

**VYSOKÁ ŠKOLA POLYTECHNICKÁ JIHLAVA**

Katedra technických studií

**Systém pro podporu tvorby studijních  
plánů**

bakalářská práce

Autor práce: Tomáš Cink

Vedoucí práce: doc. Ing. Karel Richta, CSc.

Jihlava 2019

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Autor práce:

**Tomáš Cink**

Studijní program:

Elektrotechnika a informatika

Obor:

Aplikovaná informatika

Název práce:

**Systém pro podporu tvorby studijních plánů**

Cíl práce:

Cílem práce je vytvoření aplikace pro studenty, která by jim pomáhala při volbě předmětů pro následující období studia. Studenti by měli k dispozici kompletní seznam předmětů pro toto období a mohli by si vybrat předměty, které by chtěli studovat. Aplikace by jim pomáhala při výběru v tom, že by jim kontrolovala rozvrh vybraných předmětů, počítala jejich kredity a kontrolovala důležité požadavky, jako jsou povinné předměty, kredity z povinně-volitelných a volitelných předmětů. Aplikace by dále kontrolovala, zda má student splněné požadavky na sport. Dále by kontrolovala, zda má student splněny všechny prerekvizity zvolených předmětů. Dále by obsahovala všechny důležité informační milníky jako sehnání praxe, závěrečné práce, počítání kreditů za dva po sobě jdoucí předměty a podobně. Celkově by měla aplikace ulehčit dlouhodobou přípravu na studium, zejména pro studenty, kteří z nějakého důvodu nevyužijí doporučený studijní plán. V aplikaci bude dostupné vytvořit plány od dvou do jedenácti semestrů.



**doc. Ing. Karel Richta, CSc.**  
vedoucí bakalářské práce



**doc. Ing. Zdeněk Horák, Ph.D.**  
vedoucí katedry  
Katedra technických studií

## **Abstrakt**

Plánování je dnes důležitým krokem ke všemu, co chceme dovést do zdárného konce a právě plánováním celého studia na vysoké škole se zabývá i tato práce. Tento systém pokrývá problematiku dlouhodobého plánování, kontroly podmínek k úspěšnému průchodu studiem a dodává informace o předmětech, oborech a garantech předmětů. Analytická a návrhová část využívá modely UML, které popisují funkce systému. Pro práci s databází bylo využito jazyku SQL a na programování aplikace byl použit jazyk C#. Rozhraní aplikace je vytvořeno v prostředí .NET pomocí formulářových oken. Aplikace splňuje všechna kritéria studia. Skvělým přínosem pro studenty je možnost vytvářet svoje vlastní plány v prostředí, které obsahuje jejich předměty a umožňuje jim volně plánovat studium a tím si udržovat přehled a kontrolu nad plněním všech podmínek pro zdárné ukončení.

## **Klíčová slova**

C#; .Net; UML; plán; student

## **Abstract**

A planning is today an important step to reach desired goals and a planning of a student's study on a university is what this thesis is about. This system covers problems of long-time planning, controlling requirements to successful passage through a study and it provides information about subjects, study programs and guarantees. Analytic and design part of this work uses models of the UML that describes system functions. The SQL was used for working with the database and the C# language was used for programming. The application interface is created in .NET using the Windows Form modules. The application meets all criteria of a study. A great value for the students is an opportunity to create their own plans of a study in an environment that contains all the school subject and allows them to freely do their planning of a study. Thanks to that they will have an overview and control of their successful fulfillment of study conditions.

## **Keywords**

C#; .Net; UML; plan; student

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů, v platném znění, dále též „AZ“).

Souhlasím s umístěním bakalářské práce v knihovně VŠPJ a s jejím užitím k výuce nebo k vlastní vnitřní potřebě VŠPJ.

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje AZ, zejména § 60 (školní dílo).

Beru na vědomí, že VŠPJ má právo na uzavření licenční smlouvy o užití mé bakalářské práce a prohlašuji, že **s o u h l a s í m** s případným užitím mé bakalářské práce (prodej, zapůjčení apod.).

Jsem si vědom toho, že užít své bakalářské práce či poskytnout licenci k jejímu využití mohu jen se souhlasem VŠPJ, která má právo ode mne požadovat přiměřený příspěvek na úhradu nákladů, vynaložených vysokou školou na vytvoření díla (až do jejich skutečné výše), z výdělku dosaženého v souvislosti s užitím díla či poskytnutím licence.

V Jihlavě dne 11. dubna 2019

.....  
Podpis studenta

## **Poděkování**

*Chtěl bych poděkovat svému vedoucímu doc. Ing. Karlovi Richtovi, CSc. za cenné rady, vedení při vypracování a možnost vytvářet tuto práci pod jeho vedením. Poděkování také patří kolegovi Jáchymu Hruškovi za poskytnutí dat na hromadné načítání předmětů a popisů ze školní databáze.*

## Obsah

Úvod.....	9
Motivace .....	10
Cíl práce.....	11
1 Teoretická část .....	12
1.1 Současný stav problematiky.....	12
1.2 Výběr prostředí ASP.Net vs .Net .....	13
1.3 C#.....	13
1.4 SQL .....	13
1.5 UML.....	14
1.6 NuGet.....	14
1.6.1 Dapper.....	14
2 Analýza a návrh řešení.....	15
2.1 Datový model aplikace.....	15
2.2 Datový slovník .....	16
2.2.1 Obor .....	16
2.2.2 Předmět .....	17
2.2.3 Katedra.....	18
2.2.4 Vyučující.....	18
2.2.5 Záznam.....	19
2.2.6 Plán semestr .....	19
2.2.7 Výběr .....	20
2.3 Funkční model aplikace .....	21
2.3.1 Případy užití pro aktéra v roli Uživatel.....	22
2.3.2 Případy užití pro aktéra v roli Správa .....	23
2.4 Scénáře případů užití aktéra v roli Správa .....	23
2.4.1 Správa garantů správcem: .....	24
2.4.2 Model správy garanta.....	25
2.4.3 Model správy oborů .....	26
2.4.4 Model správy předmětů .....	27
3 Popis implementace .....	28
3.1 Tvorba plánu .....	28
3.2 Přidávání předmětů do semestru .....	30
3.3 Výpočet kreditů.....	32
3.4 Třída na práci s rozbalovacími seznamy a seznamy .....	32
3.5 Vyhledávání a úprava.....	33

3.6	Hromadné plnění dat do databáze .....	34
3.7	Načtení popisů k předmětům do databáze .....	36
3.8	Návod k použití .....	38
3.9	Podmínky spuštění .....	39
3.10	Testování .....	39
4	Diskuze .....	41
	Závěr .....	42
	Seznam použité literatury .....	44
	Přílohy .....	45

## Seznam obrázků

Obrázek 1 - ER Diagram aplikace SPPSP (vlastní).....	15
Obrázek 2 - Use Case model aplikace SPPSP (vlastní).....	21
Obrázek 3 - Scénář – správa garantů (vlastní).....	25
Obrázek 4 - Scénář – správa oborů (vlastní).....	26
Obrázek 5 - Scénář – správa předmětů (vlastní).....	27
Obrázek 6 - Tvorba záznamu (vlastní) .....	28
Obrázek 7 - Tvorba nového záznamu (vlastní).....	29
Obrázek 8 - Úprava záznamu a změna počtu semestru (vlastní).....	30
Obrázek 9 - Načtení lichých předmětů do seznamu (vlastní).....	31
Obrázek 10 - Vkládání dat do komponenty rozbalovací seznam (vlastní).....	33
Obrázek 11 - Vyhledávání v komponentě rozbalovací seznam (vlastní) .....	33

## Seznam tabulek

Tabulka 1 - Datový slovník pro Obor .....	16
Tabulka 2 - Datový slovník pro Předmět.....	17
Tabulka 3 - Datový slovník pro Katedru .....	18
Tabulka 4 - Datový slovník pro Vyučující .....	19
Tabulka 5 - Datový slovník pro Záznam .....	19
Tabulka 6 - Datový slovník pro Plán semestru.....	20
Tabulka 7 - Datový slovník pro Výběr .....	20
Tabulka 8 - Scénář – správa garantů.....	24

## **Seznam použitých zkratk**

CRUD Vytvořit, číst, upravit, smazat (create, read, update, delete)

ER Vztahy entit

IIS Internetová informační služba

IS Informační systém

SPPSP Systém pro podporu tvorby studijních plánů

SQL Strukturovaný dotazovací jazyk

UML Unifikovaný modelovací jazyk

VS Visual Studio

VŠPJ Vysoká škola polytechnická Jihlava



## Úvod

Každý student kdysi zažil ten pocit nejistoty, zda bude dostatek prostředků k pokračování ve studiu a nebude nutné žádat o výjimky nebo ukončit studium. Tento problém by mnohým mohl vyřešit Systém pro podporu tvorby studijních plánů.

Hlavním cílem této práce je navrhnout a realizovat aplikaci, jenž by implementovala systém pro podporu tvorby studentských plánů. Účelem této aplikace je vytvořit prostředí, ve kterém si mohou studenti naplánovat celé jejich studium a mít přehled a plán na celou dobu studia. Aplikace umožní nastavení počtu semestrů od dvou semestrů pro studenty s dostatkem uznaných předmětů až po jedenáct semestrů. Může být užitečná pro studenty studující kratší, standartní nebo delší délku studia. Zvolené maximum je nejpravděpodobnější podle počtů semestrů potřebných ke splnění studia, kde další semestr by prakticky znamenal celé další studium. Jelikož plánování bude navrženo flexibilně, lze tak přidávat a odebírat semestry podle potřeby nastalých změn při studiu a mít tak vždy svůj aktuální studentský plán kdykoliv ke kontrole a k dispozici.

Student bude mít možnost zobrazit všechny předměty a filtrovat předměty podle toho, zda má tyto předměty již zapsané nebo zda budou dostupné v letních nebo zimních semestrech. Vybrané předměty bude následně možné přiřadit do zvolených semestrů.

Při zařazení předmětu do plánu budou přepočítávány kredity, jak semestru, tak celkového studia a budou kontrolovány prerekvizity, jako potřebnost nějakého předmětu, který musí být dokončen před vybráním daného předmětu. Dále budou kontrolována zapsání všech povinných předmětů a také povinně volitelných, volitelných předmětů a v neposlední řadě jazyků a sportů. Kromě těchto kontrol se bude zobrazovat upozornění na další důležité milníky jako výběr praxe, navolení závěrečných praxí a další. Jelikož není nic závazné navolené předměty lze kdykoliv odebírat nebo přidávat anebo vytvářet několik verzí plánu mezi kterými půjde jednoduše přepínat a tím mít připraveno několik možností, jenž pokryjí všechny potřebné body a rizikové předměty, které budou muset býti přesunuté nebo nahrazené jako většinou bývají vyměňovány povinně volitelné a volitelné předměty po zjištění obtížnosti nebo nezajímavosti těchto předmětů.

Kromě plánů bude možné v aplikaci dohledat všechny informace o předmětech a oborech. U každého oboru budou vedeny informace o předmětech, kde ve většině

případů se budou nacházet cíle předmětu, získané znalosti, dovednosti, sylabus předmětu a doporučená literatura. Dále budou vedeny údaje o garantech předmětu a kateder, pod které předmět spadá. U každého z garantů budou vedeny kontaktní informace garantů jako email, telefon a také informace o tom, kdy jsou dostupní na konzultace v konzultačních hodinách. Z každého oboru bude možné zobrazit celkový seznam předmětů, takže uživatel bude moci porovnat obory a vybrat si obor ještě před podáním přihlášek, což ušetří čas z hledání všech informací na obsáhlých stránkách všech oborů.

Celkově by aplikace měla ulehčit dlouhodobou přípravu na studium, jelikož školy umožňují většinou pouze počítání kreditů absolvovaných předmětů již ukončených semestrů a přidávání předmětů do dalšího ročníku, do kterého se zapisujete v určité době na začátku semestru. Tato možnost je dostačující pro studenty, kteří studují podle doporučeného studijního plánu a nepotřebují přetvářet plán, přesunovat předměty, kontrolovat prerekvizity a počítat kredity, aby se nedostali do situace, že nějaký z následujících semestrů nebude dostatečný počet předmětů k umožnění pokračování studia do dalších ročníků.

Díky možnosti zapnout práva správce lze tuto aplikaci využívat i na jiných školách a oborech, které nejsou zahrnuty do současné databáze. Správce bude moci hromadně načítat data do databáze podle předepsaného formátu, což ulehčí zpracování velkého množství předmětů, jež by zabralo hodiny přepisovat do aplikace ručně. Kromě hromadného načítání bude k možné i jednotlivé přidávání, upravování a mazání předmětu, oboru, kateder i garanta podle potřeby, a tak využít tuto aplikaci i pro ostatní, jenž chtějí mít předem naplánováno svoje studium.

## **Motivace**

Toto téma bylo vybráno kvůli častým problémům studentů, kteří z různých důvodů prodlužují studium, jenž má za důsledek často nedostatek kreditů ke konci studia a za důsledek ukončení studia nebo žádání o výjimku. Celkově by tato aplikace měla napomoci studentům s bezpečnějším volením předmětů bez strachování při registraci nového semestru, zda budou kredity stačit k pokračování. Vytvoření této aplikace také prověří moje schopnosti a zkušenosti nabyté z předmětů databázové systémy na práci s daty v databázi, programování a programování desktopových aplikací využité pro

implementaci této aplikace a umožnění přehledného zobrazení díky formulářovým oknům technologie .NET.

## **Cíl práce**

Hlavním cílem práce je implementace tohoto systému, který umožní studentům ulehčit jejich volbu předmětů pro celé nebo následující období studia. Studenti by měli k dispozici kompletní seznam předmětů pro toto období a mohli by si vybrat předměty, které by chtěli studovat. Aplikace by jim pomáhala při výběru v tom, že by jim kontrolovala výběr předmětů, počítala jejich kredity a kontrolovala důležité požadavky, jako jsou počty povinných, povinně-volitelných, volitelných předmětů a sportů. U předmětů budou také uvedeny prerekvizity u předmětů, aby se zajistila dostatečná znalost, která může být vyžadována před studiem určitého předmětu. V neposlední řadě bude uchovávat i všechny důležité informační milníky jako sehnání praxe, závěrečné práce, počítání minimálních kreditů v semestru podle studentského řádu. Celkově by měla aplikace ulehčit dlouhodobou přípravu na studium, zejména pro studenty, kteří z nějakého důvodu nevyužijí doporučený studijní plán. Díky této aplikaci si mohou zjistit, zda mají dostatečný počet kreditů k dalšímu pokračování ve studiu na škole. V aplikaci bude možné vytvářet plány s délkou studia od dvou do jedenácti semestrů.

## 1 Teoretická část

V této kapitole jsou zahrnuty části jako je rešerše, teoretická část práce, ve které jsou rozebrány důvody výběru prostředí, použité metody a prostředky využití k tvorbě aplikace.

### 1.1 Současný stav problematiky

V této části jsou popsány podobné systémy k tomuto tématu. Jsou zde rozebrány systémy, které implementují plánování studia v nějaké podobě. Mezi tyto systémy patří IS VŠPJ, UIS, IS/STAG a Plánovač studia.

*IS VŠPJ* je Informační systém Vysoké školy polytechnické Jihlava. Pro plánování předmětů se využívá několik kroků. Předregistrace, která kontroluje, kolik studentů má zájem o určitý předmět a podle zájmu se tento předmět přidá do registrací. Převážně pro povinně volitelné a volitelné předměty. V období registrací podle harmonogramu student si zaregistruje předměty, které by studoval v nadcházejícím semestru. U splněných předmětů se počítají kredity celkové a povinnostní. Zásadní rozdíl mezi SPPSP a tímto IS je to, že předměty lze přiřadit pouze do následujícího semestru a nikde není zřetelně vidět, zda budou navolené kredity stačit pro další pokračování po dokončení tohoto semestru. Dále nelze naplánovat celé studium a pak pouze potvrzovat navržený plán.

*UIS* je Univerzitní informační systém společnosti IS4U využívaný na čtrnácti vysokých školách v Česku a na Slovensku. Tento systém má integrovaný portál studenta k organizaci studia. Obsahuje základní funkce, jako sledování rozvrhu, přihlašování na zkoušky a mimo jiné i jejich předdefinovaný studijní plán (UIS, 2019).

*IS/STAG* je informační systém pro univerzity, vysoké a vyšší odborné školy. Tento systém byl vyvinut Centrem informatizace a výpočetní techniky a je momentálně nasazen na patnácti školách v České republice umožňující si zvolit volitelný modul Editor studijních plánů. Tato aplikace je využívána spíše správci plánu a administrátory. Finální verze v grafické podobě se posléze zobrazuje studentům. Obsahuje funkci na srovnávání a kopírování plánu a vyhledávání určitých částí (IS/STAG, 2017).

*Plánovač studia* je nástroj, který byl vytvořen jako výstup z bakalářské práce. Tato aplikace zjednodušuje kontrolu nad studiem v Informačním systému Masarykovy univerzity (Trnkova, 2017). Studentům je umožněno využívat tuto aplikaci prostřednictvím IS a dovoluje naplánování absorbovaných předmětů do jednotlivých semestrů a tento plán posléze uložit a použít k samotné registraci (Plánovač, 2017).

## 1.2 Výběr prostředí ASP.Net vs .Net

Pro implementaci aplikace v c# se nabízejí dvě možnosti: online nebo off-line verze. Zpracování na webu by bylo dostupné pro všechny bez nutnosti stažení aplikace, ale musela by se řešit autentizace uživatelů. V další řadě je za potřeby zajistit webhosting. Pro webhosting na ASP.Net je zapotřebí IIS rozšíření Windows Serveru na webový server, jenž není studentům na školních serverech momentálně k dispozici k použití a realizaci těchto webů. Nejvhodnější volba je tedy služba od Microsoft Azure, kde lze zajistit „free“ hosting po dobu 30 dnů, poté je za potřeby platit měsíční předplatné. Další možnosti, které jsou označeny jako „free“ mají omezení co do velikosti místa, tak v zastaralosti technologií.

Na druhou stranu formulářová aplikace v .Net nepotřebuje autentizaci, jelikož každý, kdo si aplikaci nainstaluje, je uživatel. Tento instalátor je generován ve VS a není zapotřebí žádných placených služeb k vytvoření nebo provozu. Nevýhoda tohoto je, že každý musí aplikaci stáhnout a nainstalovat. Výhoda je, že generování instalátoru je zdarma a není zapotřebí server k implementaci.

## 1.3 C#

Pro tvorbu aplikace byl vybrán jazyk vytvořený pro technologii .NET C#. Tento vysokoúrovňový objektově orientovaný jazyk založen na jeho předchůdcích Java a C++. Výhoda je vysoká robustnost, modernost a jednoduchost tohoto jazyka (ISO, 2017).

## 1.4 SQL

Jak se píše v odborné knize (Gennick, 2006) o strukturovaném dotazovacím jazyku (dále SQL), jedná se o průmyslový standard pro přístup k relačním databázím, který se využívá k vytváření, čtení, úpravě a mazání (dále CRUD) záznamů z databáze. Dotazy

na čtení lze využít pro vyhledávání jednoho záznamu pro specifické úkoly nebo souhrnu záznamů, které lze nahrát do datových struktur v kódu programu. Lze také využít nástroj Dapper (viz dále v odstavci „Dapper“ 1.5.1) ke zpracování do spojových seznamů.

## **1.5 UML**

Unifikovaný modelovací jazyk (dále UML) druhé verze je standardizované shrnutí grafických notací na popis a vytváření převážně objektově orientovaných systémů (Fowler, 2004). Tento jazyk se snaží sjednotit modelovací zvyky a tradice. Popisem pomocí diagramů UML umožníme čtenářům porozumět, co jsme měli na mysli, neboť se jedná o jakési esperanto v oblasti vývoje programů.

## **1.6 NuGet**

Nástroj NuGet je volně dostupný a od Visual Studio 2012 předinstalován automaticky. Tento balíčkový manager se využívá na tvorbu, sdílení a využívání kódů seskupených do balíčků spravovaných jako knihovny DLL obsahující zkompileovaný kód. Tyto balíčky se přes hostitele, většinou nuget.org, mohou rozšiřovat mezi spotřebitele. Z vybraného balíčku poté lze využít funkci, kterou potřebujeme a o její správu se postará samotný NuGet (Nandwani, 2016).

### **1.6.1 Dapper**

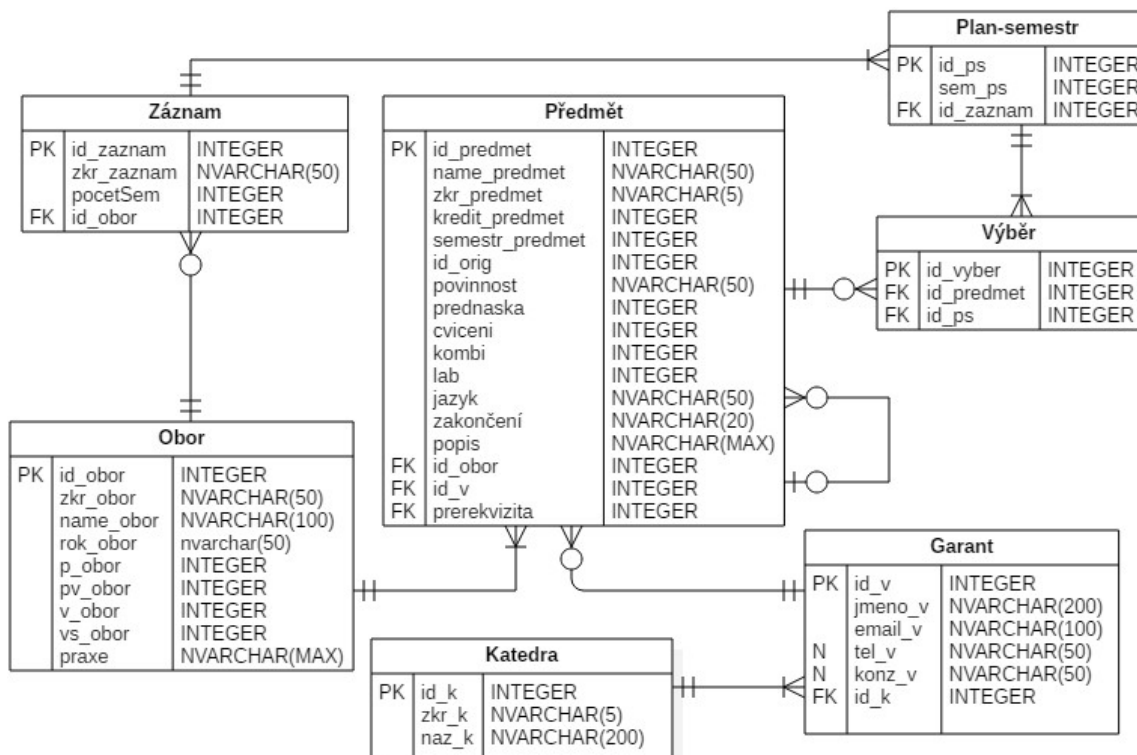
Dapper je Mikro Objektově Orientovaný Mapper, který funguje při běhu kódu mezi klientem a relační databází. Tento mezičlánek je umožněn díky dotazování a vytváření objektů za běhu. Výhoda Dapperu je jeho nenáročnost na výkonnost a zatížení projektu hlavně pro jednodušší data a datové rozhraní (Peipman, 2017). Tento NuGet balíček bude použit v následujících funkcích v kódu na vytváření a naplnění seznamů z dotazů databáze.

## 2 Analýza a návrh řešení

Tato kapitola se bude věnovat analýze a návrhu řešení problému. Podíváme se zde na funkční a datovou část projektu a postupně si rozebereme všechny důležité části a popíšeme funkčnost. Pro uváděné datové modely byla zvolena notace ER-diagramů, pro funkční modely byl zvolen modelovací jazyk UML, speciálně diagram případů užití (use case diagram) a scénáře (sekvenční diagramy) pro jednotlivé případy užití.

### 2.1 Datový model aplikace

Datový model definuje veškerá data, která jsou nutná pro zajištění požadované funkčnosti aplikace. Podrobný popis potřebných dat je uveden v odstavci „Datový slovník“ (2.2). Protože se předpokládá, že data budou uložena a spravována v relační databázi, byl pro jejich popis zvolena varianta relačního ER-diagramu, která je určena k popisu relačních tabulek databáze. Každá tabulka je podrobněji popsána v následujících bodech této kapitoly. Vztahy mezi datovými entitami jsou realizovány mechanismem primárních (PK) a cizích klíčů (FK).



Obrázek 1 - ER Diagram aplikace SPPSP (vlastní)

## 2.2 Datový slovník

V této části je slovně popsána každá z tabulek databáze a její atributy. Každá z tabulka je rozebrána v samostatném oddílu a atributy jsou popsány ve stejném oddílu.

### 2.2.1 Obor

Zvolený obor studia určuje požadavky, jenž musí být splněny k úspěšnému dokončení studia. Obsahuje všechny potřebné informace o oborech na škole. Tato tabulka obsahuje identifikační číslo, zkratku, celý název a počty kreditů potřebných k absolvování.

**Tabulka 1 - Datový slovník pro Obor**

Id_obor	Identifikační číslo oboru, které identifikuje každý obor. Umělý klíč vytvořený pouze pro databázi.
zkr_obor	Zkratka názvu oboru. Každý obor má zkrácený název oboru. Například AI pro Aplikovanou Informatiku.
name_obor	Název oboru. Celý název oboru, například obor Aplikovaná informatika a další.
P_obor	Počet potřebných kreditů z povinných předmětů. Každý obor má daný počet kreditů z povinných předmětů, jež musí student během semestru nasbírat.
Pv_obor	Počet potřebných kreditů z povinně-volitelných předmětů. Každý obor má daný počet kreditů z povinně-volitelných předmětů, jež musí student během semestru nasbírat.
v_obor	Počet potřebných kreditů volitelných předmětů. Každý obor má daný počet kreditů z volitelných předmětů, jež musí student během semestru nasbírat. Tato část může být sdílená a sčítána s kredity za sporty.
vs_obor	Počet potřebných kreditů volitelných sportů.
Praxe	V atributu praxe se ukládají základní informace o praxi na daném oboru.



### 2.2.2 Předmět

Entita Předmět uchovává informace o předmětech. Každý předmět má povinné části jako počet kreditů, název, garant atd. Předměty též uchovávají id ze školní databáze pro možnost zpětné kompatibility dat. Tato tabulka obsahuje popisy atributů, které jsou identifikační číslo názvu a zkratka předmětu, kreditové ohodnocení, doporučený semestr, počty hodin v týdnu daných částí, jazyk, zakončení a originální identifikační číslo.

**Tabulka 2 - Datový slovník pro Předmět**

Id_predmet	Identifikační číslo předmětu, které identifikuje každý předmět.
Zkr_predmet	Zkratka názvu předmětu obsahuje zkrácený název předmětu. Povinný prvek. Každý předmět má nějaký zkrácený název.
Name_predmet	Název předmětu bude obsahovat celkový název předmětu. Povinný atribut předmětu.
Kredit_predmet	Kreditní ohodnocení – každý předmět má nějaké kreditní ohodnocení, které bude zaznamenáno číselně a bude kontrolována nezápornost kreditního ohodnocení.
Id_orig	Originální id předmětu ze školních databází určující každý předmět ve škole. Převážně informativní atribut. Neslouží jako klíč pro daný předmět.
povinnost	Ukazuje, zda je předmět povinný, povinně-volitelný, volitelný anebo sport či speciální. Každý předmět by měl mít toto označení.
Přednáška	Zaznamenává kolik hodin je vyhrazeno týdně na přednášky předmětu. Nemusí být vyplněno.
Cvičení	Zaznamenává kolik hodin je vyhrazeno týdně na cvičení předmětu. Nemusí být vyplněno.
Kombi	Zaznamenává kolik hodin je vyhrazeno pro kombinovanou formu předmětu. Nemusí být vyplněno.

Lab	Zaznamenává kolik hodin je vyhrazeno týdně na laboratoře předmětu. Nemusí být vyplněno.
Jazyk	Označuje, jaký jazyk se využívá k výuce předmětu. Implicitně bude vyučován v češtině.
Zakončení	Udává, jakým stylem je ukončen předmět. Může být zápočet, zkouška nebo zápočet i zkouška.
Id_obor	Cizí klíč určující, pod jaký obor předmět spadá. Předmět má právě jeden obor a obor může mít více předmětů.
popis	Obsahuje celkový popis předmětu. Textová podoba.
Id_vyuc	Identifikační číslo garanta, cizí klíč, který ukazuje na to, jaký vyučující tento předmět garantuje.

### 2.2.3 Katedra

V tabulce Katedra jsou uloženy základní informace o katedrách ve škole jako jsou názvy a zkratky kateder a umělý primární klíč sloužící k identifikaci.

**Tabulka 3 - Datový slovník pro Katedru**

Id_k	Identifikační číslo katedry, které identifikuje každou katedru.
Naz_k	Zde je uchováváno celé znění názvu katedry.
Zkr_k	Zkratka názvu katedry. Každá katedra má zkrácený název. Například KES pro Katedru Ekonomických Studií.

### 2.2.4 Vyučující

Informace o vyučujících na Vysoké škole polytechnické v Jihlavě. Vyučující spadá pod určitou katedru. Vyučující může vyučovat více předmětů. Vyučující může garantovat určitý předmět.

**Tabulka 4 - Datový slovník pro Vyučující**

Id_v	Identifikační číslo zaměstnance, jenž identifikuje každého zaměstnance.
Jmeno_v	Jméno vyučujícího obsahující celé jméno, příjmení a tituly před a za jménem.
Email_v	Emailová adresa vyučujícího. Každý vyučující by měl mít alespoň jednu kontaktní informaci (email nebo telefon).
Tel_v	Telefonní číslo na vyučujícího. Není povinný atribut.
Konz_v	Konzultační hodiny vyučujícího. Není povinný atribut, data se mohou často měnit.
Id_k	Cizí klíč určující katedru, pod kterou vyučující spadá.

### 2.2.5 Záznam

Celkový plán shromažďující všechny semestrální plán uživatele, který může vytvořit více záznamů a mezi nimi přepínat. Není omezeno, kolik záznamů může uživatel vytvořit.

**Tabulka 5 - Datový slovník pro Záznam**

Id_zaznam	Identifikační číslo záznamu, které identifikuje každý záznam.
Zkr_zaznamu	Zvolená Identifikační zkratka/název záznamu plánu. Tato část je pouze informativní a udržuje popis, jakým si uživatel pojmenoval tento záznam.
Id_obor	Cizí klíč, jenž určuje obor vytvořeného plánu.

### 2.2.6 Plán semestr

Plán na každý semestr, jenž spojuje výběry na semestr. Plánů semestru by mělo být podle toho, kolik uživatel vybere semestrů studia. Do této tabulky se ukládají výběry předmětů pro dané semestry.

**Tabulka 6 - Datový slovník pro Plán semestru**

Id_ps	Identifikační číslo Plánu semestru, které identifikuje každý semestrální plán.
Sem_ps	Nastavený semestr pro tento plán. Identifikace, do kterého semestru se bude tento plán zobrazovat.
Id_zaznam	Cizí klíč záznamu, pod který semestrální plán spadá.

### 2.2.7 Výběr

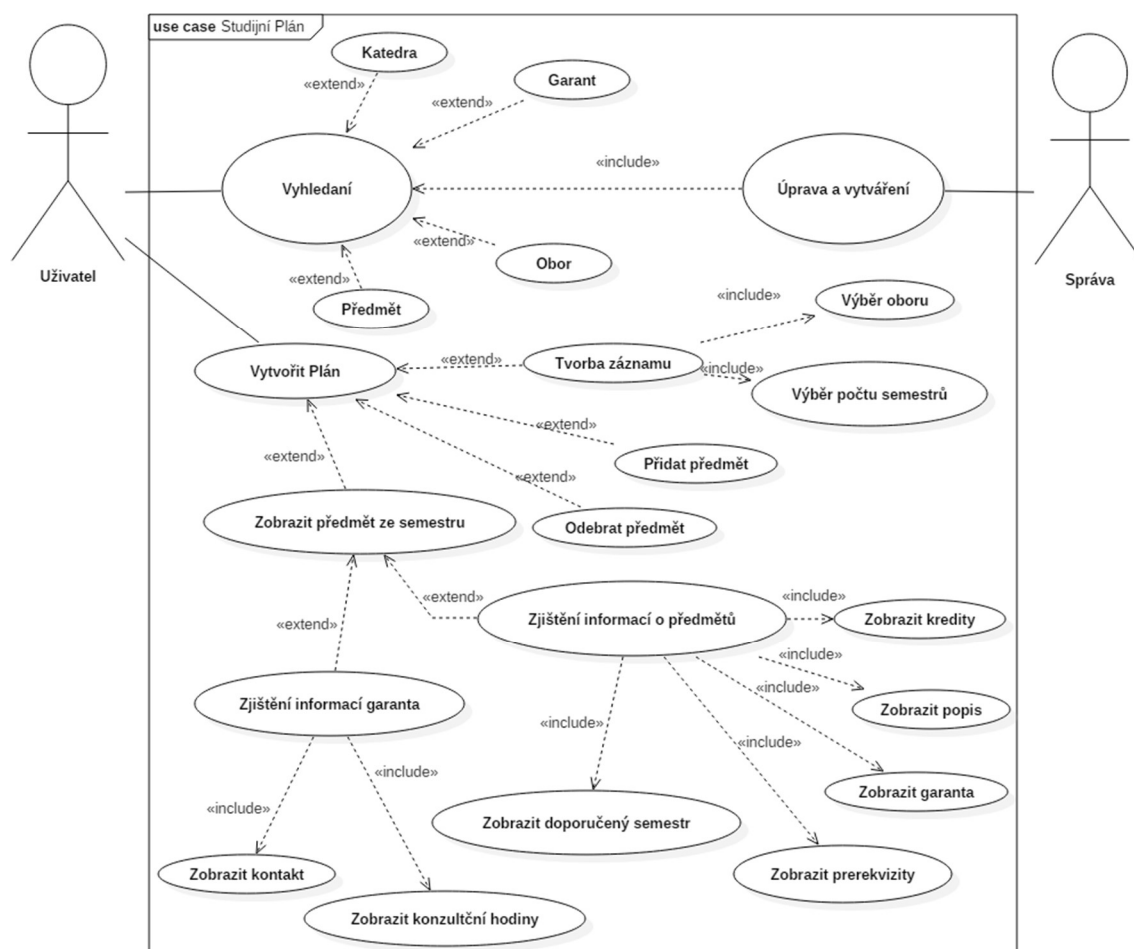
Výběr je vytvořen jako propojovací tabulka mezi semestrálním plánem a předmětem, která reprezentuje vztah mezi předměty a semestry (vztah typu M:N). Tato tabulka umožňuje přidat libovolný počet předmětů do jednoho semestru.

**Tabulka 7 - Datový slovník pro Výběr**

Id_vyber	Identifikační číslo výběru předmětu, které identifikuje každý výběr.
Id_předmět	Cizí klíč, jenž určuje, jaký je vybraný předmět.
Id_ps	Cizí klíč, jenž určuje, do jakého semestrálního plánu výběr přidává.

## 2.3 Funkční model aplikace

Funkční model aplikace shrnuje sadu služeb, které by aplikace měla uživatelům poskytovat. Při analýze potřebných služeb bylo konstatováno, že budou existovat dvě kategorie uživatelů – běžný uživatel a správce. Tyto kategorie jsou v dokumentaci reprezentovány aktéry Uživatel a Správa. Uživatel v roli Uživatel využívá aplikaci po nainstalování a může vytvářet plány, upravovat délku studia, přidávat a odebírat předměty ze semestrů a volit obory. Uživatel v roli Správa je speciální typ uživatele, který může být zvolen v nastavení aplikace. Tato role umožní navíc uživateli přidávat a odebírat předměty, garanty, katedry a obory a různě je upravovat. K dispozici bude také hromadné načítání dat. Tento problém i s hlavičkou požadovaného souboru bude popsán v kapitole o vkládání dat do databáze. Celkový přehled služeb poskytovaných aplikací je uvede na obrázku Use Case model aplikace SPPSP (Obrázek 2). K dokumentaci je použit tzv. Use Case Model UML (model jednání).



**Obrázek 2 - Use Case model aplikace SPPSP (vlastní)**

### 2.3.1 Případy užití pro aktéra v roli Uživatel

Tato část je zaměřena na případy užití pro aktéra v roli Uživatel

- I. Vytvořit plán – Umožní uživateli vytvořit studijní plán, jenž bude ukládán a bude možnost vytvořit více plánů najednou.
  - a. Tvorba záznamu – Tato funkce umožní uživateli vytvořit záznam plánu. Tento záznam obsahuje zvolený obor a semestry. Vytvořené záznamy kromě tvorby upravovat a mazat.
    - i. Vybrat počet semestrů – Tento údaj symbolizuje, kolik je zapotřebí zobrazit semestrů, které bude uživatel využívat pro přidávání a odebírání semestrů dle výběru.
    - ii. Vybrat obor – Nastavení požadovaného oboru studia, jenž je zapotřebí vybrat před vytvářením plánu, aby bylo možno filtrovat předměty podle oboru.
  - b. Přidat předmět do semestru – Přidání předmětu do plánu a požadovaného semestru. Uživatel si bude moci vybrat předmět z nabídky předmětů jeho zvoleného oboru a daného období (letní nebo zimní semestr).
  - c. Odebrat předmět ze semestru – Odebrání předmětu do plánu a požadovaného semestru. Pomocí kliknutí na daný předmět a zvolení možnosti smazání bude tento předmět odstraněn z předmětu a přibude opět do nabídky předmětů, které jdou zapsat do semestru.
- II. Vyhledat předmět – Umožní vyhledat předmět po zadání jeho názvu, povinnosti předmětu, počtu kreditů, doporučených semestrů.
- III. Zjistit informace o předmětu – Po výběru předmětu bude možnost zobrazit informace o předmětu, jako jsou zkratka předmětu, doporučený semestr studia, prerekvizita.
  - a. Zobrazit kredity – Zobrazení kreditového ohodnocení zvoleného předmětu.
  - b. Zobrazit popisy – Zobrazení popisu vybraného předmětu, jako jsou: cíle, požadavky atd.

- c. Zobrazit garanta – Zobrazení garanta daného předmětu.
  - d. Zobrazit prerekvizity – Zobrazení potřebných prerekvizity předmětu.
  - e. Zobrazit doporučený semestr – Zobrazení doporučeného semestru ve studiu požadovaného předmětu.
- IV. Zobrazit informace o garantovi předmětu – Možnost zobrazit informace o garantovi jako: místo kanceláře aj.
- a. Zobrazit kontakt – Zobrazení kontaktních informací daného garanta jako jsou telefon a email.
  - b. Zobrazit konzultační hodiny – Zobrazení konzultačních hodin a dne konzultací daného garanta.

### **2.3.2 Případy užití pro aktéra v roli Správa**

Tato část je zaměřena na případy užití pro aktéra v roli Správa. Správa je sada funkcí, jenž rozšiřuje vyhledávání a umožňuje CRUD operace pro položky kateder, oborů, předmětů a garantů.

- I. Úprava a vytváření – podle vybrané skupiny lze vytvořit položku do této skupiny a po vyhledání položky libovolné skupiny jde tuto položku upravit nebo smazat.

## **2.4 Scénáře případů užití aktéra v roli Správa**

V této části se podíváme na scénáře případů užití pro jednotlivé případy aktéra správce. Jedná se o správu garantů, správu oborů a správu předmětů. Scénáře jsou rozepsány slovně a doplněny tzv. sekvenčními diagramy UML.

### 2.4.1 Správa garantů správcem:

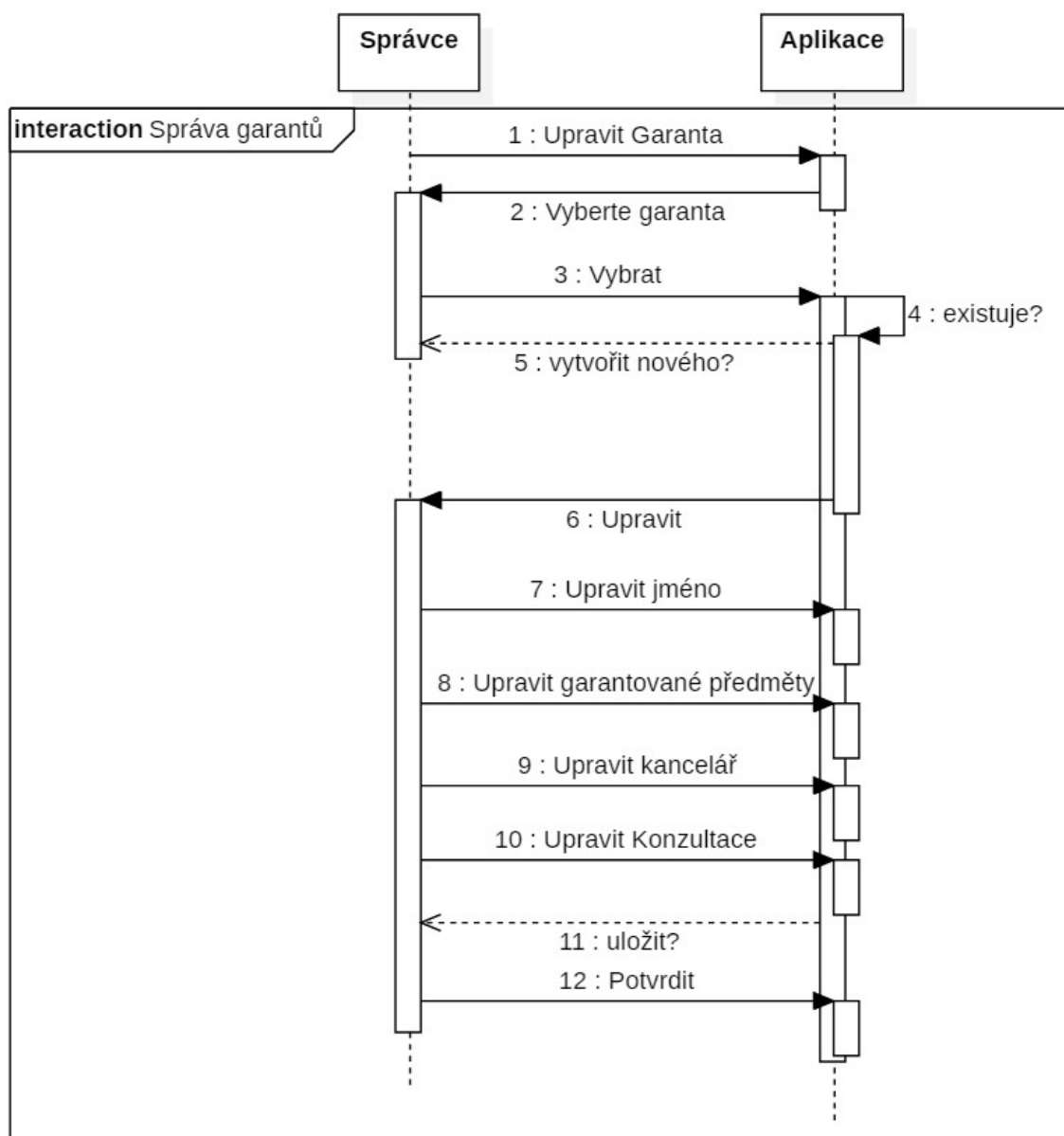
Tabulka 8 - Scénář – správa garantů

<b>Případ užití:</b> Správa garantů	
<b>Stručný popis:</b> Bude moci upravit veškeré informace po vybrání garanta nebo zadání jména upravit údaje o garantech jako jméno, kontakty, místo kanceláře.	
<b>Hlavní aktér:</b> Správce	
<b>Hlavní kroky:</b> <ol style="list-style-type: none"> <li>1. Výběr garanta                             <ol style="list-style-type: none"> <li>a) Podle jména</li> <li>b) Ze seznamu garantů</li> </ol> </li> <li>2. Jestliže(neexistuje)                             <ol style="list-style-type: none"> <li>a) Vytvořit nového garanta</li> <li>b) Zadat jiné jméno</li> <li>c) konec</li> </ol> </li> <li>3. Úprava                             <ol style="list-style-type: none"> <li>a) Jméno</li> <li>b) Příjmení</li> <li>c) Konzultační hodiny</li> <li>d) Katedra</li> <li>e) Garantovaný Předmět</li> <li>f) Smazání garanta</li> </ol> </li> <li>4. Konec</li> </ol>	
<b>Podmínky ukončení:</b> Potvrzení oprav/Zrušení správy	
<b>Alternativní kroky:</b> -	



## 2.4.2 Model správy garanta

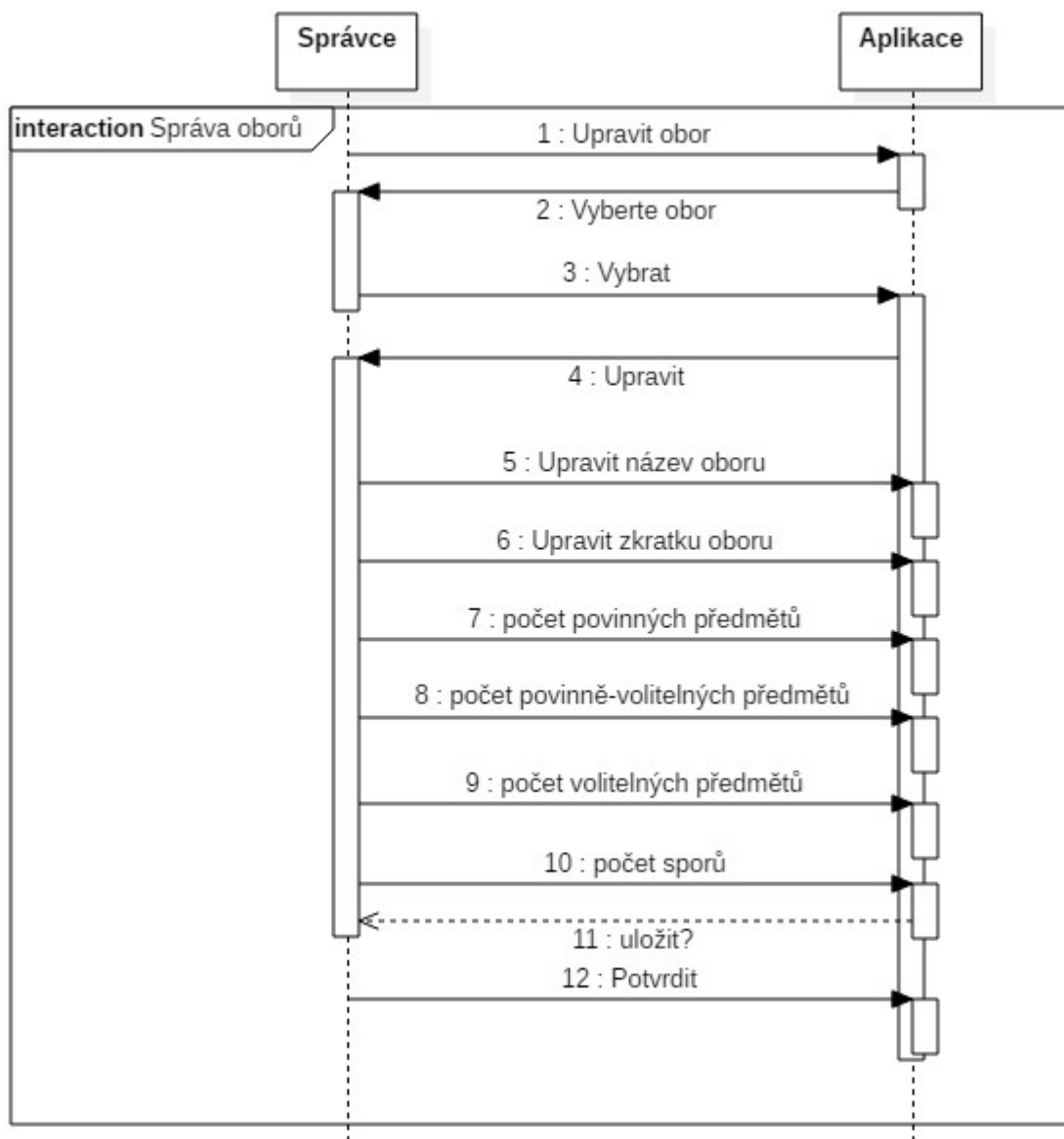
Tento model popisuje průběh funkce, který je popsán ve scénáři v předešlé kapitole.



Obrázek 3 - Scénář – správa garantů (vlastní)

### 2.4.3 Model správy oborů

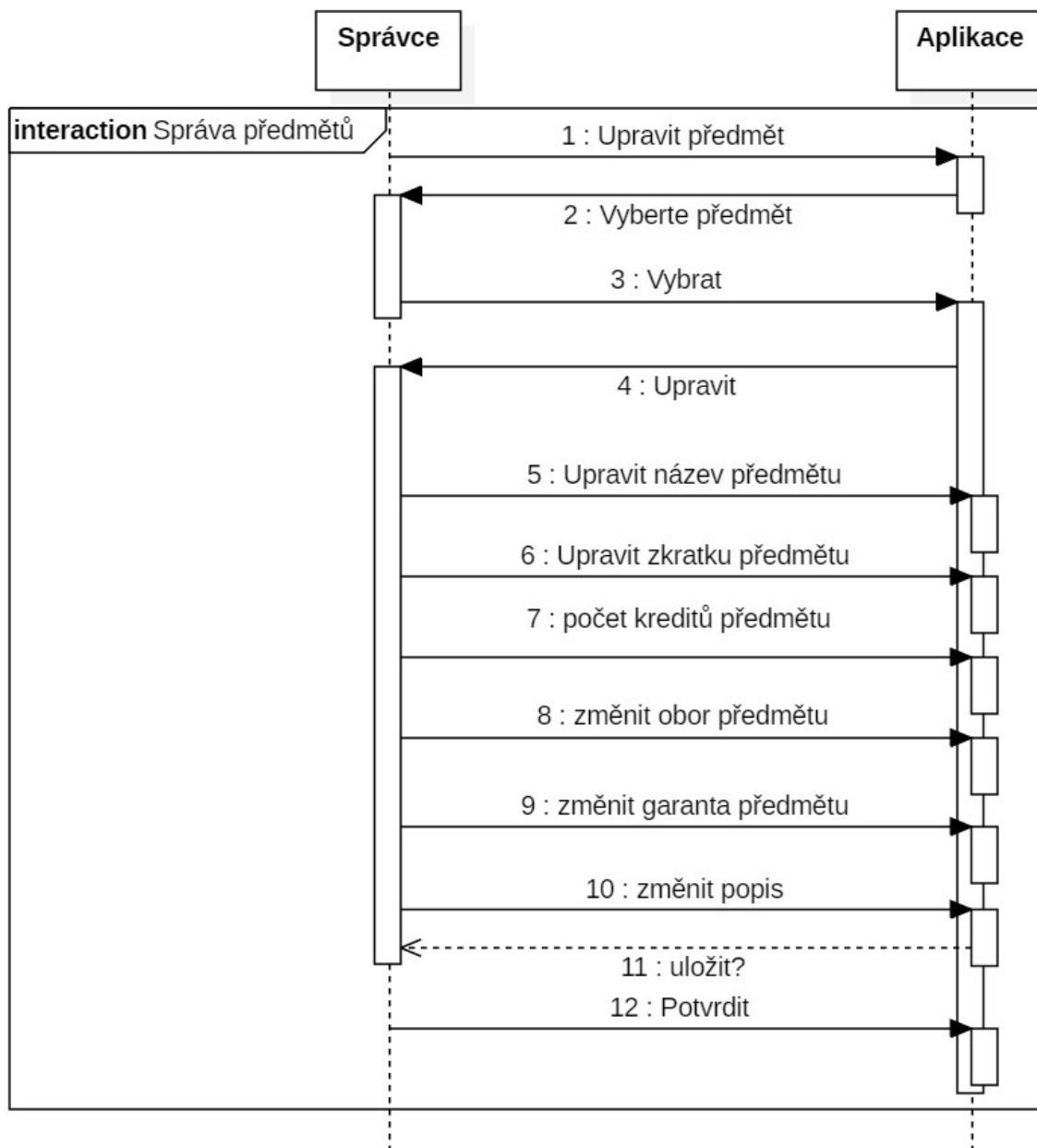
Model popisuje průběh kroků funkce, jež vykonává při úpravě oboru v programu.



Obrázek 4 - Scénář – správa oborů (vlastní)

#### 2.4.4 Model správy předmětů

Model popisující kroky k úspěšné úpravě předmětu v aplikaci. Pakliže funkce dojde do konce dojde k uložení změn údajů předmětu.



Obrázek 5 - Scénář – správa předmětů (vlastní)

### 3 Popis implementace

V této části je rozebrána implementace aplikace. Je zde využito poznatků z analytické části práce. Obsahuje rovněž popis naplnění databáze daty s předměty a vším potřebným, implementace funkcí a realizace tvorby samotného plánu.

#### 3.1 Tvorba plánu

Pro úplné využití aplikace je zapotřebí vytvořit záznam tohoto plánu. Jak je popsáno v kapitole (2.2.5) tyto záznamy se skládají z počtu semestrů, názvu a oboru. Tyto položky jsou povinné a na tuto povinnost je upozorněno pomocí komponenty error provider, která zobrazí červený vykřičník i informací o nutnosti vyplnění požadovaných položek, jestliže položky nejsou vyplněny záznam nebude vytvořen.

Obrázek 6 - Tvorba záznamu (vlastní)

Jakmile jsou položky vyplněny, tak po stisknutí OK se vrátíme do původního okna, kde se zavolá funkce, jenž vložená data, která jsou přenesena z okna záznamů pomocí přepravky Záznam, uloží do databáze. Po založení záznamu se podle počtu semestru vytvoří položky do tabulky plán semestr, využívané pro naplnění samotných plánů.

**Algoritmus:** funkce **NovyZaznam**

**Vstup:** Naplněná přepravka s daty záznamu, položka mazat se udává pouze u úpravy záznamu

**Výstup:** Přidána data do tabulky Záznam v databázi programu

**Postup:** Aplikace zkontroluje, zda se nemá před vytvořením smazat předešlý záznam, tato část je využívána při úpravě, když se změní obor záznamu, tak dojde ke smazání původního a založení nového plánu. Po kontrole mazání se zavolá funkce na vložení záznamu do databáze. Po vytvoření záznamu se připraví jednotlivé plány na každý semestr až do počtu semestru, zadaného při vytváření.

```
public void NovyZaznam(Zaznam zaz, int mazat = 0)
{
    try
    {
        DataCrud x = new DataCrud();
        if (mazat == 1)
            x.DeleteZaznam(zaz.Id_zaznam); //smazání starého
        x.InsertZaznam(zaz); //založení nového
        for (int i = 1; i <= zaz.PocetSem; i++)
            x.InsertPS(zaz.Zkr_zaznam, i);
        MessageBox.Show(Properties.Resources.Vlozeno_MESSAGE,
            Properties.Resources.Vlozeno_TITLE,
            MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(Properties.Resources.NelzeUlozit_MESSAGE + ex,
            Properties.Resources.Chyba_TITLE,
            MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Obrázek 7 - Tvorba nového záznamu (vlastní)

Záznam může mít pouze jeden obor a pakliže se tento obor v průběhu používání změní, vymažou se i veškeré zapsané předměty. Informace o těchto oborech jsou zobrazované ve vlastní komponentě detail oboru. Při změně počtu semestrů se při snížení počtu odstraní nejvyšší semestr a při navýšení přidá položka.

**Algoritmus:** na změnu počtu semestrů v záznamu

**Vstup:** Přepravka s daty záznamu

**Výstup:** Upraven záznam a plány semestrů podle nového počtu semestrů

**Postup:** Funkce zjistí původní počet semestrů zaznamenaný v databázi a poté provede úpravu záznamu podle zadaných dat z přepravky. Jestliže je počet nových semestrů menší, než původní počet tak se pomocí cyklu odstraňují položky z databáze až do doby, než jsou tyto počty semestrů stejné. Naopak pakliže je počet nových semestrů větší než původní počet, tak se přidávají další plány na semestr až do doby, než tyto počty jsou stejné.

```
id_z = Convert.ToInt32(Zaznam.Id); //převod identifikačního čísla na INT
if (oborNew == oborOld) //pakliže se obor nezmění
{
    PocSem = Zaznam.Semestr; //zjistí počet nových semestrů
    int stare = a.GetZaznamSemestr(id_z); //zjištění stávajícího počtu
    //úprava záznamu
    x.UpdateZaznam(id_z,
                    Zaznam.Zkr,
                    PocSem);
    if (stare > PocSem) //pakliže se počet semestrů sníží
        while (stare > PocSem)
            //mažeme nejvyšší semestry do doby než počty nejsou stejné
            {
                x.DeletePlanSemestr(id_z, stare);
                stare--;
            }
    if (stare < PocSem) //když se počet semestrů zvýší
        while (stare < PocSem)
            //přidáváme plán semestru od původní položky až do nové
            {
                x.InsertPS(Zaznam.Zkr, stare);
                stare++;
            }
}
```

Obrázek 8 - Úprava záznamu a změna počtu semestru (vlastní)

Pro vytváření a úpravu se využívá formulářové okno. Toto okno lze vyvolat pomocí tlačítka tvorby nového nebo úpravy vybraného anebo v podpoložce vytvořit, upravit záznam v položce Soubor.

### 3.2 Přidávání předmětů do semestru

Pro přidání předmětů je funkce potřebná k výběru předmětů do daných semestrů a je zapotřebí pro kompletní tvorbu plánu. Tato funkce je realizovaná přes formulářové okno Přidávání. Toto okno bere údaje zadaného semestru, vybraného oboru daného záznamu plánu, ze kterého se načítají sudé, liché předměty a sporty. Tyto seznamy jsou aktualizované při každém provedení přidání a při načtení plánu. U sportů se kontroluje,

zda nebyl zapsán pouze v daný semestr, jelikož lze sporty absolvovat několikrát a ostatní předměty podle toho, zda doporučený, a tedy i vyučovaný semestr je sudý nebo lichý.

**Algoritmus:** pro přidávání předmětů

**Vstup:** Vybraný záznam plánů, nastavený požadovaný semestr

**Výstup:** Přidána sada předmětů do okna seznamu předmětů daného semestru

**Postup:** Funkce zkontroluje, zda je vybraný nějaký záznam. Poté kontroluje, zda zadaný semestr je lichý (obrázek 9) nebo sudý. Po výběru tuto sadu lichý nebo sudých dostupných předmětů podle oboru v záznamu načte do okna záznamu v novém formuláři na přidávání předmětů a přidá k nim sporty, které nebyli již absolvované v zadaném semestru. Po kliknutí na tlačítko ok se přidají všechny vybrané předměty pošlou do funkce vkládání do tabulky Výběr v databázi a obnoví se okno seznamu daného semestru s přidávanými předměty.

```
public List<Predmet> GetPredmetFullLichyVyberKromeSportu(int id_obor, int izaz)
{
    using (IDbConnection connection = new SqlConnection(ConnectionString))
    {
        List<Predmet> vystup = connection.Query<Predmet>(<
            $"Select [predmet].* from predmet where id_obor='{id_obor}' " +
            $"AND (semestr_predmet=1 OR semestr_predmet=3 OR semestr_predmet=5) " +
            $"AND [predmet].id_predmet " +
            $"NOT IN(" +
            $"SELECT [predmet].id_predmet " +
            $"FROM ([predmet] JOIN [vyber] ON [predmet].id_predmet= [vyber].id_predmet) " +
            $"JOIN [plansemestr] ON [plansemestr].id_ps = [vyber].id_ps " +
            $"WHERE [plansemestr].id_zaznam='{izaz}' AND [predmet].id_predmet " +
            $"NOT IN(" +
            $"SELECT [predmet].id_predmet FROM ([predmet] " +
            $"JOIN [vyber] ON [predmet].id_predmet= [vyber].id_predmet) " +
            $"JOIN [plansemestr] ON [plansemestr].id_ps = [vyber].id_ps " +
            $"WHERE [plansemestr].id_zaznam='{izaz}' AND [predmet].semestr_predmet=0 ))" +
            $"ORDER BY semestr_predmet,povinnost").ToList();
        return vystup;
    }
}
```

**Obrázek 9 - Načtení lichých předmětů do seznamu (vlastní)**

Pro výběr předmětů slouží dva seznamy, jenž na jedné straně obsahují seznam s předměty a na druhé jsou předměty vybrané. Při kliknutí na předmět se zobrazí data daného předmětu ve vlastní komponentě detailu předmětu. Po potvrzení výběru se data zapíší do databáze a obnoví seznam na hlavním okně.

### 3.3 Výpočet kreditů

Pro výpočet počtu povinných, povinně volitelných, volitelných předmětů a sportů kreditů a celkový počet těchto kreditů je využívána pomocná třída *Kredity*, která uchovává tyto hodnoty. Při naplnění okna seznamu se zavolá funkce, která prochází atributy každého předmětu v seznamu a ty podle povinnosti roztrídí do daných skupin a počítá jejich sumy. Toto se opakuje pro všechny semestry. Po projití všech semestrů se tyto hodnoty přiřadí do stanovených oken v hlavním okně a barevně označí, zda jsou tyto hodnoty dostačující nebo nikoliv.

**Algoritmus:** pro *VypoctiPovinnostiKredity*

**Vstup:** Spojový seznam předmětů a záznam třídy *Kredit*

**Výstup:** Přičtené hodnoty do daných atributů do záznamu třídy *Kredit*

**Postup:** Funkce prochází každý předmět v seznamu, kde se kontroluje atribut *Povinnost*. Když se narazí při porovnávání na tuto povinnost, tak se této položky se vezme atribut *kredit* a ten se přičte k dané položce v *Kredit*. Další kontrola se již neprovádí a funkce je posunuta na další položku předmětu. Toto se opakuje do konce seznamu.

### 3.4 Třída na práci s rozbalovacími seznamy a seznamy

Třída *Filling* je pomocná třída využívaná na práci s komponentami rozbalovací seznam (*ComboBox*) a seznam (*Listbox*). Hlavním úkolem je zjednodušit vkládání a práci s daty v těchto komponentách. Tyto funkce využívají generické datové typy. Lze využít pro různé vstupní data podobných struktur, v tomto případě třídy *Přepravek* pro jednotlivé tabulky. Díky čemuž lze jednu funkci využít pro několik různých dat. Aby data byla zobrazena v požadovaných formátech je u každé *Přepravky* přetížena funkce *ToString()*, která upravuje výstup těchto *Přepravek*.



```

public void NaplnComboBox<T>(ComboBox cb, List<T> li)
{
    //vyčištění dat v comboboxu
    cb.DataSource = null;
    cb.Items.Clear();
    //pro každou položku v listu se přidá jako item
    try
    {
        foreach (T temp in li)
            cb.Items.Add(temp);
    }
    catch (Exception ex)
    {
        MessageBox.Show(Properties.Resources.Chyba_TITLE + ": " + ex);
    }
}

```

Obrázek 10 - Vkládání dat do komponenty rozbalovací seznam (vlastní)

Na vyhledávání je využita funkce (obr. 11), která kontroluje, zda zadaný text k vyhledávání v daném políčku k vyhledávání se nachází v databázi. Při vyhledávání se také po nalezení prvku přepočítává délka záznamu, aby všechny záznamy byly čitelné a nemuselo se využívat horizontálních posuvných lišt v okně komponenty.

```

public void NajdiVComboBoxu<T>(ComboBox x, List<T> temp)
{
    //uvolnění ComboBoxu
    x.Items.Clear();
    //nastavení defaultní šířky
    int sirka = 1;
    foreach (T o in temp)
        //prohledávání všech prvků v listu a porovnávání se zadaným textem
        {
            if (o.ToString().IndexOf(x.Text, Comp) >= 0)
                //pakliže se najde prvek obsahující x.Text, tak se přidá do ComboBoxu
                {
                    x.Items.Add(o);
                    //přepočítání šířky rozbalené komponenty
                    int tmp = TextRenderer.MeasureText(o.ToString(), x.Font).Width;
                    if (sirka < tmp)
                        sirka = tmp;
                }
        }
    x.DropDownWidth = sirka;
}

```

Obrázek 11 - Vyhledávání v komponentě rozbalovací seznam (vlastní)

### 3.5 Vyhledávání a úprava

Pro lepší přehled a informovanost o potřebných položkách slouží okno vyhledávání, kde si uživatel může vybrat požadovanou kategorii (Obor, Katedra, Garant, Předmět) a tuto

položku vyhledat podle zvolení bližších parametrů jako například je katedra u garanta a také možnost napsat část názvu nebo jména do vyhledávacího políčka (obrázek 11), které vyfiltruje všechny záznamy se znaky obsahujícími v políčku. Kromě vyhledávání lze povolit v tomto okně i režim správce, jenž přidá možnosti odebírání a upravování vyhledaných nebo vytváření nových záznamů. V tomto módu je umožněno také hromadné načítání popsané v kapitolách (3.5–3.6).

### 3.6 Hromadné plnění dat do databáze

Ze získaných dat ze školní databáze je za potřeby zpracovat textové soubory a posléze je zapsat do příslušných tabulek v databázi. Z důvodu zkrácení anotací při výpisu předmětů v ostatních daty je za potřeby vytvořit přídatný soubor, ve kterém se objevují tyto anotace v plném znění. Pro zpracování prvního souboru se používá čtení po řádku, jelikož každý předmět je zapsán na řádek a jednotlivé informace odděleny středníky. Tyto data se po přečtení přidělí do patřičných funkcí a zapíše do tabulky. Takto se projde celý soubor, jenž je načten pomocí komponenty umožňující nastavit cestu k tomuto textovému souboru. Tuto funkci lze využít na přidání dalších nových roků a předmětů za předpokladu dodržení požadované hlavičky.

Vzorek ze souboru má tvar:

*1610; Semestrální projekt; xSP; 4; ZA; P-13/14; 5; 35; f; P; Povinný předmět; PaedDr. František Smrčka, Ph.D.; Katedra technických studií; KTS;; smrcka@vspj.cz; 0;;0;;1; Česky; Czech; Semestral Project; ; t; Smrčka*

Kde jednotlivá data jsou:

ID; název; zkratka; kredity; zakončení; Rok Předmětu; doporučený Semestr; -; -; zkratka povinnosti; Povinnost; garant; katedra; zkratka katedra; -; email garanta; hodiny Přednáška; hodiny Cvičení; tutoriál; kombi CV; laboratoře; typ Předmětu; jazyk; jazyk[anglicky]; název[anglicky]; -; -;

Tento problém je řešen pomocí vytvořených funkcí zahrnuté ve třídě Načítání dat, kde se pro vložení těchto dat do databáze využívá vytvořená třída Data CRUD. Jelikož je v tomto souboru obsažena potřebná data pro tabulky obor, katedra, garant a předmět, tak se tento soubor čte po řádku a příslušné zápisy se přidávají do patřičných funkcí, které kontrolují, zda data nejsou již obsažena v databázi či nikoliv. Po tomto vyhodnocení se

tyto položky buď zahodí, nebo vloží. Jednotlivé záznamy je potřeba ukládat v pořadí katedra, obor, garant, předmět, protože garant obsahuje odkaz na cizí klíč na identifikační číslo příslušné katedry a předmět obsahuje jak identifikační číslo oboru, tak i garanta. Proto lze jediné zaměnit pozice nahrávání dat do oborů a kateder, jelikož by po přemístění mohlo docházet k chybám na volání záznamů, které ještě nejsou naplněny do databáze.

#### **Algoritmus pro funkci `Proved`**

**Vstup:** Cesta k souboru, kde jsou uložena data jako text ve výše uvedeném formátu (path)

**Výstup:** Data převedená do tabulek Garant, Předmět, Obor, Katedra v databázi.

**Postup:** Funkce čte data z textového souboru po řádcích. Data každého řádku jsou vyextrahována do pomocného pole, jenž ulehčuje přiřazení jednotlivých dat do požadovaných funkcí Insert na vkládání dat.

Vlastní kód funkce "Proved" je pak uveden v příloze A1.

#### **Algoritmus pro funkci `InsertKat`**

**Vstup:** Zkratka a název katedry

**Výstup:** Data uložená do databáze, pakliže tyto údaje již nejsou obsažena

**Postup:** Funkce zkontroluje, zda se nejedná o duplicitní data, jestliže tyto data jsou unikátní, tak se otevře propojení s databází a provede se SQL dotaz na vložení dat.

#### **Algoritmus pro funkci `InsertPředmětHromada`**

**Vstup:** Přepřavka s daty předmětu

**Výstup:** Data vložena do databáze do tabulky Předmět

**Postup:** Funkce zkontroluje, zda předmět neexistuje a pakliže se nejedná o duplikát nastane otevření připojení do databáze, jenž přiřadí data z této přepřavky do proměnných SQL dotazu na vkládání. Jestli nastane problém v uložení dojde k vytvoření varovné zprávy, která se zobrazí na monitoru.

Funkce na vkládání oborů a garantů fungují na stejném principu jako tato funkce a vlastní kód je k dispozici v příloze A2.

**Výsledkem je řádek v tabulkách:**

Katedra, kde:

název: *Katedra technických studií*

zkratka: *KTS*

Obor, kde:

rok obor: *P-13/14*

Garant, kde:

jméno: *PaedDr. František Smrčka, Ph.D.*

email: *smrcka@vspj.cz*

katedra: identifikační číslo pro *Katedra technických studií*

Předmět, kde:

název: *Semestrální projekt*

zkratka: *xSP*

kredity: *4*

orig ID: *1610*

zakončení: *ZA*

id obor: identifikační číslo pro obor *P-13/14*

doporučeny Semestr: *5*

zkratka povinnosti: *P*

povinnost: *Povinný předmět*

garant: identifikační číslo pro garanta *PaedDr. František Smrčka, Ph.D.*

katedra: identifikační číslo pro *Katedra technických studií*

hodiny Přednáška: *0*

hodiny Cvičení: *0*

tutoriál: *0*

kombi CV: *1*

laboratoře: *1*

### **3.7 Načtení popisů k předmětům do databáze**

Pro načtení popisu k předmětu je potřeba využít dalšího souboru, jelikož zápisy jsou více řádkové a komplikuje to situaci načítání po řádcích. Z toho důvodu je načítání řešeno přes nahrání celého textového souboru do řetězce a poté rozděleno po oddělovacím

znaménku‘;’ a poté naplněno v cyklu podle počtu znaků. Po zpracování se vloží text popis předmětu k patřičnému předmětu podle zbylých položek zpracované části souboru. Tento předmět je vyhledán podle dat zadaný při vytváření přepravy, kde se podle názvu a roku označení vyhledá identifikační číslo předmětu. Pro tuto problematiku je vytvořena funkce ProvedPopis v kódu aplikace.

Vzorek tohoto souboru má tvar:

*1610; Semestrální projekt; xSP; Cílem Semestrálního projektu je vytvořit pro studenta podmínky pro započítání reálné práce na zpracovávání své bakalářské práce (BP). Student individuálně a formou konzultací se svým vedoucím BP řeší úkoly a problémy související se svou BP. Garant předmětu stanovuje obecné podmínky na realizaci předmětu. Semestrální projekt lze zpracovávat i na jiné téma, než je téma budoucí bakalářské práce, tento způsob však není preferován.; P-13/14; 35*

Kde jednotlivé části jsou:

ID předmětu; název předmětu; zkratka předmětu; text popisu předmětu; obor; -

#### **Algoritmus pro funkci ProvedPopis**

**Vstup:** Cesta k souboru, kde jsou uložena data jako text ve výše uvedeném formátu (path)

**Výstup:** Přidání popisu k předmětu podle dat na řádku do tabulky Předmět v databázi.

**Postup:** Funkce zapíše celý soubor do pomocného spojovaného seznamu řetězců, kde každá položka je vložena po předem definovaném rozdělovacím znaku (zde „~“). Poté se tento seznam projde pomocí for cyklu, jenž se posouvá podle počtu položek na záznam. Tato část se vloží do přepravy předmětů a pošle jako vstupní hodnota pro funkci InsertPopis.

#### **Algoritmus vkládání dat InsertPopis**

**Vstup:** Přeprava s daty předmětu z funkce ProvedPopis

**Výstup:** Vložení popisu k předmětu

**Postup:** Funkce vytvoří připojení k databázi, kde se položky přiřadí do patřičných míst v SQL dotazu na vkládání dat.

Vlastní kód funkcí na hromadné načítání popisů je k dispozici v příloze A3.

### 3.8 Návod k použití

Veškerá okna lze uzavřít předčasně, a to buď křížkem nebo červeným tlačítkem *Zavřít* či *Cancel*. Všechny změny se ukládají po stisknutí zeleného tlačítka *OK*. Při prvním a každém dalším, kdy není vytvořen plán, spuštění aplikace vyskočí informativní okno o absenci plánu a dotazu o přesměrování k jeho vytvoření. Po potvrzení se otevře okno tvorby plánu, jenž vytvoří po vyplnění oboru, počtu semestrů a označení prázdný záznam plánu a vygeneruje všechny pod plány na semestry. Při potvrzení, zavření nebo nepotvrzení vytvoření se otevře hlavní okno aplikace.

V tomto oknu je hlavní nabídka s možnostmi tvorby a úpravy záznamů plánů, vyhledávání a nápovědy. Pod touto lištou je rozbalovací okno, ve kterém lze přepínat mezi různými plány. Na pravé straně od plánů se nachází numerické počítadlo, znázorňující vybraný semestr, jež je využíváno pro přidávání předmětů do daného semestru. Pro přidání stačí kliknout na tlačítko *Přidat*. Pod těmito komponentami je několik políček pro samotné semestry s počítadly kreditů. Tyto okna se naplní vybranými předměty zvolenými při přidávání. Na každý předmět lze kliknout a tím tento předmět vybrat. Po vybrání se zobrazí dodatečné informace k předmětu a možnosti na zobrazení informací o garantovi anebo možnost smazání tohoto předmětu z daného semestru. V pravém dolním rohu se pak zobrazují celkové kredity a další kreditní požadavky určené vybraným oborem. Veškeré kreditní ukazatele mají nastaveny barevné zobrazení, kde zelená znamená dostatek kreditů, a naopak červená nedostatek.

Po vybrání semestru a kliknutí na tlačítko *Přidej* se dostaneme do nového Okna přidávání, kde na levé straně se ukazují informace o zvoleném předmětu a na pravé straně jsou dvě okna mezi nimiž jsou šipky určující kam se předmět má přesunout. V pravém okně jsou všechny dostupné předměty pro daný semestr a na levé straně pak ty, které chceme zvolit do našeho plánu. Pod tímto oknem je také počítadlo kreditů, které slouží převážně jako informativní údaj o hodnotě přidávaných předmětů. Po kliknutí na tlačítko *OK* se tyto předměty se přidají do daného semestru a zobrazí na hlavním okně v dané položce.

Pakliže chceme přistoupit k módu správce nebo vyhledat nějaké informace o katedrách, oborech, garantech či předmětech, tak na hlavním okně v horním panelu vybereme možnost vyhledávání, která nás přesměruje na okno vyhledávání a úprav záznamů v aplikaci. Pro vyhledávání slouží výběr z položek zobrazený pomocí zatahovacích oken s názvy obor, katedra, garant, předmět. Po zvolení se v okně zobrazí rozbalovací okno

s názvy dané kategorie se všemi záznamy. Tyto záznamy lze filtrovat pomocí pomocných rozbalovacích oken, jsou-li k dispozici anebo napsáním části názvu do okna. Po rozbalení tohoto okna se zobrazí jenom záznamy obsahující zadané znaky. V horní části tohoto okna je také tlačítko na povolení modu správa, kde po stisknutí vyskočí varovná zpráva o možnosti smazání kritických částí, jenž může dospět až ke zničení a smazání již vytvořených plánů. Po potvrzení se zobrazí tlačítka na tvorbu, úpravu a mazání u okna vyhledaných položek. Dále se objeví také v horní liště možnosti hromadného načtení dat pro načítání dat ve větším množství ze souboru. Pro tvorbu a úpravu jednotlivých položek z kategorií se spustí vytvořená okna s políčky pro jednotlivé položky záznamu. Veškeré důležité položky je potřeba vyplnit, aby namísto chybné hlášky po potvrzení změn či potvrzení vytvoření se tento prvek uložil do databáze. Všechna okna aplikace jsou vyobrazeny v příloze B, kde jsou i jejich krátké popisy.

### **3.9 Podmínky spuštění**

Pro správnou funkčnost aplikace je za potřeby podporu lokálních databází a technologii Microsoft .NET Framework 4.6.1 nebo nainstalované VS 2015 nebo novější, kde je nainstalované rozšíření pro aplikace .NET technologie a pro práci se SQL databází.

### **3.10 Testování**

Testování je jedna z nejdůležitějších částí při tvorbě programu. Testování probíhá v logickém pořadí od tvorby samotného plánu do vyhledávání až po CRUD operace.

#### **Testování tvorby záznamu**

Při pokusu o uložení záznamu bez názvu nebo oboru se rozbliká červený vykřičník s informací o nevyplnění. Při úpravě záznamu se při změně objeví tlačítko „?“ informující o smazání dat z plánu. Změny počtu semestrů podle předpokladu správně přidávají či odstraňují okna seznamů na hlavním okně. Pomocí validace dat funkce nevstupují do chybových stavů.

#### **Testování přidávání předmětů do plánu**

Výběr semestru je omezen na počet podle vybraného záznamu plánu, tudíž lze přiřadit pouze od vygenerovaných seznamů. Při přidávání lze přidat libovolný počet předmětů.

Po přidání a odebrání se správně aktualizují seznamy tak, že nelze zapsat tentýž předmět dvakrát v plánu s výjimkou sporů, které lze zapsat do různých semestrů i když byl zapsán, nikoliv však do stejného semestru, kde je již zapsán. Při nevybrání záznamu plánu, či smazání všech záznamů nejde přidat do ničeho i když se může stát, že okna seznamů plánu zůstanou na hlavním okně zobrazena. Při stisknutí na tlačítko Zavřít nebo křížek se okno uzavře beze změn v plánu.

### **Testování přepočítávání a kontroly kreditů**

Kredity správně berou kreditní limity z oboru z vybraného záznamu plánu po přidání nebo dobrání předmětu se hned mění hodnoty kreditů v daných oknech a barevné změny se provádí také okamžitě a bez chyb. Při změně údajů oboru v režimu správa se po zavření okna vyhledávání správně přepočtou změny, pakliže nějaké nastaly.

### **Testování CRUD operací pro tabulky Katedra, Obor, Garant, Předmět**

Zobrazení detailů vybrané položky se zobrazuje správně a vyhledávání přes napsání části nebo celého názvu nebo jména vyhledává v pořádku. Přídavné vyhledávání jako například obor, semestr a povinnost u položek z tabulky Předmět správně aktualizuje možnosti v rozbalovacím okně název.

Pro testování mazání, tvorby a úpravy položek tabulek je nutno přistoupit k režimu správa.

Při pokusu o smazání prázdné vyhledané položky se neprovede žádná akce. Při smazání záznamu, který je duplikovaný jako jsou například předměty stejného názvu ale jiných oborů se smaže pouze položka, jenž byla vybrána a ostatní zůstanou nedotčeny.

Při tvorbě nové položky libovolné tabulky se správně otevře přidělené okno každé tabulce. Nový záznam nelze vložit do databáze, pakliže nejsou vyplněny všechny položky. Při položkách jako například email garanta se kontroluje správnost formátu tohoto emailu a upozornění správně zobrazí červený vykřičník u položky email s varováním o špatném formátu emailu.

Úprava správně načítá všechny položky a po úpravě je i správně uloží do databáze a vytvoří nový výpis k této položce ve vyhledávání. Validace údajů, jelikož se jedná o stejná formulářová okna jako při tvorbě, fungují stejně jak u předešlého testu.



## 4 Diskuze

Celkově bych práci hodnotil pozitivně i když se objevuje několik částí, které jsou řešeny v současných systémech. Největší nevýhodou je vedení této aplikace jako samostatného prvku. Uživatelé mohou mít tento systém k dispozici i když nejsou studenti, což může ulehčit plánování před studiem, ale jelikož většina studentů neplánuje svoje studium tolik dopředu nebo se drží doporučeného studijního plánu navrhnutého tak, aby bylo studium procházeno optimální cestou. Jsou zde i tací, kteří i když studují podle doporučených plánu, stále chtějí mít kontrolu nad studiem. Další slabší stránkou je nepropojenost se školím systémem. Na jednu stranu tento jev přináší volnost, co se týče zaměření na různé školy nebo podobné organizace s obdobným systémem ohodnocení. Nicméně změny, které jsou provedeny v daných organizacích, ať už změna studijních řádu na VŠPJ nebo změny detailů kolem předmětu musí uživatel prakticky hlídat a kontrolovat sám, jelikož aplikace má lokální databázi, a tudíž je nemožné tuto databázi upravovat vzdáleně. Také nemožnost aplikovat vytvořený plán při registracích a zápisech do semestrů, takže uživatel musí využít tuto aplikaci pouze informativně, jelikož předmět, jenž si zvolil nemusí být dostupný anebo nepůjde zvolit z důvodu krytí v rozvrhu s jiným předmětem, což může rozhodit celý plán, který byl vytvořen.

## **Závěr**

Hlavním cílem bakalářské práce bylo navrhnout a implementovat systém pro podporu tvorby studijních plánů. Tento cíl byl splněn.

Uživatelům je k dispozici aktuální seznam všech předmětů vyučovaných na VŠPJ, ve kterých lze vyhledávat a přidávat si je do semestrů, jak bylo zamýšleno. Při těchto operacích vznikl problém se čtením a přenášením správných dat. Tento problém byl vyřešen přepravkami dat pro dané tabulky. Dále nastal velký problém se zobrazováním správných předmětů, a to hlavně sportů. Sporty lze volit opakovaně, a tudíž podléhají jiným pravidlům než ostatní předměty. Tento problém byl vyřešen rozdělením dotazů, kde jeden filtruje všechny předměty, jež nejsou zapsané do plánu podle toho, zda jsou sudé nebo liché semestry a druhý, který kontroluje a vypisuje všechny sporty kromě těch, které nejsou zapsané do semestru, do něž se tyto předměty budou přidávat. Počítání kreditů funguje správně. Pokud nastane změna, je zapotřebí změnit konstanty určující počty mezi dvěma semestry anebo popřípadě tato pravidla upravit. Důležité požadavky jsou kontrolovány správně podle atributu Povinnost v tabulce Předmět a úspěšně se také přepočítávají podle oboru daného plánu. Ne úplně vyřešena je problematika prerekvizit. V aplikaci se nenachází automatická kontrola prerekvizit. Tyto prerekvizity předmětů jsou pouze označeny jako potřebné, i když se uživatel nemusí těmito prerekvizitami řídit a přidat si tento předmět v pořadí jakém chce. Důležité milníky jsou uchovávány v oborech a lze je sledovat při vytváření či úpravě záznamu plánu nebo po vyhledání daného oboru. Navíc byla zhotovena funkce přidávání hromadného shluku dat, jež byla využita pro prvotní načtení z dodaných souborů a jelikož by se tato funkce dala využít v budoucnu na aktualizace a přidávání nových oborů, zůstala tato funkce jako možnost pro správu. Pro zjednodušení práce a zobrazování dat byly vytvořeny vlastní komponenty pro předměty, garanty a obory, využívané v hlavním a vyhledávacím okně aplikace.

Aplikace byla navržena se snahou o přehlednost a jednoduchost, ale z důvodu množství semestrů nejsou některá okna dostatečně velká tak aby se všechny položky vešly do oken dostatečně, a tudíž je v oknech při velkém zaplnění relativně velký počet vertikálních i horizontálních posuvných lišt. Většina chyb je řešena chybovými hláškami se snahou minimalizovat problémy s použitím. Celkově by tato aplikace měla napomoci studentům s bezstarostnými starty nových semestrů, bez obavy o dostatek kreditů a splnění všech požadavků daného oboru.

Menší problém nastává při instalaci, kdy ne všem uživatelům se úspěšně podaří nainstalovat všechny potřebné věci k chodu aplikace a vygenerovaný instalátor občas selže nebo nenainstaluje vše co je nutné, a proto je nejjednodušší využívat tuto aplikaci přes spustitelný soubor přiložený na online repositáři GitHub, s nutností nainstalování VS s podporou .NET a databází, poté se aplikace spustí bez problémů. Tento problém vzniká z důvodu výběru lokální databáze. Tato databáze na rozdíl od databází na serverech vyžaduje namísto internetového připojení něco, co dokáže s touto databází pracovat.

Rozšíření aplikace by se mohlo zaměřit na vyhotovení online databáze na zlepšení aktuálnosti dat, či na provázání se školní databází s umožněním využití těchto plánů k registraci předmětů. Dále by se nabízelo rozšíření na více platform, a to buď cestou aplikací na smartphony nebo možné převedení formulářové aplikace na webovou formu pro efektivnější rozšíření aplikace mezi studenty a multiplatformní dostupnost.

## Seznam použité literatury

GENNICK, Jonathan. SQL Pocket Guide. 2nd ed. Farnham: O'Reilly, 2006. ISBN 978-0596526887.

FOWLER, Martin. UML Distilled: a brief guide to the standard object modeling language. 3rd ed. Boston: Addison-Wesley, 2004. ISBN 978-0321193681.

ISO/IEC 23270:2018. Standard ECMA-334: C# Language Specification. Závěrečný návrh 2. vydání. Geneva: ECMA, 2017. 5. edice. Dostupné také z: <http://www.ecma-international.org/publications/standards/Ecma-334.htm>

IS/STAG [online]. Plzeň: Západočeská Univerzita, 2017 [cit. 2019-03-12]. Dostupné z: [https://is-stag.zcu.cz/zajemci/moduly/plan\\_editor](https://is-stag.zcu.cz/zajemci/moduly/plan_editor)

NANDWANI, Karan. Úvod do NuGet. Microsoft [online]. United States: Microsoft, 2016, 10. 01. 2018 [cit. 2019-02-25]. Dostupné z: <https://docs.microsoft.com/cs-cz/nuget/what-is-nuget>

PEIPMAN, Gunnar. What is Micro ORM?. In: *Gunnar Peipman – Programming Blog* [online]. 2017, 2.5.2017 [cit. 2019-02-25]. Dostupné z: <https://gunnarpeipman.com/tools/micro-orm/>

Plánovač Studia [online]. Brno: MUNI, 2017 [cit. 2019-03-12]. Dostupné z: [https://is.muni.cz/napoveda/student/planovac#s\\_planovac\\_popis](https://is.muni.cz/napoveda/student/planovac#s_planovac_popis)

Stack OverFlow [online]. Stack Exchange, 2019 [cit. 2019-01-29]. Dostupné z: <https://stackoverflow.com/>

TRNKALOVÁ, Anna. Nástroj pro plánování studia v Informačním systému MU [online]. Brno, 2018 [cit. 2019-03-12]. Dostupné z: <https://is.muni.cz/th/kcokf/>. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Vedoucí práce Michal Brandejs.

UIS: UNIVERZITNÍ INFORMAČNÍ SYSTÉM [online]. Brno: IS4U, 2019 [cit. 2019-03-12]. Dostupné z: <https://www.uis-info.com/cs/index>

VLČKOVÁ, Milena, NOVÁK Milan: Informace o studiu: Prezenční a Kombinované studium. Vysoká škola polytechnická Jihlava [online]. Jihlava, 2017 [cit. 2018-11-15]. Dostupné z: <https://www.vspj.cz/student>

## **Přílohy**

*Přílohy A – Ukázky kódu aplikace.*

*Příloha A1 – Funkce na hromadné načtení Proved.*

*Příloha A2 – Vkládání dat do tabulek při hromadném načtení.*

*Příloha A3 – Hromadné načtení popisů.*

*Příloha B – Uživatelská příručka.*

## Příloha A1 – funkce na hromadné načtení Proved

```

public void Proved(string path)
{
    string line = "";
    StreamReader reader = new StreamReader(path); //vytvorení čtení řádku ze souboru
    DataCrud DC = new DataCrud();
    try
    {
        while (reader.Peek() >= 0) //testovavni zda je jeste dalsi radek
        {
            line = reader.ReadLine(); //precteni radku
            string[] fulldata = line.Split(';'); //rozlozeni prvku v line do prvku pole fulldata k jednoduššímu přístupu
            //vlození dat do insertu do databaze
            //naplnění dat do kateder
            DC.InsertKat(fulldata[13], //zkratka
                        fulldata[12]); //nazev
            //naplnění dat do oborů
            DC.InsertObor(new Obor(fulldata[5], //zkratka
                                   fulldata[5], //nazev
                                   fulldata[5], //rok
                                   154, 20, 4, 2)); //pocty kreditu defaultní hodnota
            //naplnění dat do garantů
            DC.InsertGarant(new Garant(fulldata[11], //jmeno
                                       fulldata[15], //email
                                       fulldata[12])); //katedra
            //naplnění předmětů bez popisu
            DC.InsertPredmetHromada(new Predmet(fulldata[1], //název
                                                fulldata[2], //zkratka
                                                fulldata[3], //kredit
                                                fulldata[5], //obor
                                                fulldata[11], //garant
                                                fulldata[6], //semestr
                                                fulldata[0], //orig
                                                fulldata[10], //povinnost
                                                fulldata[16], //prednaska
                                                fulldata[17], //cviceni
                                                fulldata[19], //kombi
                                                fulldata[20], //lab
                                                fulldata[22], //jazyk
                                                fulldata[4])); //zakončení
        }
        MessageBox.Show("Načtení dat do databáze proběhlo úspěšně", "Informace", MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    catch (Exception ex) //zachycení chyby
    {
        MessageBox.Show("Načtení dat skončilo s chybou: " + ex, "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

## Příloha A2 – Vkládání dat do tabulek při hromadném načtení

```
private const string NazevDB = "SystemProPodporuStudijnichPlanu.Properties.Settings.DatabaseAppConnectionString";
public void InsertKat(string zkratka, string nazev)
{
    DataAccess da = new DataAccess();
    //kontrola, zda katedra již neexistuje v databázi
    da.CheckExistKatedra(nazev, out int exist);
    if (exist <= 0)
    {
        //když neexistuje otevře se connection s databází a provede se dotaz na vložení
        using (SqlConnection conn = new SqlConnection(DataAccess.ConnValue(NazevDB)))
        {
            SqlCommand kat = new SqlCommand("insert into [katedra]([zkr_k],[naz_k]) values(@zkr_k,@naz_k)", conn);
            kat.Parameters.AddWithValue("@zkr_k", zkratka);
            kat.Parameters.AddWithValue("@naz_k", nazev);
            try
            {
                conn.Open();
                kat.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Načtení dat skončilo s chybou: " + ex, "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            conn.Close();
        }
    }
}

public void InsertObor(Obor o)//využití přepravky Oboru na přenos všech údajů
{
    DataAccess da = new DataAccess();
    //kontrola, zda obor již neexistuje v databázi
    da.CheckExistObor(o.Name_obor, out int exist);
    if (exist <= 0)//když neexistuje otevře se connection s databází a provede se dotaz na vložení
    {
        using (SqlConnection conn = new SqlConnection(DataAccess.ConnValue(NazevDB)))
        {
            const string CmdText = "insert into obor(zkr_obor,name_obor,rok_obor,p_obor,pv_obor,v_obor,vs_obor,praxe) " +
                "values(@zkr,@naz,@rok,@p,@pv,@v,@vs,@praxe)";
            SqlCommand obor = new SqlCommand(cmdText: CmdText, connection: conn);
            obor.Parameters.AddWithValue("@zkr", o.Zkr_obor);
            obor.Parameters.AddWithValue("@naz", o.Name_obor);
            obor.Parameters.AddWithValue("@rok", o.Rok_obor);
            obor.Parameters.AddWithValue("@p", o.P_obor);
            obor.Parameters.AddWithValue("@pv", o.Pv_obor);
            obor.Parameters.AddWithValue("@v", o.V_obor);
            obor.Parameters.AddWithValue("@vs", o.Vs_obor);
            obor.Parameters.AddWithValue("@praxe", o.Praxe);
            try
            {
                conn.Open();
                obor.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Načtení dat skončilo s chybou: " + ex, "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            conn.Close();
        }
    }
}
```

## Vysoká škola polytechnická Jihlava

```
public void InsertGarant(Garant g)
{
    DataAccess da = new DataAccess();
    da.CheckExistGarant(g.Jmeno_v, out int exit);
    if (exit <= 0)
    {
        using (SqlConnection conn = new SqlConnection(DataAccess.ConnValue(NazevDB)))
        {
            //kontrola jestli neexistuje již existuje
            SqlCommand garant = new SqlCommand("insert into [garant](jmeno_v,email_v,tel_v,konz_v,id_k) values(@jmeno_v,@email_v,@tel_v,@konz_v,@id_k)", conn);
            garant.Parameters.AddWithValue("@jmeno_v", g.Jmeno_v);
            garant.Parameters.AddWithValue("@email_v", g.Email_v);
            garant.Parameters.AddWithValue("@tel_v", g.Tel_v);
            garant.Parameters.AddWithValue("@konz_v", g.Konz_v);
            garant.Parameters.AddWithValue("@id_k", g.Id_k);
            try
            {
                conn.Open();
                garant.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Načtení dat skončilo s chybou: " + ex, "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            conn.Close();
        }
    }
}

public void InsertPredmetHromada(Predmet p)
{
    DataAccess da = new DataAccess();
    da.CheckExistPredmet(p.Name_predmet, p.Id_obor, out int exist);
    if (exist <= 0)
    {
        using (SqlConnection conn = new SqlConnection(DataAccess.ConnValue(NazevDB)))
        {
            SqlCommand pred = new SqlCommand(
                "INSERT INTO [predmet]" +
                "([name_predmet],[zkr_predmet],[kredit_predmet],[id_obor],[id_v],[semestr_predmet],[id_orig],[povinnost],[prednaska],[cviceni],[kombi],[lab],[jazyk],[zakonceni])" +
                "VALUES(@name_predmet,@zkr_predmet,@kredit_predmet,@id_obor,@id_v,@semestr_predmet,@id_orig,@povinnost,@prednaska,@cviceni,@kombi,@lab,@jazyk,@zakonceni)"
                ,conn);
            pred.Parameters.AddWithValue("@name_predmet", p.Name_predmet);
            pred.Parameters.AddWithValue("@zkr_predmet", p.Zkr_predmet);
            pred.Parameters.AddWithValue("@kredit_predmet", p.Kredit_predmet);
            pred.Parameters.AddWithValue("@id_obor", p.Id_obor);
            pred.Parameters.AddWithValue("@id_v", p.Id_v);
            pred.Parameters.AddWithValue("@semestr_predmet", p.Semestr_predmet);
            pred.Parameters.AddWithValue("@id_orig", p.Id_orig);
            pred.Parameters.AddWithValue("@povinnost", p.Povinnost);
            pred.Parameters.AddWithValue("@prednaska", p.Prednaska);
            pred.Parameters.AddWithValue("@cviceni", p.Cviceni);
            pred.Parameters.AddWithValue("@kombi", p.Kombi);
            pred.Parameters.AddWithValue("@lab", p.Lab);
            pred.Parameters.AddWithValue("@jazyk", p.Jazyk);
            pred.Parameters.AddWithValue("@zakonceni", p.Zakonceni);
            try
            {
                conn.Open();
                pred.ExecuteNonQuery();
            }
            catch (Exception ex)
            {
                MessageBox.Show("Načtení dat skončilo s chybou: " + ex, "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
            }
            conn.Close();
        }
    }
}
```



## Příloha A3 – Hromadné načtení popisů

```
public void ProvedPopis(string path)
{
    DataCrud DC = new DataCrud();
    List<string> fulldata = new List<string>();
    //načtení celého textu do stringu
    fulldata = File.ReadAllText(path).Split('~').ToList();
    //procházení pole a zápis do db krok po každých šesti datech
    for (int i = 0; i < fulldata.Count(); i += PocetPrvku)
    {
        try
        {
            //načtení dat vždy 1., 3. a 4. v pořadí od 0+(n*V) kde n začíná od 0 až do konce souboru
            DC.InsertPopis(new Predmet(fulldata[i + 1], //navez predmetu
                                     fulldata[i + 3], //text popisu
                                     fulldata[i + 4])); //rok oboru/označení
            MessageBox.Show(Properties.Resources.SuccNact_MESSAGE,
                            Properties.Resources.Info_TITLE,
                            MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        catch (Exception ex) //zachycení chyby
        {
            MessageBox.Show("Načtení dat skončilo na indexu: " + i + "s chybou: " + ex,
                            Properties.Resources.Chyba_TITLE,
                            MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
```

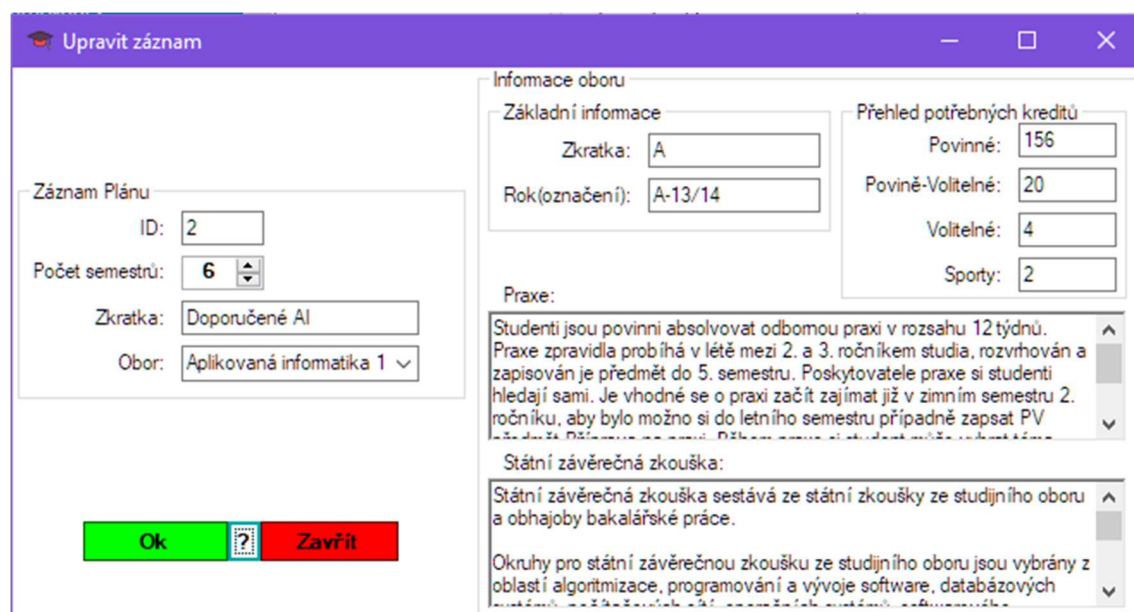
```
public void InsertPopis(Predmet p)
{
    using (SqlConnection conn = new SqlConnection(DataAccess.ConnValue(NazevDB)))
    {
        SqlCommand pop = new SqlCommand("update [predmet] set [popis]=@popis where [id_predmet]=@id_predmet", conn);
        pop.Parameters.AddWithValue("@popis", p.Popis);
        pop.Parameters.AddWithValue("@id_predmet", p.Id_predmet);
        try //zkouska zapisu
        {
            conn.Open();
            pop.ExecuteNonQuery();
        }
        catch (Exception ex) //kdyz neproběhne, tak se vypíše chyba
        {
            MessageBox.Show("Načtení dat skončilo s chybou: " + ex, "Chyba", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
        conn.Close();
    }
}
```

## Příloha B – Uživatelská příručka

V tomto dokumentu budou rozebrány všechny okna dostupná v aplikaci s cílem seznámit uživatele s aplikací pro snazší orientaci a ovládání

### Tvorba záznamů

Při zapnutí aplikace se zobrazí nabídka na vytvoření nového plánu, když žádný plán není vytvořen. Při potvrzení se otevře okno na vytváření s informací o vybraném oboru. Nastavení pojmenování záznamu a výběr počtu semestrů.



**Upravit záznam**

**Záznam Plánu**

ID:

Počet semestrů:

Zkratka:

Obor:

**Informace oboru**

Základní informace

Zkratka:

Rok(označení):

**Přehled potřebných kreditů**

Povinné:

Povinně-Volitelné:

Volitelné:

Sporty:

**Praxe:**

Studenti jsou povinni absolvovat odbornou praxi v rozsahu 12 týdnů. Praxe zpravidla probíhá v létě mezi 2. a 3. ročníkem studia, rozvířován a zapisován je předmět do 5. semestru. Poskytovatele praxe si studenti hledají sami. Je vhodné se o praxi začít zajímat již v zimním semestru 2. ročníku, aby bylo možno si do letního semestru případně zapsat PV.

**Státní závěrečná zkouška:**

Státní závěrečná zkouška sestává ze státní zkoušky ze studijního oboru a obhajoby bakalářské práce.

Okruhy pro státní závěrečnou zkoušku ze studijního oboru jsou vybrány z oblasti algoritmizace, programování a vývoje software, databázových

**Ok** **?** **Zavřít**

## Hlavní okno

Toto okno se zobrazí po vytvoření záznamu nebo po zapnutí s již vytvořeným záznamem. Zde se zobrazují semestry, kam lze přidávat/odebírat předměty podle potřeby a zobrazit detail vybraného předmětu a garanta. Kromě zobrazení lze zvolit Přidání při čemž se zobrazí okno na přidávání anebo vybrat něco z nabídky. Soubor obsahuje vytváření, úpravu plánu a ukončení. Nabídka vyhledávání otevírá okno s na hledání, popřípadě úpravu dat. V poslední řadě nápověda, ve které se budou nacházet informace o aplikaci a možná i návod.

**Systém pro podporu tvorby studijních plánů**

Soubor Vyhledávání Nápověda

Výběr plánu: Doporučené AI

**Semestr 1** (30 kreditů): Anglický jazyk /PSAI-u1, Základy lineární algebry, Teoretické základy informatiky, **Programování 2**

**Semestr 2** (28 kreditů): Matematika 2, Anglický jazyk /PSAI-u2, Architektura počítačů, Základy elektrotechniky, Základy efektivní komunikace

**Semestr 3** (26 kreditů): Sport - Kondiční cvičení, Webové technologie 1, Numerical Methods, Statistika pro techniky, Datové struktury a algoritmy, Databázové systémy 1

**Semestr 4** (26 kreditů): Anglický jazyk /PSAI-u4, Programování desktopových aj, Ekonomika podniku 2, Operační systémy, Databázové systémy 2

**Semestr 5** (49 kreditů): Sport - Florbal, Informační systémy 1, Bezpečnost a ochrana dat, Pokročilé programovací techniky, Odborná praxe

**Semestr 6** (21 kreditů): Programování pro mobilní plat, Programovací jazyky a překlad, Právo v technické praxi, Seminář k bakalářské práci, Projektový management

**Detaily vybraného předmětu**

Základní informace:  
 Zkratka: xPRG2  
 Semestr: 1  
 Kredity: 5  
 Povinnost: Povinný předmět  
 Zkončení: ZK  
 Prerekvizita: Není  
 Jazyk: Česky  
 Týdenní dotace hodin: 2  
 Přednášky: 2  
 Cvičení: 2  
 Kombi: 8  
 Laboratoře: 0

Garant: doc. Ing. Karel Richta, CSc.  
 Popis: Předmět je úvodem do algoritmizace a základů programování v jazyce C s využitím procedurálního přístupu. Kromě základních pojmů jsou probrány datové typy, řídicí příkazy, výrazy a funkce. Student se naučí, jak pomocí základních programovacích technik sestavit jednoduchý program. Znalosti: Student zná principy procedurálního programování, zná obvyklé datové typy, řídicí příkazy, princip volání funkcí a předávání parametrů. Dovednosti: Student umí použít teoretické poznatky při řešení konkrétní úlohy. Umí rozdělit problém na podproblémy a řešit jej systémem procedur (funkcí) v imperativním jazyce C.

**Detail Garanta**

Informace garanta:  
 Konzultace: Čtvrtek - 17:00-18:00 - Na konzultace je třeba  
 Email: karel.richta@vspj.cz  
 Telefon: 567141241  
 Katedra: Katedra technických studií

**Summary:**  
 Celkem Kreditů: 180  
 Kredity z povinných předmětů: 156  
 Kredity z povinně-volitelných: 20  
 Kredity z volitelných předmětů: 4

Toto okno může využít několika klávesových zkratk, jako například po výběru předmětu v libovolném semestru stačí zmáchnout *ENTER* a poté se zobrazí detail o garantovi předmětu. Po stisknutí *BACKSPACE* na vybraném předmětu se tento předmět odstraní z plánu.

## Přidávání

V tomto okně lze přidávat předměty do vybraného semestru. Při zvolení se zobrazí informace o tomto předmětu. Lze tyto vybrané předměty přidat anebo zrušit výběr a navrátit se do předešlého okna.

Pro výběr nebo vrácení předmětu z jedné strany na druhou lze využít šipek určující kam se má vybraný předmět přesunout anebo stačí stisknout klávesu SPACE a tato operace se také provede.

Výběr plánu:

Doporučené AI

Výber semestr:

2

Přidat

Smaž vybraný

Přidání předmětu do 2. semestru

Detaily vybraného předmětu

Základní informace

Zkratka: xPPX

Semestr: 4

Kredity: 4

Povinnost: Povinně volitelný

Zakončení: ZA

Prerekvizita: Není

Jazyk: Český

Týdenní dotace hodin

Přednášky: 0

Cvičení: 0

Kombi: 0

Laboratoře: 0

Garant:

Mgr. Antonín Příbyl

Popis:

Smyslem předmětu je vytvořit studentovi podmínky pro přípravu na plnohodnotné absolvování odborné praxe, tzn. pro studium nástrojů, SW apod., které na praxi bude používat ve zvolené organizaci. Během předmětu se student seznamuje s problematikou, kterou bude řešit během Odborné praxe, absolvuje potřebná školení a provádí další případné přípravy pro zdáné absolvování Odborné praxe.

Výbrat semestr: Všechny

Multimedia

Český jazyk-u2

Matematický seminář 2/AP2

Prostředky průmyslové automatizace

Připrava na praxi

Webové technologie 2

Český jazyk-u4

Finanční matematika

Multimediální tvorba I

Multimediální a studiová technika

Elektronické obchodování

International Business Week

Měření v systémech managementu jak

Počítačová grafika

Počítačové řídící systémy

Podnikání i fyzické osoby

Psychologie

Simulační manažerské hry

Společnost s ručením omezeným

Sport - Aerobic

Sport - Badminton

Sport - Basketbal

Optimalizační metody

Ok

Zavřít

Kredity 4

## Vyhledávání (Obor, Garant, Předmět, Katedra)

Po zvolení Vyhledávání v nabídce se otevře okno s výběrem. Po vybrání lze vyhledat požadovaný prvek. Lze vybrat z možností anebo napsat část názvu a po rozbalení nabídky se zobrazí všechny záznamy s podobným textem.

**Vyhledávání**

Zakázat správu   Hromadné načítání

Vyberte s čím chcete pracovat

- ☒ Obor
- ☐ Garant
- ☐ Předmět
- ☐ Katedra

Výběr a vyhledávání oborů

Obor:

**Aplikovaná informatika 13/14** ▼

Informace oboru

Základní informace

Zkratka:

Rok(označení):

Přehled potřebných kreditů

Povinné:

Povinně-Volitelné:

Volitelné:

Sporty:

Praxe:

Studenti jsou povinni absolvovat odbornou praxi v rozsahu 12 týdnů. Praxe zpravidla probíhá v létě mezi 2. a 3. ročníkem studia, rozvrhován a zapisován je předmět do 5. semestru. Poskytovatele praxe si studenti hledají sami. Je vhodné se o praxi začít zajímat již v zimním semestru 2. ročníku, aby bylo možno si do letního semestru případně zapsat PV.

Státní závěrečná zkouška:

Státní závěrečná zkouška sestává ze státní zkoušky ze studijního oboru a obhajoby bakalářské práce.

Okruhy pro státní závěrečnou zkoušku ze studijního oboru jsou vybrány z oblasti algoritmizace, programování a vývoje software, databázových

**Zavřít**

**Vyhledávání**

Zakázat správu   Hromadné načítání

Vyberte s čím chcete pracovat

- ☐ Obor
- ☒ Garant
- ☐ Předmět
- ☐ Katedra

Výběr a vyhledávání garantů

Katedra:

Katedra technických studií ▼

Garant:

**doc. Ing. Karel Richta, CSc.** ▼

Informace garantů

Konzultace:

Email:

Telefon:

Katedra:

**Zavřít**



Vyhledávání

Zakázat správu

Hromadné načítání

Vyberte s čím chcete pracovat

☐ Obor

☐ Garant

☒ Předmět

☐ Katedra

Výběr a vyhledávání předmětů

Obor:

Aplikovaná informatika 13/14

Název:

Matematika 2

Povinnost:

"Povinný předmět"

Semestr:

Zavřít

Detaily vybraného předmětu

Základní informace

Zkratka:

xMAT2

Semestr:

2

Kredity:

5

Povinnost:

Povinný předmět

Zakončení:

ZK

Prerekvizita:

Matematika 1

Jazyk:

Česky

Garant:

doc. RNDr. Petr Gurka, CSc.

Popis:

Cílem předmětu je vybavit posluchače základními znalostmi z oblasti matematické analýzy, a to především diferenciálního a integrálního počtu funkcí více proměnných, obyčejných diferenciálních rovnic a základních integrálních transformací. Znalosti: Absolvent zná základní věty a definice, které jsou obsahem předmětu Matematika 2, zná souvislosti a vztahy mezi studovanými objekty a dokáže je vysvětlit. Získaných poznatků umí využít při řešení příslušných matematických problémů. Dovednosti: Absolvent předmětu umí využívat získaných matematických znalostí oň dalším.

Týdenní dotace hodin

Přednášky:

2

Cvičení:

2

Kombi:

8

Laboratoře:

0

V...

Zakázat správu

Vyberte s čím chcete pracovat

☐ Obor

☐ Garant

☐ Předmět

☒ Katedra

Výběr a vyhledávání kateder

Katedra:

Katedra zdravotnických studií

Katedra technických studií

Katedra cestovního ruchu

Katedra ekonomických studií

Katedra sociální práce

Katedra matematiky

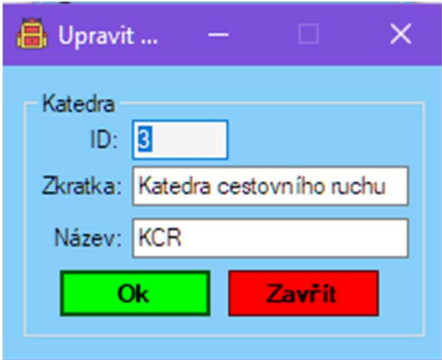
Katedra jazyků

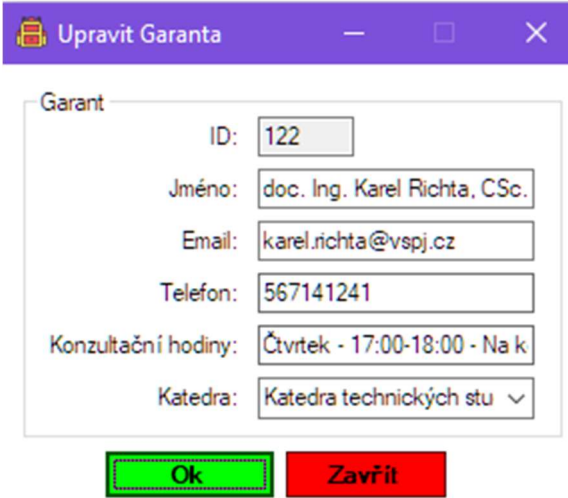
Kabinet sportů

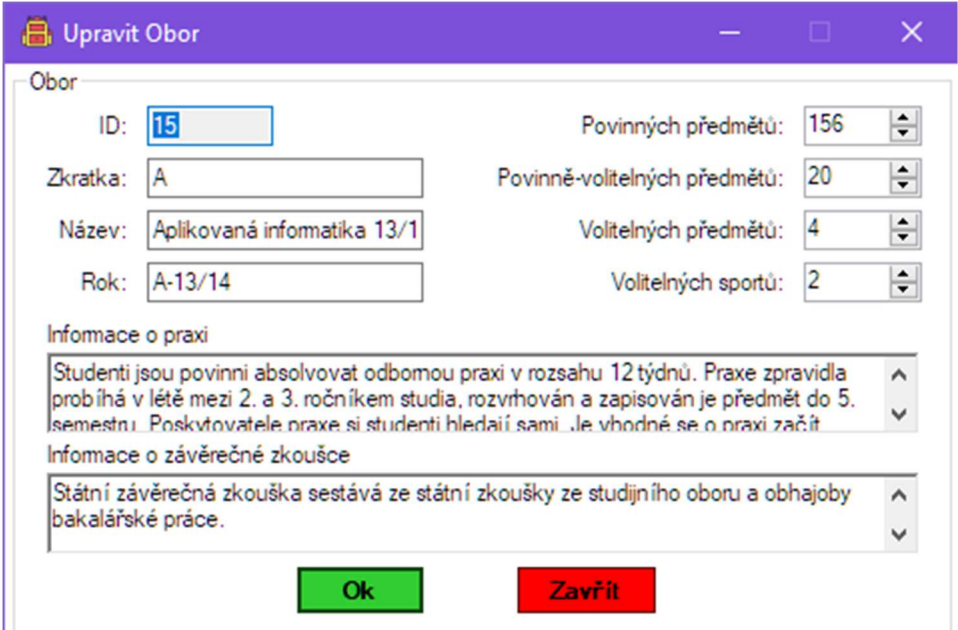
Zavřít

## Vytváření a úprava garanta, katedry, oboru a předmětu

Po povolení správy lze vybrat možnosti vytvořit nový anebo upravit, popřípadě smazat vybraný (vyhledaný prvek)







**Upravit Předmět**

Předmět

ID:

Zkratka:

Název:

Garant:

Obor:

Orig. ID:

Povinnost:

Zakončení:

Jazyk:

Prerekvizita:

Kredity:

Semestr:

Přednášky:

Cvičení:

Laboratoře:

Cvičení kombi:

Popis:

Předmět biofyzika obsahuje znalosti z fyziky a specifické učivo zaměřené na zdravotnickou problematiku.  
Cíl: Studenti získají základní informace o fyzikálních odlišnostech jednotlivých metod a jejich biologických účincích  
Znalosti: Student zná strukturu a vlastnosti biomembrán, biofyziku tkání orgánů, biomechaniku kostí a kloubů, biomechaniku krevního oběhu, biofyziku dýchání a biofyziku smyslového vnímání.  
Dovednosti: Student provede natočení ekg u klienta s onemocněním srdce, zvládá provedení a edukaci klienta při spirometrii.  
Obecné způsobilosti: Seznámí se s jednotlivými druhy těchto metod a přístroji, kterými jsou tato vyšetření prováděna.

Dokumentace, modely a kód aplikace se nachází na přiloženém CD nebo na: [HTTPS://GITHUB.COM/CINK01/SPPSP](https://github.com/CINK01/SPPSP)