (`reply_to`)
REFERENCES `is`.`discussion_comment` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`discussion_category` (
`id` INT NOT NULL AUTO_INCREMENT,
`name` VARCHAR(45) NULL,
PRIMARY KEY (`id`))
ENGINE = InnoDB;

CREATE TABLE IF NOT EXISTS `is`.`discussion_to_category` (
`id` INT NOT NULL AUTO_INCREMENT,
`discussion_id` INT NULL,
`category_id` INT NULL,
PRIMARY KEY (`id`),
INDEX `discussion_id_idx` (`discussion_id` ASC),
INDEX `discussion_category_id_idx` (`category_id` ASC),
CONSTRAINT `discussion_id`
FOREIGN KEY (`discussion_id`)
REFERENCES `is`.`discussion` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION,
CONSTRAINT `discussion_category_id`
FOREIGN KEY (`category_id`)
REFERENCES `is`.`discussion_category` (`id`)
ON DELETE NO ACTION
ON UPDATE NO ACTION)
ENGINE = InnoDB;

```