**Střední průmyslová škola a Vyšší odborná škola, Písek,**

**Karla Čapka 402**

**397 01 Písek**

**Školní rok: 2021/2022**

**Obor vzdělání: 18-20-M/01 Informační technologie**

**Maturitní práce**

**Oracle – SW pro modely třívrstvé architektury**

Téma číslo:6

Jméno žáka: Jiří Daněk

Třída:B4.I

Vedoucí práce: Mgr. Jana ten Hagenová

**Střední průmyslová škola a Vyšší odborná škola, Písek,**

**Karla Čapka 402**

**397 01 Písek**

**Školní rok: 2021/2022**

**Obor vzdělání: 18-20-M/01 Informační technologie**

**Specializace: Manažerská informatika**

**Zadání maturitní práce**

**Oracle – SW pro modely třívrstvé architektury**

Jméno žáka: Jiří Daněk Třída: B4.I

Téma číslo: 6 Vedoucí práce: Mgr. Jana ten Hagenová

Termín odevzdání: 31. březen 2022

Zadání:

1. Popište možné technologie a proveďte jejich porovnání (včetně ekonomického) pro vytvoření modelů a výstupu kódu DB
2. Vytvořte příklad třívrstvé architektury a popište přechody jednotlivých vrstev
3. Na navržené databázi sestavte příklady sestav a možných SQL dotazů (např. i v MS Access)
4. Sestavte prezentaci možné vizualizace jednotlivých modelů a jejich výstupů

**Kritéria hodnocení maturitní práce:**

nutné parametry práce, které musí být splněny, aby práce byla uznána a byla hodnocena:

1. splněn požadovaný minimální rozsah vlastního textu práce v rozsahu 15 stran textu s přiměřeným množstvím obrázků a tabulek **nezbytně nutných** k popisu/výkladu problému řešeného v textu
2. splněna struktura práce:
   1. teoretický úvod k problematice řešené v práci v rozsahu max. 6 strany
   2. popis autorského řešení zadaného úkolu, doplněného výpočty, výkladem algoritmů, obrázky, které jsou nezbytně nutné k vyřešení částí zadání, v rozsahu min. 8 stran
   3. závěr hodnotící dosažené výsledky v rozsahu min. 1 normované strany
3. předvedeny výstupy realizační části práce vedoucímu práce a oponentovi práce

**Pokud práce nesplňuje předchozí tři kritéria, je hodnocena: *nedostatečně.* Pokud jsou předchozí kritéria splněna, je práce hodnocena**:

**vedoucí oponent**

* odpovědnost a přístup žáka v průběhu řešení zadání: (0 – 10) % 0 %
* dodržení obsahové/grafické struktury maturitní práce: (0 – 10) % (0 – 10) %
* originalita a vhodnost řešení: (0 – 25) % (0 – 35) %

(***konkretizuje vedoucí práce ve 2 až 5 bodech podle***

***požadovaných výstupů práce***)

* + Žák popsal a porovnal SW platformy pro modelování DB
  + Je vytvořen návrh relační databáze pomocí třívrstvé architektury
  + Je vytvořeno srovnání SW a jejich grafických výstupů Microsoft a Oracle
* funkčnost řešení: (0 – 30) % (0 – 30) %

(***vedoucí práce ve 2 až 5 bodech konkretizuje podle***

***požadovaných kritérií funkčnosti***)

* + Žák použil minimálně 3 SW Oracle a MS Access
  + Na základě modelu je vytvořena DB, včetně možných výstupů
  + Je vytvořena prezentace postupu vývoje a reverzního inženýringu v třívrstvé architektuře
* vlastní obhajoba: (0 – 25) % (0 – 25) %

**Klasifikační stupnice:**

nedostatečný (0 – 30) %, dostatečný (31 – 47) %, dobrý (48 – 65) %, chvalitebný (66 % – 83) %, výborný (84 % – 100) %.

**Způsob zpracování a pokyny k obsahu a rozsahu maturitní práce:**

* práce bude zpracována podle platného metodického pokynu pro zadání a realizaci maturitní práce Č. j.:SPŠ/1211/2013(dostupný na: n:\!maturita\MetodickýPokyn\...)
* kompletní práce se odevzdává do informačního střediska školy v jednom tištěném exempláři doplněném elektronickým nosičem dat (CD, DVD, USB flash disk, SD karta), na kterém bude uvedena kompletně zpracovaná práce včetně příloh. V případě tvorby software, také zdrojový kód navrženého software. V případě projektu, také projektová dokumentace (podrobná technická zpráva, úplná výkresová dokumentace, podrobný rozpočet)
* délka obhajoby maturitní práce před zkušební maturitní komisí je stanovena na 15 minut
* náklady na materiál bude hradit: škola/~~firma~~/~~žák[[1]](#footnote-1)~~
* funkční vzorek bude majetkem: školy/~~firmy~~/~~žáka~~

V Písku 16. listopadu 2021 Ing. Jiří Uhlík

*ředitel SPŠ a VOŠ Písek*

**Anotace**

Cílem práce je prezentace vývoje databáze pomocí třívrstvé architektury, popis a srovnání softwaru pro tvorbu jednotlivých modelů včetně daných výstupů. Na příkladech databáze rezervačního systému ubytování je předvedena tvorba databáze až po výstupy pomocí SQL dotazů. Opačný postup tvorby databáze pro rezervační systém je demonstrován na reverzním inženýringu.

**Klíčová slova:** Třívrstvá architektura, databáze, Oracle, UML, SQL

**Annotation**

The aim of this thesis is to present the development of the database using a Three-tier architecture, description and comparison of the software for the creation of individual models including the given outputs. Using examples of an accommodation reservation system database, the development of the database to the outputs using SQL queries is demonstrated. The reverse process of creating a database for a reservation system is demonstrated using reverse engineering.

**Keywords:** Three-tier architecture, database, Oracle, UML, SQL

**Poděkování**

Děkuji touto cestou vedoucí mé maturitní práce Mgr. Janě ten Hagenové, za poskytnuté rady, cenné připomínky a mimořádnou vstřícnost, s jakou mě vedla nejen v průběhu této práce, ale po celou dobu studia na SPŠ a VOŠ Písek.

Obsah

[Úvod 9](#_Toc97210725)

[1. Software 10](#_Toc97210726)

[Aplikační software 10](#_Toc97210727)

[Systémový software 10](#_Toc97210728)

[Programovací software 10](#_Toc97210730)

[2. Společnosti ve světě databází 11](#_Toc97210731)

[Oracle Corporation 11](#_Toc97210732)

[Microsoft Corporation 11](#_Toc97210733)

[Lucid Software Inc. 11](#_Toc97210734)

[3. Databáze 12](#_Toc97210735)

[Uživatelé 12](#_Toc97210736)

[4. Modely databáze 13](#_Toc97210737)

[Flatfile model 13](#_Toc97210738)

[Hierarchický model 13](#_Toc97210739)

[Relační model 14](#_Toc97210740)

[5. Architektury databáze 15](#_Toc97210741)

[Jednovrstvá 15](#_Toc97210742)

[Dvouvrstvá 15](#_Toc97210743)

[Třívrstvá 15](#_Toc97210744)

[6. Návrh databází pro třívrstvou architekturu 16](#_Toc97210745)

[7. Definování business rules 17](#_Toc97210746)

[8. Konceptuální vrstva 18](#_Toc97210747)

[UML 18](#_Toc97210748)

[MS Visio 18](#_Toc97210749)

[a) Sekvenční diagram 19](#_Toc97210750)

[b) Diagram případu užití 20](#_Toc97210751)

[Lucidchart 20](#_Toc97210752)

[a) Diagram tříd 22](#_Toc97210753)

[b) Entitně-relační diagram (UML) 22](#_Toc97210754)

[c) Stavový diagram 23](#_Toc97210755)

[Porovnání Visio vs Lucidchart 24](#_Toc97210756)

[9. Databázová vrstva 25](#_Toc97210757)

[Oracle SQL Developer Data Modeler 25](#_Toc97210758)

[MySQL Workbench 26](#_Toc97210759)

[Porovnání SQL Data Modeler vs MySQL Workbench 27](#_Toc97210760)

[10. Fyzická vrstva 29](#_Toc97210761)

[MS Access 30](#_Toc97210762)

[MySQL 34](#_Toc97210763)

[Porovnání MySQL vs MS Access 37](#_Toc97210764)

[11. Dotazování v SQL 38](#_Toc97210765)

[Příklady možných dotazů 38](#_Toc97210766)

[12. Reverzní inženýrství 41](#_Toc97210767)

[Závěr 42](#_Toc97210768)

# Úvod

Téma práce bylo zvoleno ze zájmu o databáze a způsob jakým probíhá práce s daty, tedy komunikace mezi webovým rozhraním a systémem zprostředkovávajícím SŘBD. Nabyté vědomosti a vytvořená práce budou základem pro rozvoj se studiem na vysoké škole.

Úkolem práce je vybrat řadu softwarů pro tvorbu databáze pomocí třívrstvé architektury (konceptuální, databázový a fyzický model). Dále porovnat a zhodnotit, ať už cenové či uživatelsky přívětivé vazby programu s uživatelem.

Část práce popíše tvorbu databáze na téma rezervačního systému pro ubytování. Na databázi bude předvedeno několik SQL dotazů, užitečných například z byznysového hlediska, a zároveň ukázána prezentace výstupů. Představen bude také postup při reverzním inženýringu databáze a výstup jeho výsledku.

Bude prezentován příklad registrace uživatele na webovém, které spolupracuje s platformou XAMPP (Apache, MySQL). Data budou ukládána přes vytvořený formulář do připravené tabulky „vlastnik“ na MySQL Serveru.

Jedním z úkolů je také tvorba prezentace přechodů a výstupu jednotlivých vrstev, která bude přiložena v přílohách dokumentace a prezentována.

# Software

Software je soubor instrukcí, dat nebo programů, které se používají k ovládání počítačů a provádění konkrétních úkolů. Je opakem hardwaru, který popisuje fyzické aspekty počítače. Software je obecný termín, který se používá pro označení aplikací, skriptů a programů, které jsou spuštěny v zařízení. Lze jej považovat za proměnlivou část počítače, zatímco hardware je jeho neměnnou součástí. [1]

## Aplikační software

Balík počítačového softwaru, který vykonává určitou funkci pro uživatele nebo v některých případech pro jinou aplikaci. Mezi příklady aplikací patří kancelářské balíky, grafický software, databáze, webové prohlížeče, textové procesory, nástroje pro vývoj softwaru, grafické editory a komunikační platformy. [1]

## Systémový software

Tyto programy jsou určeny ke spouštění aplikačních programů a hardwaru počítače. Kromě toho řídí činnost hardwaru počítače a poskytuje prostředí nebo platformu pro práci všech ostatních typů softwaru. Příkladem jsou operační systémy či firmware. [1]

## Programovací software

Programovací software a programovací nástroje umožňují vývojářům vyvíjet, psát, testovat a ladit další programy. Mezi příklady programovacího softwaru patří assemblery, kompilátory, ladicí programy a interprety. [1]

# Společnosti ve světě databází

## Oracle Corporation

Oracle Corporation je celosvětová korporace se sídlem v Kalifornii, která vyvíjí a nabízí počítačový software pro podnikání. Společnost se proslavila svým softwarem pro databáze (MySQL) a SŘBD. Mimo jiné se také podílela na vývoji operačního systému Solaris, nebo objektově orientovaného programovacího jazyka Java. [2]

## Microsoft Corporation

Microsoft Corporation je vedoucí společnost ve světě vývoje softwarových systémů a aplikací pro osobní počítače. Společnost je známá především kvůli jednomu ze světově nejrozšířenějších operačních systému Windows. Další známé produkty této korporace jsou databázová služba Azure, balíček služeb MS Office (Access, Excel, Word, Visio…), nebo internetovým vyhledávačem Microsoft Edge. [3]

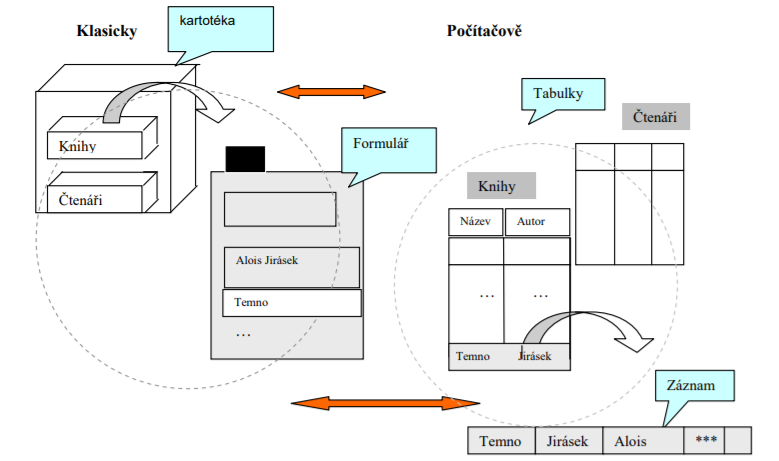
## Lucid Software Inc.

Lucidchart je chráněná platforma založená na webovém rozhraní, která umožňuje uživatelům spolupracovat na znázorňování, revizi a sdílení schémat a diagramů (UML).

# Databáze

Databáze je uspořádaný soubor strukturovaných informací nebo dat, obvykle uložených elektronicky v počítačovém systému. Databáze je obvykle řízena systémem pro správu databází (DBMS). Data a DBMS spolu s aplikacemi, které jsou s nimi spojeny, se společně označují jako databázový systém, často zkráceně jen databáze. [9]

Pro laické porovnání je uváděn příklad z reálného života, kdy lze vidět obrovské zjednodušení při vypůjčování knih. Viz Obrázek č. 1 Porovnání možností databáze [4].



Obrázek č. 1 Porovnání možností databáze [4]

## Uživatelé

V DB (databázových) systémech se data neodstraňují, jsou subjektově orientována, časově rozlišitelná a integrovaná, je nutné určit uživatelská oprávnění, tedy privilegia. Proto se entity, jež přistupují do daného SW s databázovým systémem, dělí do skupin:

* administrátor
* aplikační programátor
* uživatel

# Modely databází

Soubor prostředků a konceptů, popisujících data (sémantiku, strukturu, vztahy,integritní omezení) na určité úrovni abstrakce. Obvykle rozlišujeme tři komponenty – strukturální, operační a specifikace integritních omezení. Konkrétní datový model souvisí s úrovní abstrakce, s pohledem na data v procesu vývoje aplikace. [4]

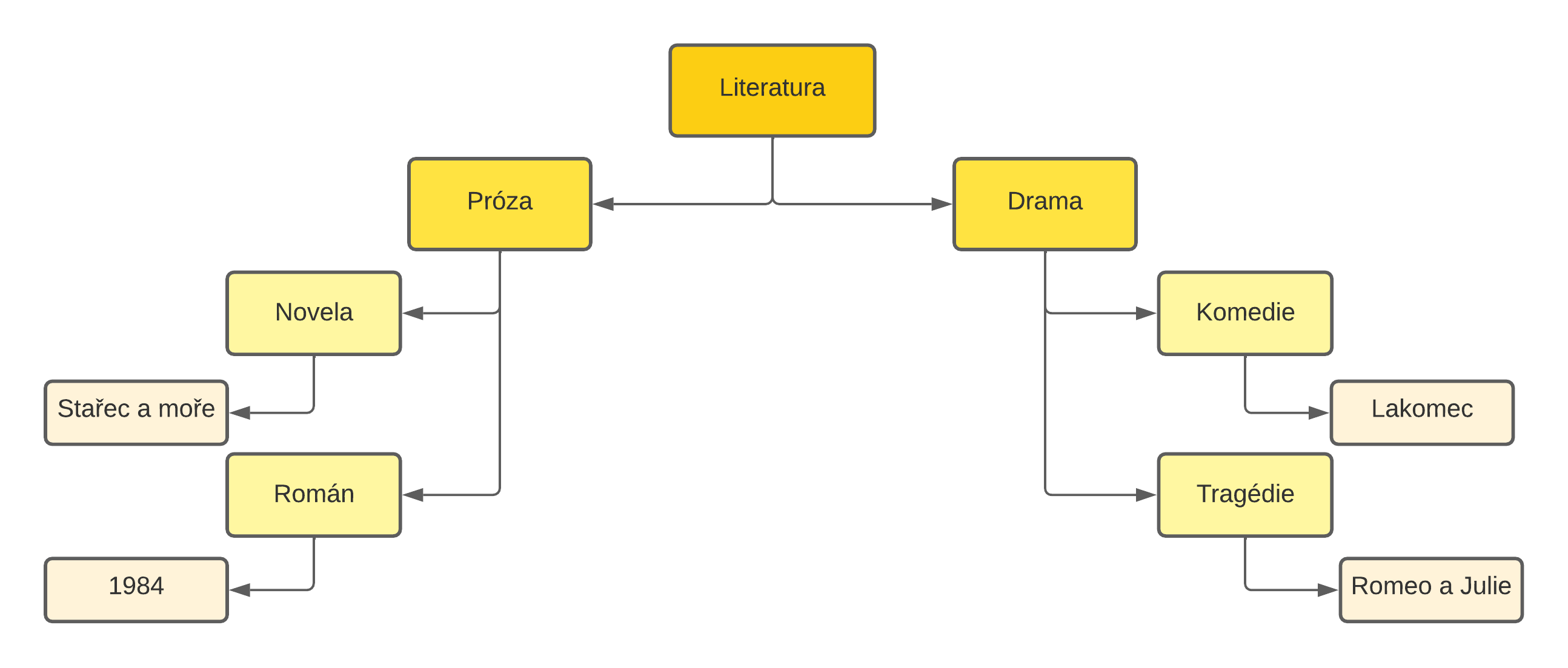
## Flat-file model

Prostý databázový soubor je jednoduchá databáze uložená v textovém souboru ve formě prostého textu.

Příkladem takové databáze je dnes často využívaný formát ukládání dat v MS Excel, či jemu aplikace podobné. Takto tvořené DB jsou sice velmi jednoduché a přívětivé pro běžného uživatele, naopak se zde ale vyskytuje mnoho nevýhod. Mezi ně patří integrita, nekonzistence, redundance a minimální bezpečnost dat.

## Hierarchický model

Data jsou reprezentována ve stromové struktuře. V tomto modelu jsou data ukládána ve formě záznamů, které jsou sbírkou polí. Záznamy jsou propojeny prostřednictvím odkazů a typ záznamu říká, které pole obsahuje záznam. Každé pole může obsahovat pouze jednu hodnotu. Model musí mít pro každý podřízený uzel pouze jeden nadřazený uzel, ale nadřazené uzly mohou mít více než jeden podřízený. Více rodičů není povoleno. Viz Obrázek č. 2 Hierarchický model. Celý model viz Příloha č. 1 Hierarchický model. [5]



Obrázek č. 2 Hierarchický model

## Relační model

Záznamy stejného entitního typu jsou logicky organizovány ve formě dvoudimenzionálních tabulek, vztah mezi záznamy je definován hodnotami vazebních atributů (cizích klíčů), obecně v samostatných tabulkách. Relační databázi tvoří jedna, nebo několik tabulek. Tabulka uchovává informace o skupině podobných objektů reálného světa, např. o knihách. Informace o jednom objektu je na jednom řádku tabulky. Pořadí řádků v tabulce není důležité, tzn. nenese žádnou informaci. Sloupec tabulky uchovává informace o jedné nestrukturované vlastnosti objektu. [4]

# Architektury databází

Postup návrhu pomáhá při návrhu, vývoji, implementaci a údržbě systému správy databází. Architektura DBMS umožňuje rozdělit databázový systém na jednotlivé komponenty, které lze nezávisle upravovat, měnit, nahrazovat a měnit. [6]

Rozděluje se na základní tři varianty dle přístupu k datům jednovrstvá, dvouvrstvá a třívrstvá.

## Jednovrstvá

Dle SŘBD je nejjednodušší architekturou databáze, ve které se klient, server a databáze nacházejí na stejném počítači. Jednoduchým příkladem jednoúrovňové architektury je do systému nainstalována databáze a do které se přistupuje za účelem procvičování dotazů SQL. Taková architektura se však v produkčním provozu používá jen zřídka. [6]

## Dvouvrstvá

Dvouúrovňová architektura v SŘBD je architektura databáze, kde prezentační vrstva běží na klientovi (PC, mobil, tablet atd.) a data jsou uložena na serveru. Dvouúrovňová architektura poskytuje SŘBD vyšší bezpečnost, protože není vystavena přímému kontaktu s koncovým uživatelem. Poskytuje také přímou a rychlejší komunikaci. [6]

## Třívrstvá

Tříúrovňová architektura v SŘBD je nejoblíbenější architektura klient-server v SŘBD, ve které se vývoj a údržba funkčních procesů, logiky, přístupu k datům, ukládání dat a uživatelského rozhraní provádí nezávisle jako samostatné moduly. Tříúrovňová architektura obsahuje prezentační vrstvu, aplikační vrstvu a databázový server. Návrh třívrstvé architektury databáze je rozšířením dvouvrstvé architektury klient‑server. [6]

Třívrstvá architektura má následující vrstvy:

* Prezentační – jen uživatelé, zajišťuje vstup požadavků a prezentaci výsledků
* Aplikační/Funkční – middleware, operuje s daty a požadavky (aplikační server)
* Datová vrstva/Databázová – SŘBD, práce s daty (agregace, integrita, audit…)

# Návrh databáze pro třívrstvou architekturu

Návrh vybraného modelu databáze má být sekvenční. Pro správný chod a naplnění požadavků tvorby je nutné dodržovat strukturu návrhu.

Důležitý je styk se zákazníkem, při kterém se domluví pomocí metodiky SLA (Service level agreement) přesné parametry databáze (funkce dle zákazníkovi představy, zaznamenání vztahů ve podniku). Při zasedání se smluví úlohy a oprávnění uživatelů. Tato fáze se nazývá definování Business Rules.

Dalším důležitým článkem je tvorba konceptuálního modelu databáze, při kterém je pro zákazníka připravena řada různých UML diagramů (Diagram případu užití, Sekvenční diagram, Diagram aktivit, Diagram komponent a další…). Ty slouží pro svou jednoznačnost a srozumitelnost k domluvu s koncovým zákazníkem. V této práci jsou vybrány dva softwary (Lucidchart a Visio), které jsou porovnány z mnoha pohledů. Při tomto procesu se doupraví návrh databáze dle požadavků a určí se akceptační testování, které zahrnuje body, při nichž se databáze považuje za předatelnou zákazníkovi.

Při tvorbě logického (databázového) modelu, vzniká prvotní relační schéma databáze. Vizualizace výstupů databáze je připravena v softwarech Oracle SQL Data Modeler a MySQL Workbench. Tyto softwary pro tvorbu druhé vrstvy třívrstvé architektury jsou představeny a porovnány včetně výstupů.

Modelování databáze takzvaného fyzického (relační) modelu se vytváří pomocí SQL kódu přímo tabulky včetně daných atributů provázaných relacemi. Zde bude představen open-source software se SŘBD, tedy MySQL Server. Příklad v něm bude předveden přímo v CLI (Command Line Interface).

Aplikačního server zajišťuje základní služby pro provoz databáze a softwaru. Příkladem je vyhledávání a ukládání, bezpečnost – autentizace, uživatelská oprávnění, přístup k datům, zprostředkování operace mezi front-end a back-end. Důležité je zmínit samotné programování a design webového rozhraní, pro které databáze slouží. Tyto kroky jsou předvedeny na localhost pomocí Apache a MySQL databáze.

# Definování business rules

Řídí každodenní rozhodování v podnicích tím, že popisují vztahy mezi objekty, jako jsou jména zákazníků a jim odpovídající objednávky. Bez nich může být aktualizace procesů náročnější a zdlouhavější a dokumenty mohou být vystaveny většímu množství lidských chyb a nekonzistencí.

Důležité je si navrhnout jednotlivé objekty databáze, jaké role budou hrát a jaký vliv budou mít na konečný koncept. Proto byly definovány tyto entity:

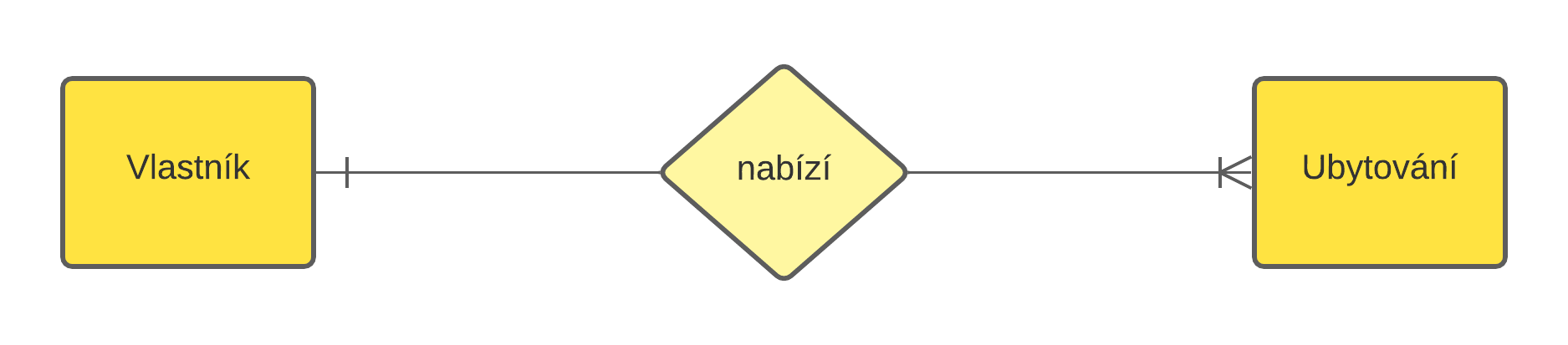
* Vlastník
* Ubytování
* Typ domu
* Hodnocení
* Město
* Země
* Pokoj
* Rezervace
* Host
* Stav
* Kupon
* Měna

Poté následuje fáze definice pravidel databáze. Toto jsou příklady business rules při tvorbě vlastní databáze:

* každý vlastník má svůj účet
* host přistupuje k ubytování bez účtu, vždy vyplní své osobní údaje
* pokoj může být rezervován v jeden moment pouze jedním hostem
* vlastník může nabízet více než jedno ubytování
* v ubytování se může vyskytovat více než jeden pokoj k rezervaci
* hodnocení hotelu upravuje admin DB dle získané zpětné vazby
* měnou může být zvolena CZK/EUR/USD
* kupóny zpřístupňuje admin DB
* …

# Konceptuální vrstva

Na příkladu vztahu mezi dvěmi entitami (Vlatník a Ubytování) je vztah difinován slovesem (kosočtverec), kardinalitou (symbol na konci relace) a optionality (vzhled relace). V tomto případě diagram vyjadřuje následující: Jeden vlastník nabízí jedno nebo více ubytování. S tím, že optionality vysvětluje, že výskyt údajů vlastník i ubytovnání je povinný, tedy musí být vložena data.



Obrázek č. 3 ER diagram – Konceptuální model

Obdobně byly vytvořeny vztahy mezi ostatními objekty databáze, která slouží systému rezervace ubytování. Viz Obrázek č. 8 Entitně–relační diagram.

V této práci jsou pro popis konceptuální vrstvy užity a porovnány softwary Lucidchart a MS Visio, ve kterých jsou vytvořeny diagramy sloužící k základnímu pochopení činností a chování databáze.

## UML

UML je pevně definovaný a srozumitelný jazyk. Existuje množství diagramů, které se dělí do tří skupin:

* Diagramy struktury (př. Diagram tříd)
* Diagramy chování (př. Diagram případu užití, Stavový diagram)
* Diagramy interakce (př. Sekvenční diagram)

## MS Visio

Software od společnosti Microsoft Inc. Visio je součástí balíčku Office 365, tedy minimálně jeho nižší, pouze webová varianta Visio Standard. Existuje i vyšší konfigurace Visio Professional, která uživateli nabídne desktopovou aplikaci, ve které je modelování možné i off-line. Mobilní aplikace poskytuje pouze režim pro čtení a je propojena s účtem MS. [15]

Visio Standard bohužel není možné používat bezplatně ani v omezené míře a nabízí za svoji cenu jen zlomek modelovacích možností, proto je v této práci téměř nepoužitelný. Naopak vyšší verze Visio Professional obsahuje nespočetné množství možností. Bohužel je pro běžného uživatele nepřívětivá jak svým vzhledem, tak rozmístěním ovládacích prvků.

Příkladem vizualizace jsou dva diagramy popisující interakce a chování databáze – Sekvenční diagram a Diagram případu užití.

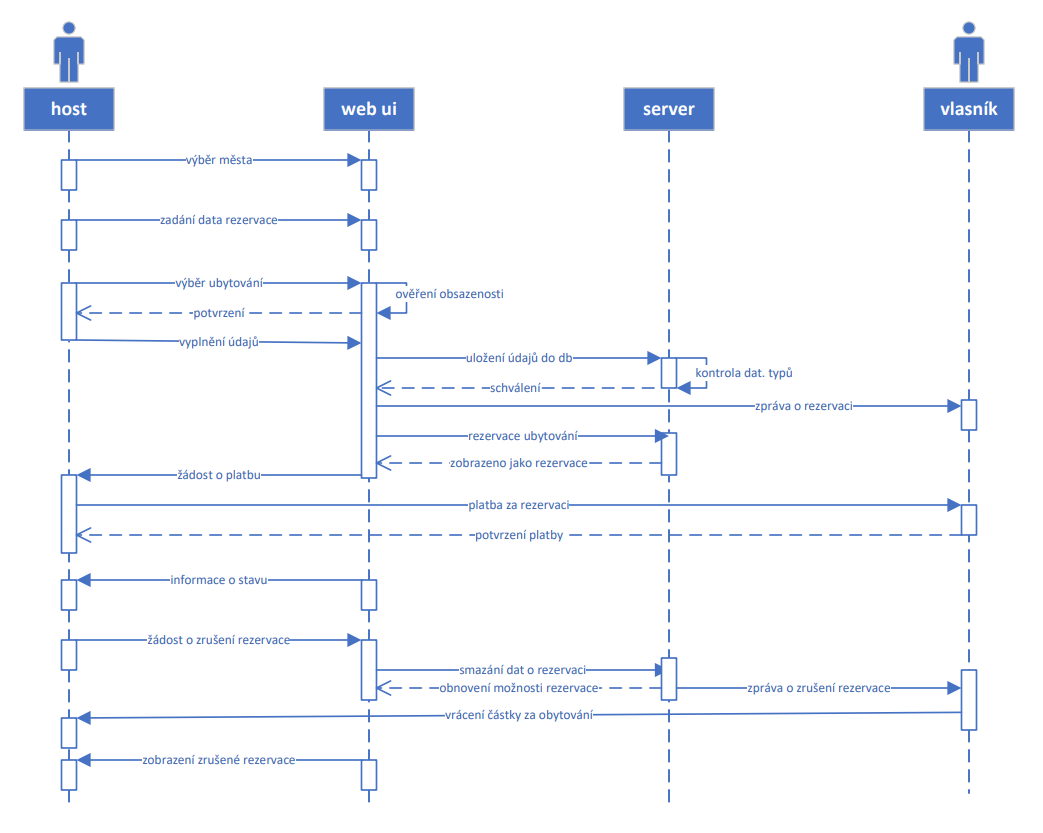
## Sekvenční diagram

Tento diagram znázorňuje komunikaci mezi objekty v daném čase. Je používán ve vývojové fázi specifikace požadavků a návrhů implementace. [13]

Na příkladu Sekvenčního diagramu je předvedena interakce mezi čtyřmi objekty (host, webový rozhraní, aplikační server a vlastník). Celý průběh jednotlivých transakcí je rozvržen časově po příchodu hosta na rezervační web. Nejdříve host definuje základní parametry pro ubytování (kde, kdy, cena, počet lůžek…), pomocí nichž se filtruje zobrazené ubytování.

Dále je znázorněno vyplnění dat o ubytované osobě, které se uloží do databáze. Následně, aby byla rezervace považována za schválenou, se ověří platba za rezervaci, která je systémem přeposlána na účet vlastníka. Také je mu odeslána informativní

zpráva o rezervaci. Popsán je také postup při zrušení rezervace ze strany hosta. Viz Obr. č. 4 Sekvenční diagram – Visio. Celý diagram viz Příloha č. 2 Sekvenční diagram – Visio.



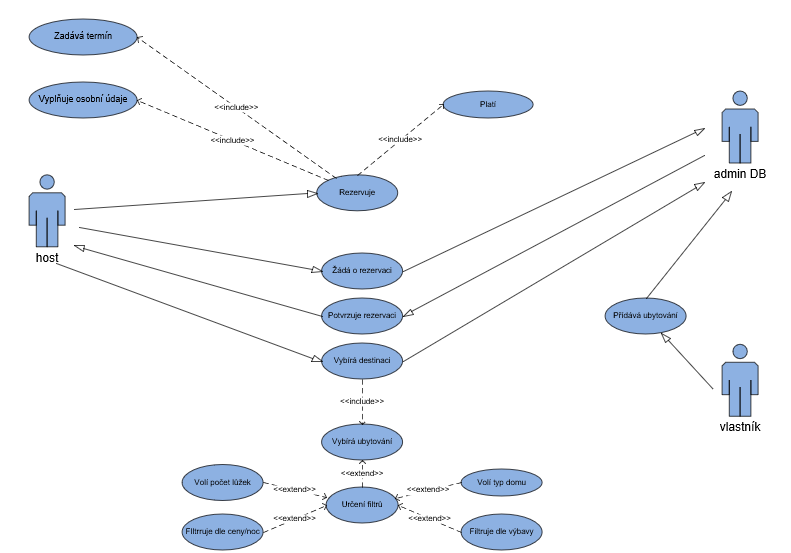
Obrázek č. 4 Sekvenční diagram – Visio

## Diagram případu užití

Diagram případu užití diagramem chování poskytující funkční náhled na systém, tedy kdo s kým a jak pracuje. Užívá se pro realizaci DFD (Data Flow diagramy), nebo ve fázi specifikace požadavků. [13]

Druhý výstup ze softwaru MS Visio popisuje chování entit (host, admin a vlastník), které přistupují k systému. V diagramu je znázorněno, že host při vybírání rezervace má možnost aplikovat různé filtry (vztah extend), při kterých si volí pokoj dle výbavy, jaký typ domu upřednostňuje, cenu pokoje za noc atd.

Z vyhovujících pokojů rezervuje vybraný a doplní potřebné osobní informace (vztah include) a nakonec zaplatí. Admin databáze potvrzuje rezervaci a tím zprostředkuje požadované informace vlastníkovi ubytování. Vlastník má v systému úlohu zpřístupňování nového ubytování hostům, které validuje admin databáze. Viz Obrázek č. 5 Diagram případu užití – Visio. Celý diagram viz Příloha č. 3 Diagram případu užití – Visio.



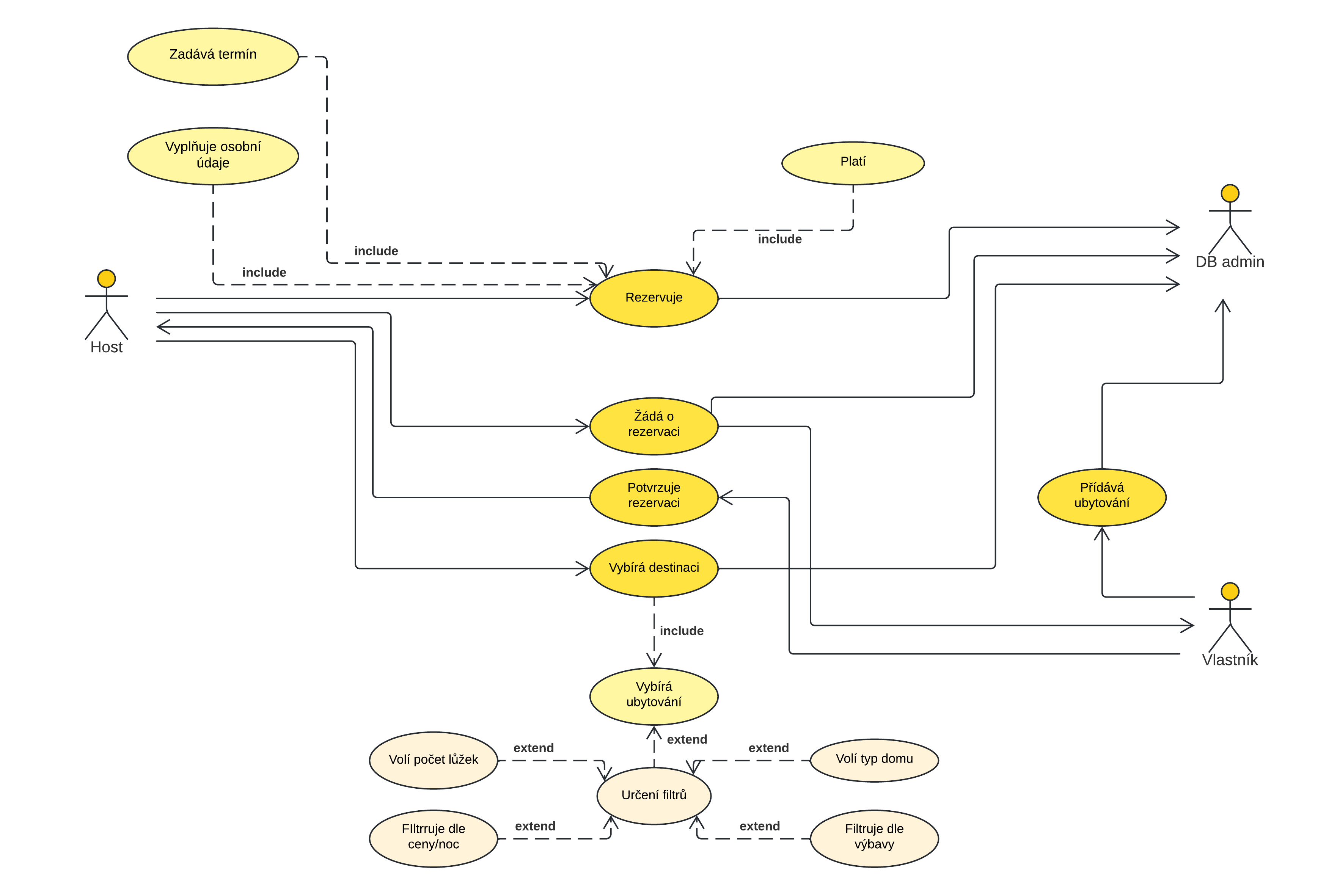
Obrázek č. 5 Diagram případu užití – Visio

## Lucidchart

Software pro tvorbu UML diagramů Lucidchart má pouze webové rozhraní, je tedy nutné připojení k internetu. Cenově se rozděluje na individuální (9,75 $/osoba) a týmový balíček pro minimálně tři pracovníky (6,67 $/osoba). Ten je po přepočtu na osobu mnohem výhodnější, a to nejen cenou, ale i možností komunikace s ostatními členy a větším uložištěm. Dále také nabízí mobilní aplikaci, která je synchronizována s účtem. [14]

Možností je využívat v omezeném režimu software plně zdarma za podmínek tvorby maximálně tří diagramů. Pro studenty je nabízen plán Education, který rozšiřuje možnosti modelování. Samotný software je uživatelky velmi přívětivý, modelování a orientace je jednoduchá. V placené verzi je možnost exportovat diagramy například do zmiňovaného MS Visio, kde se dají posléze upravovat.

Pro porovnání použitých modelovacích softwarů je diagram případu užití vytvořen v obdobné verzi jako u MS Visio. Viz Obrázek č. 6 Diagram případu užití – Lucidchart. Celý diagram viz Příloha č. 4 Diagram případu užití – Lucidchart.



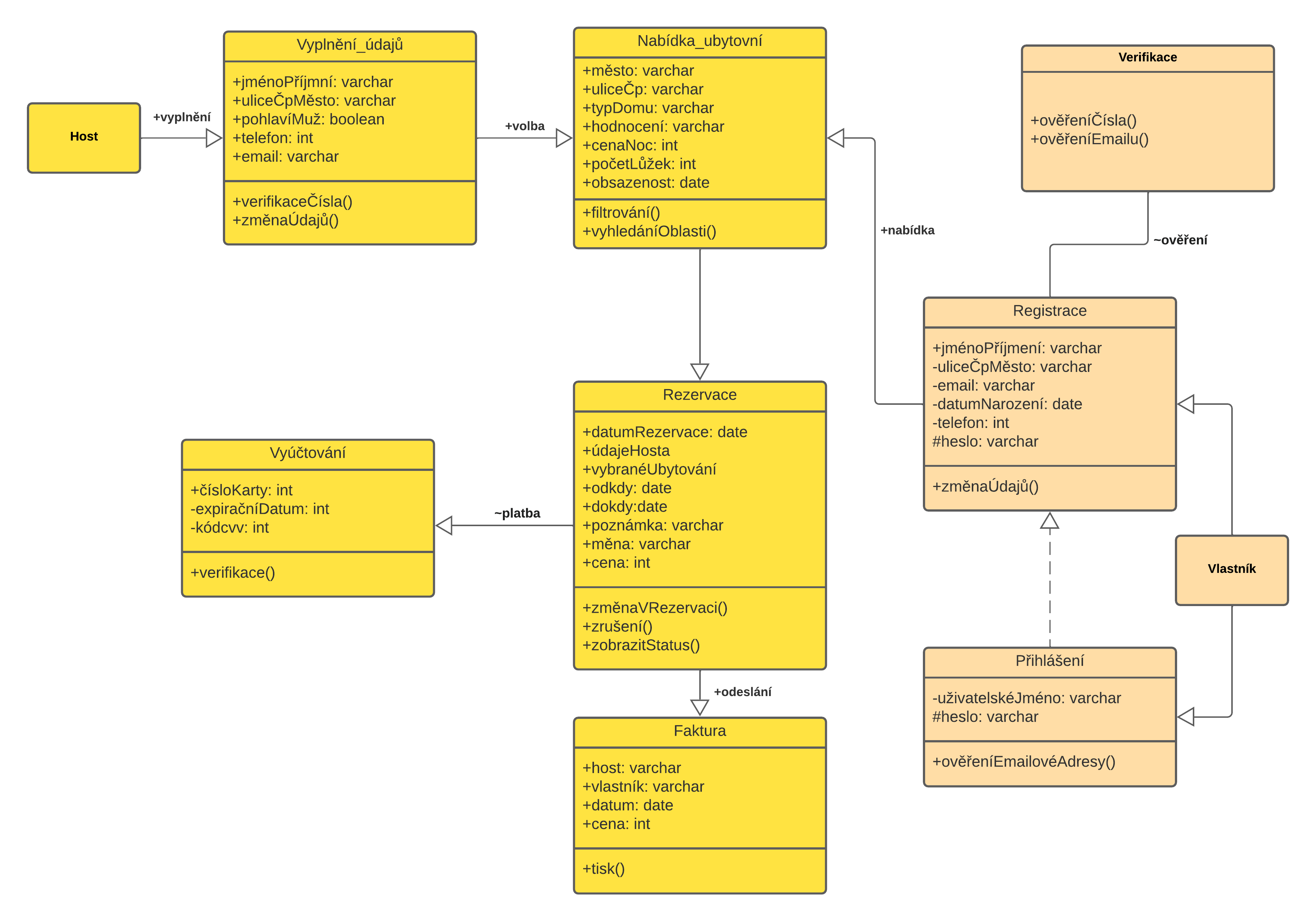
Obrázek č. 6 Diagram případu užití – Lucidchart

Ilustrace práce se softwarem Lucidchart je prezentována následujícími diagramy popisující interakce a chování databáze:

## Diagram tříd

Diagram tříd poskytuje logický náhled na systém. Znázorňuje datové struktury, operace u objektů a také relace mezi nimi. Je využíván pro tvorbu entitně-relačních diagramů a při návrhu implementace. [13]

V Diagramu tříd pro modelování struktury zvolené databáze vystupují aktéři (host, vlastník), kteří vykonávají popsané činnosti. Vlastník po přihlášení do svého účtu, pokud účet nemá registruje se. Poté se verifikují údaje (email a telefonní číslo). Znaky v tabulkách (před atributy) znázorňují viditelnost údajů vložených do určitých polí. Například veřejný údaj (+), privátní data (-) a šifrované (#). Po shledání nabídky s poptávkou dojde k vytvoření konečné sestavy – faktury, která je tisknutelná. Viz Obrázek č. 7 Diagram tříd – Lucidchart. Celý diagram viz Příloha č. 5 Diagram tříd – Lucidchart.

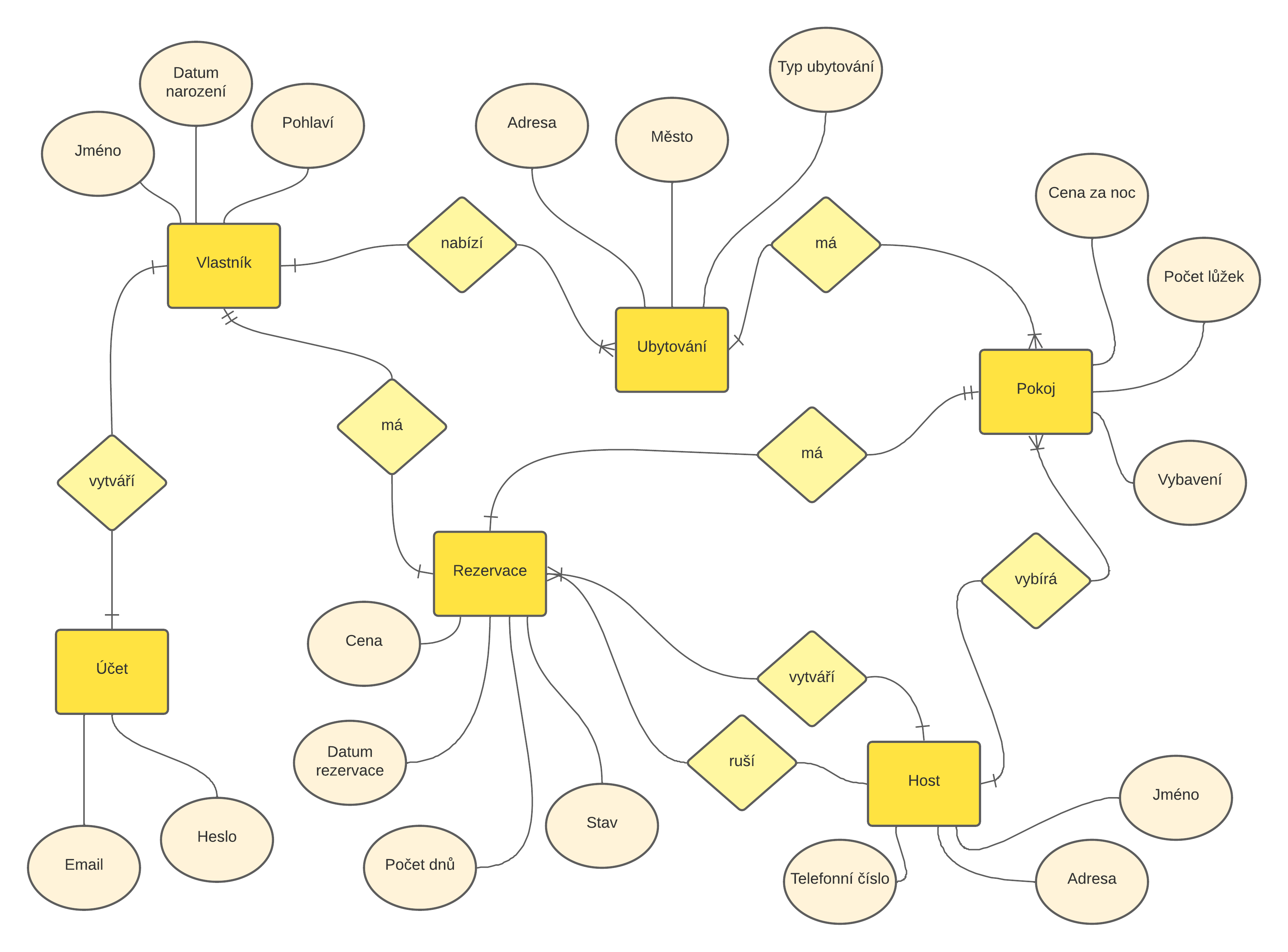


Obrázek č. 7 Diagram tříd – Lucidchart

## Entitně–relační diagram (UML)

ERD znázorňuje objekty včetně jejich atributů a vztahy mezi nimi. Použití nalezne pro abstraktní a konceptuální modelování dat, které vývojáři pomáhá při modelování shora dolů, nebo domluvě se zákazníkem. Obrázek č. 3 ER diagram – Konceptuální model. [13]

Tento E–R diagram představuje rozšířenou variantu, která je popsána na začátku této kapitoly. Viz Obrázek č. 3 ER diagram – Konceptuální model. Navíc se zde objevují atributy entit, které ji definují. Tedy například entita ubytování je popsána těmito atributy: ID, vlastnik\_id, ulice, cp, mesto, typ\_ubytovani, hodnoceni\_id. Viz Obrázek č. 8 Entitně–relační diagram – Lucidchart. Celý diagram viz příloha č. 6 Entitně–relační diagram – Lucidchart.



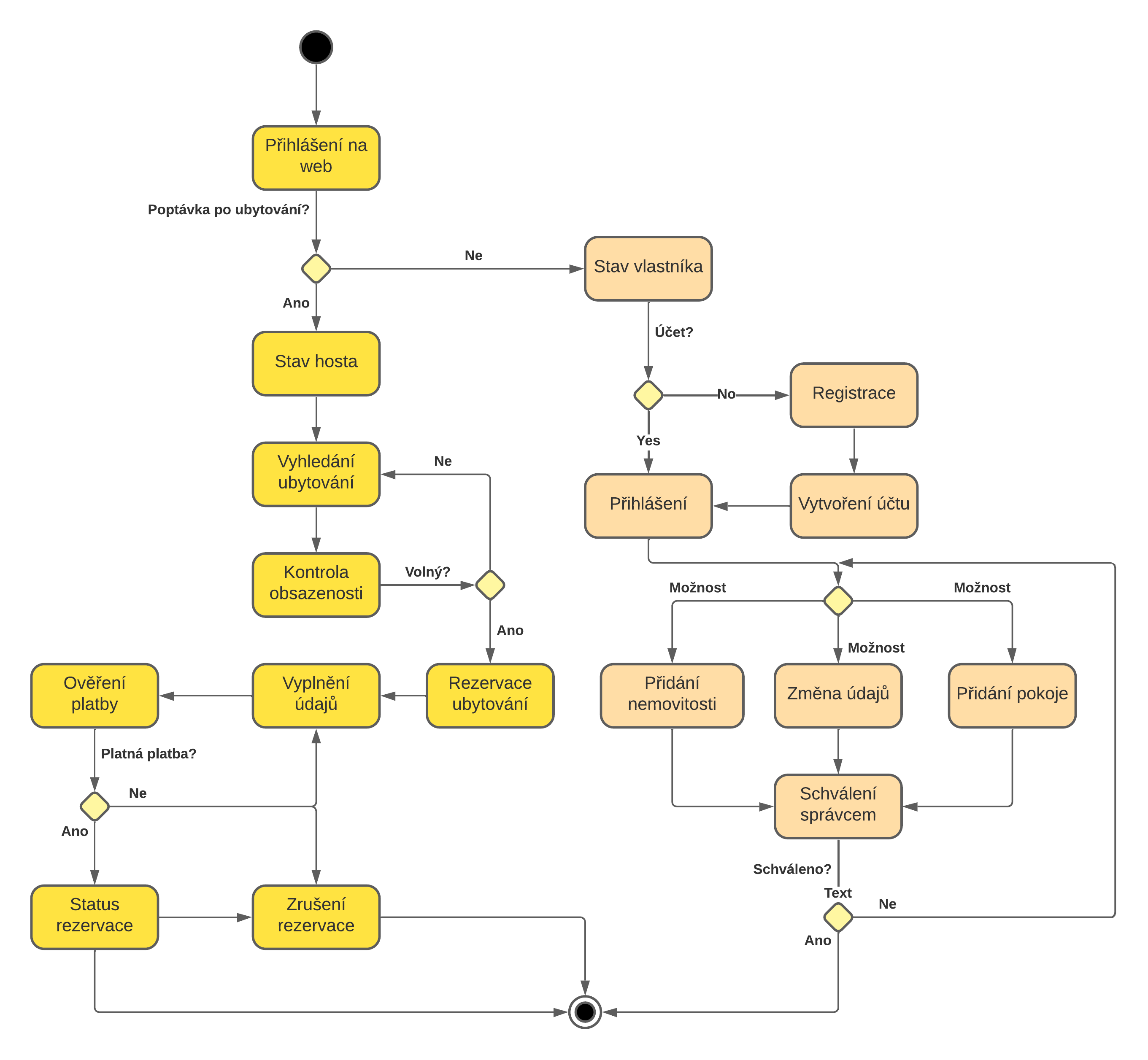
Obrázek č. 8 Entitně–relační diagram – Lucidchart

## Stavový diagram

Stavový diagram zobrazuje životní cyklus objektů a stavy do kterých se během života dostávají. Hlavní využití nalezneme při návrhu implementace. [13]

Diagram popisuje chování webového rozhraní a serveru s příchodem osoby, která se ihned identifikuje tím, jestli má zájem o ubytování. Na začátku vznikne rozdělení na entity hosta a vlastníka. Každý host již má pomocí tohoto diagramu definovanou cestu, kterou se bude ubírat celý životní cyklus jeho rezervace.

Naopak vlastník se nejdříve přihlásí, případně vytvoří nový účet. Poté provádí operace, mezi které patří přidání ubytování/pokoje nebo změna údajů. Tuto transakci, aby byl cyklus ukončen a údaje uloženy do databáze, musí schválit admin databáze. Viz Obrázek č. 9 Stavový diagram – Lucidchart. Celý diagram viz příloha č. 7 Stavový diagram – Lucidchart.



Obrázek č. 9 Stavový diagram – Lucidchart

## Porovnání Visio vs Lucidchart

Ať už vzhledem k ceně, možnosti práce v reálném čase, či přístupu do softwaru zdarma vyhrává Lucidchart. Upoutává také svým přívětivým uživatelským rozhraním. K vyzkoušení práce je možnost týdenního free trialu prémiového rozhraní. Nevýhodou přetrvává nemožnost vlastnit licenci, tedy jednorázový nákup softwaru. Přechod mezi vybranými softwary je velmi autentický. Pro srovnání základních informací při porovnávání byla vytvořena tabulka. Viz Tabulka č. 1 Srovnání MS Visio vs Lucidchart.

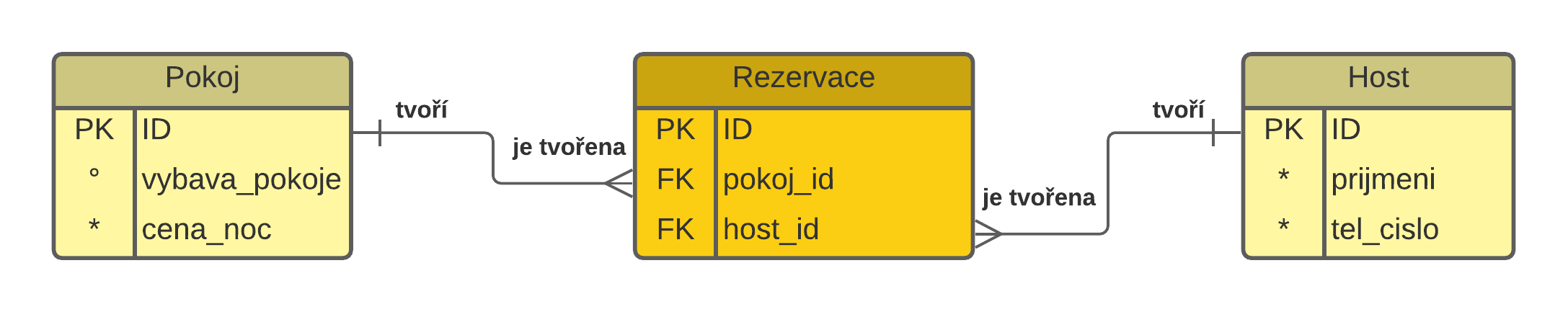
Tabulka č. 1 Srovnání MS Visio vs Lucidchart

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Měsíčně | Ročně | Licence | Free trial | Zdarma |
| Visio Standard | 5 $ | 60 $ | 280 $ | 30 dní | Ne |
| Visio Professional | 15 $ | 180 $ | 530 $ | 30 dní | Ne |
| Lucidchart | 8 $ | 95 $ | - | 7 dní | Ano |

# Databázová vrstva

Při modelování databázové vrstvy se ERD z konceptuální vrstvy změní do formy tabulek. Relace je zobrazena vztahem primárního klíče (ID entity) a cizího klíče (odkaz na ID v jiné tabulce. Hlavní tabulka Rezervace, je tvořena unikátním ID, které je pro každou z nich jiné. Dále se zde nachází dva sekundární klíče, pomocí nichž lze zjistit, jaký host je v rezervaci a následně všechny jeho údaje.

Kdyby došlo k změně emailu, není nutné údaj nahrazovat všude, stačí pouze v té tabulce. V kolonce pro primární klíče se objevují symboly u atributů, a to, jestli je nutné je vyplňovat, či nikoli. To je definováno dle požadavků na databázi. Viz Obrázek č. 10 ER diagram – Databázový model.



Obrázek č. 10 ER diagram – Databázový model

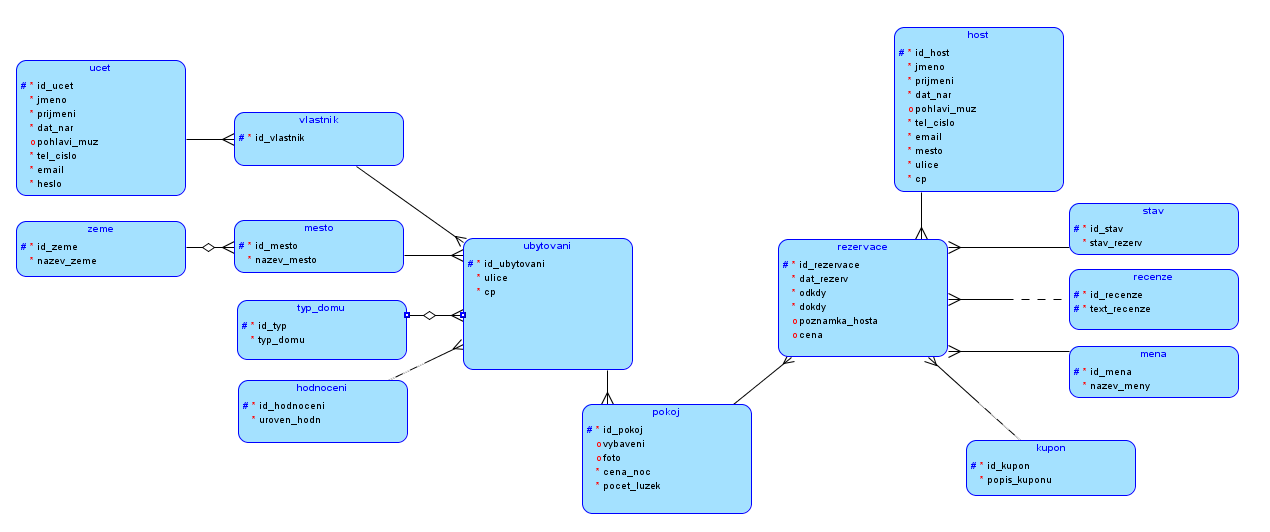
V této práci jsou pro popis Databázové vrstvy užity a porovnány softwary Oracle SQL Developer Data Modeler a MySQL Workbench, ve kterých jsou vytvořeny ER diagramy sloužící k pochopení vztahů a kardinality mezi jednotlivými entitami vytvořené databáze.

## Oracle SQL Developer Data Modeler

Oracle SQL Developer Data Modeler je grafický modelovací nástroj databázové vrstvy při návrhu třívrtvé architektury. Možností je vytvářet, upravovat či procházet dokonce i vícerozměrové modely, kterým se postupně přidělují datové typy pro správnou integritu. Mezi tabulkami jsou tvořeny relace, které mimo jiné zobrazují kardinalitu či volitelnost. [7]

Software je bezplatný a má podporu všech základních operačních systémú. Poskytuje možnost forward a reverse inženýringu, podporuje vývoj pomocí integrované konzoly v SQL kódu. Pomocí forward inženýringu je možné ze správně vytvořeného modelu databázové vrstvy zformovat vrstvu fyzickou, která je také reprezentována ER diagramem.

Pro prezentaci výstupu vizualizace z tohoto software je zhotoven entitně-relační diagram, který popisuje vztahy mezi objekty databáze. Atributy se dělí na povinné (\*) a dobrovolné (°). U relací je povinnost vyznačena přerušovanou čarou, jako například u vztahu mezi rezervací a recenzí (do entity rezervace vložena recenze později/vůbec). Viz Obrázek č. 11 ERD Databázová vrstva – SQL Data Modeler. Celý diagram viz Příloha č. 8 ERD Databázová vrstva – SQL Data Modeler.



Obrázek č. 11 ERD Databázová vrstva – SQL Data Modeler

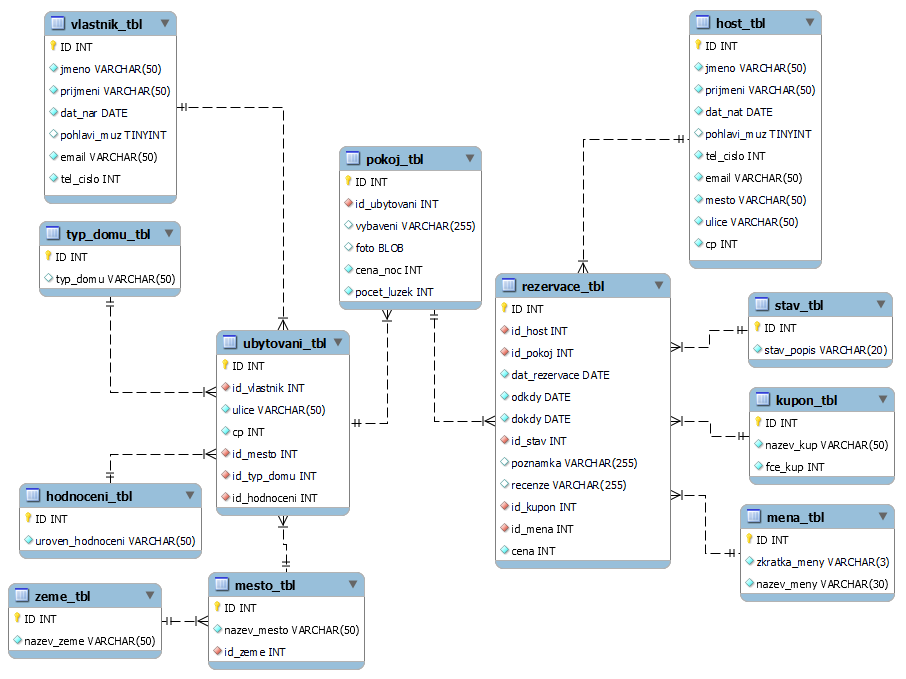
## MySQL Workbench

Dalším software od společnosti Oracle je MySQL Workbench. Podporu má u všech základních operačních systémů. Software je dostupný ve třech vydáních. Community Edition je open-source, zdarma a také je světově nejvíce rozšířena.

Standard Edition je placená verze, která dává možnost podporovat vysoce výkonné a škálovatelné aplikace pro zpracování OLTP (Online Transaction Processing). Cenová relace se opírá o množství používaných serverů, tedy 1–4 (2 000 $/rok) a 5 a více (4 000 $/rok). [8]

Enterprise Edition, která obsahuje řadu pokročilých funkcí pro dosažení nejvyšší škálovatelnosti, zabezpečení a spolehlivosti. Platební plán se také dělí dle využití serverů, tedy 1–4 (5 000 $/rok) a 5 a více (10 000 $/rok). [8]

V softwaru se v databázové vrstvě zobrazují datové typy daných atributů, které je typické spíše již pro vrstvu fyzickou. Volitelnost vkládání dat do atributů je zde reprezentována modrým (nutný – NOT NULL), nebo bílým kosočtvercem (dobrovolný – NULL). Pro porovnání vizualizace s konkurenčním softwarem Oracle byla vytvořen obdobný ERD databáze rezervačního systému. Viz Obrázek č. 12 ERD Databázová vrstva – MySQL Workbench. Celý diagram viz Příloha č. 9 ERD Databázová vrstva – MySQL Workbench.



Obrázek č. 12 ERD Databázová vrstva – MySQL Workbench

## Porovnání SQL Data Modeler vs MySQL Workbench

Oba softwary patří mezi nejlepší ve svém oboru. Záleží pouze na finální integraci databázového serveru, který je pro podnikové účely používán. Každý ze zmíněných modelovacích programů navazuje na svůj obdobný server. SQL Data Modeler je odnoží Oracle Database, naopak MySQL Workbench se váže k MySQL Serveru.

Díky své přehlednosti, open-source variantě a spoustě dalších funkcí byl pro tuto práci zvolen MySQL Server, kterému je dán prostor v další kapitole. Pro srovnání základních informací při porovnávání balíčků MySQL byla vytvořena tabulka. Viz Tabulka č. 2 Srovnání základních balíčků MySQL.

Tabulka č. 2 Srovnání základních balíčků – MySQL

|  |  |  |  |
| --- | --- | --- | --- |
| **MySQL** | **Community** | **Standard** | **Enterprise** |
| Cena (1-4 Servery/rok) | Zdarma | 2 000 $ | 5 000 $ |
| Databáze (Database) | Ano | Ano | Ano |
| Připojení (Connectors) | Ano | Ano | Ano |
| Migrace dat (Replication) | Ano | Ano | Ano |
| Rozdělování dat (Partitioning) | Ne | Ne | Ano |
| Workbench Server | Ne | Ano | Ano |
| MyISAM (Storage engine) | Ano | Ano | Ano |
| InnoDB (Storage engine) | Ne | Ano | Ano |
| Enterprise Monitor | Ne | Ne | Ano |

# Fyzická vrstva

U fyzické vrstvy se při modelování ERD diagramů přidává množství užitečných informací pro správné uchování dat, které podléhá normování. Na příkladu vytvořené databáze třetí (nový) sloupec slouží k upřesnění datových typů. Viz Obrázek č. 13 ER diagram – Fyzický model.

Mezi datové typy v této práci patří:

* INT – číselná hodnota, převážně u primárních a cizích klíčů
* VARCHAR () – textové pole s možností omezeností
* CURRENCY – pole pro měnu, většinou v USD
* DATE – pole pro vložení data ve formátu YYYY-MM-DD
* TINYINT – malé číslo, nahrazuje funkci BOOLEAN
* BLOB – ukládá jakákoliv binární data, př. foto, soubor .txt. a další

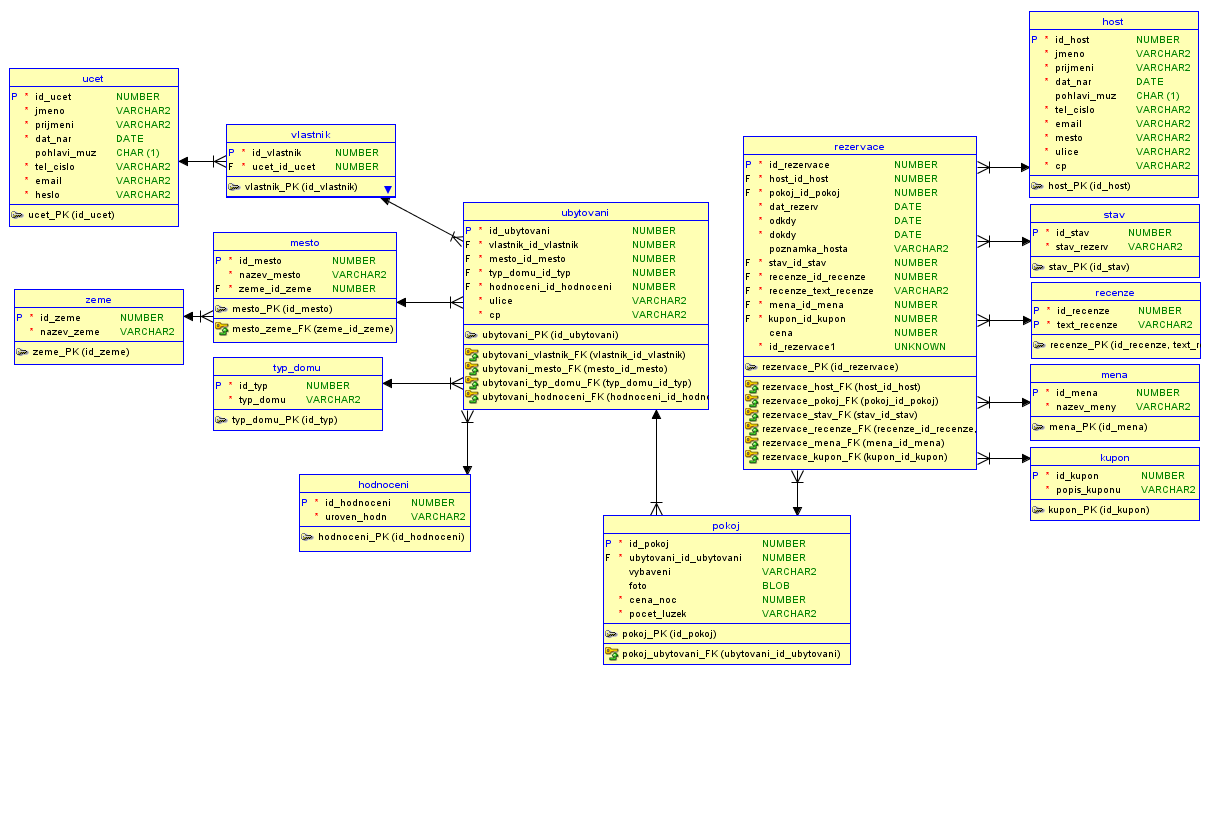
Obsah obrázku stůl

Popis byl vytvořen automaticky

Obrázek č. 13 ER diagram – Fyzický model

Software Oracle SQL Developer Data Modeler je schopen transformovat pomocí forward inženýringu z logického modelu plnohodnotný fyzický model. Z něhož je možné extrahovat datový soubor DDL (Data Definition Language), který připraví na fyzické vrstvě podobu tabulek, kde jsou ukládána a udržována data.

Z logického modelu, viz Obrázek č. 11 ERD Databázová vrstva SQL Data Modeler, byla provedena transformace. Pro prezentaci vizualizace výstupu procesu je vložena ilustrace. Viz Obrázek č. 14 Fyzický model SQL Data Modeler. Celý diagram viz příloha č. 10 Fyzický model SQL Data Modeler.



Obrázek č. 14 Fyzický model SQL Data Modeler

## MS Access

MS Access označuje databázi vyvinutou společností Microsoft, která je součástí softwaru Microsoft Office. Hlavním úkolem aplikace Microsoft Access je tzv. RDBMS – relační systém správy databází (organizace dat ve formě tabulek, které jsou propojeny s obecnými poli). Jedná se o nástroj pro správu informací, který lze použít k ukládání informací pro referenční účely, sestavy a analýzy. [11]

Tento software je možné používat pouze na operačním systému Windows. Výhodou je možnost jeho použití v balíku Office 365, který je dostupný za měsíční částku 5,99 $. Samotná licence je přístupná za cenu 139,99 $ na jeden počítač. Předností je jednoduchá práce s databázemi, použití bez znalosti programovacích jazyků či vizualizace výstupů (ERD, formuláře, sestavy, …). Na rozdíl od programů pro modelování logické vrstvy se liší přístupem k datům. Možné je, jak pracovat s ukládáním záznamů do předem vytvořených tabulek (např. pomocí formulářů), tak je následně prezentovat v sestavách či některé vybírat pomocí dotazů. [11]

Za pomocí již sestavených ERD v logickém modelu databáze byl i zde vytvořen soubor tabulek a vztahů mezi nimi. MS Access používá jiné značení pro kardinalitu vztahů (1: n). Viz Obrázek č. 15 ERD Databázová vrstva – MS Access. Celý diagram viz příloha č. 11 ERD Databázová vrstva – MS Access.

Obsah obrázku text, snímek obrazovky, počítač, interiér

Popis byl vytvořen automaticky

Obrázek č. 15 ERD Databázová vrstva – MS Access

Při tvorbě tabulek byl brán ohled na datové požadavky, které byly určeny v předchozí (databázové) vrstvě. Primární klíče jsou nastaveny na číselnou hodnotu (INT), která není nijak upravitelná. Vždy při vložení nových záznamů se zvětší o hodnotu 1, neboli se inkrementuje. Výjimkou je ID rezervace, kde z byznysového hlediska není vhodné, aby bylo viditelné číslo rezervace. Zde je nastaveno automatické číslo, které zabezpečuje důvěrnost platformy.

Například u entit se jedná o tyto atributy a jim přidělené datové typy:

* ID – INT (automatické/inkrementace)
* Datum narození – DATE
* Jméno – VARCHAR (počet znaků)
* Cena – CURRENCY
* Pohlaví muž – TINYINT (1 –muž/0 –žena)
* Foto pokoje – BLOB

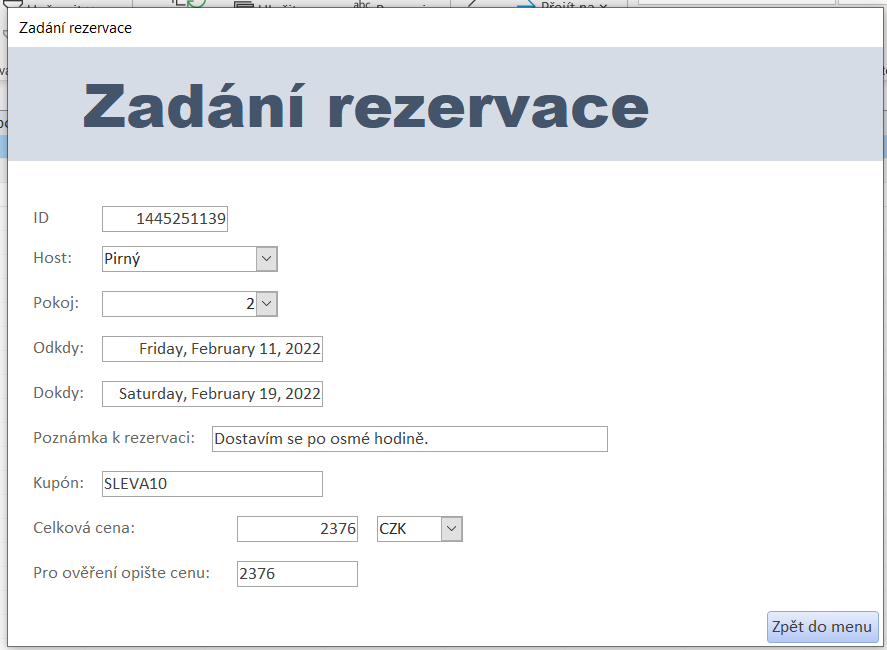
**Formuláře**

Dále byla vytvořena sada formulářů, které reprezentují webové rozhraní. V základním menu se nachází tři sloupce (vlastník, host, admin), přičemž v každém jsou tlačítka, na omezené působení daných osob v databázi. U vlastníka jsou možnosti vyplnit formulář o sobě, čímž je přidán do databáze (tabulka vlastnik). Poté je možnost pomocí dvou tlačítek přidat ubytování a následně jeden či více pokojů. Host má možnost přidat své údaje do databáze a následně vytvořit rezervaci k ubytování.

Všechny tyto transakce jsou schvalovány adminem, který je může odstranit, upravit, či ponechat. Poslední tlačítko admin je přístupné pouze po zadání hesla (admin). Ten zde má oprávnění upravovat různé objekty (např. přidat kupón, upravit stav rezervace, upravit hodnocení ubytování, vytisknout fakturu, …)

Formulář pro tvorbu rezervace obsahuje množství polí. ID je generováno automaticky a je zvoleno náhodné číslo. Pole host, pokoj a měna jsou ve formě vyjíždějícího výběrů instancí, které již byly vloženy do databáze. Datumové pole odkdy a dokdy se zadávají pomocí kalendáře, který se objeví při kliknutí. Je ošetřeno zadání zpětného data, při kterém by počet dní vyšel v záporných hodnotách. Kupón a poznámka k rezervaci jsou textová pole a obě zároveň i dobrovolná, tedy nemusí se zadat žádná hodnota.

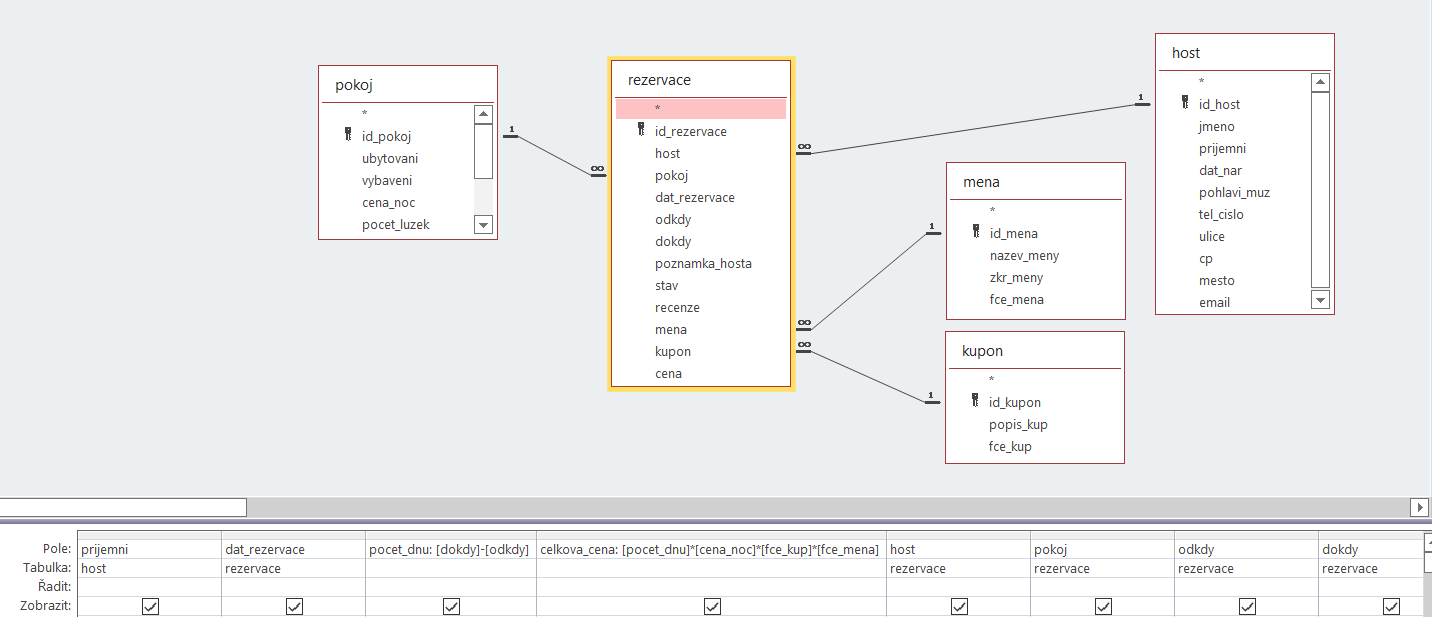
Celková cena se přepočítává automaticky podle vzorce v kapitole Dotazy. Pro správné zapsání rezervace do databáze je nutné správně opsat číselnou hodnotu celkové ceny. Jeli stisknuté tlačítko, které slouží pro navrácení do menu, rezervace se uloží. Viz Obrázek č. 16 Formulář rezervace – MS Access. Celý formulář viz Příloha č. 12 Formulář rezervace – MS Access.

****

Obrázek č. 16 Formulář rezervace – MS Access

**Dotazy**

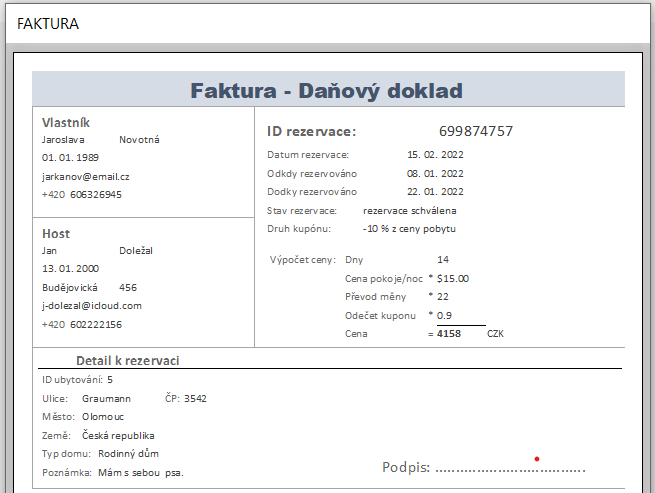
Dotazy jsou v práci používány k doplňujícím informacím, či tvorbě formulářů a sestav. Použity byly na příkladu tvorby celkové ceny objednávky, která se skládá z více samostatných atributů. Nejdříve byl spočten počet dnů (rozdíl data odjezdu a data započetí ubytování), při kterých bude host ubytován ve vybraném zařízení, viz sloupec pocet\_dnu. Dále se tato hodnota vyskytuje v dalším výpočtu pro určení celkové ceny za rezervaci. Ta se skládá z použitého vzorce na určení dnů, dále ceny za noc (tabulka pokoj – cena\_noc) v USD, poté je-li užit kupón, tak z jeho koeficientu (např. SLEVA20 – koeficient 0,8). Není‑li použit je automaticky přidělena hodnota 1. Také je obsažen převod na měnu, v níž je vystavena faktura (v základu – USD). Možnost převést měnu na CZK či EUR. Viz Obrázek č. 17 Dotaz výpočet ceny – MS Access.



Obrázek č. 17 Dotaz výpočet ceny – MS Access

**Sestavy**

Soustavy slouží pro vizuální výstup dat k smysluplnému účelu (soustava faktury). Použito bylo co nejvíce atributů z existujících entit vytvořené databáze. Faktura slouží jako výstup z formuláře rezervace, kterou zprostředkovává admin po přijetí zaplacené objednávky. Sestava je složena ze čtyř částí, tedy údaje o vlastníkovi a hostovi, základní informace o rezervaci a níže poté detailněji rezervace. Viz Obrázek č. 18 Sestava faktura – MS Access. Celá faktura viz Příloha č. 13 Sestava faktura – MS Access.

****

Obrázek č. 18 Sestava faktura – MS Access

## MySQL

MySQL je open source systém pro správu relačních databází (RDBMS) založený na strukturovaném dotazovacím jazyce (SQL). MySQL běží prakticky na všech platformách včetně Linuxu, UNIXu a Windows. Ačkoli jej lze použít v široké škále aplikací, je nejčastěji spojován s webovými aplikacemi a online publikování. [12]

Díky své open-source variantě, tedy užití zdarma, v Community verzi, která má ovšem svá omezení, je software použit v této práci. Je možné využít i placené balíčky, vyžaduje-li to podniková infrastruktura. Viz Tabulka č. 2 Srovnání základních balíčků MySQL. V této práci je software představen v příkazové řádce pomocí softwaru XAMPP, který slouží jako multiplatforma kombinující práci s databází (MySQL) a serverem (Apache), jež spolu komunikují na základě kódu.

Nejdříve byla vytvořena databáze pomocí příkazu (CREATE DATABASE sleepover), do které postupně byly vloženy tabulky představované v logickém modelu třívrstvé architektury. Například tabulka vlastník vznikla s použitím příkazu (CREATE TABLE vlastnik …), zde si lze všimnout zápisu entity bez diakritiky, kde by při převodu kódování znaků mohl vzniknout problém. Data (instance) byly do tabulek vloženy příkazem (INSERT INTO vlastnik (id, …) VALUES (1, …)). Byly dodrženy datové typy z logického návrhu a relace jsou tvořeny cizími klíči. Primární klíč je vždy ID, který má nastavenou hodnotu AUTO\_INCREMENT.

Pro prezentaci ukládání dat do MySQL databáze přes formulář založený na webovém rozhraní, byla vytvořena stránka, která by mohla reprezentovat část možné aplikace. Formulář představuje registraci k novému účtu vlastníka, díky němuž by bylo nabízeno ubytování pro potenciální hosty.

Nejdříve byla vytvořena databáze – userform, a v ní tabulka – vlastnik, do které budou údaje o registrovaném uživateli vkládány. To bylo provedeno viz Obrázek č. 19 Vytvoření tabulky vlastník – MySQL. Celý příklad kódu viz Příloha č. 14 Vytvoření tabulky vlastník – MySQL.

Obsah obrázku text

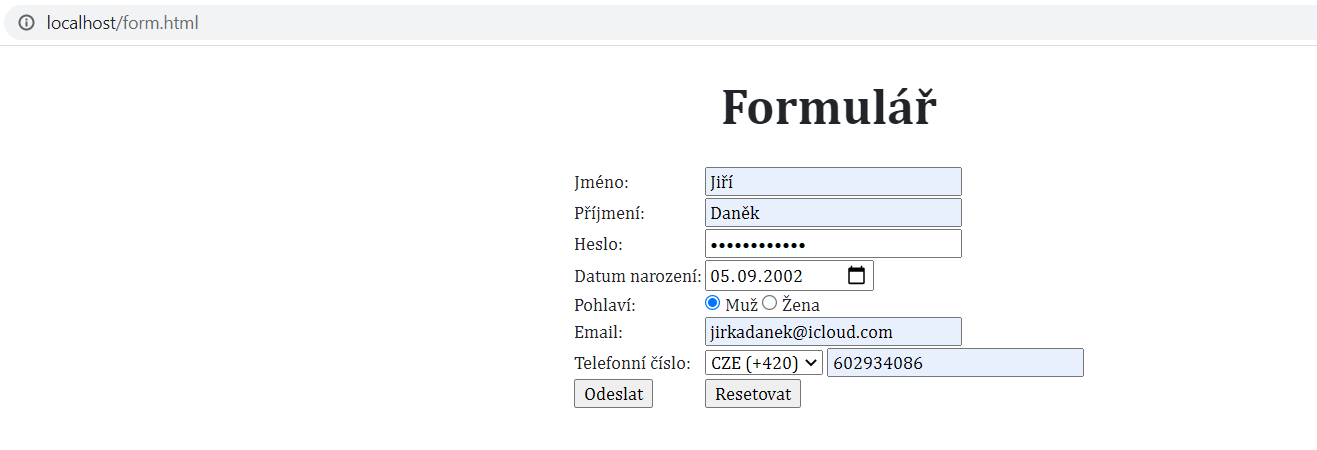
Popis byl vytvořen automaticky

Obrázek č. 19 Vytvoření tabulky vlastník – MySQL

Dále byl v softwaru Visual Studio napsán kód, který reprezentuje formulář, do něhož budou zadávána data. První atribut a zároveň primární klíč (id) je číselný a nezadává se do pole, vždy je k účtu automaticky přiřazen. Atributy jmeno, prijmeni, heslo, email, tel\_cislo jsou zadávány jako textové pole, s tím, že tel\_cislo přijímá pouze číselné hodnoty. Dat\_nar je datový údaj, který je nutné vyplnit ve správném pořadí (DD‑MM‑YYYY), či vybrat pomocí ikony kalendáře. Také je zde omezení, které nepřijímá budoucí datumy. Výběr pohlaví a předvolba je ve formuláři omezena na dvě možnosti, z kterých musí být jedna vybrána.

Při nesplnění nějaké z podmínek, nelze formulář odeslat. U daného pole se objeví upozornění. Nachází se zde dvě tlačítka, jedno pro odeslání dat do databáze a přístupu na účet a druhé slouží k vymazání celého dotazníku. Zároveň při vyplnění údajů (email, tel\_cislo), které se již v tabulce jednou nachází, se zobrazí stránka s upozorněním.

Data se neuloží a formulář musí být vyplněn znovu. Prezentace formuláře viz Obrázek č. 20 Formulář – localhost/form.html. Celý formulář viz Příloha č. 15 Formulář – localhost/form.html. Kód programu viz Příloha č. 16 /form.html.



Obrázek č. 20 Formulář – localhost/form.html

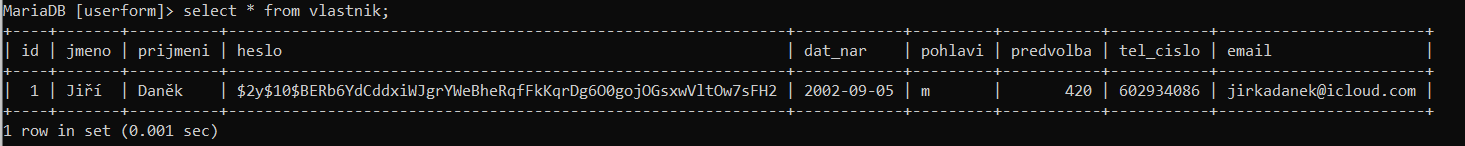
K uložení dat do databáze je potřeba také PHP kód, který pracuje na straně serveru. Nejdříve naváže spojení s vytvořenou databází pomocí příkazu ($conn = new mysqli()) a poté přebere vložená data z formuláře a dosadí je do proměnných. Ty jsou následně uloženy do připravené tabulky vlastník. Heslo je zároveň přeměněno z bezpečnostních důvodů, například zcizení dat, na tzv. hash, který je poté zapsán v atributu heslo.

Vyskytne-li se během procesu chyba, vždy je program přerušen a vypsána odůvodňující hláška. Po úspěšném zadání je uživatel přesměrován na stránku, kde je přivítán svým jménem. Dále je mu nabídnuto, zda chce odejít, nebo pracovat se svým ubytováním. Viz Obrázek č. 21 Účet – localhost/insert.php. Kód programu viz Příloha č. 17 /insert.php. Kód programu při zadání stejného emailu/čísla, viz Příloha č. 18 /message.php.



Obrázek č. 21 Účet – localhost/insert.php

Následně můžeme zkontrolovat v CLI, který je spojen s MySQL databází, že jsou údaje správně uloženy. To provedeme pomocí příkazu (SELECT \* FROM vlastnik), viz Obrázek č. 22 Výpis dat z tabulky – MySQL.



Obrázek č. 22 Výpis dat z tabulky – MySQL

## Porovnání MySQL vs MS Access

MS Access je v případě integrace databáze do aplikací téměř nepoužitelný a není zde možné používat SQL kód, proto byl v této práci primárně použit software MySQL. Jediné využití MS Accessu je při tvorbě osobní databáze, kvůli nezbytnosti znát základy SQL a srozumitelnosti.

Naopak MySQL je ideálním programem pro prezentaci této práce. To především díky své užitečnosti při implementaci reálné aplikace, či open-source variantě. Při použití MySQL Workbench dokážeme vizualizovat tabulky, či přímo ovládat funkce serveru.

# Dotazování v DB

Slouží k výčtu požadovaných hodnot z vybrané tabulky či sady tabulek, mezi kterými jsou vytvořeny vazby. Tímto způsobem je možné zjistit požadované informace z tabulky prostřednictvím příkazu:

SELECT (atributy) FROM (hlavní tabulka) WHERE (omezení);

## Příklady možných dotazů

První dotaz je zaměřen na objednávky za minulý rok, který by mohl být potřebný při ověřování dat před podáváním daňového přiznání. Nebo pouze k porovnání, jak si podnik vede s rezervacemi při porovnání s jinými lety. Z celé řady rezervací, které jsou v tabulce rezervace\_tbl uloženy tedy vybereme ty, u kterých je dat\_rezervace v rozmezí prvního a posledního dne minulého roku, zároveň se ale kvůli symbolu (\*) zobrazí všechny sloupce tabulky. Viz Obrázek č. 23 Dotaz počet rezervací za rok – MySQL.

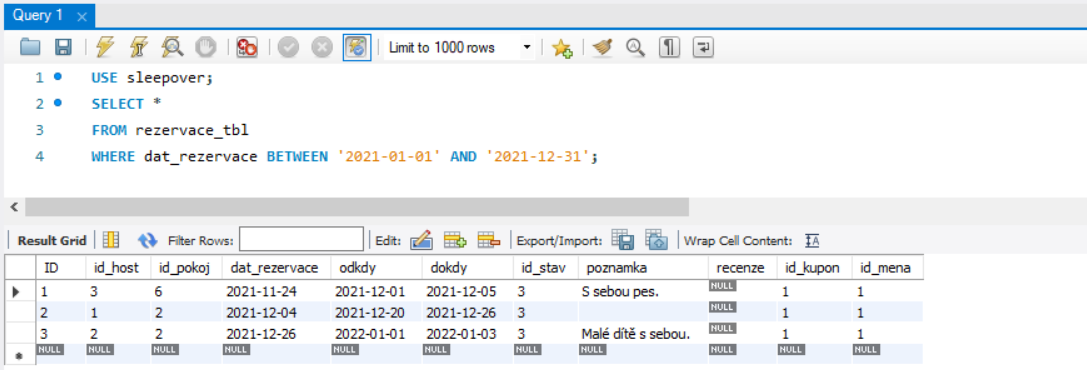
Kód:

SELECT \*

FROM rezervace\_tbl

WHERE dat\_rezervace

BETWEEN '2021-01-01' AND '2021-12-31';



Obrázek č. 23 Dotaz počet rezervací za rok – MySQL

Druhým dotazem je vytvořen žebříček uživatelů, s největším počtem rezervací, který je zároveň uveden v posledním sloupci pod názvem pocet. Tato informace by se mohla uplatnit při zpřístupnění výhod pro aktivní uživatele. Tím by se zvýšilo množství rezervací a také spojený vyšší zisk. Dotaz propojuje dvě tabulky, kterými jsou rezervace a host, kvůli lepší identifikaci hosta (jméno, příjmení). Viz Obrázek č. 24 Dotaz počet vytvořených objednávek – MySQL.

Kód:

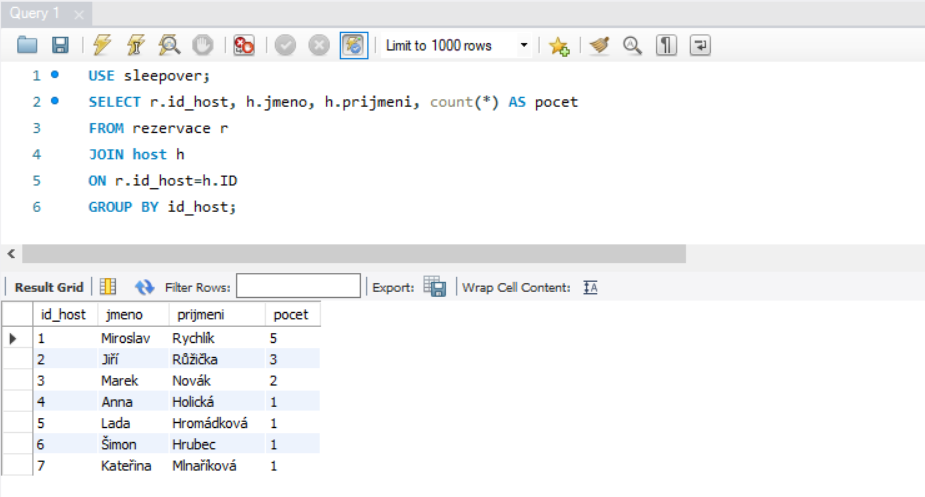
SELECT r.id\_host, h.jmeno, h.prijmeni, count(\*) AS pocet

FROM rezervace r

JOIN host h

ON r.id\_host = h.ID

GROUP BY id\_host;



Obrázek č. 24 Dotaz počet vytvoření objednávek – MySQL

Třetí dotaz představuje rovněž žebříček informací o rezervacích a cenách pobytu, který je spočten pomocí tří proměnných (pocet\_dnu, cena\_noc a fce\_kup). Nejdříve bylo nutné spočítat, pomocí funkce DATEDIFF(), počet dnů rezervace a také propojit tři tabulky prostřednictvím cizích klíčů. Následně jsou hodnoty pouze seřazeny sestupně. Z výsledku je posléze možné vypočítat průměrnou cenu, která je také užitečným indikátorem v podnikatelském prostředí. Viz Obrázek č. 25 Dotaz výpočet ceny – MySQL.

Kód:

SELECT r.ID, dat\_rezervace, datediff(dokdy, odkdy) AS pocet\_dnu, p.cena\_noc, k.nazev\_kup, datediff(dokdy, odkdy)\*p.cena\_noc\*fce\_kup AS cena

FROM rezervace r

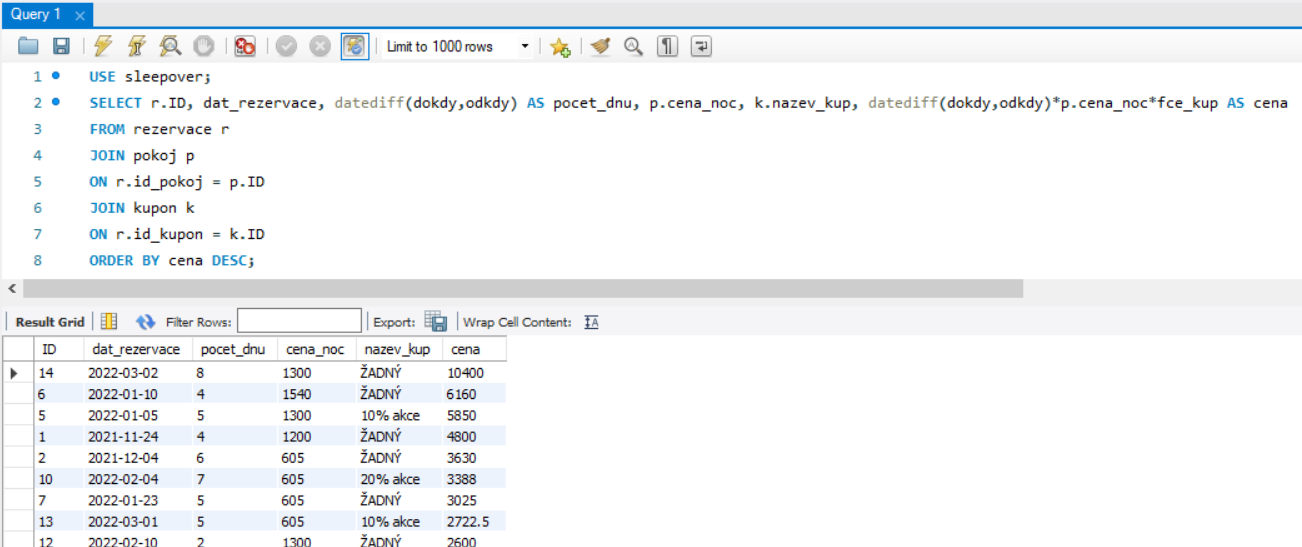
JOIN pokoj p

ON r.id\_pokoj = p.ID

JOIN kupon k

ON r.id\_kupon = k.ID

ORDER BY cena DESC;

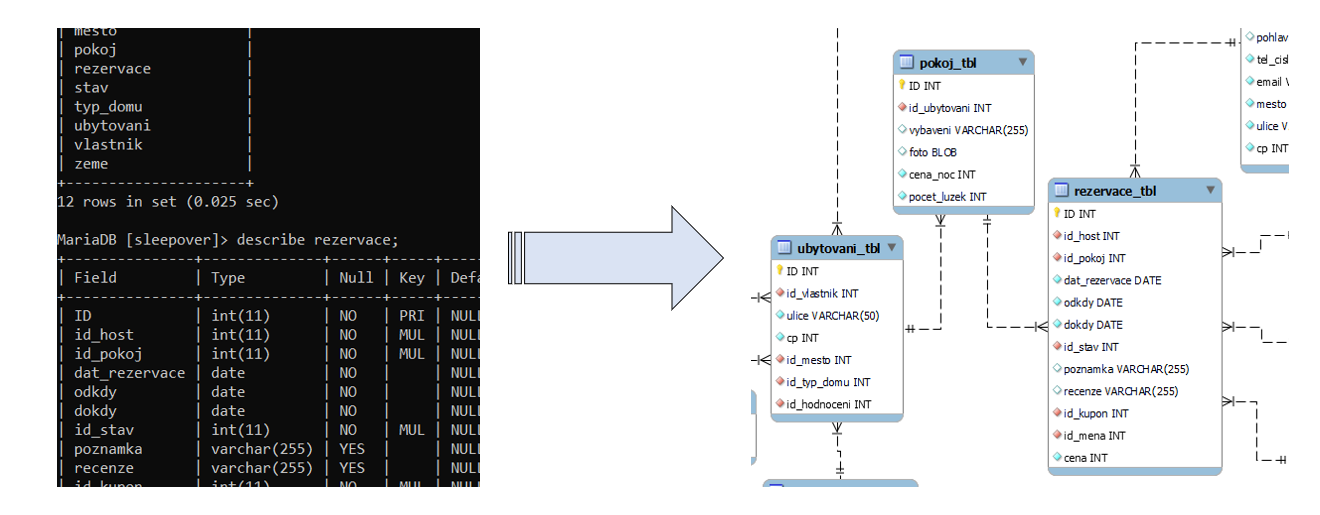


Obrázek č. 25 Dotaz výpočet ceny – MySQL

# Reverzní inženýrství

Způsob, jak z již stvořené databáze (SQL kódu) dostat ER diagram prezentován v kapitole Databázová vrstva. K tomuto účelu existují funkce, jak v MySQL Workbench, tak i v SQL Data Modeleru. Oba softwary jsou z tzv. DDL (Data Definition Language) kódu schopny vytvořit tabulky s relacemi včetně zadaných datových typů.

Postup reverzního inženýringu na vytvořené databázi v MySQL pomocí SQL kódu byl proveden následujícím způsobem. V softwaru MySQL Workbench byla zvoleno pole Database a zde funkce Reverse Engineer. Byla vyplněna potřebná data pro spojení s databází. Dále byla označena databázi, která byla následně převedena na ERD. Posléze zobrazí požadovaný výsledek, viz Obrázek č. 26 Reverzní inženýring – MySQL.



Obrázek č. 26 Reverzní inženýring – MySQL

Obdobně pracuje i software SQL Data Modeler, který pomocí naprogramovaných operací vytvoří z fyzického modelu logický. Oba diagramy lze porovnat viz Příloha č. 8 a Příloha č. 10.

# Závěr

Byla vytvořena vlastní relační databáze, pomocí třívrstvé architektury, reprezentující rezervační systém ubytování. Ta obsahuje jednotlivé entity, které mají také určité role:

* Vlastník
* Ubytování
* Typ domu
* Hodnocení
* Město
* Země
* Pokoj
* Rezervace
* Host
* Stav
* Kupon
* Měna

Hlavní (výstupní) entitou byla zvolena „Rezervace“. Příklad výstupu je: „v jeden moment může být ubytování rezervováno pouze jedním hostem“.

Konceptuální model byl předveden na softwarech Visio od korporace Microsoft a Lucidchart od stejnojmenné společnosti. Výstupy jsou prezentovány převážně pomocí softwaru Lucidchart, kvůli své jednoduchosti tvorby, ceně a spoustě definovaných vzorů. UML diagramy (Sekvenční, Use Case, ER, Class, Stavový) jsou popsány přímo na reálném využití na rezervačního systému. Viz přílohy.

Databázový model byl předveden na softwarech SQL Data Modeler a MySQL Workbench od korporace Oracle. Zde vytvořená tabulka přímo popisuje verzové odchylky včetně ceny zvoleného softwaru MySQL. Ten byl vybrán nejen kvůli spolupráci s MySQL Workbench, kde jsou prezentovány výstupy databázového modelu, ale především u důvodu open-source Community balíčku a možnosti integrace aplikací.

MS Access představuje prvotní návrh fyzického modelu, kde byla vytvořena databáze rezervačního systému. Jako rozhraní pro vkládání dat byla vytvořena sada formulářů, která umožňuje například zadat údaje o vlastníkovi a jeho ubytování, či po vložení informací o hostovi vytvořit z předchozích entit samotnou rezervaci. Příkladem prezentace výstupů pro rezervace je sestava „Faktura“, která kombinuje množství atributů a dotazů.

Pomocí platformy XAMPP (Apache, MySQL) byla vytvořena ukázková databáze pro vkládání dat skrz formulář na webovém rozhraní, prostřednictvím napsaného PHP kódu. Následně byla data ukládána do připravené tabulky. Formulář reprezentuje možné rozhraní např. pro registraci vlastníka do systému. Zde je možnost další integrace pro přihlašování uživatelů, či přímé nasazení na rezervační systém.

Po vložení záznamů o fiktivních osobách jsou v práci prezentovány dotazy včetně SQL kódu, které by mohly mít reálné využití v byznysovém prostředí. Příkladem dotazu je: „Kolik objednávek zprostředkoval systém minulý rok?“.

Pro ukázku převodu SQL kódu databáze (DDL) do formy ER diagramu – reverzního inženýringu, byl využit software MySQL Workbench. Ten byl propojen přímo se serverem, kde byla databáze uložena a následně zde byl předveden výsledný diagram.

V přílohách je také přiložena prezentace, která přehledně shrnuje a popisuje vytvořenou práci.

**Literatura a zdroje**

*[1] Software. TechTarget [online]. Newton: Rosencrance, 2021 [cit. 2022-02-16]. Dostupné z: https://www.techtarget.com/searchapparchitecture/definition/software*

*[2] Oracle Corporation. Britannica [online]. Chicago: Lotha, 2019 [cit. 2022-02-16]. Dostupné z: https://www.britannica.com/topic/Oracle-Corporation*

*[3] Microsoft Corporation. Britannica [online]. Chicago: Bauer, 2020 [cit. 2022-02-16]. Dostupné z: https://www.britannica.com/topic/Microsoft-Corporation/Further-developments-in-Windows-OS*

*[4] Databáze [online]. Olomouc [cit. 2022-02-16]. Dostupné z: http://www.jardaz.cz/soubory/db/databaze.pdf. Učební materiál. Univerzita Palackého, Přírodovědecká fakulta, Katedra informatiky.*

*[5] Hierarchický databázový model. Cs.education-wiki [online]. [cit. 2022-02-16]. Dostupné z: https://cs.education-wiki.com/2247971-hierarchical-database-model*

*[6] Database Architecture in DBMS: 1-Tier, 2-Tier and 3-Tier. Guru99 [online]. Gujarat: Peterson, 2022 [cit. 2022-02-16]. Dostupné z: https://www.guru99.com/dbms-architecture.html*

*[7] SQL Developer Data Modeler. Oracle [online]. Austin, [cit. 2022-02-17]. Dostupné z: https://www.oracle.com/cz/database/technologies/appdev/datamodeler.html*

*[8] MySQL Workbench. JavaTpoint [online]. [2022] [cit. 2022-02-17]. Dostupné z: https://www.javatpoint.com/mysql-workbench*

*[9] Database. Oracle [online]. Austin, [2022] [cit. 2022-02-17]. Dostupné z: https://www.oracle.com/database/what-is-database/*

*[10] The Three-level ANSI-SPARC Architecture. GeeksforGeeks [online]. rn540, 2020 [cit. 2022-02-17]. Dostupné z: https://www.geeksforgeeks.org/the-three-level-ansi-sparc-architecture/*

*[11] MS Access. TheItBase [online]. Hyderabad: theitbase, 2020 [cit. 2022-02-21]. Dostupné z: https://www.theitbase.com/definitions/microsoft-access/amp/*

*[12] MySQL. TechTarget [online]. Newton: Moore, 2018 [cit. 2022-03-01]. Dostupné z: https://searchoracle.techtarget.com/definition/MySQL*

*[13] UML. Visual Paradigm [online]. Hong Kong [cit. 2022-03-06]. Dostupné z: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-uml/*

*[14] Lucidchart. PC Mag [online]. New York: Duffy, 2021 [cit. 2022-03-06]. Dostupné z: https://www.pcmag.com/reviews/lucidchart*

*[15] Visio. Microsoft [online]. Redmond: Microsoft, [2022] [cit. 2022-03-06]. Dostupné z: https://www.microsoft.com/en-us/microsoft-365/visio/flowchart-software/*

**Seznam obrázků**

[Obrázek č. 1 Porovnání možností databáze [4]](#_Toc97464545)

[Obrázek č. 2 Hierarchický model](#_Toc97464546)

[Obrázek č. 3 ER diagram – Konceptuální model](#_Toc97464547)

[Obrázek č. 4 Sekvenční diagram – Visio](#_Toc97464548)

[Obrázek č. 5 Diagram případu užití – Visio](#_Toc97464549)

[Obrázek č. 6 Diagram případu užití – Lucidchart](#_Toc97464550)

[Obrázek č. 7 Diagram tříd – Lucidchart](#_Toc97464551)

[Obrázek č. 8 Entitně–relační diagram – Lucidchart](#_Toc97464552)

[Obrázek č. 9 Stavový diagram – Lucidchart](#_Toc97464553)

[Obrázek č. 10 ER diagram – Databázový model](#_Toc97464554)

[Obrázek č. 11 ERD Databázová vrstva – SQL Data Modeler](#_Toc97464555)

[Obrázek č. 12 ERD Databázová vrstva – MySQL Workbench](#_Toc97464556)

[Obrázek č. 13 ER diagram – Fyzický model](#_Toc97464557)

[Obrázek č. 14 Fyzický model SQL Data Modeler](#_Toc97464558)

[Obrázek č. 15 ERD Databázová vrstva – MS Access](#_Toc97464559)

[Obrázek č. 16 Formulář rezervace – MS Access](#_Toc97464560)

[Obrázek č. 17 Dotaz výpočet ceny – MS Access](#_Toc97464561)

[Obrázek č. 18 Sestava faktura – MS Access](#_Toc97464562)

[Obrázek č. 19 Vytvoření tabulky vlastník – MySQL](#_Toc97464563)

[Obrázek č. 20 Formulář – localhost/form.html](#_Toc97464564)

[Obrázek č. 21 Účet – localhost/insert.php](#_Toc97464565)

[Obrázek č. 22 Výpis dat z tabulky – MySQL](#_Toc97464566)

[Obrázek č. 23 Dotaz počet rezervací za rok – MySQL](#_Toc97464567)

[Obrázek č. 24 Dotaz počet vytvoření objednávek – MySQL](#_Toc97464568)

[Obrázek č. 25 Dotaz výpočet ceny – MySQL](#_Toc97464569)

[Obrázek č. 26 Reverzní inženýring – MySQL](#_Toc97464570)

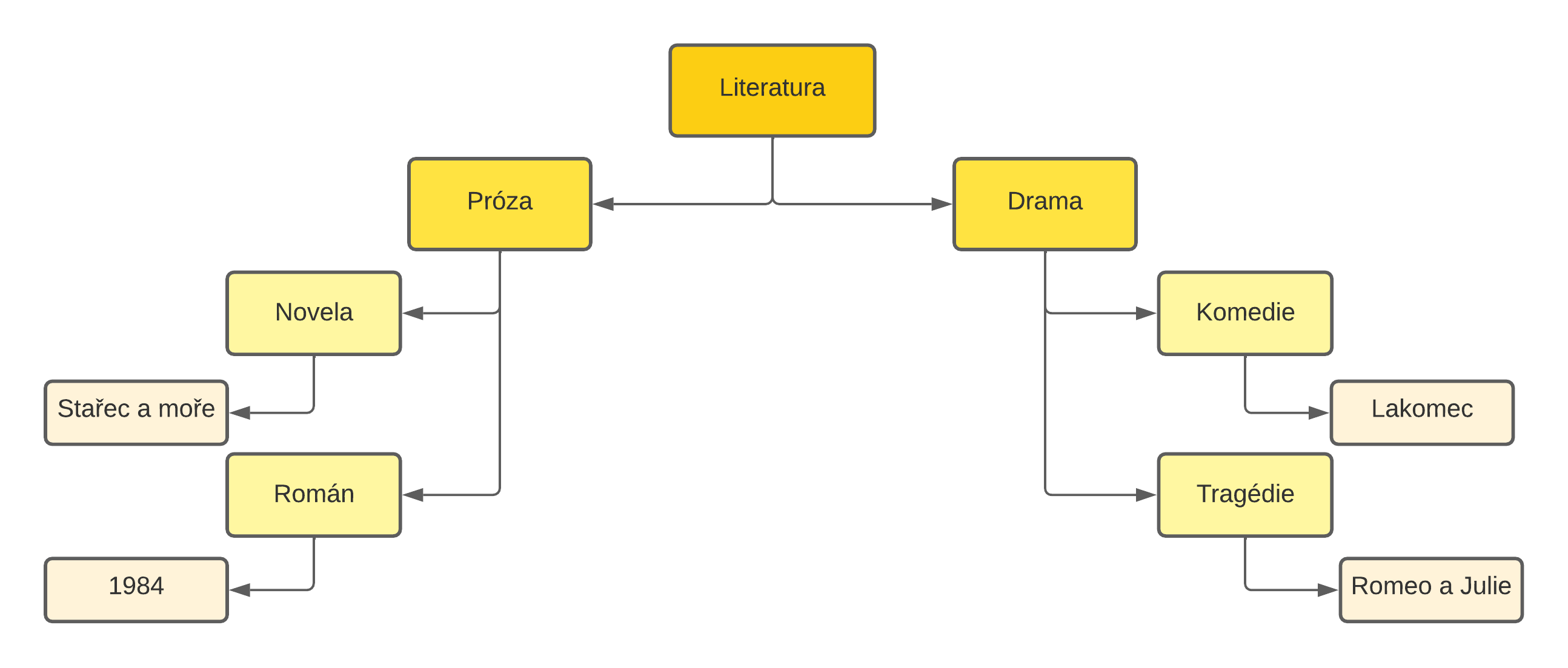
**Seznam tabulek**

[Tabulka č. 1 Srovnání MS Visio vs Lucidchart](#_Toc97105937)

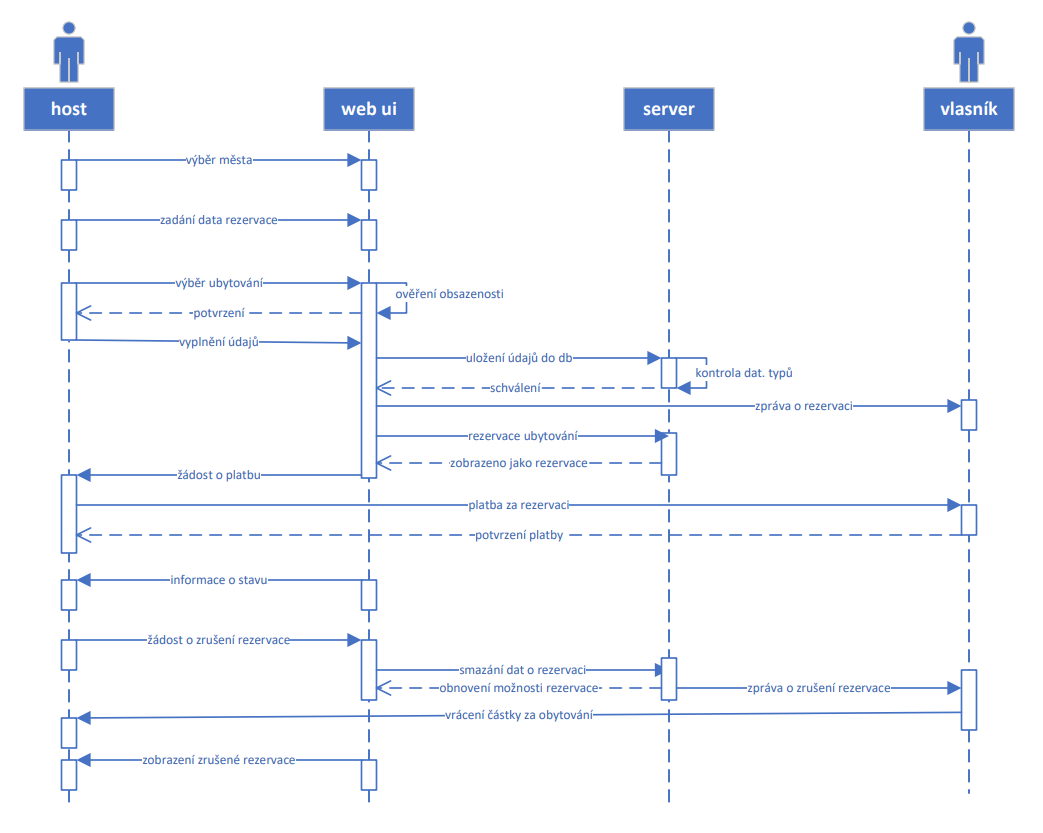
[Tabulka č. 2 Srovnání základních balíčků – MySQL](#_Toc97105938)

**Seznam příloh**

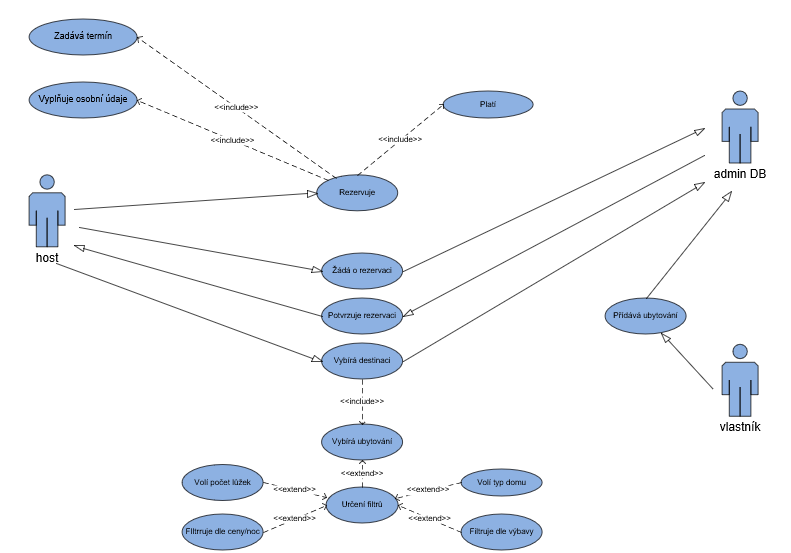
**Příloha č. 1 Hierarchický model**



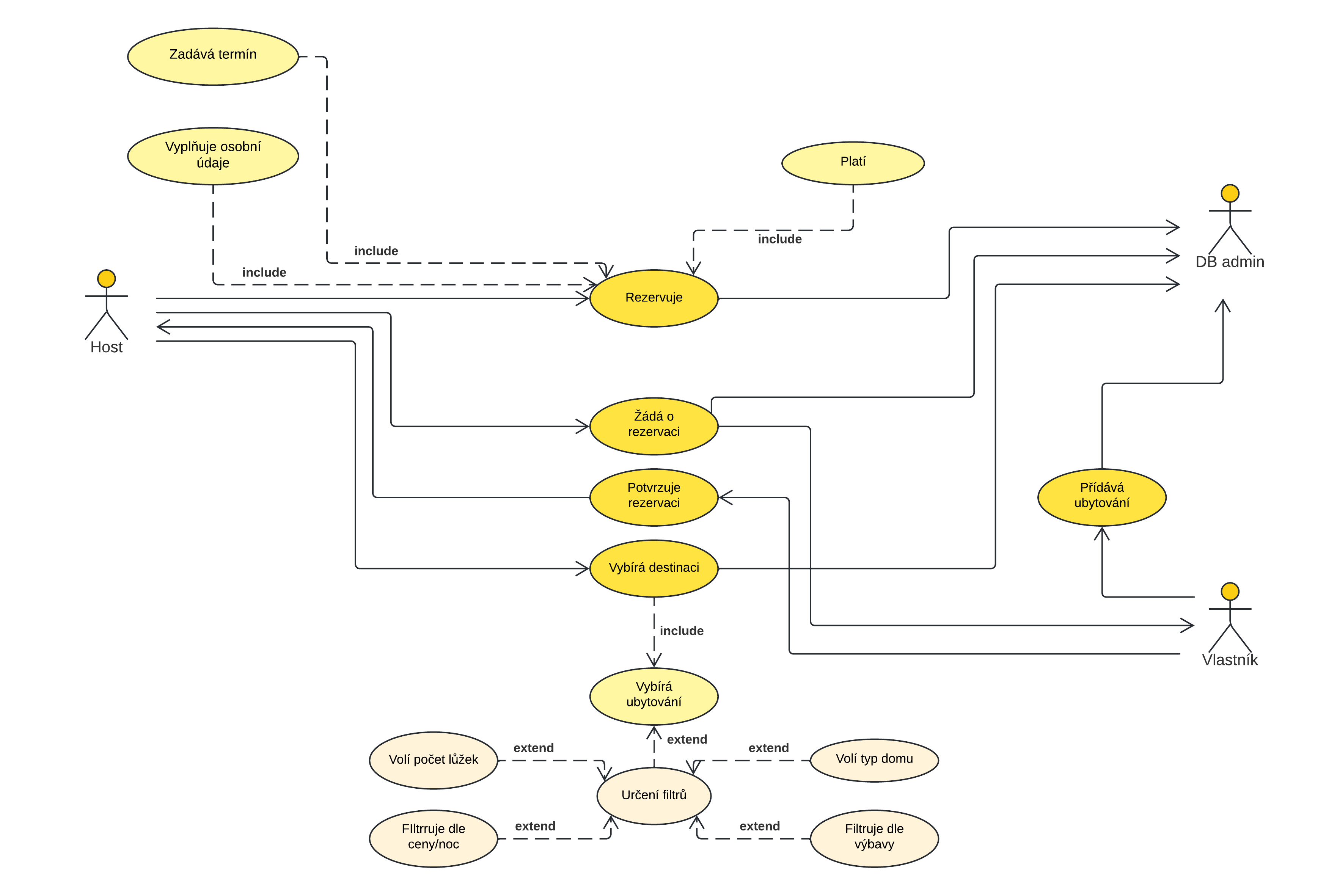
**Příloha č. 2 Sekvenční diagram – Visio**



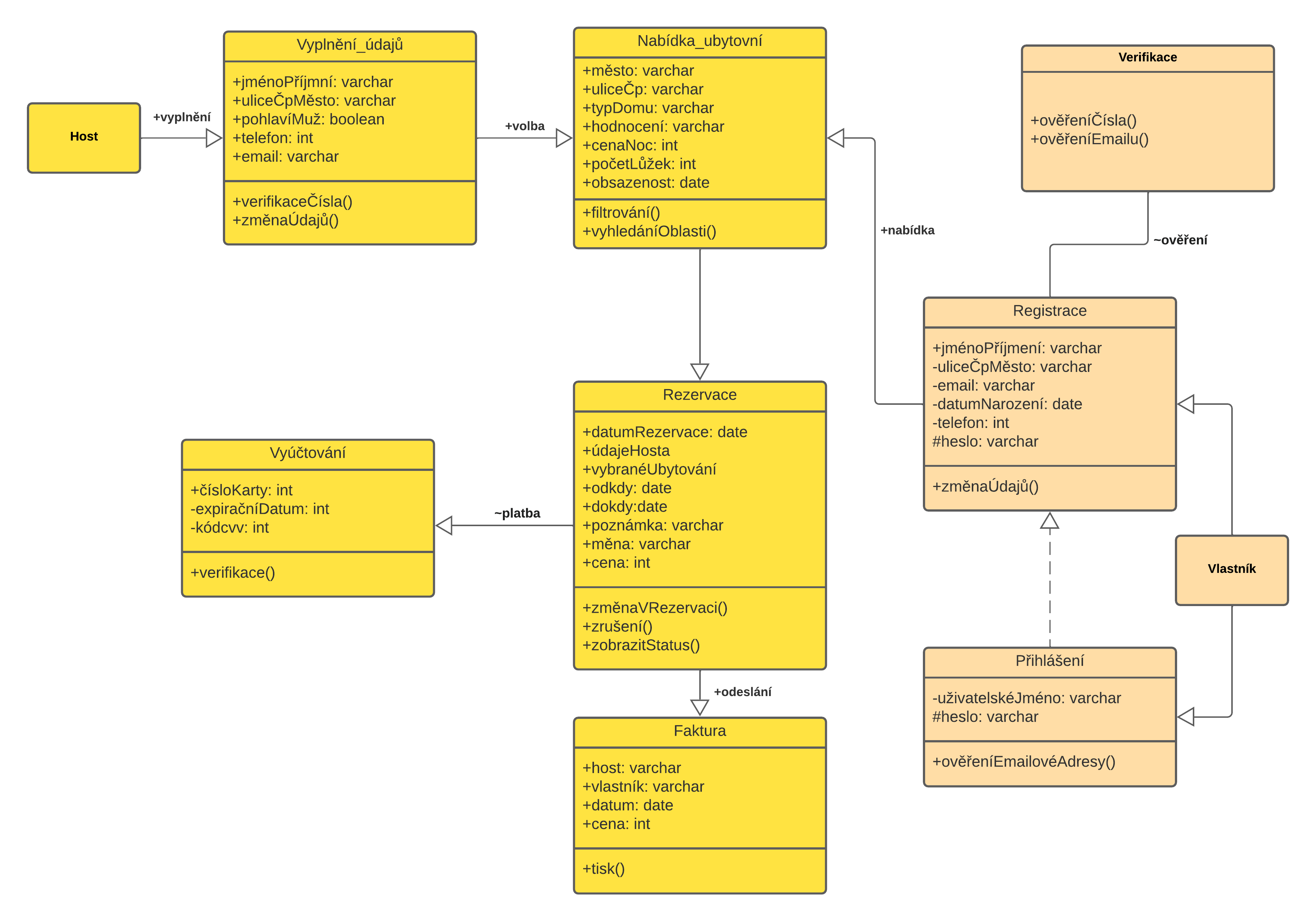
**Příloha č. 3 Diagram případu užití – Visio**

****

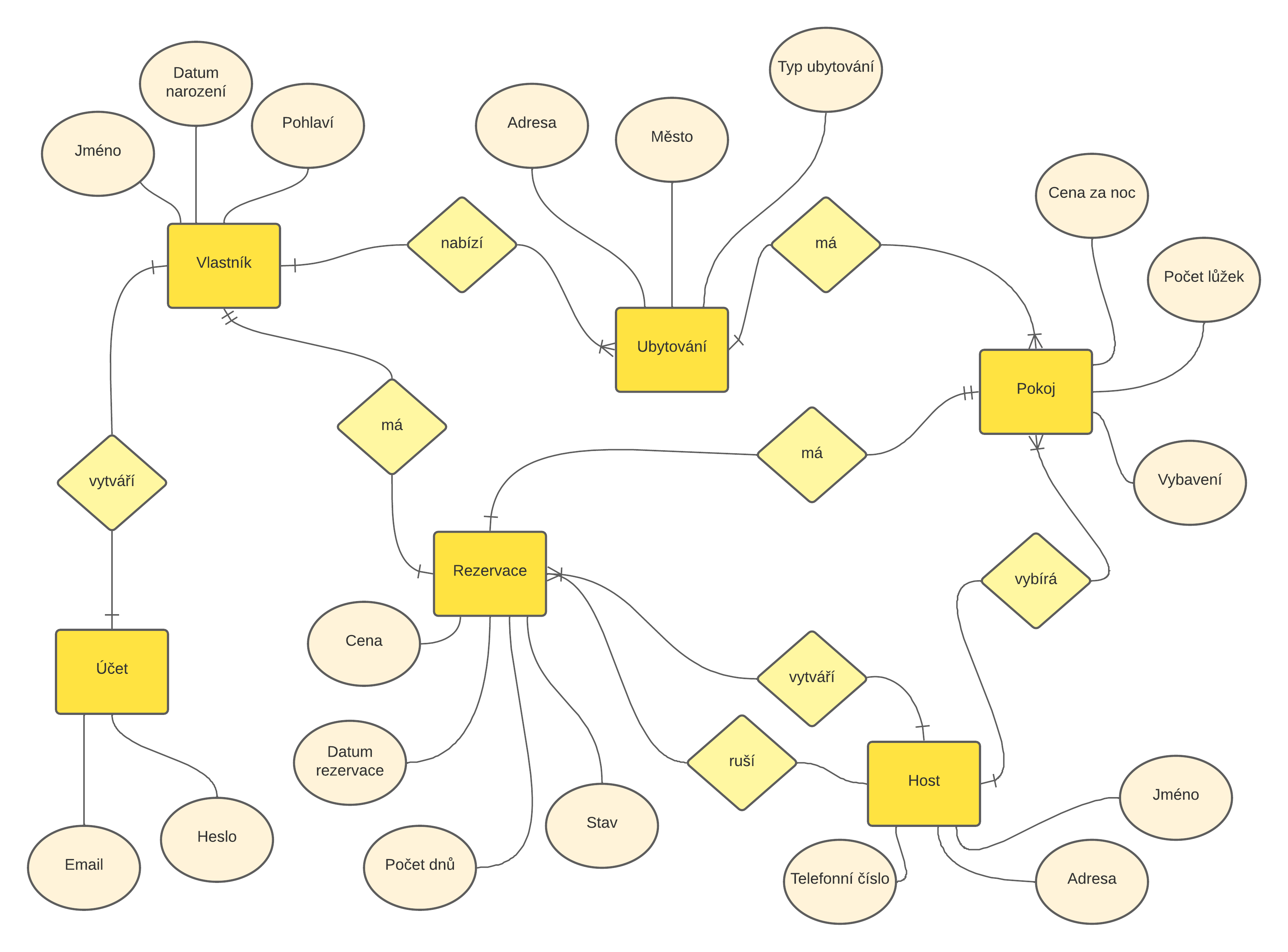
**Příloha č. 4 Diagram případu užití – Lucidchart**

****

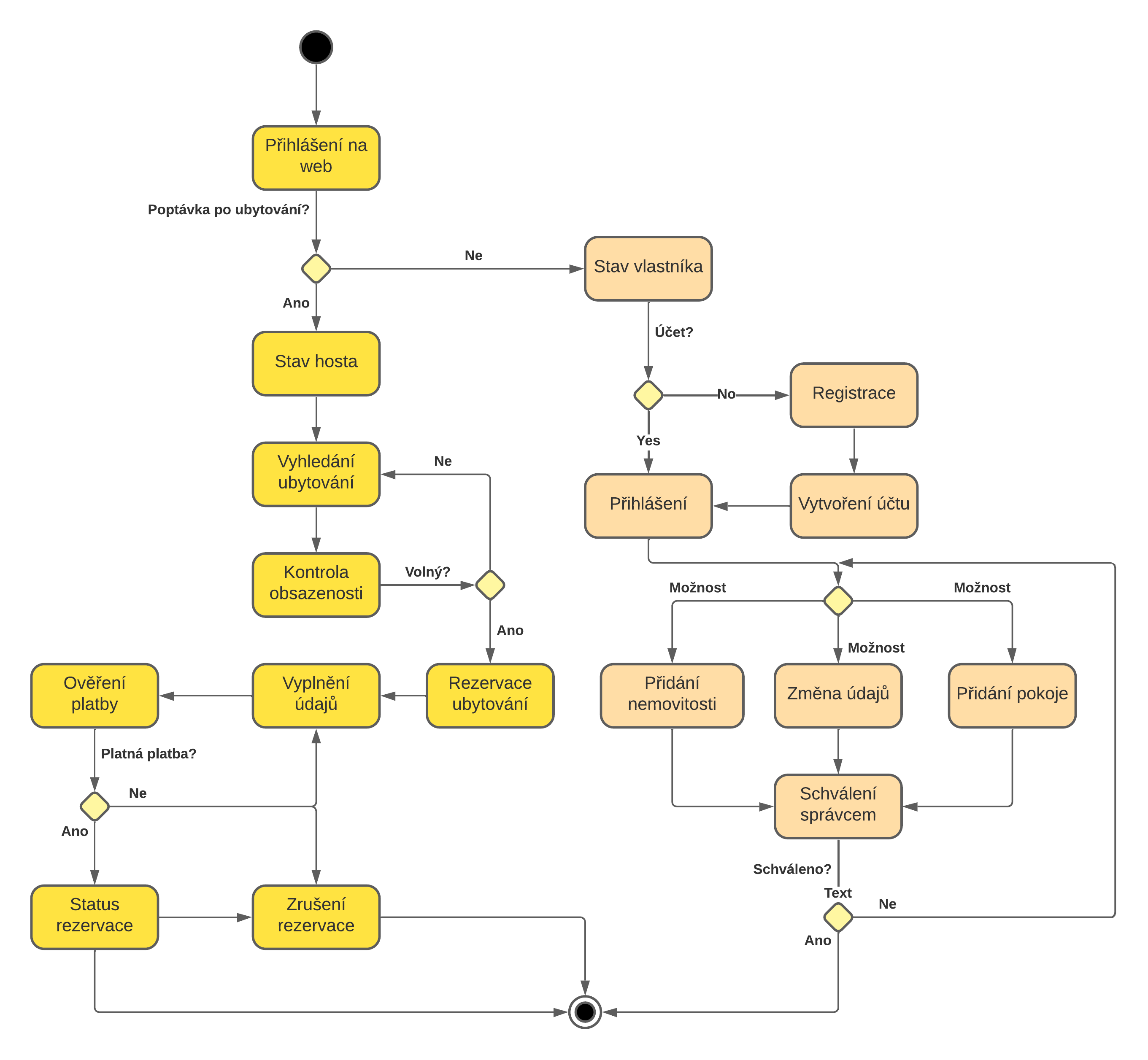
**Příloha č. 5 Diagram tříd – Lucidchart**

****

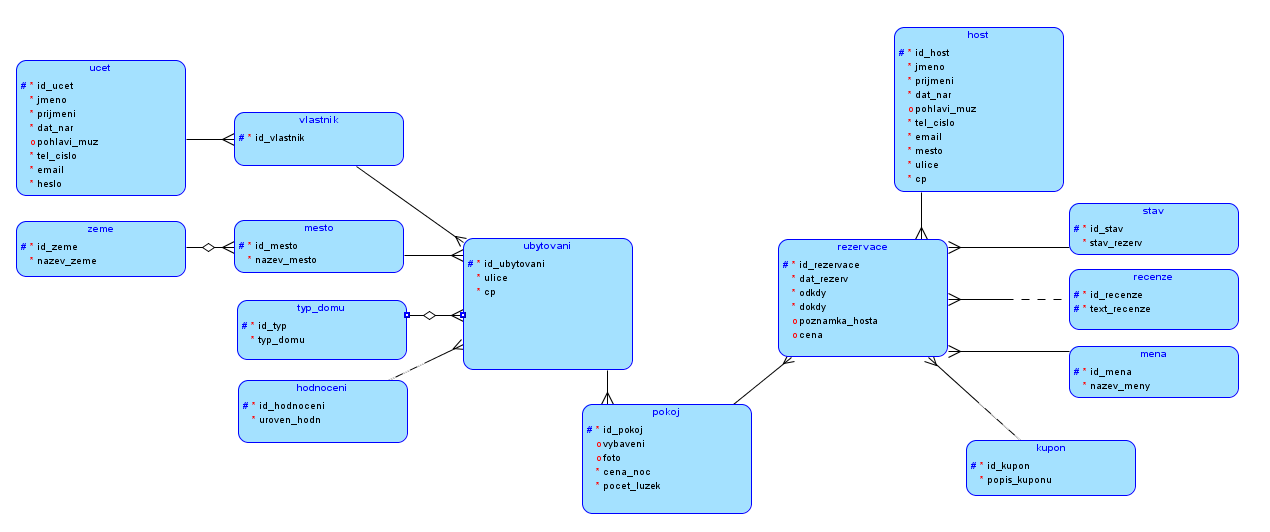
**Příloha č. 6 Entitně–relační diagram – Lucidchart**

****

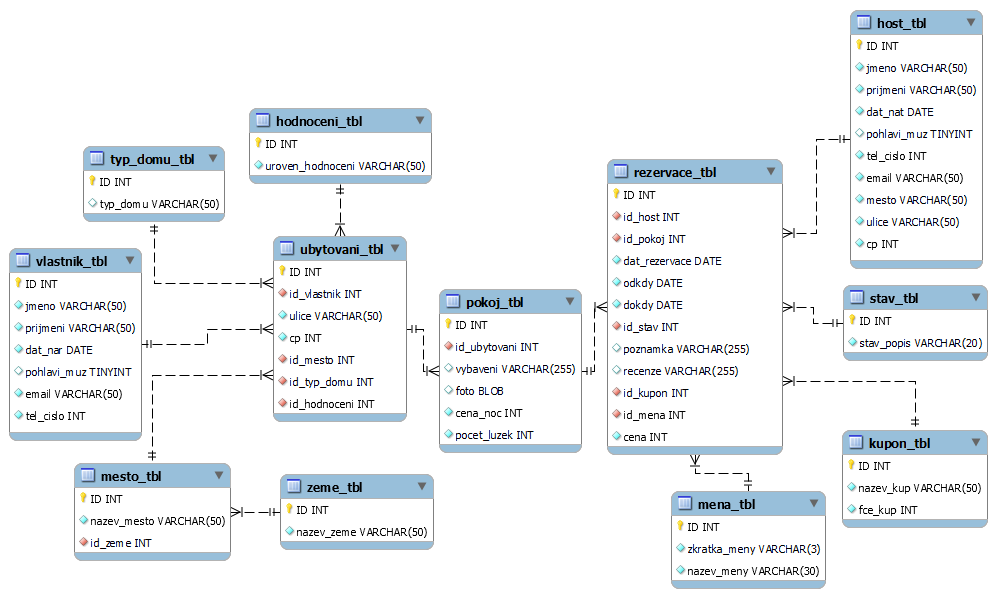
**Příloha č. 7 Stavový diagram – Lucidchart**

****

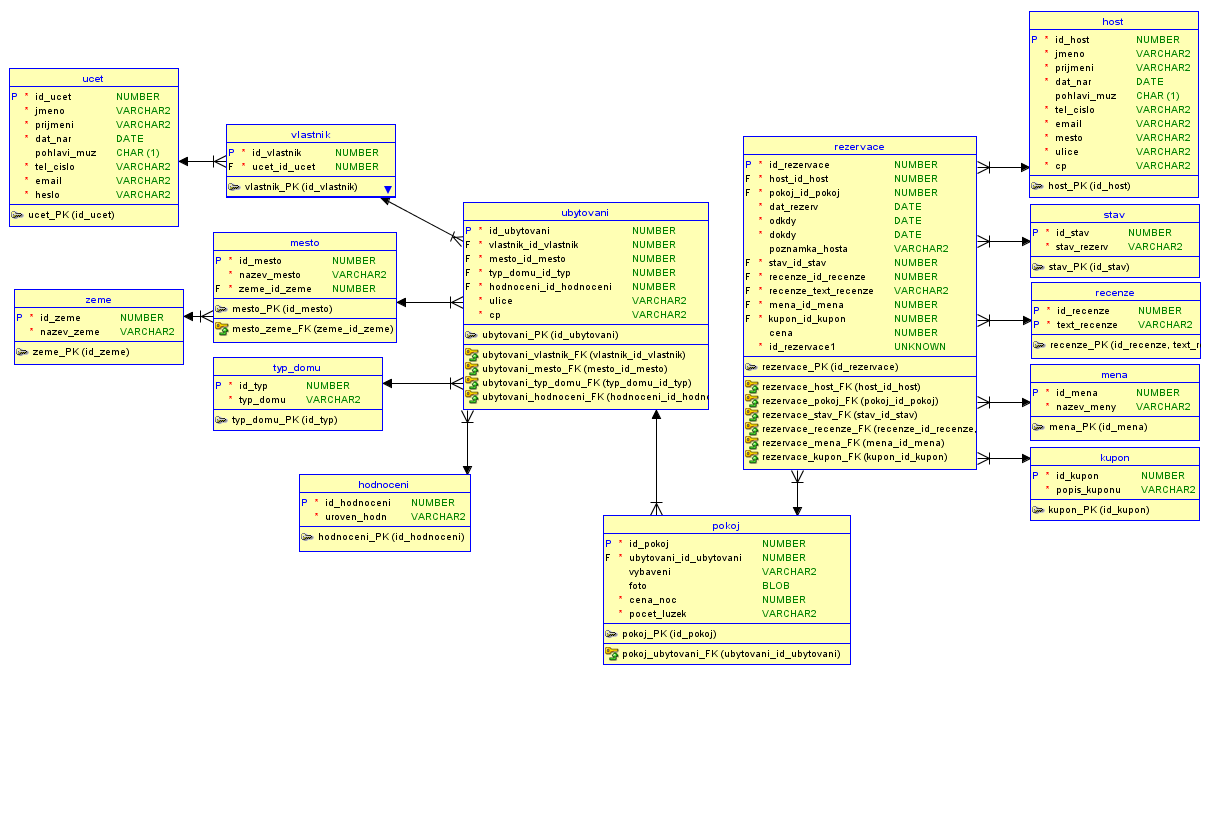
**Příloha č. 8 ERD Databázová vrstva – SQL Data Modeler**

****

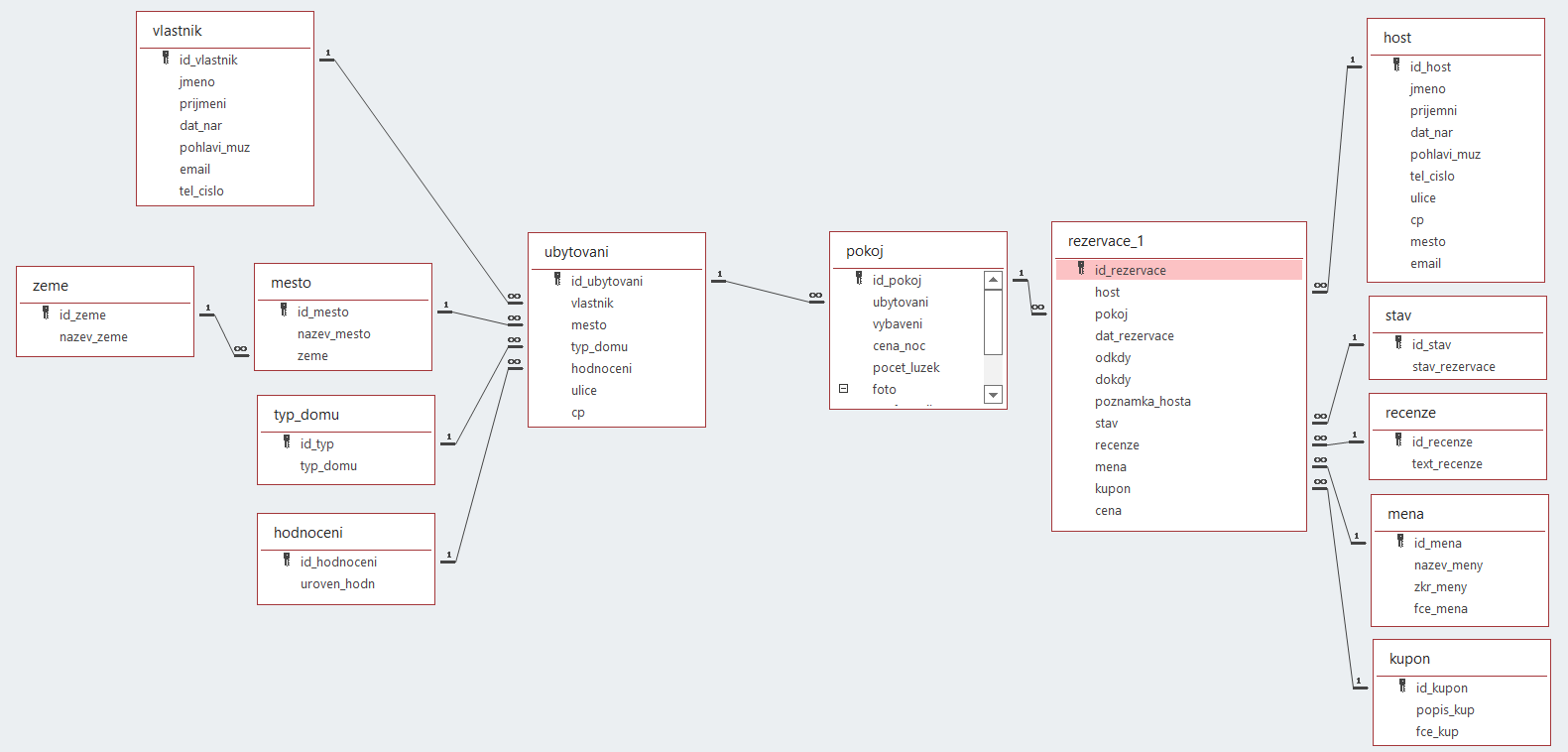
**Příloha č. 9 ERD Databázová vrstva – MySQL Workbench**

****

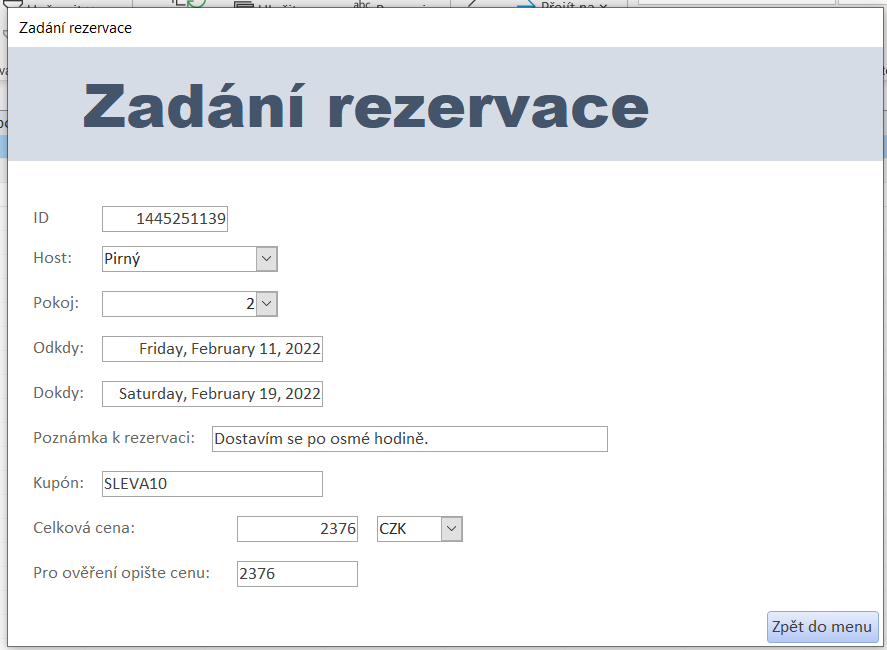
**Příloha č. 10 Fyzický model SQL Data Modeler**

****

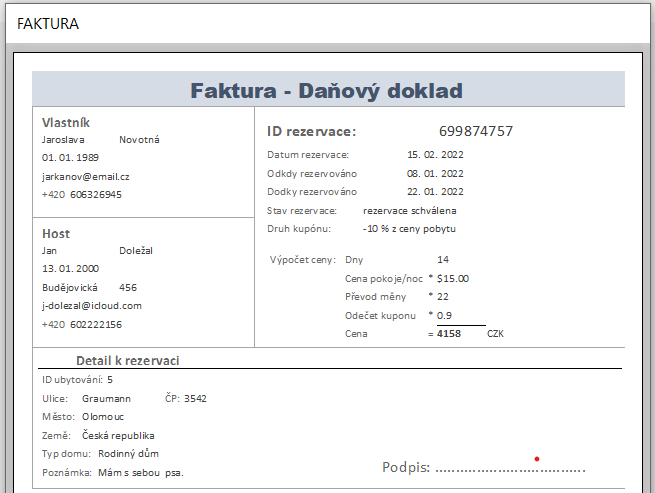
**Příloha č. 11 ERD Databázová vrstva – MS Access**

****

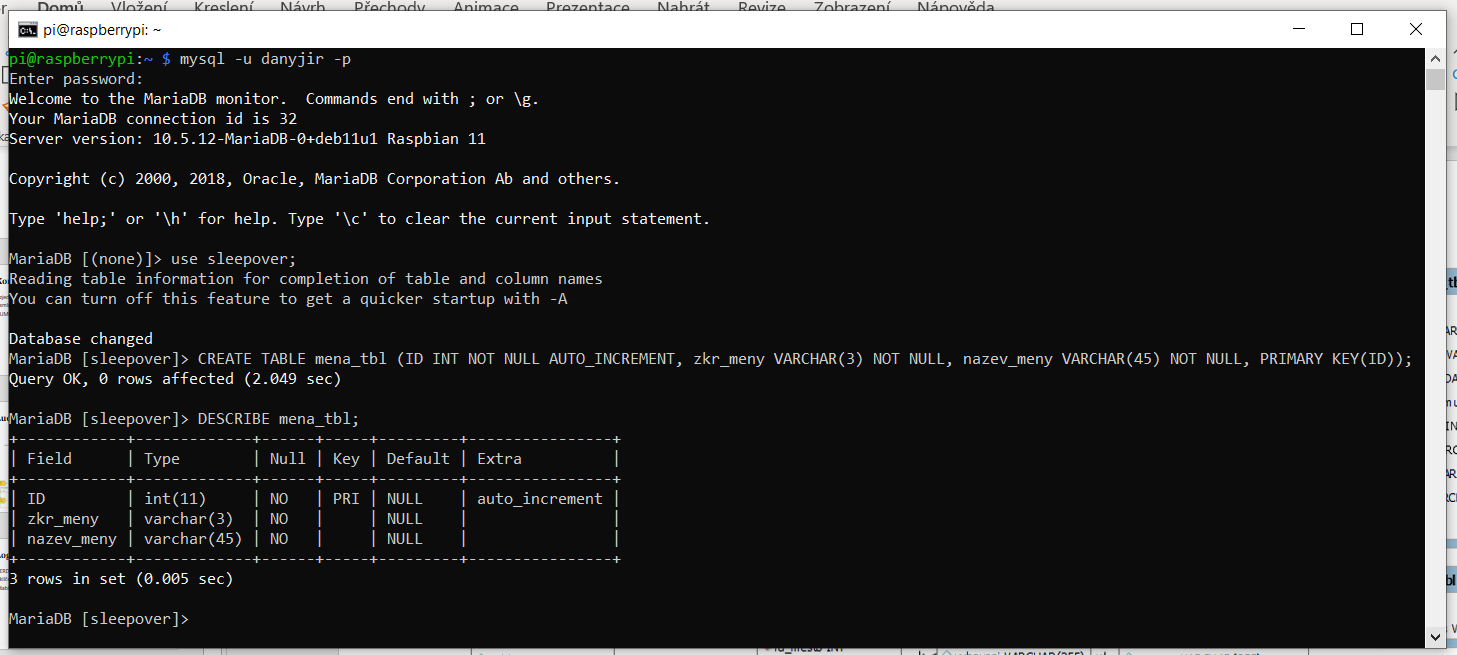
**Příloha č. 12 Formulář rezervace – MS Access**

****

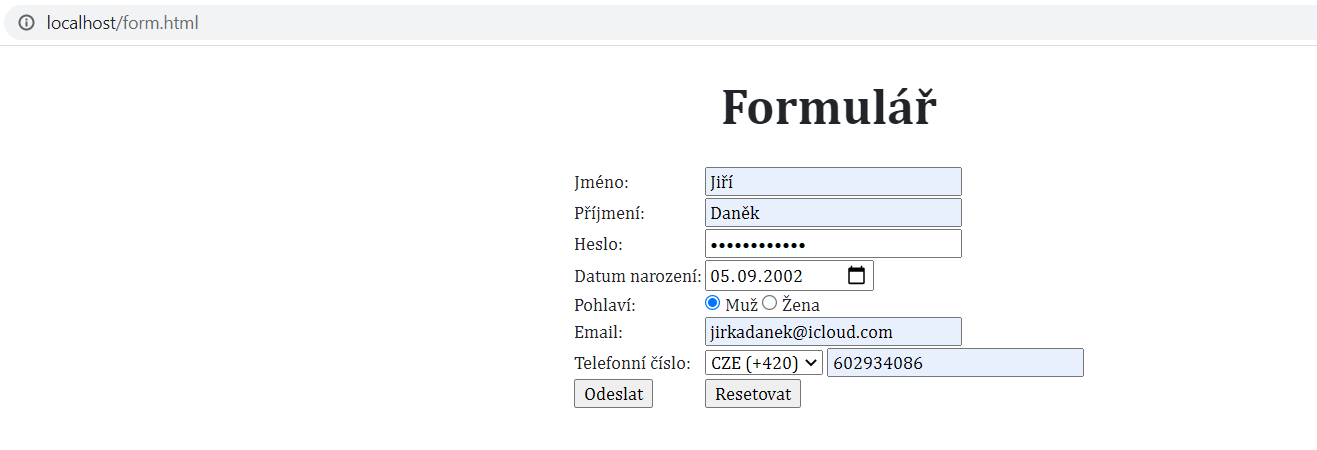
**Příloha č. 13 Sestava faktura – MS Access**

****

**Příloha č. 14 Vytvoření tabulky vlastník – MySQL**

****

**Příloha č. 15 Formulář – localhost/form.html**

****

**Příloha č. 16 /form.html**

<!DOCTYPE HTML>

<html lang="cs">

<head>

<meta charset="utf-8">

<title>Register Form</title>

</head>

<body>

<h1> <b>Formulář</b> </h1>

<center>

<form action="insert.php" accept-charset="utf-8" method="POST">

<table>

<tr>

<td>Jméno:</td>

<td><input type="text" name="firstname" required></td>

</tr>

<tr>

<td>Příjmení:</td>

<td><input type="text" name="lastname" required></td>

</tr>

<tr>

<td>Heslo:</td>

<td><input type="password" name="password" required></td>

</tr>

<td>Datum narození:</td>

<td><input type="date" id="datePickerId" name="birth" required></td>

<tr>

<td>Pohlaví:</td>

<td>

<input type="radio" name="gender" value="m" required> Muž

<input type="radio" name="gender" value="f" required> Žena

</td>

</tr>

<tr>

<td>Email:</td>

<td><input type="text" name="email" required></td>

</tr>

<tr>

<td>Telefonní číslo:</td>

<td>

<select name="phoneCode" required>

<option selected hidden value=""> Předvolba</option>

<option value="420">CZE (+420)</option>

<option value="421">SVK (+421)</option>

</select>

<input type="text" name="phone" required>

</td>

</tr>

<tr>

<td><input type="submit" name="submit"></td>

<td><input type="reset"></td>

</tr>

</table>

</form>

</center>

<script>

datePickerId.max = new Date().toISOString().split("T")[0];

</script>

</body>

</html>

**Příloha č. 17 /insert.php**

<?php

header('Content-Type: text/html; charset=utf-8');

if (isset($\_POST['submit'])) {

if (isset($\_POST['firstname']) && isset($\_POST['lastname']) && isset($\_POST['password']) &&

isset($\_POST['birth']) && isset($\_POST['gender']) && isset($\_POST['email']) &&

isset($\_POST['phoneCode']) && isset($\_POST['phone'])) {

$jmeno = $\_POST['firstname'];

$prijmeni = $\_POST['lastname'];

$password = $\_POST['password'];

$heslo = password\_hash($password, PASSWORD\_DEFAULT);

$dat\_nar = date('Y-m-d', strtotime($\_POST['birth']));

$pohlavi = $\_POST['gender'];

$email = $\_POST['email'];

$predvolba = $\_POST['phoneCode'];

$tel\_cislo = $\_POST['phone'];

$host = "localhost";

$dbUsername = "root";

$dbPassword = "";

$dbName = "userform";

$conn = new mysqli($host, $dbUsername, $dbPassword, $dbName);

if ($conn->connect\_error) {

die('Nebylo možné se spojit s databází.');

}

else {

$Select\_email = "SELECT email FROM vlastnik WHERE email = ? LIMIT 1";

$Select\_tel = "SELECT tel\_cislo FROM vlastnik WHERE tel\_cislo = ? LIMIT 1";

$Insert = "INSERT INTO vlastnik(jmeno, prijmeni, heslo, dat\_nar, pohlavi, predvolba, tel\_cislo, email) values(?, ?, ?, ?, ?, ?, ?, ?)";

$stmt = $conn->prepare($Select\_email);

$stmt->bind\_param("s", $email);

$stmt->execute();

$stmt->bind\_result($resultEmail);

$stmt->store\_result();

$stmt = $conn->prepare($Select\_tel);

$stmt->bind\_param("s", $tel\_cislo);

$stmt->execute();

$stmt->bind\_result($resultTel\_cislo);

$stmt->store\_result();

$stmt->fetch();

$rnum = $stmt->num\_rows;

if ($rnum == 0) {

$stmt->close();

$stmt = $conn->prepare($Insert);

$stmt->bind\_param("sssssiis",$jmeno, $prijmeni, $heslo, $dat\_nar, $pohlavi, $predvolba, $tel\_cislo, $email);

if ($stmt->execute()) {

$message = "Údaje úspěšně uloženy databáze";

echo "<script type='text/javascript'>alert('$message');</script>";

}

else {

echo $stmt->error;

}

}

else {

header("location: message.php");

}

$stmt->close();

$conn->close();

}

}

else {

echo "Všechna pole nejsou vyplněna.";

die();

}

}

else {

echo "Tlačítko k odeslání selhalo.";

}

?>

<!DOCTYPE html>

<html lang="cs">

<head>

<meta charset="utf-8">

<title>Vítej</title>

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">

<style>

body{ font: 14px Cambria; text-align: center; }

</style>

</head>

<body>

<h1 class="my-5">Zdravím <b><?php echo $jmeno; ?></b>, vítej ve svém účtu.</h1>

<p>

<a href="add\_accom.php" class="btn btn-warning">Pokračovat na ubytování</a>

<a href="form.html" class="btn btn-danger ml-3">Odejít</a>

</p>

</body>

</html>

**Příloha č. 18 /message.php**

<?php

echo "<center>

<br><br><br><br><br><br><br>

<h1>Tento email či číslo je již použito. :(</h1>

<br> Zadejte prosím takové, které není v databázi.

</center>";

echo '<script>

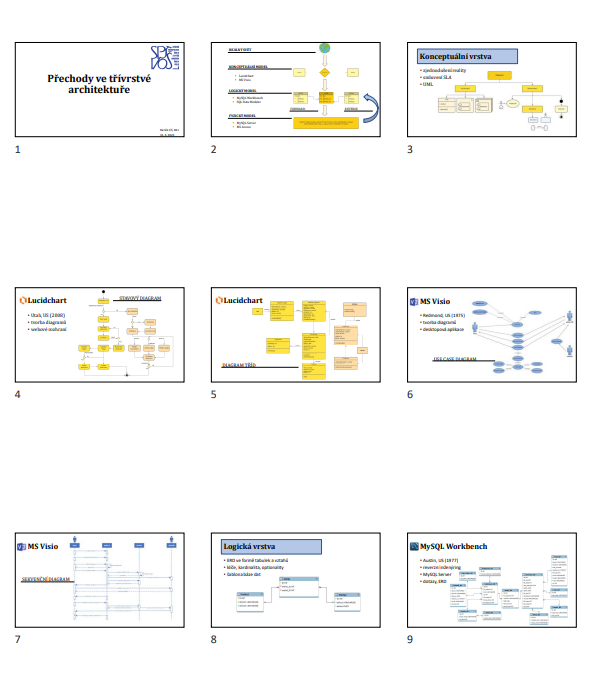
setTimeout(function()

{window.location.href ="form.html"}, 4 \* 1000);

</script>';

?>

 **Příloha č. 19 Prezentace – Přechody třívrstvé architektury**

****

**Obsah obrázku stůl

Popis byl vytvořen automaticky**

1. Nehodící škrtněte [↑](#footnote-ref-1)