Univerzita Pardubice
Fakulta elektrotechniky a informatiky

# Rozšíření modulu Testy na portálu UPa

Bc. Filip Borovec

Diplomová práce

2010

**Univerzita Pardubice**
**Fakulta elektrotechniky a informatiky**
Akademický rok: **2009/2010**

# ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

| | |
|---|---|
| Jméno a příjmení: | **Bc. Filip BOROVEC** |
| Studijní program: | **N2646 Informační technologie** |
| Studijní obor: | **Informační technologie** |
| Název tématu: | **Rozšíření modulu Testy na portálu UPa** |
| Zadávající katedra: | **Katedra softwarových technologií** |

Zásady pro vypracování:

- Na Univerzitě Pardubice se v současné době rozjíždí nová část podpory e-learningu - modul Testy v portálu UPa. Tento modul byl navržen jak pro specifické potřeby Jazykového centra, tak pro potřeby ostatních učitelů na UPa. Komunikace o potřebách uživatelů ale nakonec probíhala výhradně s Jazykovým centrem. V současné době, kdy ostatní učitelé také chtějí začít používat tento systém nastává situace, kdy vyvstávají nové požadavky na systém. - Úkolem studenta bude zpracovat všechny požadavky uživatelů tohoto systému, vytvořit případy užití, zjistit rozdíl mezi současnou situací, navrhnout a podílet se na implementaci rozšíření modulu o nové funkčnosti. - Pro to, aby mohl student navrhovat a implementovat změny v této aplikaci, musí mít přehled podpory e-learningu používaný v univerzitních prostředích, na Univerzitě Pardubice a zároveň musí v práci prokázat pochopení základních technologií, na kterých stojí portálové řešení, ve kterém byl modul Testy implementován (jedná se o specifikaci JSR-168). Předpoklady: - Student analyzuje a implementuje možnosti lepšího zabezpečení běhu testu (spouštění v konkrétní učebně, omezení kopírování otázek a odpovědí, ...). - U rozřazovacích testů zatím chybí zpětná kontrola na jaký předmět / rozvrhovou akci se student na základě výsledku rozřazovacího testu zapsal, bude nutné tuto novou funkčnost implementovat. - V současné době chybí statistiky testu - např. jaká otázka byla pro studenty v testu nejtěžší apod. - Další požadavky vyplynou z analýzy.

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

**[1] HORTON, William. E-Learning by Design. 1st edition. [s.l.] : Pfeiffer, 2006. 640 s. ISBN 0787984256. [2] COLE, Jason, FOSTER, Helen. Using Moodle: Teaching with the Popular Open Source Course Management System. 2nd edition. [s.l.] : O'Reilly Media, Inc., 2007. 282 s. ISBN 059652918X. [3] ECKEL, Bruce. Thinking in Java. 4th edition. [s.l.] : Prentice Hall, 2006. 1150 s. ISBN 0131872486. [4] SIERRA, Kathy. Head First Servlets and JSP. 2nd edition. [s.l.] : O'Reilly Media, Inc., 2008. 911 s. ISBN 0596516681. [5] Online zdroje na Internetu**

Vedoucí diplomové práce: **Ing. Jiří Pinkas**
Katedra informatiky v dopravě

Datum zadání diplomové práce: **30. října 2009**

Termín odevzdání diplomové práce: **21. května 2010**

prof. Ing. Simeon Karamazov, Dr.
děkan

L.S.

doc. Ing. Antonín Kavička, Ph.D.
vedoucí katedry

V Pardubicích dne 10. listopadu 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou vedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20.8.2010

Bc. Filip Borovec

Poděkování

Rád bych poděkoval vedoucímu mé diplomové práce panu Ing. Jiřímu Pinkasovi za cenné rady a připomínky, které mi pomáhaly nejen při tvorbě této diplomové práce.

## SOUHRN

Tato práce se zabývá problematikou e-learningu a jeho využití jako alternativní formy výuky i jako doplnění klasické výuky. V práci najdeme souhrn výhod a nevýhod jeho použití. Důraz je kladen na testování jako nedílnou součást e-learningu. V praktické části se práce zaměřuje na analýzu požadavků na rozšíření testovacího modulu používaného na pardubické Univerzitě. Najdeme zde ale i to, jak byly požadavky implementovány, a stručný popis technologií, na kterých testovací modul stojí.

## KLÍČOVÁ SLOVA

E-learning, testování, analýza, Java, Java EE, portály, portlety

## TITLE

Extension of Test module on UPa portal.

## ABSTRACT

This work deals with an e-learning questions and its using like alternative form of teaching and like complement of classical teaching methods. We also find summary of advantages and disadvantages of using e-learning in this work. Emphasis is placed on the testing like atomic part of e-learning. In practical part this work deals with analysis of requirement to extension of Test module using on University of Pardubice. But we also find brief description of requirements implementation and short description of technologies on which Test module is based on.

## KEYWORDS

E-learning, testing, analysis, Java, Java EE, portals, portlets

# Obsah

# Úvod

V dnešní uspěchané době, kdy se všichni snaží optimalizovat své aktivity, je patrná snaha zefektivnit vzdělávání, ať už zaměstnanců nebo studentů na Univerzitě či v jiné instituci. Nové informace navíc přibývají velkou rychlostí, zvláště v oboru informačních technologií. A tak je třeba volit vhodné formy výuky.

Mnoho firem a institucí tak dává přednost v dnešní době modernímu e-learningu. Tato forma výuky slibuje oproti klasické formě studia efektivnější využití vynaložených nákladů a přitažlivější a individuálnější přístup k novým informacím. Dá se říci, že za cenu vyšších počátečních nákladů na přípravu kurzu je možno vyškolit větší množství studentů s menšími náklady a v kratším čase než je tomu v případě klasické „učebnové“ výuky.

Nedílnou součástí e-learningu je testování získaných znalostí studentů. To umožňuje provádět vyhodnocení pokroku jednotlivých studentů, definovat kritéria pro ukončení kurzu a případnou certifikaci. Testy také poskytují studentům ve vzdělávacím procesu velice důležitou zpětnou vazbu a mohou přispívat k hodnocení kvality samotného kurzu. Tato práce se proto zaměřuje na testování dosažených znalostí. Důraz je kladen na správné zařazení testů do výukového procesu a na možnosti navržení testu a jeho otázek.

V praktické části si tato práce klade za cíl analyzovat požadavky lektorů pardubické Univerzity na rozšíření modulu pro testování studentů, který je zprovozněn v rámci portálu IS/STAG. Dále navrhnout možné postupy, jak by bylo nejlepší tyto požadavky implementovat. A nakonec rozšířit testovací modul tak, aby měli lektoři pardubické Univerzity k dispozici bezpečné, pohodlné a efektivní testovací prostředí, které jim bude ulehčovat práci.

# 1 E-learning

## 1.1 Vymezení pojmu E-learning

E-learning zažívá v poslední době svůj velký „boom“. Důvodem je zcela nepochybně snaha o efektivnější předání znalostí, a to jak z časového hlediska, tak i z hlediska vynaložených nákladů.

Na e-learning je poměrně často nahlíženo jako na nový směr vzdělávání – v duchu „nové ekonomiky“ a představy internetu jako nového trhu. Jsou však autority, které odmítají koncepci „nové ekonomiky“ s odkazem, že se ekonomika řídí stále stejnými zákonitostmi a internet není novým trhem, ale jenom novým médiem. Pro ně e-learning probíhá v alternativním prostředí a užívá jen jiné učební pomůcky. Nemá podstatný význam pro změnu paradigmatu vzdělávání [2].

Definic e-learningu existuje celá řada. Jsou různé podle toho, z jakého úhlu na problematiku každý jednotlivý autor nahlíží. Po přečtení několika definic, si lze udělat poměrně ucelený obrázek o tom, co e-learning znamená. Uvedu zde tři definice:

- E-learning je vzdělávání formou prezentace s texty a odkazy, animovanými sekvencemi, video snímky. E-learning poskytuje sdílení pracovní plochy, hlasové komentáře, komunikaci s lektorem a spolužáky, testy, elektronické modely procesů [5].
- E-learning jsou systémy pro správu znalostí a řízení vzdělávání (LMS) a systémy pro správu obsahu a poskytování elektronických kurzů (LCMS) [5].
- E-learning je vzdělávací proces, využívající informační a komunikační technologie k tvorbě kurzů, k distribuci studijního obsahu, komunikaci mezi studenty a pedagogy a k řízení studia [3].

Jak je zřejmé z těchto definici, e-learning je formou vzdělávání, tedy určitým procesem, ve kterém se snažíme v maximální míře a pokud možno efektivně a smysluplně využít dostupných prostředků výpočetní techniky. A to tak, aby se co možná nejvíce odlehčila a zjednodušila práce zainteresovaným stranám, tedy zejména pedagogům a studentům.

Jako důležitá a nosná vlastnost e-learningu se též uvádí možnost studenta zvolit si vlastní formu vzdělávání a hlavně vlastní tempo. Pomocí e-learningu tedy můžeme (alespoň částečně) eliminovat přizpůsobování tempa výkladu lektora v extrému na „úroveň“ nejhoršího nebo naopak nejlepšího studenta přítomného na kurzu, což se často děje v prezenční formě výuky.

## 1.2 Vztah e-learningu a klasické výuky

Nástup e-learningu byl přijímán s velkým očekáváním. Ta byla spojena s představou, že 80 – 90 % firemního vzdělávání bude mít tuto formu a že se výrazně sníží náklady na vzdělávání. Po počátečním nadšení následovalo zklamání. E-learning nesplnil očekávání, účastníci kurzů přece jen upřednostňují osobní setkání a nechtějí trávit hodiny u obrazovky, zvláště když u ní tráví většinu své pracovní doby. Proto se objevily nové trendy. Zejména kombinace e-learningu a prezenčního studia se ukazuje jako nosná [2].

Tato kombinace klasické výuky a e-learningu se nazývá **Blended Learning**. Blended learning může nabývat dvou základních podob [5]:

- „**e-learningový kurz** – prezenční kurz“, kde je e-learningový kurz svým charakterem **startovací kurz**.
- „**prezenční kurz** – e-learningový kurz“, kde má e-learningový kurz charakter **udržovacího či oživovacího kurzu**.

V univerzitním prostředí, na které je tato práce zaměřena, je v kombinovaných ale i prezenčních formách studia více využíván druhý uvedený způsob kombinace klasické výuky s e-learningovými kurzy. Důraz v e-learningových kurzech je ale kladen spíše než na oživování znalostí na jejich rozšiřování.

Zde je ještě vhodné podotknout, že jednotlivé moduly e-learningových systémů mohou s výhodou doplňovat i klasické prezenční studium. Tím se zjednoduší, zpříjemní a zrychlí práce zúčastněných osob. Řeč je zejména o elektronické distribuci studijního obsahu a doprovodných materiálů a elektronickém testování studentů.

Je tedy zřejmé, že cílem nasazení e-learningu by nemělo být úplné nahrazení klasické prezenční výuky, i když v některých oblastech toto lze s úspěchem aplikovat. Zejména v oblastech, kde je přímý kontakt s lektorem nezastupitelný, se ukazují e-learningové nástroje jako vhodným doplněním a distribučním kanálem.

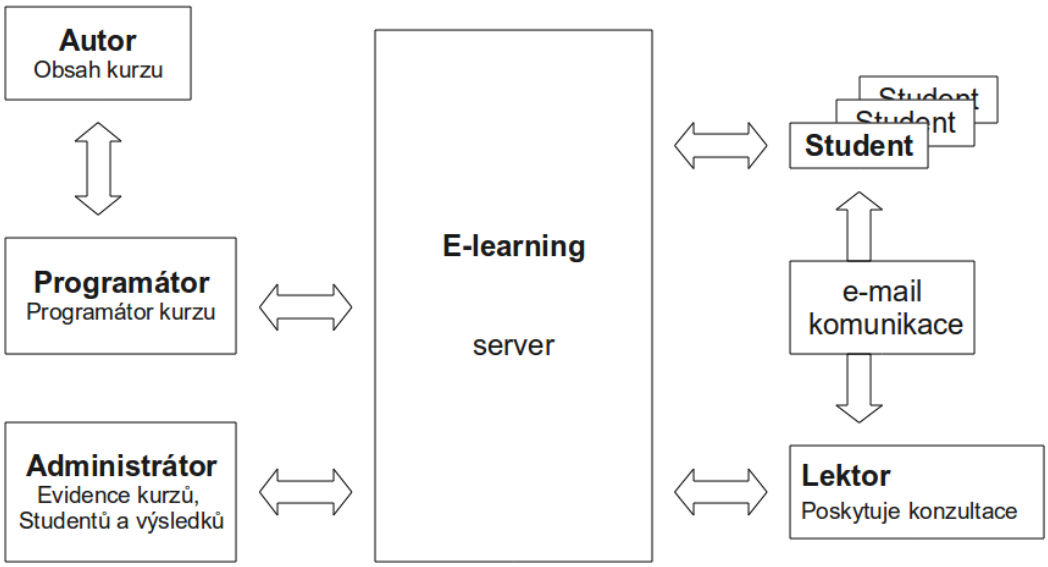
Někdy se také setkáváme s určitým předsudkem, že v e-learningové výuce není učitel. Vyučující zde samozřejmě existuje. Rozdíl je v jeho jiných možnostech v přístupu ke studentům. Díky řadě nástrojů jako je e-mail, videokonference nebo elektronické testování může učitel intenzivněji a hlavně individuálněji komunikovat se studenty. Díky výsledkům studentských pre-testů a re-testů (bude rozebráno v kapitole 2.2) uložených v databázi má vyučující k dispozici daleko lepší zpětnou vazbu o vývoji znalostí studentů, než je tomu v případě klasické výuky. V e-learningové výuce je rovněž prostor pro sledování různých dalších statistik, např. kolik času tráví studenti při studiu jednotlivých kurzů, atd.

Toto vše poskytuje individuálnější přístup ke studentům oproti klasické výuce, kde je čas, který je možné věnovat individuálním konzultacím, striktně omezen délkou výuky kurzu. Navíc je lektor ve vyučovací místnosti sdílen více studenty, což opět snižuje možnosti individuálního přístupu.

## 1.3 Účastníci e-learningu

Jak již bylo řečeno, e-learning je didaktický proces. Aby tento proces správně a efektivně fungoval, musí se ho účastnit kromě informačního systému další účastníci. V této kapitole si specifikujeme role účastníků e-learningu. Mezi účastníky existují logické vazby, které si můžeme dokumentovat na přiloženém obrázku.

Obrázek č. 1: Vazby mezi účastníky e-learningu




Zdroj: HLAVÁČEK, Tomáš. *E-learning a jeho přínosy pro organizaci*. Pardubice, 2005. 85 s. Diplomová práce. Univerzita Pardubice, Dopravní fakulta Jana Pernera

Následující tabulka blíže specifikuje jednotlivé role a jejich úkoly:

| **Role** | **Úkoly** |
|---|---|
| Autor | Tvorba odborného obsahu kurzu.<br>Odpovědnost za pedagogickou správnost.<br>Někdy tvorba multimediálního obsahu.<br>Tvorba testů. |
| Programátor resp. vývojář | Převod obsahu do e-learningového systému.<br>Tvorba počítačové grafiky.<br>Programátorské práce.<br>Tvorba interaktivního obsahu na základě podnětů autora.<br>Pilotní ověření e-kurzu. |
| Lektor | Vedení výuky.<br>Konzultace, rady ke studiu. |

| | |
|---|---|
| | Podpora usnadnění studia.<br>Vyhodnocování výsledků testů |
| Student | Studium kurzů.<br>Vykonávání testů. |
| Administrátor | Zajišťování evidence kurzů, studentů a výsledků.<br>V moderních systémech je tato role v maximální míře nahrazována samotným systémem. |

Tabulka č. 1: Účastníci e-learningu a jejich úkoly

Někdy se též uvádí další role „Manažer“, ten zabezpečuje celkový návrh strategie, koordinuje projekt a provádí analýzu a vyhodnocení výuky.

Toto rozdělení úkolů je nutno brát spíše jako příklad než jako dogma. Konkrétní tým si může úkoly přizpůsobit podle svých možností a schopností. Někdy se také role „Autor“ může rozpadnout na autora a didaktického pracovníka, i když je to většinou jedna a ta samá osoba.

## 1.4 Úrovně e-learningu

E-learningové kurzy můžeme rozdělit podle míry využití možností výpočetní techniky, připojení k internetu a interaktivních prvků do následujících třech skupin.

### 1.4.1 CBT

**Computer-Based Training** [13] – Vzdělávání s podporou výpočetní techniky můžeme označit za první úroveň elektronického vzdělávání. Jedná se o tzv. vzdělávání off-line. Veškeré materiály, kurzy, doprovodné cvičení a jiné programy jsou distribuovány off-line pomocí datových nosičů (např. CD nebo DVD). V této souvislosti ještě nelze mluvit o e-learningu.

### 1.4.2 WBT

**Web-Based Training** [14] – Druhá úroveň elektronického vzdělávání. Jde o vzdělávání s podporou webových technologií. V této on-line formě vzdělávání je veškerý obsah distribuován přes síť internet nebo např. přes firemní intranet. To má oproti CBT výhodu v možnosti okamžité aktualizace obsahu bez dalších dodatečných nákladů v podobě redistribuce datových nosičů. On-line připojení studentů je kromě zobrazení obsahu kurzu důležité rovněž pro komunikaci (e-mail, chat) a to jak studenta s lektorem, tak i studentů mezi sebou.

### 1.4.3 LMS

**Learning Management system** [7] – Systém řízení výuky již představuje kompletně propracovaný informační systém. Přistupuje se k němu stejně jako u WBT pomocí webového prohlížeče. Systém řízení výuky ovšem poskytuje sadu nástrojů nejen pro studenty, ale i pro ostatní účastníky e-learningu, kterým umožňuje efektivní správu kurzů, tvorbu testů a správu jejich výsledků a ostatních statistik. Pomocí nich a dalších nástrojů je možné analyzovat a hodnotit celý výukový proces. Kromě toho by měl poskytovat širokou škálu komunikačních kanálů mezi studenty, lektory, případně i autory kurzů a manažery vzdělávání.

Můžeme říci, že LMS promyšleně integruje jednotlivé e-learningové nástroje pro efektivní správu e-learningových kurzů a všeho, co s nimi souvisí, tak aby bylo možné řídit a organizovat výuku a kompetence účastníků. Tyto nástroje, včetně samotného obsahu kurzu a dalších doprovodných aktivit, jsou prezentovány účastníkům prostřednictvím webu.

LMS se zabývá aktivitami z hlediska kompetencí, správou výukových aktivit, jejich distribucí. Nezabývá se však tvorbou obsahu. Proto je nutné v souvislosti s LMS zmínit ještě pojem LCMS.

**Learning Content Management system** (**LCMS**) [16] – Systémy pro tvorbu obsahu kurzu. Jelikož obsahem kurzu může být prakticky jakýkoli dokument, můžeme označit jakýkoli nástroj, kterým tento dokument vytvoříme jako LCMS. V e-learningu se ovšem většinou za LCMS považují specializované systémy podporující aplikaci různých výukových strategií, zpětnovazebné prvky v kurzu a e-learningové standardy pro integraci do LMS. Tyto nástroje se většinou nezaměřují jen na technickou tvorbu kurzu, ale i na týmovou spolupráci, sdílení a distribuci obsahu.

## 1.5 Formy elektronické výuky

E-kurzy můžeme z hlediska časování rozdělit na následující čtyři formy [2]:

- **Synchronní s lektorem** – Vztahuje se k reálnému času. Všichni účastnící mohou navzájem ve stejném čase reagovat na stejné podněty a vstřebávat prezentované vědomosti. Je to v podstatě virtuální třída, kde mohou studenti společně s lektorem spolupracovat a komunikovat pomocí různých technologií (např. videokonference).
  Tento typ kurzu je poměrně náročný, protože vykazuje skoro všechny nároky, jaké jsou u prezenčních kurzů. Znamená to vyčlenit si určitý čas v pevně stanovenou dobu. Jedinou výhodou je, že odpadá doprava do určeného místa [2].

- **Synchronní bez lektora** – Podobně jako u synchronního kurzu s lektorem. S tím rozdílem, že kurz není „moderován“ lektorem, ale je řízen přesným programem. Tyto kurzu nejsou příliš oblíbené, protože působí strojově. Výjimku mohou tvořit různá nemoderovaná on-line diskuzní fóra.

- **Asynchronní s lektorem** – Nevztahuje se k reálnému času. Tato forma umožňuje studentům přistupovat individuálně k probírané látce a „projít“ kurz vlastním tempem, kdy uznají za vhodné. Studenti komunikují s lektorem a ostatními studenty asynchronně pomocí e-mailu, v off-line diskuzích či individuálně (i mimo systém).

  Výhodou oproti synchronním formám je levnější výuka, jednoduchá distribuce a modifikace obsahu.

Nevýhodou naproti tomu může být slabší vedení studentů a tím slabší motivace ke studiu. V asynchronních formách výuky rovněž nemůže lektor improvizovat pro lepší vysvětlení látky.

- **Asynchronní bez lektora** – Kurz je vytvořený jako výhradně samostudijní. Může být pojat výrazně interaktivně. Interaktivita se většinou týká obsahu. V kurzu jsou různé animace a další prvky, které mají učební materiál učinit rozmanitější a podporovat tak studium samotné [2].
  Tato forma je vhodná pro výuku jednoduchých fakt a konceptů. Při nepochopení části probírané látky se student nemá na koho obrátit. Velkou výhodou je možnost najednou a v podstatě za jedny náklady naučit velké množství lidí danou problematiku.
  Proto dnes tato forma kurzů převládá. Někdy (zvláště pokud se touto formou prezentuje složitější problematika) jsou do některých míst kurzu vkládány prvky synchronicity nebo účasti lektora.

## 1.6 Výběr vhodné formy výuky

Při výběru správné metody se přihlíží ke kriteriím jako je komplikovanost výuky, stabilita obsahu, struktura obsahu, časové hledisko, počty zúčastněných a podobnost s realitou [19].

Cílem samozřejmě je vybrat takové metody a formy výuky, aby se při aplikaci na daný obsah (vyučovanou problematiku) v maximální možné míře eliminovaly jejich nevýhody a využili jejich výhody. To nemusí být jednoduché a záleží to na zkušenostech didaktických pracovníků, ale i na finančních možnostech organizace.

Platí, že téměř vždy lze vhodnou kombinací různých typů dodávaní výuky dosáhnout řešení, které je efektivnější a lepší než pouhé použití jedné metody, představované na jednom pólu klasickou učebnovou výukou a na druhém elektronickým kurzem [19].

### 1.6.1 Komplikovanost výuky

Komplikovanost výuky je základním měřítkem při určování metodiky. Pokud se výuka týká jednoduchých fakt, lze s výhodou použít asynchronní formu výuky pomocí elektronického kurzu, ke kterému bude moci student individuálně přistupovat bez potřeby interakce s jinými studenty nebo lektorem.

Pokud se naopak výuka týká složité problematiky jako je aplikování složitých paradigmat v komplexních situacích, bude nejspíše zapotřebí předávání vědomostí a komunikace v reálném čase, tedy synchronní výuka s lektorem.

### 1.6.2 Stabilita obsahu

Vzhledem k vysokým počátečním nákladům na výrobu asynchronního kurzu se nezdá být vhodné jeho použití pro vyučování nestabilního obsahu s krátkou dobou životnosti. Tady bude vhodná synchronní výuka s menšími počátečními náklady a většími náklady na konkrétní kurz.

Naopak pro výuku kurzu se stabilním obsahem s dlouhou dobou životnosti bude výhodné vytvořit asynchronní kurz.

Pokud má obsah dlouhou dobu životnosti, ale je nestabilní, je vhodné vytipovat místa nestability a vytvořit asynchronní kurz, kde se dají snadno měnit vytipovaná místa. Tím lze těžit z výhod jako je jednotná správa obsahu, rychlost reakce na změnu, či možnosti studia v době, kdy to vyhovuje studentovi.

### 1.6.3 Struktura obsahu

Strukturovaný obsah většinou představuje velice přesně definovaná fakta s utříděnými otázkami a zpětnou vazbou. To je vhodné pro asynchronní výuku. Nestrukturovaný obsah naopak implikuje použití synchronní metody výuky, protože studenti musí aplikovat vědomosti v konkrétních situacích a odpovědi na otázky jsou většinou otevřené (viz. kapitola 2.2).

### 1.6.4 Časové hledisko

Pokud jsme v časové tísni, většinou nám nezbývá než zorganizovat synchronní výuku s lektorem, která vyžaduje méně času než tvorba obsahu pro asynchronní kurz.

### 1.6.5 Počet zúčastněných

Velké počty zúčastněných studentů implikují tvorbu asynchronního kurzu, kdy se nám vykompenzují nákladu na tvorbu kurzu, zejména pokud by byly s kurzem spojeny náklady na cestování nebo zabírá-li výuka hodně času. Naopak při menších počtech studentů se jeví jako výhodnější synchronní výuka.

### 1.6.6 Podobnost s realitou

Výuka by měla být pokud možno co nejbližší realitě. Pokud vyučujeme konfiguraci síťových prvků je nejlepší to vyučovat přímo v reálné počítačové síti, nebo pokud vyučujeme jak komunikovat se zákazníky, bude to nejefektivnější v téměř identickém simulovaném prostředí.

Volba formy, prostředí a média pro výuku bude vždy kompromisem mezi snahou o co nejlepší výuku a proveditelností. V našem příkladu bude zřejmě nereálné vyučovat konfiguraci síťových prvků na produkční počítačové síti. Můžeme použít síťovou laboratoř nebo v asynchronním kurzu počítačové programy, které simulují daný síťový prvek. Komunikaci se zákazníkem bude nejefektivnější vyučovat učebně s lidmi, kteří budou představovat zákazníky, tedy synchronně.

## 1.7 Tvorba e-learningového kurzu

Při tvorbě e-learningového kurzu se doporučuje držet se následující šablony:

- Ze všeho nejdříve definovat výukové cíle. Výukové cíle poté určují nejen odborný obsah kurzu, ale i formu a volbu výukové strategie pro prezentaci obsahu kurzu.

- Tvorba pre-testu, pomocí kterého student zjistí své dosavadní znalosti a pomůže mu identifikovat oblasti problematiky, kterým je třeba se věnovat pro doplnění znalostí na požadovanou úroveň.
- Dodání obsahu (včetně implementace výukových strategií). V kurzu by měly být často kladeny praktické otázky a zadávána praktická cvičení. Ve velké míře by se měla využívat interaktivita a obsah by měl být poutavý.
- Tvorba závěrečného testu (tzv. re-testů), na kterém si student ověří dosažené znalosti a ověří svůj pokrok (oproti pre-testu).

E-learningový kurz by měl být také vytvořen v takovém systému nebo prostředí, které umožňuje studentům, aby si zvolili vlastní tempo průchodu kurzem a měli kdykoli k dispozici materiály týkající se vykládané problematiky a testy.

Při volbě výukového postupu se většinou aplikují různé výukové teorie, principy a koncepty a jejich konkrétní volba závisí u každého kurzu na studentech, obsahu a situaci. Je proto nezbytné porozumět slabým a silným stránkám jednotlivých výukových teorií, aby se mohl zvolit jejich optimální mix k dosažení požadovaného výsledku [9].

Obsah je specifická informace, kterou je třeba studentovi podat (prezentovat ji, procvičit a otestovat její přijetí) k dosažení daného výukového cíle. Obsah lze dělit do obsahových domén a ty dále do jemnějších kategorií. Toto dělení umožňuje studentovi dobře identifikovat a zvolit požadovaný kurz, lekci či přímo učební objekt [9].

Výklad výukových teorií a volba výukového postupu včetně didakticky správné tvorby obsahu jde za rámec této práce, proto nám postačí toto rozdělení. Více lze nalézt v literatuře [1], [2] nebo [12].

## 1.8 Výhody e-learningu

- **Úspora nákladů** za lektory, pronájem nebo správu studijních prostor. Další úspory se skrývají v ušetřených cestovních nákladech studentů (u geograficky rozlehlých organizací je cestování zaměstnanců často nerealizovatelné) nebo ve faktu, že školený pracovník nemusí pobývat delší dobu mimo pracovní proces a může (alespoň částečně) plnit své úkoly.
- **Rychlost**, se kterou se dostává studentům školení. S použitím e-learningu je možné školit pracovníky prakticky okamžitě, zatímco při použití klasické výuky se musí čekat až se jich sejde více. A neproškolený zaměstnanec má samozřejmě menší přínos pro společnost.
- Rychlá reakce na změnu obsahu.
- Přizpůsobení výuky i tempa, student si může studovat pouze problematiku, ve které má mezery, čímž se pro něj stává studium atraktivnější a snáze udržuje pozornost.
- Zahrnutí širokého spektra studentů najednou.
- **Lepší** možnost správy znalostí a řízení procesu vzdělávání včetně jeho evaluace pomocí testů. E-learning umožňuje objektivně nastavit požadované cíle a změřit jejich dosažení (např. dosažením 90% úspěšnosti v průběžných testech nebo v závěrečném testu).
- Interaktivní, multimediální a zajímavější vzdělávání.
- Jednoduše opakovatelné kurzy s nízkými náklady na opakování.

## 1.9 Nevýhody a problémy spojené s e-learningem

- **Větší počáteční náklady** na tvorbu kurzu, počáteční investice do řídícího systému a případně do vybavení výpočetní technikou.

- Kurz není tak silně veden lektorem jako v klasické formě vzdělávání. Je potřeba **silná vnitřní motivace** studentů pro dokončení kurzu. Tento problém se daří eliminovat, pokud je kurz správně navržen a pokud může student nabyté znalosti v krátkém časovém horizontu aplikovat.
- Často bývají kurzy dlouhé, což vede k nižší pozornosti a motivaci studenta a v konečném důsledku k malému zapamatování obsahu. Je lepší kurzy **strukturovat do krátkých kapitol**, ve kterých se snáze orientuje. Student může přeskočit kapitoly, které již zná a může snáze nalézt a případně se vracet k určité části kurzu.
- Celková nevhodnost pro určité typy kurzů, zejména pro kurzy, které jsou postavené na vzájemném předávání zkušeností a učení se stylem „pokus-omyl“.
- Možné technické problémy.

# 2 Testy jako integrální součást e-learningu

E-learning přináší nové možnosti nejen v distribuci studijního obsahu a studijních opor, ale rovněž umožňuje využívat nových forem při zajištění zpětnovazebních prvků a to hlavně při samostudiu [4].

O tom, že zpětnovazebné prvky jsou v e-learningu potřeba není třeba vézt dlouhé debaty. Měření efektivity e-learningového kurzu a zajištění jeho didaktické a obsahové správnosti se bez zpětné vazby neobejde. Kontrolní otázky, příklady, cvičení a v neposlední řadě testy slouží nejen k hodnocení práce studenta, ale též k jeho aktivizaci a motivaci. Nelze také opomenout, že dosažené výsledky v testech, příkladech a cvičeních slouží dobře jako zpětná vazba pro samotné studenty. Studenti se tak snáze orientují v tom, které učivo již dobře zvládli a kde mají naopak mezery.

Zvláště při distanční výuce musí učební materiál obsahovat ve zvýšené míře zpětnovazební prvky. Ty jsou většinou řešeny formou aktivizačních úkolů k zamyšlení, praktickými příklady a didaktickými testy. Tyto aktivizační a zpětnovazebné prvky musí v určitém směru nahradit roli učitele, který v prezenční výuce může studenty a jejich pokrok sledovat například pomocí ústního zkoušení a otázek [4].

U zpětnovazebních a aktivizačních testů a cvičení je důležité, aby student nemusel čekat na odpověď a komentáře od tutora, ale aby byl test vyhodnocen hned po jeho dokončení.

## 2.1 Kontrolní otázky a úkoly

Na konci každé kapitoly učiva (ta by měla být přiměřeně krátká) nebo na konci kratších úseků by měl student najít sadu krátkých a většinou jednoduchých otázek a úkolů. Tyto jednoduché otázky a úkoly slouží k aktivizaci studenta, jeho lepšímu soustředění na probírané učivo. Dále pomáhají lepšímu pochopení látky pomocí praktických příkladů. V neposlední řadě slouží studentům jako zpětná vazba pochopení a zvládnutí probírané tematiky.

Nejjednodušší úroveň představují kontrolní úkoly, které se týkají přímo textu výukové lekce a v tomto textu student rovněž nalezne správnou odpověď. Náročnější formou pro studenta jsou úkoly, které na základě vysvětlení v textu vyžadují vlastní úsudek a případné řešení musí student sestavit sám [4]. Tyto otázky a úkoly je dobré formulovat tak, aby odpověď byla jednoduchá a strojově porovnatelná. Jedině tak mohou být otázky a úkoly vyhodnoceny ihned po dokončení studentem.

## 2.2 Didaktické testy

Didaktické testy na sebe často berou podobu tzv. „sebetestování“ [10], kdy má student k dispozici okamžitou zpětnou vazbu, jeho test je tedy automaticky vyhodnocen bezprostředně po jeho dokončení.

Hlavní význam didaktických testů spočívá ve sdělení studentům jak zvládli učivo. Automatické vyhodnocení hned po dokončení a zobrazení výsledků studentům je tedy velice důležité.

V testech se mohou vyskytovat tyto typy úloh [15]:

- Úloha s výběrem odpovědi – jedna správná odpověď
- Úloha s výběrem odpovědi – jedna nejpřesnější odpověď
- Úloha s výběrem odpovědi – jedna nesprávná odpověď
- Úloha s výběrem odpovědi – více správných odpovědí
- Úlohy dichotomické (Ano, ne)
- Přiřazovací úlohy
- Uspořádací úlohy
- Otevřené úlohy – jednoduchá odpověď
- Otevřené úlohy – rozsáhlá odpověď

Testy mohou být vytvářeny jako uzavřené nebo otevřené [11]. Uzavřené úlohy jsou takové, ke kterým jsou předloženy odpovědi, z kterých student vybírá. Nebo obecněji takové úlohy, kde je množina řešení a jeho vyjádření konečná. Naproti tomu u otevřených úloh musí student sám formulovat svou odpověď.

Největší problém v testech představují otevřené úlohy, kde student sám formuluje svoji odpověď. Ty jsou, vzhledem k bohatým vyjadřovacím možnostem jazyka, možnosti rozsáhlého výběru slov a volnosti slovosledu, velice špatně strojově vyhodnotitelné. Proto je nutné formulovat úlohu tak, aby odpověď mohla být jednoznačně a přesně zapsána.

Z výše uvedených důvodu se pro sebetestování hodí spíše úlohy uzavřené. Nejčastěji se používají úlohy s výběrem odpovědi. Tyto typy úloh je vhodné zařazovat jako vstupní testy každé lekce, tzv. „pre-testy“ [2], i jako závěrečné testy, tzv. „re-testy“ [2]:

- **Pre-test** – Student absolvuje tento test jednou před samotným kurzem nebo před každou kapitolou. Pomocí testu může student identifikovat obsah, který zvládá na požadované úrovni a který je potřeba se naučit. Pre-test též poskytne studentovi základní představu co může od kurzu očekávat a slouží pro vyhodnocování pokroku. Někdy je pre-test kritériem k účasti v daném e-kurzu.
- **Re-test** – Jedná se o závěrečný test vykonaný po výuce. Účelem re-testu je ohodnotit jak student zvládl danou tématiku. V porovnání s pre-testem slouží k vyhodnocení pokroku studenta. A v porovnání s určitým očekávaným výsledkem může být použit pro potvrzení požadované úrovně nebo certifikaci.

Někdy používají pre-testy a re-testy stejnou sadu otázek, někdy úplně odlišnou nebo jsou to i testy odlišného druhu. Každopádně je doporučeno náhodné generování otázek ze sady do testu.

## 2.3 Hodnocené testy

Testy rovněž slouží k hodnocení studentů. Z výsledků může lektor vidět v jaké míře studenti zvládají učivo. Na základě výsledků testů může lektor rovněž posoudit zda-li již student dosáhl požadované úrovně a zda spěje ke zdárnému zakončení kurzu. Tyto testy mohou nabývat různých forem.

První forma řešení hodnocených testů může být zcela vzdálená. Student pracuje u svého počítače na libovolném místě a v libovolném čase. Jeho řešení se pak uloží v databázi a lektor si výsledky prohlédne a případně okomentuje. U této formy není možné zcela jednoznačně zajistit, že student vykonal test samostatně. Získané výsledky proto nejsou dostatečně objektivní a reliabilní. Přesto má tato forma smysl, a to zvláště v distanční formě studia. Testy je většinou nutné splnit do určitého data a nutí tak studenta pracovat průběžně a nenechávat studium až na konec semestru nebo kurzu [4].

Formou, u které lze zajistit větší reliabilitu výsledků, je vykonávání testů v učebně v určené době a pod dohledem lektora. Testy mohou mít časové omezení pro jejich vykonání. Nevýhodou, zvláště pak u distanční formy studia, je samozřejmě to, že se studenti musí nutně v určený čas dostavit do vzdělávací instituce. Tato forma je tedy spíše vhodná jako součást závěrečné zkoušky.

Opět platí, že testovací systém by měl zajistit náhodné generování otázek, včetně míchání pořadí otázek a odpovědí tak, aby test nevypadal vizuálně vždy stejně a student se musel nad otázkou zamyslet.

Hodnotící testy lze využívat i mimo rámec e-learningu v klasické prezenční „učebnové“ formě vzdělávání. Zjednoduší se tak práce lektora při závěrečném zkoušení studentů. Hodnotící testy lze použít i v průběhu studia se stejnou motivací jako u distanční formy vzdělávání, tedy nutit studenta, aby studoval průběžně.

Někdy se používají velice podobně koncipované testy jako jsou hodnotící testy k prvotnímu rozřazení studentů do skupin podle dosavadních znalostí, dovedností nebo studijních předpokladů tak, aby mohla další většinou prezenční výuka probíhat se studenty, jejichž úroveň je co možná nejvíce homogenní v rámci skupiny.

## 2.4 Výhody a nevýhody elektronického testování

Elektronické testovací systémy mají následující výhody a nevýhody [10]:

**Výhody**:

- Objektivnost při hodnocení výsledků testů, stejná pravidla pro všechny studenty.
- Variabilita testů, pokud se z databáze otázek na základě určených pravidel generují náhodné testy.
- Možnost „sebetestování“ studentů jak přímo na vyučovací hodině (v prezenční formě), tak i doma.

- Eliminace (alespoň v maximalizované míře) podvádění a opisování. Každý student dostane vlastní variantu testu s jinými otázkami v různém pořadí.
- Ulehčení lektorovi od zdlouhavého opravování někdy velkého počtu testů.
- Vždy aktualizované statistiky úspěšnosti.

**Nevýhody**:

- Absence individuálního přístupu lektora ke studentovi.
- Někdy je velmi těžké formulovat otázku tak, aby očekávaná odpověď byla jednoznačná (otevřené úlohy s textovými odpověďmi).
- Neexistence ekvivalentu některých typů otázek v standardním papírovém testu (např. Nakresli ...).
- Nároky na výpočetní techniku, server s testovacím systémem, vybavenost studentů počítači a připojením k internetu.
- Někdy je technický problém zajistit autentizaci, tedy jestli skutečně test vykonává zkoušený student.

# 3 Některé řídící systémy (LMS)

Univerzita Pardubice rozhodla pro testování studentů použít na zakázku vyrobený modul „Testy“ do IS/STAG (viz. Kapitola IS/STAG). Může tedy s výhodou využívat systém, který je navržen na míru pro potřeby lektorů na pardubické Univerzitě. Ti, jak se ukázalo při analýze požadavků, očekávají od Testů nejvíce podporu klasické prezenční výuky a ulehčení práce a až v druhé řadě podporu e-learningu. Výhoda spočívá i v integraci na Univerzitou využívaný systém IS/STAG a tedy úzkém spojení se studijní agendou a rozvrhovými akcemi (tedy kurzy) Univerzity.

Ačkoli se pardubická Univerzita rozhodla takto, popíši zde pro přehled ještě několik ostatních systémů pro řízení studia, jejichž součástí je i testování studentů.

## 3.1 Moodle

Moodle, zkratka pro Modular Object Oriented Dynamic Learning Environment (tedy modulární objektově orientované dynamické výukové prostředí), je softwarový produkt určený pro podporu výuky prostřednictvím on-line kurzů dostupných přes webové rozhraní. Je možné jej použít pro podporu klasické prezenční i distanční výuky.

Obrázek č. 2 : Logo Moodle


Zdroj: *Moodle* [online]. 2010 [cit. 2010-07-24]. Dostupné z WWW: <http://moodle.org/>.

Výuka v Moodlu je organizována do kurzů, ty mohou být rozděleny do menších částí, témat. Kurzy tvoří a upravuje lektor pomocí nezávislých modulů systému. Lze vytvořit pro studenty úkoly, diskuzní fóra, testy, hodnotit je. Součástí otázek v testu mohou být i obrázky. Veškeré aktivity studentů se zaznamenávají do protokolů, ze kterých lze zjistit mimo jiné i například četnost návštěv konkrétního kurzu.

Moodle je vyvíjen komunitou programátorů z celého světa, jde o volně šiřitelný software s otevřeným kódem podle licence GNU/GPL. Systém stojí na technologii PHP, proto je běh možný na platformách Unix, Linux, Windows, MacOS a dalších, které podporují PHP webové servery. Datovou vrstvu může představovat nejlépe databáze MySQL nebo PostgreSQL, ale také Oracle, Interbase a další. Kromě toho může být systém hostován i na serveru moodle.cz.

Systém Moodle je provozován na Ústavu systémového inženýrství a informatiky na Fakultě ekonomicko-správní Univerzity Pardubice. Problémem je nízká míra integrace na IS/STAG (viz. následující kapitola), z čehož plyne dvojí práce při zakládání studijních skupin (rozvrhových akcí), definování uživatelských privilegií, navazování testů na předměty, atd. Dalším problémem je zřejmá datová závislost aplikace, při updatu na novou verzi si musí všichni aktéři systému (zejména lektoři) provést export dat a po updatu zase import, jinak o data přijdou.

## 3.2 eDoceo

Jedná se o LMS určený pro správu jak prezenčních, tak i distančních elektronických kurzů, včetně testování, vyhodnocování výsledků e-kurzů, certifikování studentů, atd. Systém lze provozovat v rámci podnikového prostředí (intranetu) nebo v rámci sítě internet a je možné ho propojit přímo na personální databáze nebo ERP systémy.

Systém je vyvíjen českou firmou Trask solutions s.r.o. Je ale lokalizován i do angličtiny a slovenštiny. EDoceo stojí na osvědčených a otevřených technologiích Java, J2EE a XML, díky tomu je jeho běhové prostředí nezávislé na platformě. Systém doplňují desktopové aplikace Autor a Off-line Student.

Obrázek č. 3 : Logo eDoceoe


Zdroj: *EDoceo* [online]. 2010 [cit. 2010-07-24]. Dostupné z WWW: <http://www.edoceo.cz>.

### 3.2.1 Autor

Jedná se o externí desktopovou aplikaci určenou pro přípravu e-kurzů z již existujících elektronických dokumentů a materiálů. Pomocí této aplikace lze vytvořit jak kurzy určené pro import do systému eDoceo, tak i kurzy, které fungují samostatně na osobním počítači. Většinou se ale bude jednat o převod do HTML podoby včetně interaktivních prvků jako jsou animace, obrázky se zvukem a interaktivních JavaScriptových prvků nebo Flash animací.

### 3.2.2 Off-line Student

Off-line Student je samostatná aplikace umožňující studium kurzů na klientském počítači bez nutnosti připojení k internetu nebo k systému eDoceo. Aplikace poskytuje prostředí, které je vizuálně shodné s prostředím LMS eDoceo, a to včetně testovacího

modulu. Data, která aplikace během studia generuje, tedy zejména informace o pokroku v kurzu, hodnocení studia a testů studenta, je možné importovat do LMS eDoceo a zajistit tak synchronizaci studijních dat.

Systém eDoceo byl používán Jazykovým centrem pardubické Univerzity. Problémem se ovšem ukázala vysoká cena za provoz systému, malá přizpůsobitelnost potřebám Jazykového centra a to, že systém není integrován na IS/STAG. Tato integrace by byla proveditelná, ovšem za cenu, která nebyla akceptovatelná.

## 3.3 WebCT

Historicky první systém pro řízení výuky, který došel masového rozšíření po celém světě (používá ho více než 10 milionů studentů v 80 zemích světa [25]). Systém byl vyvinut na University of British Columbia v Kanadě [25]. Jedná se o komplexní systém pro správu e-kurzů nebo jen pro doplnění klasické prezenční výuky.

Obrázek č. 4 : Logo Web-CT


Zdroj: *Univerzita Hradec Králové* [online]. 2010 [cit. 2010-07-24]. WebCT Campus Edition. Dostupné z WWW: <http://oliva.uhk.cz/webct/entryPage.dowebct>.cz>.

Jednoznačnou výhodou systému je jeho flexibilita, je tak možné opravdu vytvářet kurzy podle uživatelových přestav. Další výhodou, která plyne z rozšířenosti systému, je existence zdrojů kvalitních materiálů pro tento systém. Je tak možné zvýšit kvalitu svých kurzů.

Někteří uživatelé WebCT kritizují kvůli složitosti používání. To je z části daň za vysokou míru flexibility. Kritika se též snáší na webové rozhraní, které používá příliš mnoho HTML framů (tedy rámců oken). To způsobuje odlišné chování v různých prohlížečích.

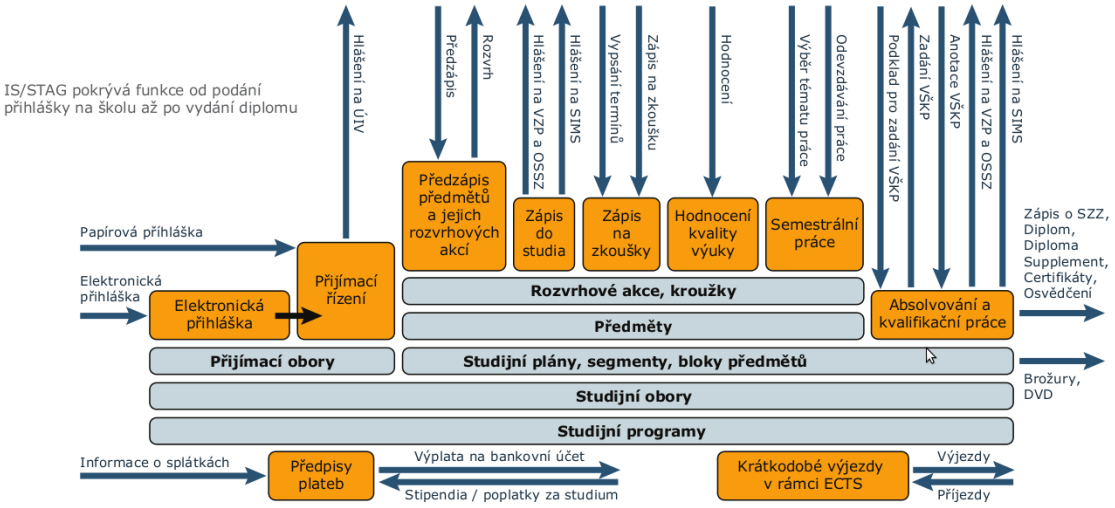
Dnes již WebCT vlastní firma Blackboard Inc., které se zabývá vývojem LMS Blackboard. Oba systémy tak byly sloučeny.

# 4 IS/STAG

**IS/STAG** je Informační Systém STudijní AGendy vysoké školy, univerzity nebo vyšší odborné školy. Systém vznikl a je vyvíjen Centrem informatizace a výpočetní techniky - Střediskem informačních systémů na Západočeské univerzitě v Plzni [26].

Poprvé se IS/STAG použil v roce 1993 na ZČU. V současnosti je na ZČU provozována v rutinním provozu již třetí vývojová verze tohoto systému. IS/STAG je celouniverzitní systém určený pro administraci studia, nikoliv pro podporu studia. IS/STAG eviduje jak kreditní, tak i nekreditní systém studia [26]. Systém podporuje administraci procesu studia od přihlášení studenta po vydání diplomu. Přehledně je to vidět na schématu IS/STAGu:

Obrázek č. 5 : Schéma systému IS/STAG




Zdroj: *IS/STAG [online]. 2009 [cit. 2010-07-26]. Dostupné z WWW: <http://is-stag.zcu.cz>.*

Systém v současnosti používá 16 škol v České republice. Z toho 11 je veřejnoprávních vysokých škol či univerzit, 4 jsou soukromé vysoké školy a 1 je vyšší škola odborná [26]. Systém používá i Univerzita Pardubice.

Aplikace je provozována nad databází Oracle a vyhovuje portálové specifikaci JSR-168 (viz. kapitola Portály). Je tvořena jádrem systému a několika dalšími moduly. Škola může systém provozovat na svém serveru nebo může využít hostování jak databáze tak i portálového serveru na serverech Univerzity Tomáše Bati (UTB).

Databázi IS/STAG reprezentuje poměrně rozsáhlý diagram, jehož součástí je nepřeberné množství tabulek, které mají poměrně velký počet atributů, jejichž název většinou moc nenapoví o jejich významu. Proto firma Seico s.r.o. provedla ve spolupráci s Informačním centrem pardubické Univerzity prvotní analýzu databáze a vytvořila zjednodušené schéma, kde jsou pouze tabulky, které se jeví jako potřebné pro integraci modulu Testy na IS/STAG (viz příloha č. 1).

Z diagramu jsou vidět zejména vazby mezi studenty, učiteli, předměty, obory a rozvrhovými akcemi.

# 5 Modul Testy – použité technologie

Modul Testy je vystavěn pomocí Java technologií, vyhovuje portálové specifikaci JSR-168, aby byl integrovatelný do portálu IS/STAG a je napojen na CAS (centrální autentizační službu) Univerzity Pardubice. Tato kapitola si klade za cíl seznámit čtenáře (byť telegraficky) s technologiemi, na kterých je vystavěn a které používá modul Testy.

## 5.1 Platforma Java

Jedná se počítačovou platformu, která v sobě agreguje různé části Javy určené pro vývoj různých typů aplikací. Platforma Java zastřešuje v současné době tyto dílčí platformy:

- **Java SE** – pro vývoj aplikací používaných na stolních počítačích,
- **Java ME** – pro aplikace provozované na mobilních zařízeních (telefony, PDA, atd.)
- **Java EE –** pro aplikace pro podnikové a rozsáhlé informační systémy.

## 5.2 Java SE

Java SE (standart edition) představuje základ pro ostatní dílčí platformy, jedná se o jeden z nejpoužívanějších programovacích jazyků na světě. Je to objektově orientovaný jazyk vyvinutý firmou Sun Microsystems, která ho představila 23. května 1995. Od května roku 2007 byla uvolněna většina zdrojových kódů a nadále je Java vyvíjena jako open source.

Důležité vlastnosti jazyka Java:

- **Objektově orientovaný jazyk** - s výjimkou 8 primitivních datových typů jsou všechny ostatní datové typy objektové.
- **Interpretovaný** jazyk – místo strojového kódu se vytváří tzv. bajtový kód, který je nezávislý na architektuře počítače a operačním systému. Tento kód je spouštěn pomocí interpretru Javy, tzv. JVM (Java Virtual Machine – virtuální stroj Javy).
- **Jednoduchá syntaxe** – syntaxe je zjednodušenou a drobně upravenou syntaxí jazyka C a C++.
- **Distribuovaný** – podporuje různé úrovně síťové komunikace.
- **Bezpečný** – neumožňuje některé programátorské konstrukce, které vedou často k chybám (příkaz goto, používání ukazatelů, ruční správa paměti). Java je silně typová, každá proměnná musí mít pevně definován svůj datový typ, který se po celou dobu života proměnné nemění.
- **Přenositelný** – vedle výše zmíněné nezávislosti na architektuře počítače a operačním systému díky interpretru bajtového kódu poskytuje Java nezávislost i na úrovni datových typů. Každý primitivní datový typ má explicitně stanovenou velikost nezávislou na cílové platformě).
- **Vícevláknový** – podporuje vícevláknové programování.

- **Automatická správa paměti** – programátor pouze konstruuje nové objekty, o jejich odstranění z paměti se stará tzv. Garbage Collector, který hledá již nepoužívané objekty a odstraňuje je z paměti.

Tím, že je program v Javě překládán do bajtového kódu, který se musí interpretovat, jsou programy v Javě pomalejší než programy psané v nativních jazycích. Ovšem i toto se daří eliminovat. V posledních verzích Javy se používá mechanismus JIT (just in time - „právě v čas“) a HotSpot, kdy se překládají jen ty části kódu, které jsou opravdu potřeba a často prováděné části kódu se přeloží do strojového kódu. Více viz. [28] nebo [18].

## 5.3 Java EE

Java Enterprise Edition (dříve též J2EE – Java 2 Enterprise Edition) je, jak již bylo zmíněno, součástí platformy Java a je určena pro vývoj podnikových aplikací a informačních systémů. Základem pro Java EE je platforma Java SE, nad ní jsou definovány všechny součásti platformy Java EE. Jedná se o :

- **Vývoj webových aplikací** – Java **Servlety** a **JSP** (Java Server Pages),
- vývoj sdílené business logiky – EJB (Enterprise Java Beans),
- přístup ke zprávovému middleware – JMS (Java Messaging Service),
- podpora technologií webových služeb,
- **Portlety** – integrace webových aplikací a portálů
- přístup k jiným systémům a zdrojům dat – JCA (Java Connector Architecture).

Zvýrazněné technologie jsou použity v modulu Testy.

Pomocí Servletů a JSP je umožněn snadný vývoj webových aplikací podle návrhové vzoru MVC (Model View Controller). Je tak možné oddělit business logiku aplikace od prezentační vrstvy a vrstvy přístupu k datům. Samozřejmostí je využívání

„dobrých objektově orientovaných principů“ z platformy Java SE. Servlety navíc zpracovávají standardně každý požadavek klienta v samostatném vlákně, což je rychlejší než zpracování samostatným procesem (jak je tomu např. u PHP). Vícevláknové zpracování také umožňuje zavést do aplikací některé další užitečné návrhové vzory jako je například tzv. „pooling“ databázových spojení, tedy vytvoření několika připojení do databáze a jejich následné přidělování. To umožňuje snížit režii při přístupu k databázi. Více viz. [27].

## 5.4 Spring framework

Spring framework je sadou užitečných tříd a postupů pro programování nejen podnikových aplikací a informačních systémů. Tento framework si klade za cíl zefektivnit vývoj aplikací a umožnit flexibilnější architekturu aplikací. Jedním z hlavních principů Springu je zjednodušení komplexnosti vývoje, neinvazivnost (framework neovlivňuje kód aplikační logiky), zaměření na správnou architekturu aplikace, modulárnost – lze použít jen část.

Spring framework pokrývá širokou škálu aplikačních domén. Od zjednodušení tvorby webových formulářů a prezentační vrstvy, přes architekturu controllerů, zjednodušení práce s JMS, JavaMail a dalších až po přístup k datům a objektově relační mapování.

V modulu Testy bylo ze Springu použito pouze zjednodušení přístupu k datům do databáze a objektově relační mapování, deklarativní správa databázových transakcí a správa některých servisních tříd aplikační logiky aplikace. O Springu více viz. [29].

## 5.5 JSR-168 Portálová specifikace

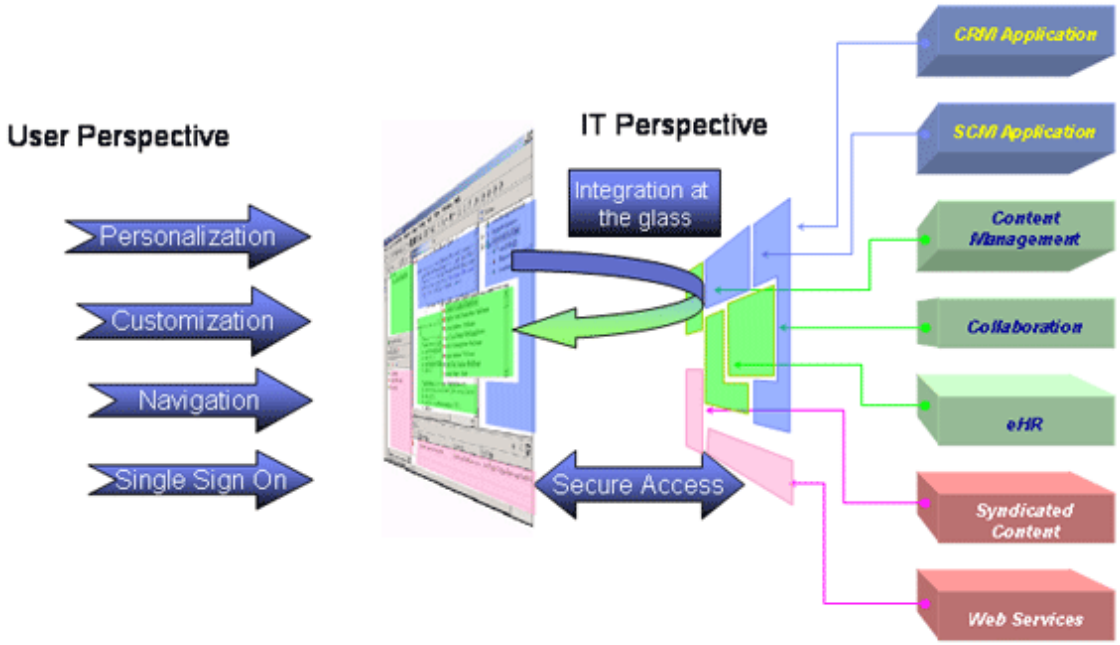
**Portál** je možné vnímat jako webovou aplikaci, která umožňuje agregaci obsahu z různých zdrojů a počítačů s možností personalizace, poskytující jednotnou prezentační vrstvu všem agregovaným aplikacím. Portál na jedné stránce obsahuje různé portlety. Portál většinou umožňuje jediné přihlášení uživatele pro přístup k různým aplikacím (single sing-on).

**Portlet** je Java komponenta, která zpracovává požadavky klientů a generuje obsah. Na stránce se zobrazuje v samostatném okně (fragment stránky), které lze podle nastavení portálu minimalizovat, maximalizovat podobně jako desktopové aplikace.

**Portletová aplikace** je webová aplikace sdružující několik souvisejících portletů. Portletovou aplikaci lze nasadit jako modul na portál. Portlety z jedné aplikace se mohou zobrazovat i na více různých stránkách portálu.

Z pohledu uživatele je to nejlépe vidět na obrázku:

Obrázek č. 7 : Integrace aplikací v portálu




Zdroj: *HEPPER, Stefan; LIESCHE, Stefan. IBM [online]. 2005 [cit. 2010-07-26]. Exploiting the WebSphere Portal V5.1.0.1 programming model. Dostupné z WWW: <http://www.ibm.com/developerworks/websphere/library/techarticles/0512_hepper/0512_hepper.html>.*

Portletovou aplikací je například modul Testy, který agreguje několik portletů (např. administrace otázek a množin otázek, testování, výsledky testů, atd.). Tato aplikace je nasazena na portálový server, kde běží portálová aplikace IS/STAG.

## 5.6 CAS

CAS (Central Authentication Service – centrální autentizační služba) je protokol pro zprovoznění single sing-on na webu. Umožňuje uživatelům, aby přistupovali k několika zabezpečeným aplikacím a poskytli své přihlašovací údaje (uživatelské jméno a heslo) pouze jednou. V některých případech umožňuje protokol autentizaci (tedy získání identity uživatele) i bez získávání hesla.

V CAS protokolu se předpokládají nejméně tři strany. Klientský webový prohlížeč, webová aplikace vyžadující autentizaci uživatele a CAS server. Ve chvíli kdy klient přistoupí poprvé k aplikaci, ta ho přesměruje na CAS server. CAS server ověří klientské identifikační údaje (nejčastěji uživatelské jméno a heslo) proti databázi, LDAPu, Active Directory, aj.

Pokud autentizace proběhla úspěšně, CAS poskytne klientovi tzv. „security ticket“ a vrátí klienta na webový server s aplikací. Aplikace potom pomocí zabezpečeného spojení ověří „ticket“ proti CAS serveru. Pro připojení k CAS serveru musí webová aplikace poskytnout serveru vlastní identifikátor a vlastní CAS „ticket“. Poté CAS poskytne aplikaci důvěryhodnou informaci o tom, jestli je daný uživatel úspěšně autentizován.

Na Univerzitě Pardubice se používá jako CAS server CoSign a portál IS/STAG (tedy i modul Testy) tento server využívá. Mezi ostatní často používané autentizační služby patří např. JaSig nebo OpenID.

# 6 Výchozí stav na Univerzitě Pardubice

Před nasazením modulu Testy nebyl na pardubické Univerzitě jednotný testovací systém, který by mohli snadno využívat všichni lektoři a studenti Univerzity.

Někteří lektoři nepoužívali žádnou aplikaci a prováděli testování na papíře. Na některých katedrách vyvíjeli své aplikace, někdo používal již existující aplikace pro testování studentů. Na Dopravní fakultě Jana Pernera (DFJP) se vyvíjela aplikace pro

testování studentů. Na Fakultě elektrotechniky a informatiky (FEI) se jich vyvíjelo dokonce několik, někteří učitelé dodnes používají své aplikace pro testování. Na Ústavu systémového inženýrství a informatiky Fakulty ekonomicko-správní (ÚSII FES) se používá e-learningový systém Moodle popsaný v kapitole 3.1. Jazykové centrum používalo systém eDoceo (popsaný v kapitole 3.2).

Žádná z těchto aplikací nebyla integrována na IS/STAG, tudíž nemohla využívat výhod, které tato integrace přináší, čímž je zejména napojení testů na předměty, vypisování testů pro studenty na rozvrhových akcích, vypisování rozřazovacích testů pro první ročníky, atd. Tyto aplikace byly vyvíjeny na technicky zaměřených fakultách. Méně technicky zaměřené fakulty většinou nepoužívali žádnou aplikaci.

Nasazením aplikace Testy na portál Univerzity se zpřístupnila funkcionalita testování všem učitelům pardubické Univerzity. Ti si tak bez dalších činností mohou vytvořit sadu testů a vypisovat je pro studenty na jejich rozvrhových akcích.

# 7 Výchozí stav modulu Testy

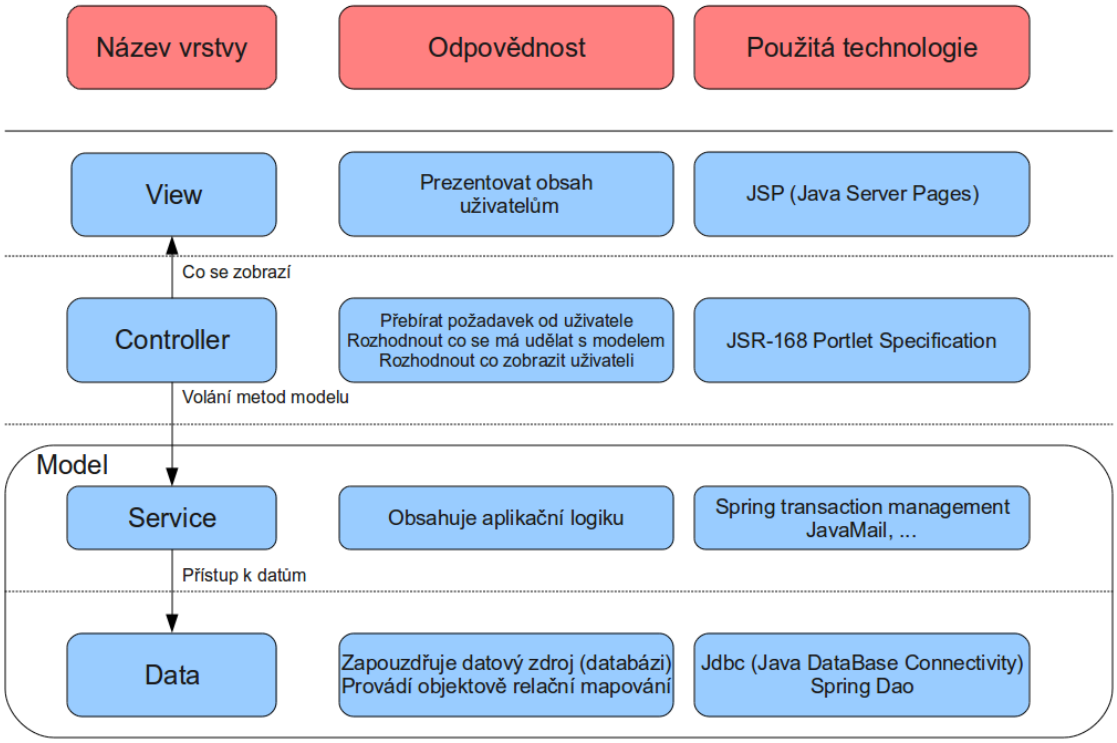
Jak je uvedeno v zadání, počáteční analýza a komunikace o potřebách uživatelů ohledně testování studentů probíhala téměř výhradně s Jazykovým centrem Univerzity Pardubice. Z toho vyplývá výchozí stav modulu Testy. Před zadáním této diplomové práce bylo hotovo:

- jádro aplikace Testy,
- definice otázek a množin otázek,
- definice a plánování testů, včetně rozdělení naplánovaných testů na hodnotící (pro hodnocení studentů) a rozřazovací (zejména pro potřeby Jazykového centra, kvůli rozřazení studentů na skupiny podle úrovně znalostí),
- frontend pro studenty, kde vykonávají test,
- prohlížení výsledků testů pro učitele i studenty.

## 7.1 Architektura Testů

Aplikace Testy je od začátku postavená jako čtyřvrstvá portletová aplikace a ctí návrhový vzor MVC (Model View Controller). Odpovědnosti jednotlivých vrstev a jejich vzájemná komunikace je nejlépe vidět na následujícím diagramu:

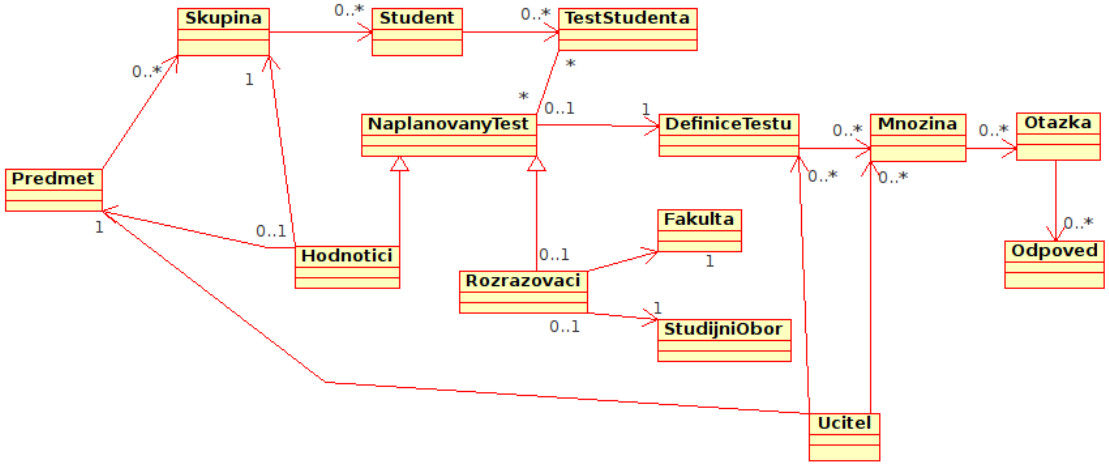
Obrázek č. 8 : Čtyřvrstvá architektura aplikace Testy




Zdroj: vlastní tvorba

Vztah entit (tříd) je přehledně vidět na analytickém třídním diagramu výchozího stavu. Je zde vidět, že definice testu se skládá z několika množin otázek, z nichž mohou být generovány otázky do testu. Každá množina otázek má několik otázek a každá otázka několik možných odpovědí. Učitel může mít několik definic testů, které může naplánovat na určitý čas pro vybranou skupinu studentů (na předmětu, rozvrhové akci, na oboru, fakultě). Naplánováním testu vytvoří „Naplánovaný test“. Při spuštění naplánovaného testu studentem se vytvoří „Test studenta“.

Obrázek č. 9 : analytický diagram tříd modulu Testy ve výchozím stavu




Zdroj: Upravený diagram z firemních materiálů Seico s.r.o.

Podrobnější pohled na entity včetně jejich atributů aplikace Testy nabízí databázový model (e-r diagram) v příloze č. 2.

# 8 Analýza požadavků na rozšíření Testů

V této kapitole se podíváme na požadavky lektorů, kteří začali nebo chtějí začít používat modul Testy na portálu pardubické Univerzity. Kromě analýzy požadavků zde budu prezentovat i návrhy postupů jak řešit konkrétní požadavky a skupiny požadavků.

Požadavky a jejich analýza byla předána firmě Seico s.r.o. V součinnosti s Informačním centrem pardubické Univerzity proběhl schvalovací proces a bylo zvoleno, které požadavky se implementují a jakým způsobem. V každé kapitole je podkapitola implementace, kde je popsáno jestli a jak se daný požadavek implementoval.

## 8.1 Kontrola zápisů předmětů

Tento požadavek vzešel od Jazykového centra jako praktické doplnění již hotových rozřazovacích testů. Rozřazovací testy slouží pro rozřazení studentů na kurzy podle úrovně znalostí. Student tedy vykoná test a na základě výsledku testu si zapíše rozvrhovou akci příslušného kurzu.

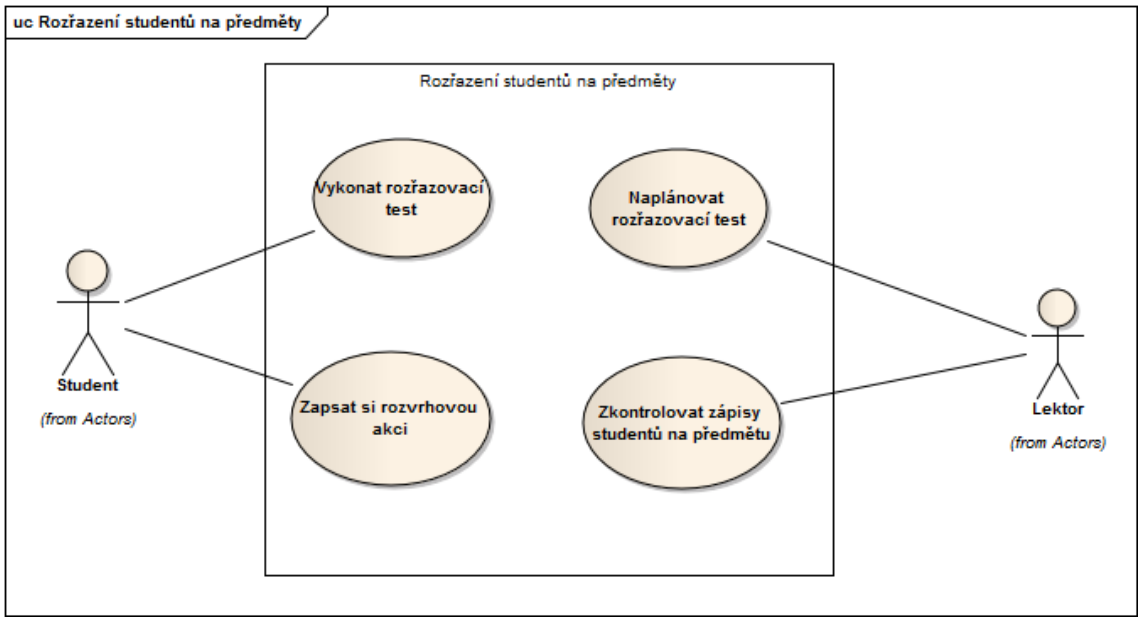
Chybí ovšem zpětná kontrola, jestli si studenti zapsali ten správný předmět podle výsledků testů. Základní potřeba tedy je moci zajistit zápis studentů na správný předmět nebo toto následně zkontrolovat.

Na jeden předmět se mohou studenti zapsat na základě ne pouze jednoho, ale i více rozřazovacích testů. Například na předmět „Angličtina – pokročilá úroveň“ se zapisují studenti na základě rozřazovacího testu vypsaného pro 1. ročník FES, též se na něj ale zapisují studenti na základě rozřazovacího testu vypsaného pro 1. ročník Fakulty filozofické (FF), tedy dvou různých testů.

Jedním z možných řešení by bylo napojit rozřazovací test již při naplánování na předměty pro které slouží a definovat intervaly úspěšnosti pro zápis jednotlivých předmětů. Následně pak nedovolit zapsání jiného předmětu než toho správného podle dosažené úspěšnosti v testu. Problém tohoto přístupu je, že v době vypsání testu ještě nemusí být v IS/STAGu tyto předměty (pro aktuální akademický rok) definovány. Navíc by se toto omezení muselo udělat v portletu pro zápis rozvrhových akcí, který je vyvíjen Západočeskou Univerzitou (ZČU) v Plzni. To znamená přesvědčit vývojáře IS/STAGu o změně funkcionality, což není příliš reálné.

Další možností je implementovat tu samou logiku ale pomocí databázového triggeru. To by ovšem generovalo pro uživatele portálu dost nesrozumitelné chybové hlášení (pokud to není ošetřeno ještě v aplikaci). V obou případech by bylo také problematické zapsat studenta po nějaké době na předmět jiné úrovně než dosáhl v testu (může se stát, že se v průběhu výuky zjistí, že má student ve skutečnosti jinou úroveň znalostí, než na kterou napsal test). Řešení by tedy mělo být spíše ve formě následné kontroly než automatického omezení:

Obrázek č. 10 : diagram případů užití rozřazení studentů a kontroly zápisu předmětů

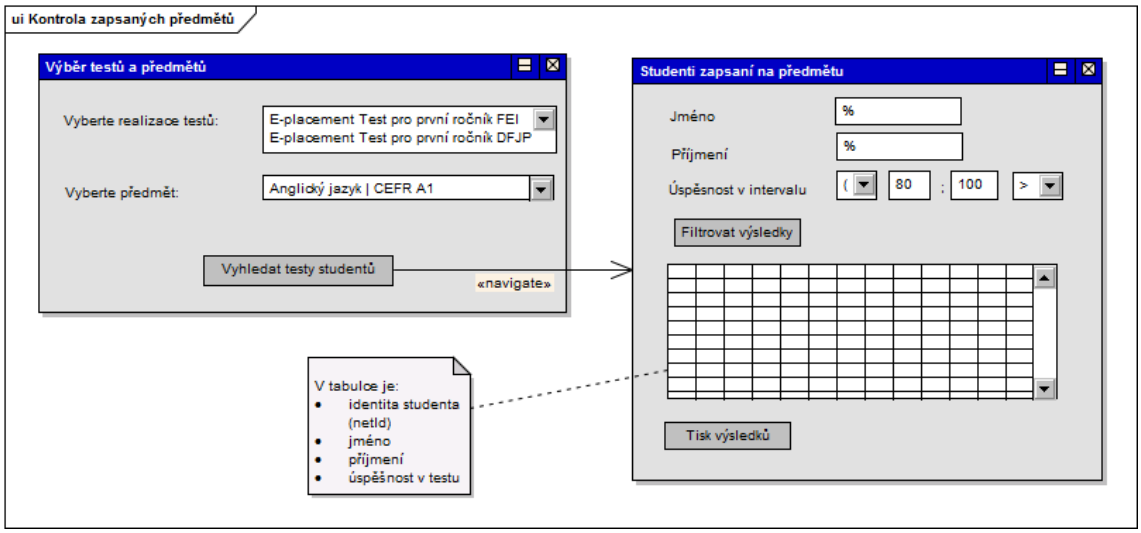



Zdroj: vlastní tvorba

Protože ani tak nejde o kontrolu studentů, kteří si zapsali správný předmět, ale spíše studentů, kteří si zapsali jiný předmět než měli, mohlo by řešení vypadat následovně:

Pracovník Jazykového centra vybere předmět, u kterého chce kontrolovat zápisy studentů, a všechny rozřazovací testy na základě jejichž výsledků si mohou studenti zapsat vybraný předmět. Potvrdí výběr a zobrazí se seznam absolvovaných testů (které byly ve výběru) studentů, kteří jsou zapsáni na vybraný předmět. Tento seznam bude možno filtrovat podle jména a příjmení studenta a hlavně podle intervalu úspěšnosti testu, který vykonal. Takto bude možno odfiltrovat studenty, jejichž výsledek testu je v požadovaném intervalu, a v seznamu zbudou pouze studenti, kteří si předmět zapsali neoprávněně. Nejlépe je to vidět na diagramu uživatelského rozhraní:

Obrázek č. 11 : diagram uživatelského rozhraní kontroly zápisů předmětů




Zdroj: vlastní tvorba

Toto řešení nevyžaduje zásahy do jiných aplikací portálu než Testů. Nebrání ani změně zápisu studentů při zjištění odlišné úrovně znalostí od těch, které byly zjištěny v rozřazovacím testu. Dokonce nevyžaduje ani žádné změny v databázi aplikace Testy, což je v běžícím systému vždy nepříjemné kvůli nutnosti následné migrace dat. Ta by v případě prvního zmíněného řešení ani nešla udělat automaticky (automaticky zjistit, kvůli kterým předmětům jsou rozřazovací testy vypsány).

### Implementace

Požadavek byl implementován způsobem zpětné kontroly zápisů předmětů proti rozřazovacím testům, tak jak bylo navrženo.

## 8.2 Zabezpečení spuštění testu heslem

Na tomto požadavku se shodli lektoři napříč fakultami. Jde o zamezení spuštění testu studentem, který nebude znát heslo pro spuštění. Heslo se vygeneruje při vypsání testu. To může sloužit zejména jako dodatečné zúžení skupiny studentů, kteří mohou psát test, v případě, že nebudou stačit standardní volby napojení (na předmět, rozvrhovou akci u hodnotících testů, fakultu, obor, ročník, typ studia u rozřazovacích testů).

Jelikož je implementace tohoto požadavku součástí návrhu řešení jiného požadavku, bude se tomuto tématu mimo jiné věnovat kapitola 8.3 Zabezpečení spouštění testu v učebně.

### Implementace

Zabezpečení spuštění testu heslem je součástí implementace požadavku z následující kapitoly.

## 8.3 Zabezpečení spouštění testu v učebně

Tento požadavek vzešel od lektorů FEI, pro které je to zcela zásadní a prioritní požadavek a bez jeho implementace je pro ně použitelnost Testů výrazně omezena. Jde o zajištění skutečnosti, že všichni studenti, kteří vykonávají test, ho vykonávají v učebně, kde je přítomen lektor.

Tím se zajistí spolehlivost testu, eliminuje se tedy opisování nebo vykonávání testu pod jinou identitou.

Nejlepším řešením by bylo naplánovat test přímo pro konkrétní učebnu, kterou lektor vybere ze seznamu při plánování testu. Systém by pak sám zajistil, že se test nespustí z jiného místa než z vybrané učebny. Po konzultaci s Informačním Centrem Univerzity jsem zjistil, jaký je manévrovací prostor při případné implementaci tohoto řešení.

### 8.3.1 Ověřování proti Active Directory

Jedna možnost je napojit Testy na systém AD (Active Directory), který je na Univerzitě provozován. Jedná se o adresářovou službu vyvinutou firmou Microsoft. Tato adresářová služba mimo jiné poskytuje přístup přes protokol LDAP a autentizaci počítačů v síti založenou na protokolu Kerberos.

Po drobných úpravách struktury a obsahu AD ze strany Informačního Centra by šlo za pomoci této službu získat seznam učeben. V aplikaci Testy by šlo identifikovat jestli byl test spuštěn z počítače, který je přihlášen v doméně pardubické Univerzity, a v které je tento počítač učebně. To znamená v podstatě ideální řešení, bohužel realizovatelné pouze u učeben s počítači, na kterých je operační systém Windows a které jsou v doméně. To nejsou zdaleka všechny učebny na Univerzitě.

V některých učebnách jsou operační systému Unixového typu. V těchto učebnách by se musela implementovat autentizace proti jinému autentizačnímu serveru (např. přes Kerberos). Zřejmě by na Unixových systémech nešlo dost dobře implementovat přihlášení a autentizace v doméně Windows, tak aby šlo následně použit AD. To znamená, že by se v Testech ověření spuštění testu z učebny rozpadlo na více dotazů na více různých služeb.

Na některé učebnách není přihlašování do univerzitní domény (např. síťové laboratoře). Je to kvůli příliš restriktivní politice nastavené v doméně pro účely síťových laboratoří, kde je potřeba experimentovat s konfigurací počítače a sítě.

Do budoucna by navíc mohl nastat problém při spouštění testu studentem, který nemá roli „student“ v rámci pardubické Univerzity (tento požadavek byl nakonec vznesen viz. Kapitola 8.12). To by při tomto již tak dost složitém řešení vyžadovalo užší spolupráci Informačního centra.

### 8.3.2 Ověřování pomocí adres IP

Toto řešení již bylo implementováno v aplikaci, která se pro účely testování studentů vyvíjela v minulosti na FEI. Principem je zjištění adresy IP počítače, ze kterého se někdo snaží spustit test. Následně tuto adresu porovnat proti nějakému seznamu, kde jsou uchovány učebny a jejich počítače s adresami IP.

V univerzitním síťovém prostředí se adresy IP přidělují automaticky pomoci DHCP serveru. Ovšem ani z DHCP serveru a ani z jiného systému nejde automaticky zjistit adresu, která byla přidělena konkrétnímu počítači v učebně. Adresy IP se nepřidělují po rozsazích logicky stejných jako jsou skupiny počítačů v učebnách.

Na FEI byly seznamy adres udržovány ručně, což by zřejmě musely být v tomto systému také. Problém je v tom, že se adresy přidělované DHCP serverem mohou každý rok změnit. Další problém spočívá v tom, že opět ne všechny počítače ve všech učebnách mají adresy IP přidělené z univerzitního DHCP serveru (opět např. síťové laboratoře).

### 8.3.3 Poloautomatické řešení pomocí kombinace omezení

Jako nejlepší řešení, které je otevřené budoucím změnám ať už v univerzitním prostředí nebo v samotné aplikaci Testy, se jeví sada logických pravidel, které, když budou splněné, zajistí, že všichni studenti spustili test z učebny. Nebude tak zapotřebí aplikaci napojovat na další systémy, které podléhají změnám v čase. Na druhé straně toto řešení vyžaduje součinnost lektora.

Návrh řešení je následující. Lektor vypíše test (vytvoří naplánovaný test). Ještě před vypisováním musí lektor spočítat počet studentů, kteří jsou přítomni v učebně a budou psát test. V naplánovaném testu nastaví maximální počet studentů na počet studentů, kteří jsou v učebně. Též nastaví, aby bylo spuštění testu chráněno heslem.

Následně lektor sdělí studentům v učebně heslo. Studenti test spustí. Poté musí lektor fyzicky (pohledem na monitory) zkontrolovat, jestli mají všichni studenti spuštěn test.

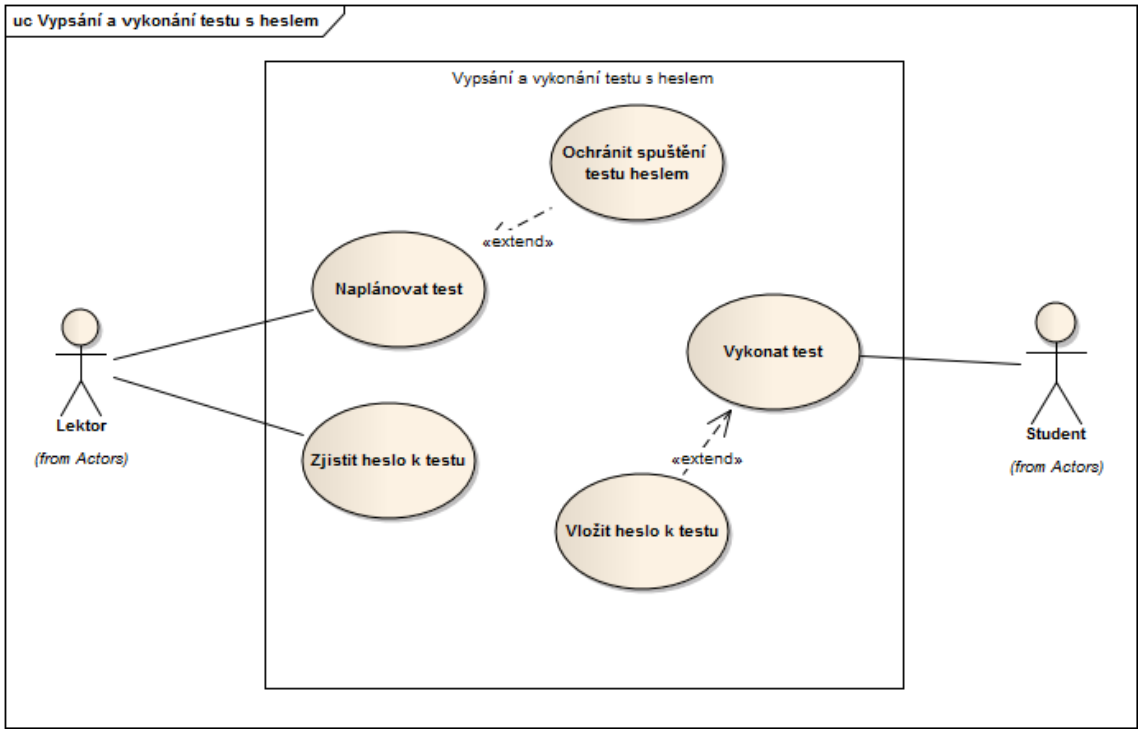
Pokud někdo test nepíše, mohl např. emailem poslat kód pro spuštění někomu jinému, kdo za něj test vykoná. Ale jelikož student může psát jeden test pouze jednou, tak už mu nejde spustit.

Jestli mají všichni studenti test spuštěn, tak, jelikož student může psát test pouze jednou a všichni v učebně test vykonávají a na test se nemůže přihlásit více studentů než je maximální počet (ten je stanoven na počet studentů v učebně), je zabezpečeno, že test psali všichni studenti v učebně a nikdo ho nepsal mimo učebnu ať už za sebe nebo za někoho jiného.

Lektor může i v průběhu vykonávání testů studenty kliknout na příslušný naplánovaný test v portletu „Prohlížení výsledků testů“ (naplánovaný test se objeví poté co první student test spustí). Následně se zobrazí aktuální seznam studentů, kteří test vykonávají s jejich průběžnými výsledky. To je dobré pro kontrolu, kdo test vykonává.

K tomu aby mohlo být toto implementováno je třeba dodělat ochranu spuštění testu heslem, což byl jeden z požadavků. Při vypisování testu se zvolí, jestli má být test chráněn heslem (a vyplní se heslo nebo se může nechat generování hesla na systému). Pokud ano, student bude muset zadat před spuštěním testu správné heslo.

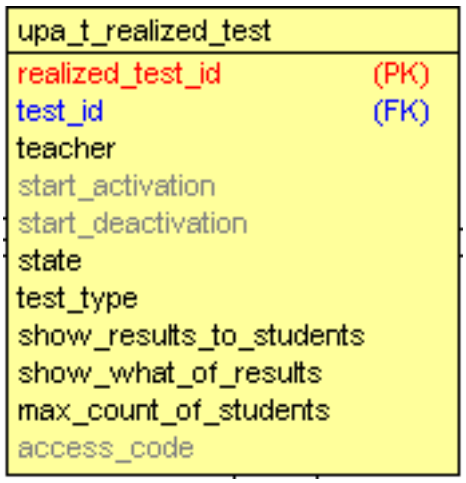
Obrázek č. 12 : diagram případu užití vypsání a vykonání testu chráněného heslem




Zdroj: vlastní tvorba

K tomu, aby mohla být implementována ochrana testu heslem, je třeba přidat do databáze do tabulky s naplánovanými testy atribut, který ponese heslo. Pokud nebude tento atribut vyplněn, test nebude chráněn heslem.

Obrázek č. 13 : e-r diagram ochrana testu heslem




Zdroj: vlastní tvorba

Samozřejmě pokud chce mít lektoru jistotu o tom, kdo psal test, je potřeba zkontrolovat identity studentů, kteří se zobrazují v systému jako studenti, kteří vykonali test, proti identitám fyzicky přítomných studentů v učebně. To platí ať už se tento požadavek implementuje jakýmkoli způsobem.

### Implementace

Tento požadavek byl nakonec implementován způsobem popsaným v kapitole 8.3.3. Důvodem byla zejména nezávislost implementace tohoto požadavku na ostatních systémech Univerzity. Implementace tohoto řešení je tedy časově stálá.

## 8.4 Generování známek podle úspěšnosti v testu

Jedná se o požadavek zobrazení známek studentů na základě jejich úspěšnosti v testu. Úspěšnost v testu je nyní vyjádřena procentuálně.

Tento požadavek je v podstatě již implementován. Na stránce „Prohlížení výsledků testů“ si může lektor pro každý vypsaný test zobrazit výsledky testů jednotlivých studentů. Ty si může exportovat do formátu PDF. Může si též seřadit testy podle procentuální úspěšnosti nebo si exportovat do PDF pouze ty testy, jejichž procentuální úspěšnost je ve zvoleném intervalu.

Je tak možné exportovat si pouze ty studenty a jejich výsledky testu, kteří dosáhly požadované úspěšnosti pro danou známku. Takto se vytvoří seznamy studentů se stejnou známkou.

Co se týče přímo zapisování známek do IS/STAGu na základě výsledků testů, tomuto tématu se věnuje kapitola 10.2.

### Implementace

Generování známek podle úspěšnosti v testu již v podstatě bylo implementováno (jak již bylo popsáno výše). Tudíž se v tomto neudály žádné změny.

## 8.5 Viditelnost testů vypsaných pro předmět

Tento požadavek se týká viditelnosti naplánovaných testů ostatními lektory. Jeden vyučující z FEI vznesl požadavek, aby garant předmětu automaticky a vždy viděl všechny naplánované a realizované testy napojené na předmět, který garantuje. Garant by tak měl přehled o výsledcích studentů a o obsahu vykonávaných testů.

To by šlo realizovat změnou aplikační logiky, respektive DAO (data access object) vrstvy tak, aby se z databáze vybíraly nejen realizované testy, jejichž je přihlášený uživatel autor nebo mu jsou sdíleny, ale i ty, které jsou napojeny na předmět nebo rozvrhovou akci předmětu, který přihlášený uživatel garantuje.

Pokud garant garantuje více předmětů, mohla by mu tato funkcionalita poměrně dost znepřehlednit prohlížení výsledků testů. Automatické zviditelnění naplánovaných testů a jejich výsledků také nemusí vyhovovat všem lektorům.

### Implementace

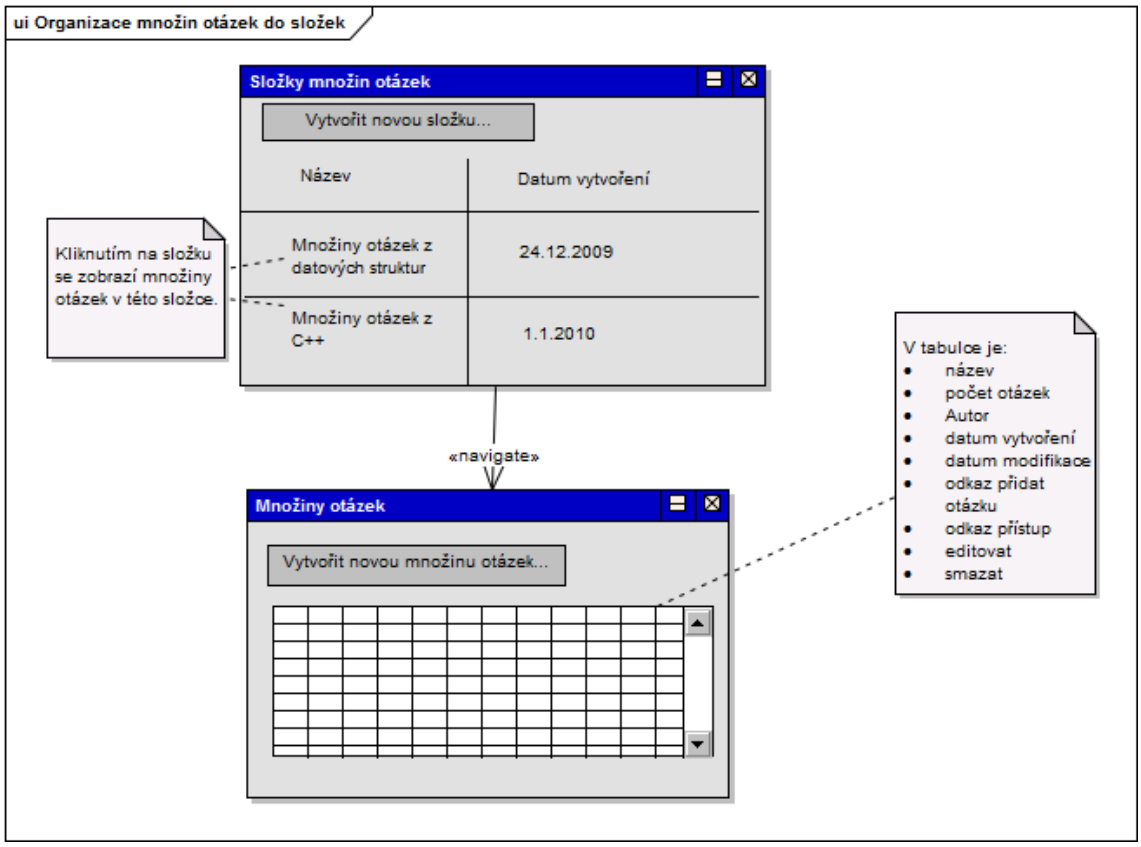
O požadavku, aby garant předmětu viděl automaticky (bez sdílení autorem) testy vypsané pro jeho předmět, bylo rozhodnuto, že se nebude implementovat.

## 8.6 Organizace entit do složek

Jde o nápad nebo potřebu lépe organizovat spravované entity v aplikaci Testy. Jde o entity množina otázek, definice testu, naplánovaný/realizovaný test. Seznam množin otázek, definic testů a naplánovaných testů je ve výpisu stránkován, položky v seznamech jdou řadit podle libovolného atributu vzestupně nebo sestupně. Standardně jsou řazeny tak, aby byly na první stránce neaktuálnější záznamy a nejstarší na poslední (množiny otázek a definice testů podle data poslední modifikace a naplánované/realizované testy podle data spuštění testu).

I přesto vznikl požadavek lépe organizovat tyto entity a to do složek, ve kterých lze mít související entity (např. složka pro definice testů pro předmět datové struktury, další složka pro definice testů pro předmět modelování). Po rozkliknutí složky by se zobrazil obsah složky v kontextu aktuálního portletu, tak jak je to vidět z diagramu uživatelského rozhraní:

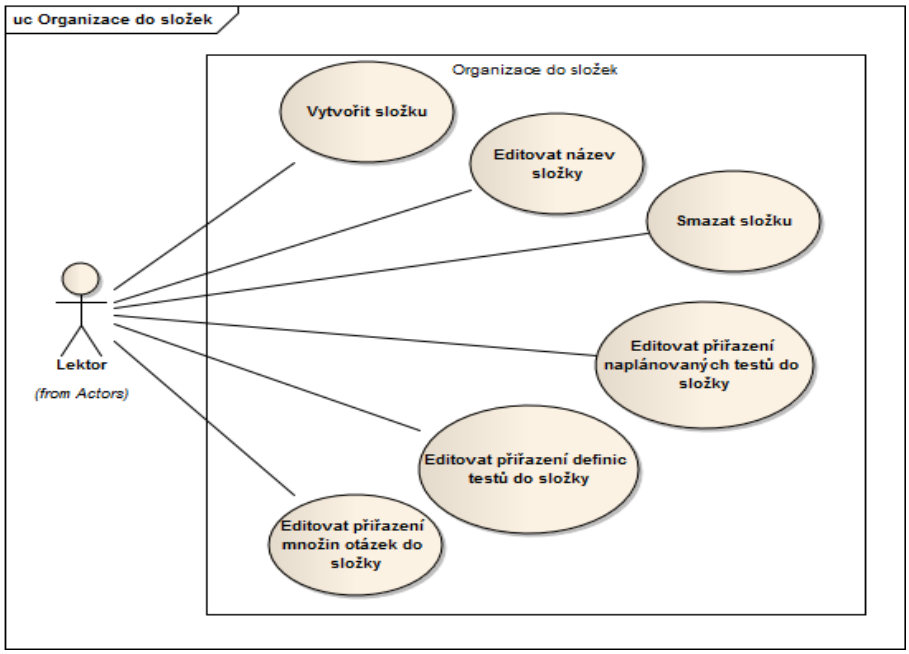
Obrázek č. 14 : diagram uživatelského rozhraní organizace množin otázek do složek




Zdroj: vlastní tvorba

Obdobně jako je to na obrázku č. 14 by se chovala organizace do složek definic testů a naplánovaných/realizovaných testů. Nejlépe je to vidět na případech užití:

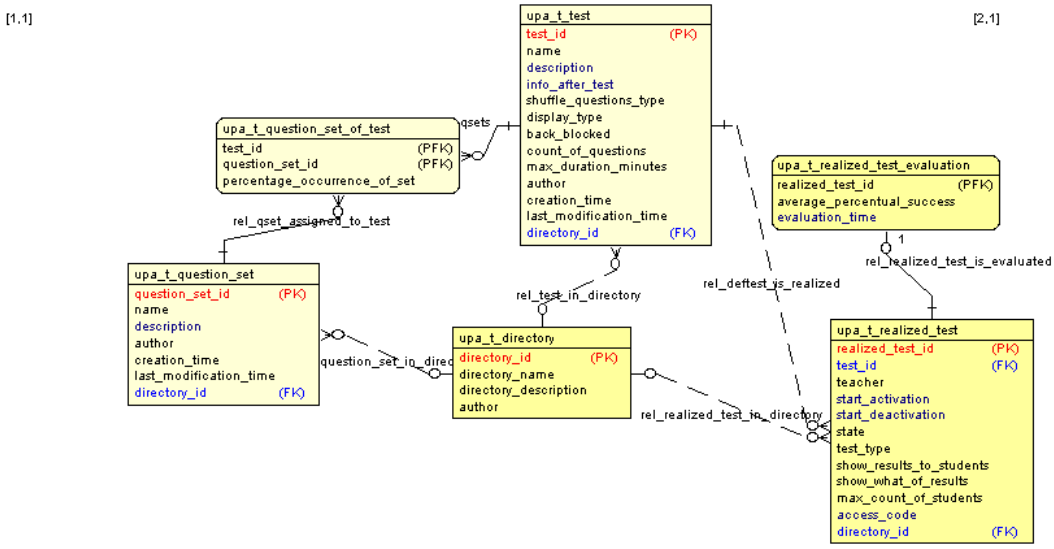
Obrázek č. 15 : diagram případů užití organizace do složek




Zdroj: vlastní tvorba

Pro implementaci těchto případů užití by bylo potřeba přidat do databáze tabulku uchovávající vytvořené složky tak, jak je to zobrazeno na následujícím diagramu:

Obrázek č. 16 : e-r diagram přidání složek do databáze




Zdroj: vlastní tvorba

Odlišným přístupem by bylo vytvořit tři tabulky, nebo lépe v jedné tabulce uchovávat informaci jestli se jedná o složku pro množiny otázek, definice testů nebo naplánované/realizované testy. Myslím, že tak jak jsem to navrhl je to výhodnější v tom, že si mohou lektoři do jedné složky dávat např. množiny otázek určených pro nějaký předmět, stejně tak definice testů a naplánované testy pro tento předmět. V jednotlivých portletech by se stejně ve složkách zobrazovaly jenom entity jednoho typu. Např. v portletu „Množiny otázek a otázky“ by se ve složce zobrazovaly pouze množiny otázek, které tam jsou. Nezobrazovaly by se tam naplánované testy, i přesto že jsou ve stejné složce.

Samozřejmě by bylo nutné upravit celou aplikaci Testy všude tam, kde se zobrazují výše uvedené entity, které by se organizovaly do složek. To je bohužel prakticky v celé aplikaci na úvodních stránkách. Samotné aplikační logiky by se toto příliš nedotklo, jinak by ale bohužel bylo nutné přepsat všechny ostatní vrstvy. V DAO vrstvě by bylo nutné parametrizovat dotazy podle složky, v které se uživatel nachází. Bylo by nutné patřičně upravit controllery a hlavně poměrně radikálně upravit prezentační vrstvu.

Navíc by bylo nutné udělat novou funkcionalitu (zřejmě nový portlet) pro tvorbu a správu složek uživatele. Tento nový portlet by byl poměrně jednoduchý a obsahoval by operace pro zobrazení, vytvoření, mazání, editaci složek a přiřazení entit do složek.

## Implementace

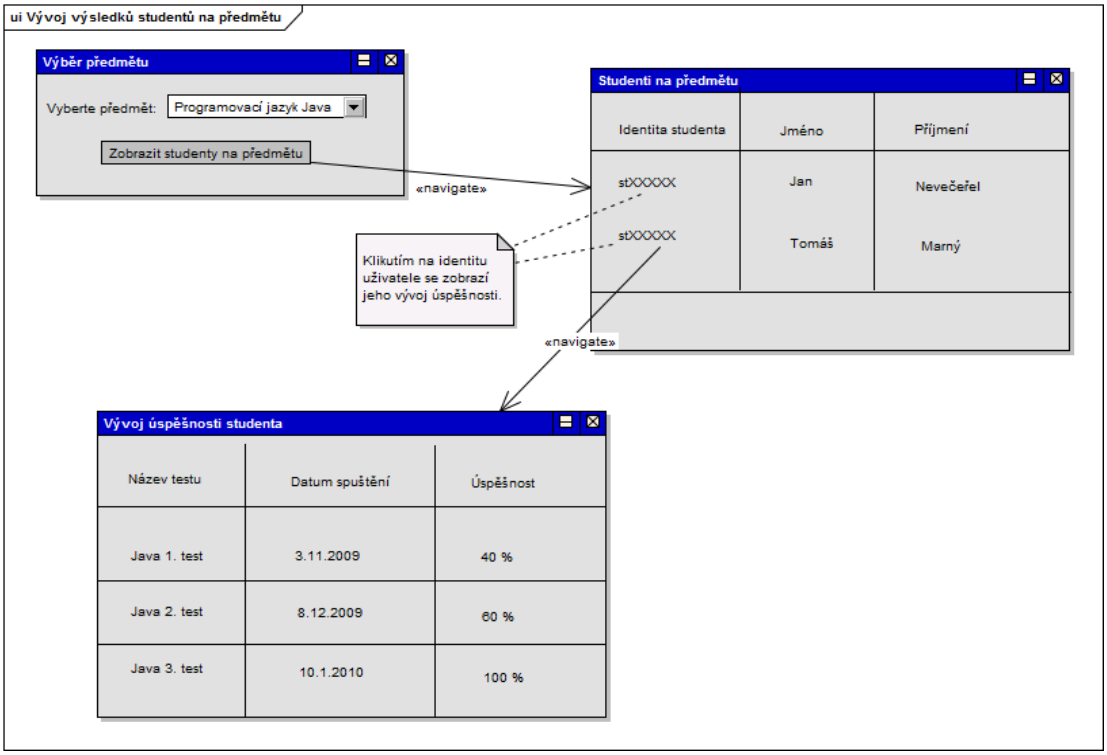
Implementace požadavku organizovat množiny otázek, definice testů a naplánované/realizované testy do složek byla zamítnuta. Důvodem je zřejmě poměrně velký a nákladný zásah do celé aplikace a drobné znepřehlednění uživatelského rozhraní pro lektory, kteří nemají vypsáno více než dvacet různých testů za semestr (těch bude zřejmě většina).

## 8.7 Vývoj výsledků studenta na předmětu

Požadavek na zobrazení vývoje výsledků testů studenta v rámci předmětu je zcela v souladu s e-learningovým pojetím testů. V aplikaci Testy je možné již v současné době zobrazit výsledky všech studentů na testu. Je tu tedy možnost evaluace testu a tedy zvládnutí látky, které se test týkal. Nicméně chybí pohled z druhé strany. Tedy pohled na testy konkrétního studenta v rámci kurzu nebo předmětu. Po implementaci tohoto požadavku budou moci lektoři jednoduše hodnotit pokrok studenta na předmětu a vývoj jeho znalostí.

Návrh uživatelského rozhraní by mohl být následovný. Lektor vybere předmět ze seznamu předmětů, které učí resp. na kterých participuje (je jako přednášející nebo cvičící). Po potvrzení výběru se zobrazí studenti, kteří jsou zapsáni na předmět. Po výběru studenta se zobrazí jeho vykonané testy s výsledky v rámci vybraného předmětu seřazené chronologicky od nejstaršího po nejnovější. Tak bude vidět vývoj výsledků studenta v rámci předmětu. Nejlépe je to opět patrné na diagramu uživatelského rozhraní:

Obrázek č. 17 : diagram uživatelského rozhraní vývoj výsledků studenta na předmětu




Zdroj: vlastní tvorba

Jelikož jde pouze o prezentaci již uchovaných výsledků, změny v databázi pro implementaci tohoto požadavku nebudou potřeba. Nebude potřeba ani upravovat stávající kód aplikace. Bude nutno přidat tuto funkcionalitu ať už do nového portletu, nebo ji vhodně umístit do nějakého stávajícího portletu.

### Implementace

Požadavek na prohlížení vývoje výsledků studentů na předmětu byl implementován navržených způsobem.

## 8.8 Vypisování testů na termín zkoušky

Hodnotící test je v současné době možno navázat na předmět nebo na rozvrhovou akci. Tedy vypsat test pro studenty, kteří jsou zapsáni na předměty nebo na rozvrhové akci. Tento požadavek se týká rozšíření možností při vypisování testu a mít možnost vypsat test pro studenty, kteří jsou zapsáni na termín zkoušky.

Při vypisování testu pro studenty na zkouškovém termínu bude v naprosté většině případů chtít lektor omezit spuštění testu na učebnu, tak jak to popisuje kapitola 8.3 Zabezpečení spouštění testu v učebně.

Tím, že se omezí spuštění testu na učebnu, ve které bude probíhat zkouška, se zároveň logicky omezí vypsaný test pouze pro studenty, kteří na té zkoušce jsou přítomni. Takže lektor může vypsat test pro předmět, ze kterého se koná zkouška, a postupovat tak, aby se dal test spustit pouze z učebny.

Pokud nechce lektor z nějakého důvodu takto důsledně zajistit spuštění testu v učebně, stačí vypsat test, ochránit jeho spuštění heslem a následně sdělit heslo pouze studentům na zkoušce.

### Implementace

Požadavek na navázání testů na termín zkoušky nebyl implementován z důvodu nahraditelnosti jiným postupem (popsáno výše).

## 8.9 Zabezpečení průběhu testu proti kopírování otázek

Na FES (Fakultě ekonomicko-správní), Ústavu systémového inženýrství a informatiky používají e-learningový systém Moodle. Požadavky vzešlé od lektorů z FES, mezi něž patří i tento, se týkají většinou doplnění funkcionalit, které jsou v systému Moodle a chybí v aplikaci Testy na portálu pardubické Univerzity.

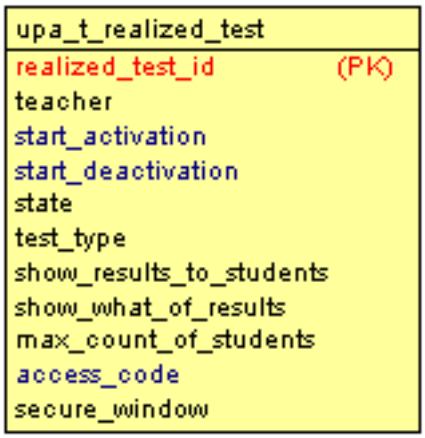
Zde je požadována ochrana proti kopírováním otázek z běžícího testu přes schránku nebo pomocí snímání obrazovky (přes klávesu Print Screen).

Tento požadavek naráží na technologické omezení webových aplikací a webu jako takového. Ve webových aplikacích, ať už jde o klasické dynamické weby nebo portletové aplikace jako v případě aplikace Testy, je vždy prezentační vrstva tvořena nějakou technologií, která nakonec pošle klientskému prohlížeči jako odpověď na požadavek statickou html stránku. Webový prohlížeč klienta ji zobrazí, a to, jak ji zobrazí a hlavně, jestli je možno ze stránky kopírovat text, je v kompetenci klientského prohlížeče.

Pro řešení tohoto problému jsem se nechal inspirovat e-learningovým systémem Moodle. Zjistil jsem, že tam je to řešeno pomocí JavaScriptu na stránce běžícího testu.

Vyřešil jsem to obdobně. Nejdříve je třeba doplnit do uživatelského rozhraní plánování testů možnost výběru, zda spouštět test v režimu „zabezpečená obrazovka“ nebo klasicky bez této ochrany. Také je samozřejmě nutné přidat do databáze do tabulky naplánovaných/realizovaných testů atribut, který udává jestli spouštět test v režimu „zabezpečená obrazovka“.

Obrázek č. 18 : e-r diagram přidání podpory „zabezpečené obrazovky“




Zdroj: vlastní tvorba

Když následně student spustí test, který má nastaveno spouštění v zabezpečeném režimu, tak bude na každé stránce testu s otázkami přidán JavaScriptový kód, který zamezí:

- vybrání textu myší nebo pomocí klávesnice,
- kopírování textu do schránky,
- zobrazení kontextového menu (pomocí pravého tlačítka myši),
- snímání obrazovky pomocí klávesy Print Screen,
- uložení HTML stránky s textem nebo zobrazení zdrojového kódu stránky pomocí schování menu s těmito funkcionalitami v prohlížeči.

Myslím, že tato ochrana zamezí kopírování otázek z testu u většiny méně technicky zdatných studentů. Ale je třeba říci, že tato ochrana se dá lehce obejít. Stačí vypnout v prohlížeči JavaScript. Navíc jak bylo zmíněno výše, to jak je HTML stránka včetně JavaScriptu interpretována je záležitost prohlížeče. Takže tato ochrana funguje dobře u dvou nejvíce používaných prohlížečů (Internet Explorer a Firefox). Jiné prohlížeče část ochranného kódu ignorují (např. v Opeře jde zobrazení kontextového menu, v linuxové verzi Firefoxu i snímání obrazovky).

Komplexní zabezpečení proti kopírování otázek a opisování z různých materiálů na počítači nebo na webu by zajistila učebna, kde by bylo implementováno testovací prostředí (viz. Kapitola 10.1).

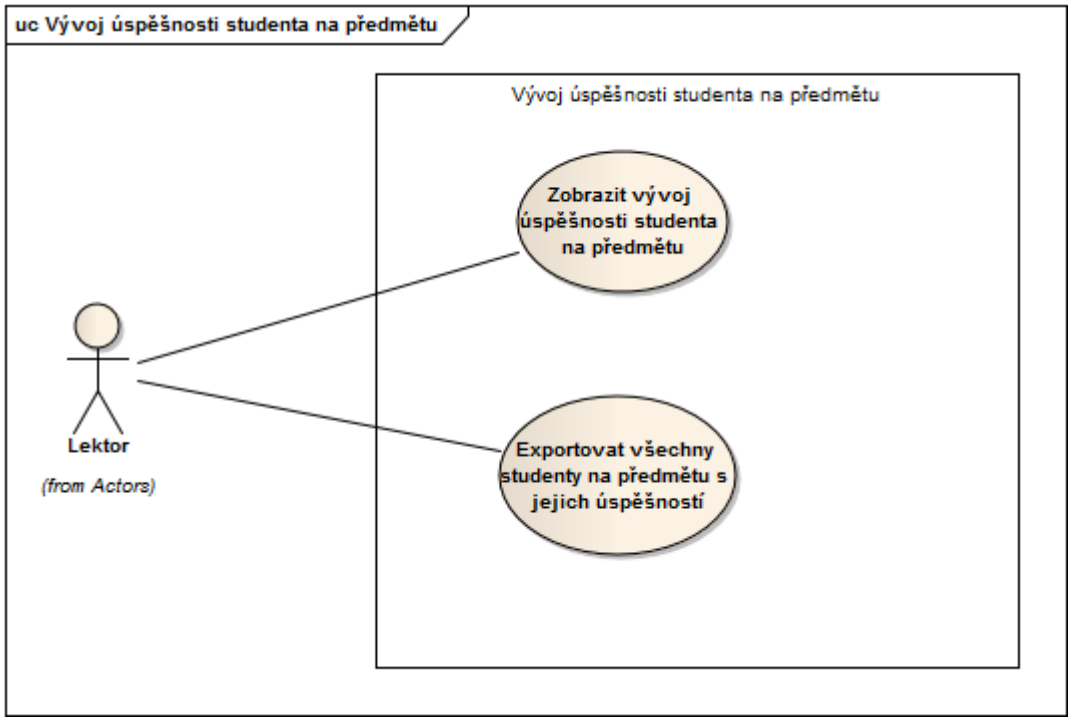
### Implementace

Zabezpečení průběhu testu proti kopírování otázek bylo implementováno tak, jak bylo navrženo. Nicméně, jak bylo popsáno, tento způsob implementace zůstává pouze „na půli cesty“. Komplexní zabezpečení proti kopírování otázek je popsáno v kapitole 10.1.

## 8.10 Export výsledků studentů v rámci předmětu

Tento požadavek je doplnění požadavku z kapitoly 8.7 Vývoj výsledků studenta na předmětu. Jde o hromadný export výsledků všech testů všech studentů na vybraném předmětu.

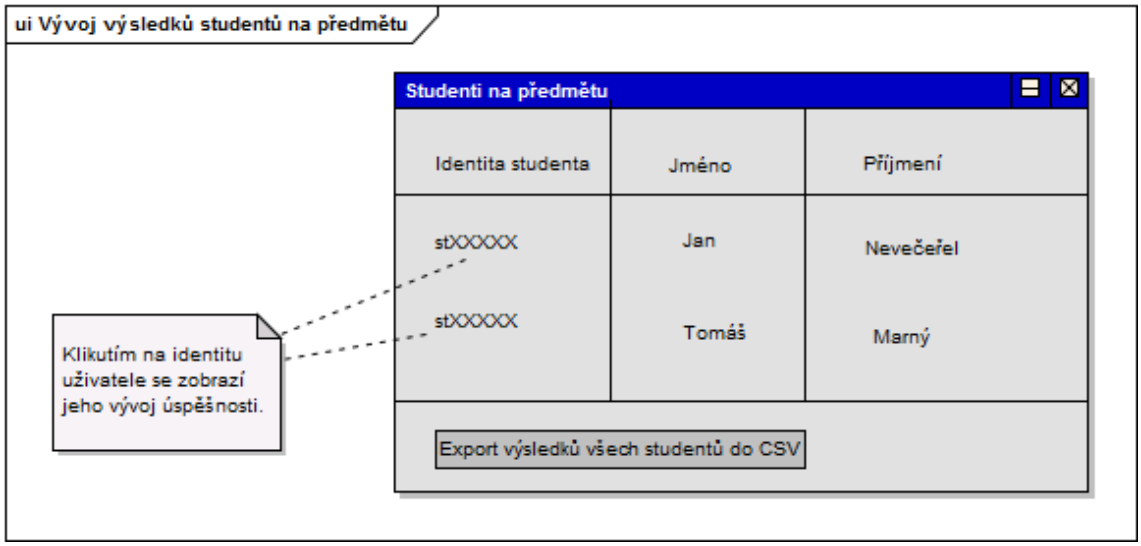
Obrázek č. 19 : diagram případů užití vývoj úspěšnosti studentů na předmětu




Zdroj: vlastní tvorba

Myslím, že by bylo vhodné tento požadavek realizovat na stránce, kde je již vybrán předmět a zobrazují se na ni studenti, pro které je možno následným rozkliknutím zobrazit vývoj úspěšnosti testů na vybraném předmětu. To znamená, že zde stačí dát tlačítko po jehož stisknutí se nabídne ke stažení soubor s výsledky všech studentů na předmětu.

Obrázek č. 20 : diagram uživatelského rozhraní export výsledků všech studentů na předmětu




Zdroj: vlastní tvorba

Výsledky všech testů všech studentů na předmětu se dají reprezentovat tabulkou, kde ve sloupcích mohou být názvy a data spuštění testů, v řádcích jména a identifikátory studentů a v buňkách jednotlivé výsledky konkrétních testů konkrétního studenta. Jelikož většina lektorů na pardubické Univerzitě pracuje s aplikací Microsoft Excel, ale nemůžeme ignorovat ani ty, kteří pracují s jiným produktem, tak se jako vhodný formát jeví data oddělená čárkou (CSV – Comma-Separated Values). A protože většina lektorů používá jako operační systém Microsoft Windows, tak je dobré data exportovat v kódování „windows-1250“. Ostatní lektoři, kteří používají jiný operační systém bývají technicky zdatnější a neměli by s tím mít problém.

## Implementace

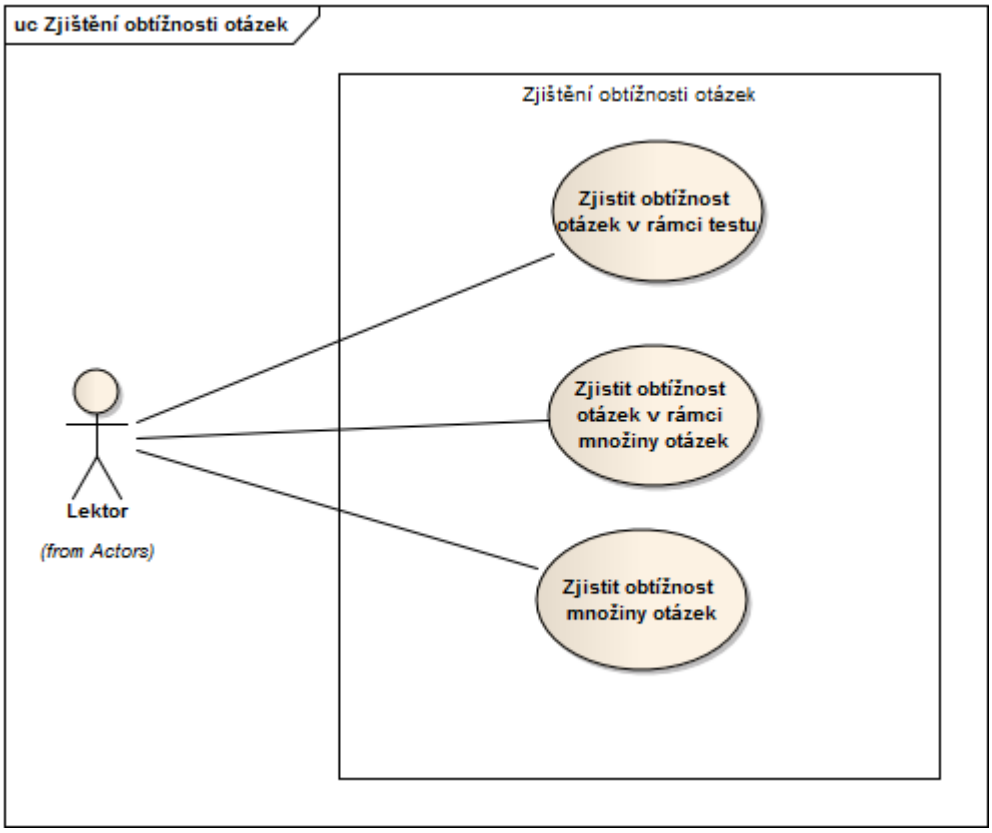
Požadavek na export výsledků všech studentů v rámci předmětu byl implementován tak, jak bylo navrženo.

## 8.11 Nejlehčí a nejtěžší otázka v testu

Výsledky testů je třeba vyhodnocovat nejen z pohledu úspěšnosti studentů a jejich pokroku při studiu. Je též potřeba vyhodnocovat jestli jsou otázky v testu správně položeny a jestli mají v daném kurzu smysl. V tom může pomoci vyhodnocení, která otázka byla v rámci testu realizovaného testu nejtěžší a která naopak nejlehčí. Jeden lektor na pardubické Univerzitě dokonce toto každý rok vyhodnocuje a nejlehčí otázku z testu pro příští rok vždy vyřazuje.

Tento požadavek by se dal zobecnit na zobrazení seznamu otázek použitých v testu, přičemž tento seznam bude seřazen podle obtížnosti otázek. Takto bude patrná obtížnost všech otázek, nejen nejtěžší a nejlehčí. Bylo by též vhodné kromě zobrazení obtížnosti otázek v rámci testu zobrazit ještě obtížnost otázek v rámci množin otázek použitých v testu.

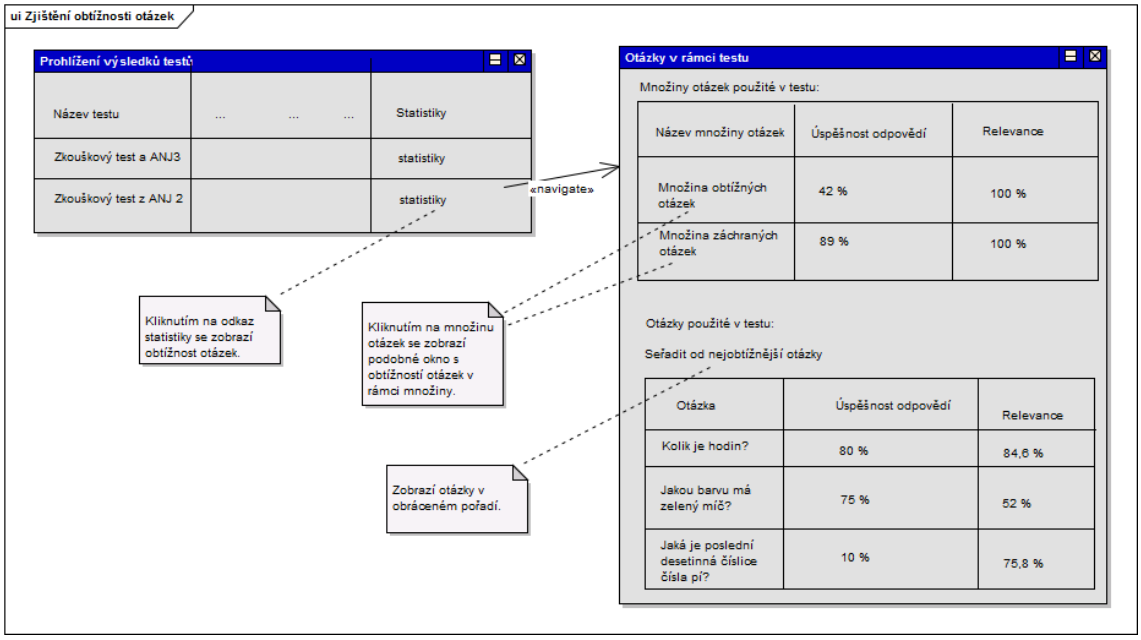
Obrázek č. 21 : diagram případů užití zjištění obtížnosti otázek




Zdroj: vlastní tvorba

Co se týče uživatelského rozhraní, tak je asi logické udělat přístup k této funkcionalitě z portletu „Prohlížení výsledků testů“ po kliknutí na příslušný realizovaný test tak, jak je to patrné z diagramu uživatelského rozhraní.

Obrázek č. 22 : diagram uživatelského rozhraní zjištění obtížnosti otázek




Zdroj: vlastní tvorba

Při pohledu na schéma databáze Testů (v příloze č. 2) jsou vidět tabulky, kde je definice množin otázek, otázek a odpovědí (upa_t_question_set, upa_t_question a upa_t_answer). Záznamy v těchto tabulkách jsou identifikovány pomocí syntetického identifikátoru (id). Je zde též vidět napojení definice testu (upa_t_test) na množiny otázek. Množiny otázek, otázky a odpovědi mohou autoři nebo ostatní lektoři, kterým jsou sdíleny s příslušnými právy, měnit a pro účely statistik je zároveň nutné uchovat otázky a odpovědi tak, jak se vyskytly v testech studentů. Proto jsou na test studenta (obecně uživatele upa_t_user_test) navázány tabulky s použitými množinami otázek, otázkami a odpověďmi (upa_t_used_question_set, upa_t_used_question a upa_t_used_answer).

Tyto tabulky uchovávají podobné údaje jako tabulky, kde se množiny otázek, otázky a odpovědi definují, navíc drží odpovědi studentů a jsou navázány na odpovídající definiční tabulky. Tato vazba je tu kvůli možnosti měnit definici otázek a odpovědí tak,

aby se tato změna promítla i v ještě ne definitivně ohodnocených uživatelských testech (v případě, že si např. lektor uvědomí další možnou odpověď na základě odpovědí studentů). Pokud se test definitivně ohodnotí klíčové atributy sloužící pro tuto vazbu se nastaví na null a už nejde výsledky testu měnit.

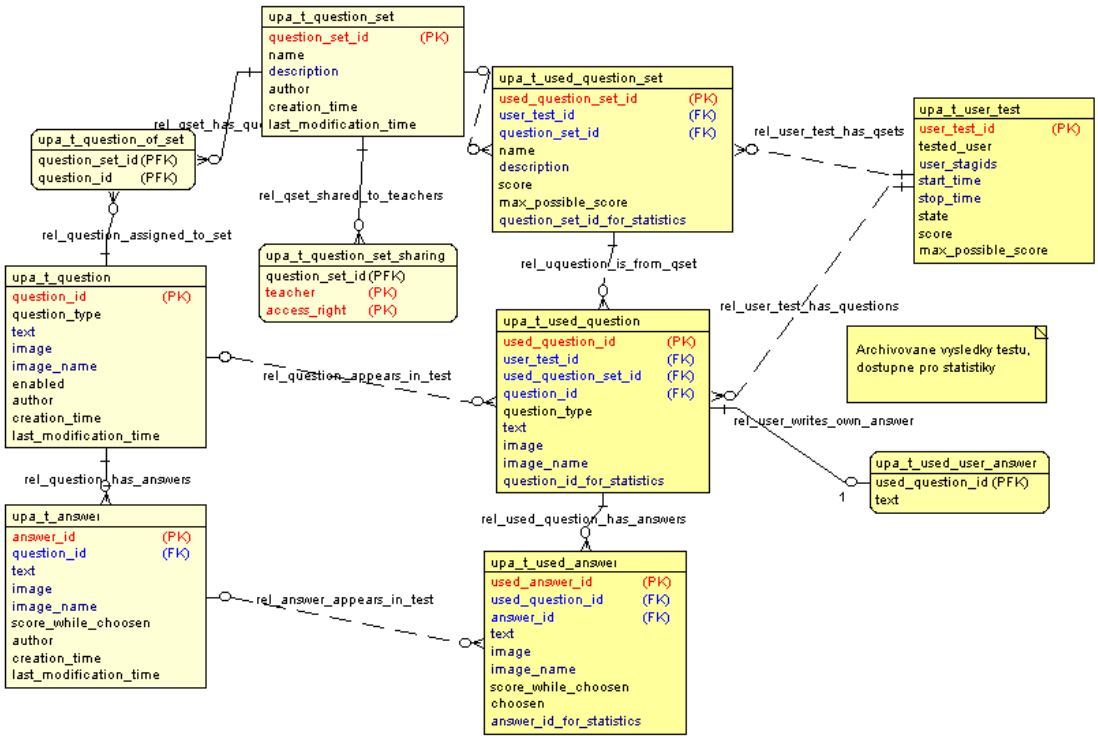
Pro hodnocení obtížnosti otázek je ale problém jak poznat, které otázky jsou stejné. U otázek v definitivně ohodnocených testech již není žádný identifikátor, pomocí kterého bychom zjistili, že dvě otázky v různých testech různých studentů ale stejném vypsaném testu jsou totožné.

### 8.11.1 Možnosti identifikace otázek

Jedna možnost jak identifikovat totožné otázky beze změn v databázi a aplikační logice je porovnat záznamy v tabulkách podle všech neklíčových atributů a stejným způsobem porovnat i všechny odpovědi na otázku. Pokud definice otázky i všech odpovědí souhlasí, mohli bychom prohlásit, že se jedná o stejné otázky. Zafungovalo by to v 99 % případů. Mohou se ovšem vyskytnout dvě odlišné otázky se stejně nadefinovaným textem a porovnávat záznamy v databázi podle neklíčových atributů je obecně spíše cesta do pekel.

Druhou možností je v tabulkách upa_t_used_question_set, upa_t_used_question a upa_t_used_answer zdvojit vazební atributy odkazující na příslušné definiční tabulky upa_t_question_set, upa_t_question a upa_t_answer. Nicméně jeden z těchto atributů nedefinovat jako klíčový a nenastavovat ho na null při definitivním hodnocení realizovaného testu. Tento atribut by sloužil obecně pro vyhodnocování statistik v rámci definitivně ohodnocených testů. Tedy pro porovnání zda se jedná o stejnou množinu otázek, otázku a odpověď (myšleno podle definice množiny, otázky a odpovědi). Je to vidět na upraveném e-r diagramu, jedná se o atributy question_set_id_for_statistics, question_id_for_statistics a answer_id_for_statistics:

Obrázek č. 23 : e-r diagram identifikace množin otázek, otázek a odpovědí v testech studentů pro účely statistik




Zdroj: vlastní tvorba

Tato možnost také není ideální, protože generuje drobné duplicity v databázi. Nicméně vyžaduje zcela minimální změnu aplikační logiky (při generování testů studentů, tedy při spuštění testu je potřeba tyto atributy vyplnit) a nevyžaduje migraci dat.

Další možností je předělat logiku editace množin otázek, otázek a odpovědí tak, aby se nemazaly vazby. To jestli již použité otázky nejsou editovatelné by se dalo zjišťovat podle stavu realizovaného testu. To by ovšem znamenalo předělat značnou část aplikační logiky a znovu otestovat. Navíc by vznikl problém při mazání definice množiny otázek, otázky nebo odpovědi. Ty by byly nesmazatelné díky zachování vazeb nebo by se musela řešit viditelnost dalším atributem (např. deleted).

### 8.11.2 Relevance obtížnosti otázek

Jelikož se do studentských testů generují otázky náhodně (podle zadaných kritérií při definování testu), musíme zavést relevanci obtížnosti otázek. Uveďme příklad: Mějme naplánovaný test, který spustilo 100 studentů, má tedy 100 studentských testů. Jedna otázka se vygeneruje pouze do jednoho studentského testu (student na ni odpoví špatně). Tudíž tato otázka má úspěšnost 0 % a je vyhodnocena jako nejtěžší, ale má relevanci pouze 1 %.

Relevanci otázky tedy můžeme definovat jako procentuálně vyjádřený poměr mezi počtem studentských testů, kde se vyskytla, a celkovým počtem studentských testů naplánovaného/realizovaného testu.

#### Implementace

Požadavek na zobrazení nejlehčí a nejtěžší otázky v testu byl transformován na zobrazení seznamu otázek seřazeného podle obtížnosti (popsáno výše), navíc bylo přidáno vyhodnocení obtížnosti otázek v rámci množiny otázek použité v testu a vyhodnocení obtížnosti celé množiny otázek použité v testu. Co se týče identifikace množin otázek a otázek, byla zvolena možnost přidání dalšího identifikátoru definice množiny otázky a otázky do tabulek s množinami otázek a otázkami použitými v testu (též popsáno výše).

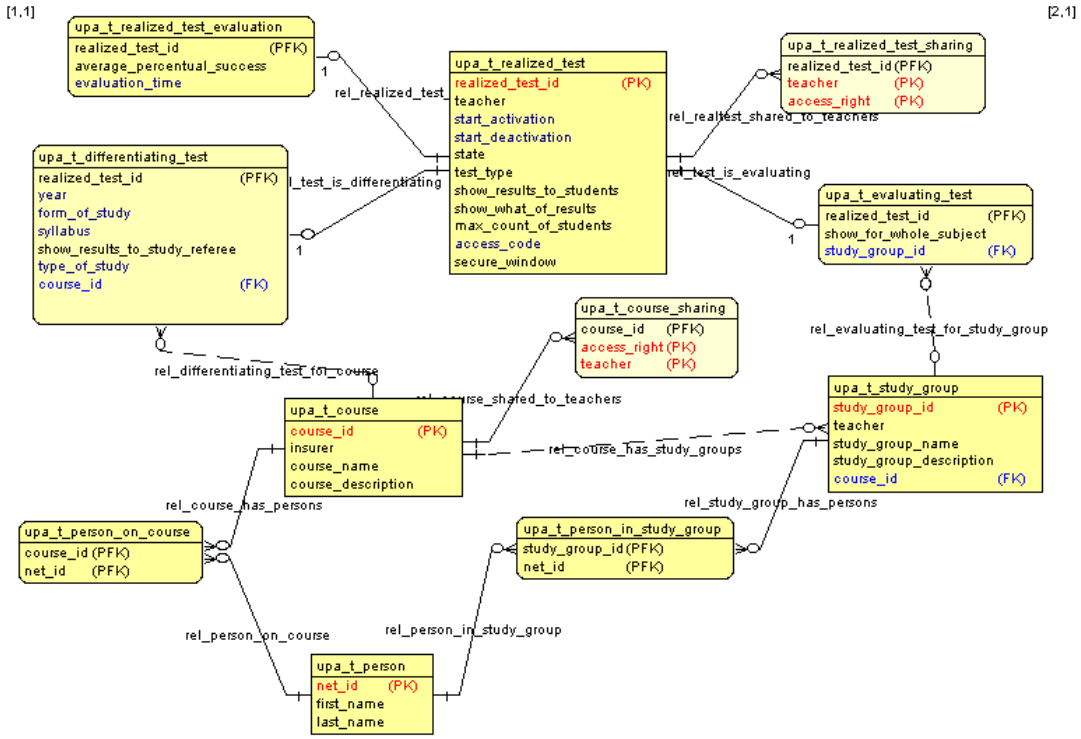
## 8.12 Testování zaměstnanců v rámci kurzů

Na Jazykovém centru se chystá otevření jazykových kurzů pro pracovníky Univerzity Pardubice. Jedná se o pracovníky různých fakult ale i různých organizačních útvarů Univerzity (Menza, OVV, IC, atd.). V rámci těchto kurzů je třeba studenty (v tomto případě pracovníky Univerzity) rozřadit na skupiny podle úrovně znalostí. Následně zhruba v půlročním intervalu vypisovat hodnotící testy pro jednotlivé skupiny. Plánovaná doba kurzů jsou zhruba tři roky.

Zásadní problém je, že ve IS/STAGu jsou evidováni pouze učitelé a studenti. Ostatní zaměstnanci Univerzity v systému nejsou a pokud v minulosti potřebovali do IS/STAGu přístup, byli tam přidání jako fiktivní učitelé. Pokud mají zaměstnanci vykonávat testy, je potřeba je někde v systému evidovat. Ať už ve IS/STAGu nebo v Testech.

Dále je potřeba evidovat kurzy a studijní skupiny, do kterých budou studenti rozřazeni na základě rozřazovacího testu. Rozřazovací testy vypsány pro tyto účely budou napojeny na kurz a následné hodnotící testy na studijní skupinu. Přehledně je to vidět v e-r diagramu:

Obrázek č. 24 : e-r diagram testování zaměstnanců v rámci kurzu




Zdroj: vlastní tvorba

V mém návrhu jsou osoby, které budou vykonávat testy, v tabulce upa_t_person. Respektive obecně jakékoli další osoby, které v budoucnu přijdou do styku s aplikací Testy a nebudou to učitelé nebo studenti. Uložení identifikace osob a jejich dalších údajů by bylo úplně nejlepší sjednotit do jedné tabulky (v současné době jsou v IS/STAGu studenti v jedné tabulce a učitelé v jiné). Nicméně to je zřejmě nereálné.

Dále je třeba udělat v portálu roli „studentKurzu“ a přiřadit ji studentům kurzu, aby se mohli zaměstnanci Univerzity, kteří budou vykonávat testy v rámci kurzu, přihlásit do portálu.

Bude také potřeba rozšířit možnost plánovat rozřazovací testy pro studenty na kurzu a rozšířit možnost plánovat hodnotící testy pro studenty ve studijních skupinách kurzu. Dále vytvořit nový portlet, ve kterém budou lektoři moci vytvořit, editovat, mazat kurzy a sdílet je ostatním lektorům. Autor kurzu nebo ten, kdo má právo editace kurzu, bude moci ke kurzu vytvářet studijní skupiny a napojovat na kurz a na studijní skupiny studenty.

Kurzy a studijní skupiny na kurzech by v budoucnu měla zřejmě spravovat jiná aplikace určená pro správu kurzů jiných než akreditovaných předmětů vyučovaných podle zákona o Vysokých školách.

### Implementace

Implementace požadavku na testování zaměstnanců Univerzity v rámci kurzů nebyla nakonec schválena Informačním centrem Univerzity.

# 9 Implementace ostatních požadavků

Mimo implementace požadavků uvedených v kapitole 8 jsem opravil některé chyby. Dále jsem provedl časovou optimalizaci některých SQL dotazů týkajících se zejména zobrazení souhrnných výsledků testů podle složitějších kritérií. Tato optimalizace vedla k několikanásobnému zrychlení práce s aplikací. Dále jsem rozšířil uživatelskou příručku Testů a vytvořil jednoduchý a rychle použitelný návod „Jak na to“ ve formě často kladených otázek.

V následujícím seznamu jsou uvedeny funkcionality, které jsem implementoval a které nejsou uvedeny v kapitole 8. Jsou to drobnější funkcionality, požadavky na jejich implementaci nepocházely přímo od uživatelů nebo nevyžadovaly podrobnější analýzu.

- Zpřístupnění výsledků testů i studijním refentkám/referentům, pokud si to autor testu přeje.
- K otázkám s výběrem správné odpovědi doplněna automaticky možnost „Nechci odpovědět (0 b.)“. Student totiž nemusí vybrat u otázky žádnou možnost, pak je otázka hodnocena 0 b. Pokud ovšem student již nějakou možnost vybral, tak neměl možnost zrušit výběr (byly použity webové prvky „radio button“). Nyní tuto možnost má.
- Zamezení určitého druhu editace otázek (mazání odpovědi, změna typu otázky), pokud by to mělo ovlivnit otázky již použité v testu (ještě ne definitivně ohodnoceného). Při takovýchto změnách by se pak otázky již použité v testu změnily takovým způsobem, že by neměly význam.
- Implementace vyhledávání studentského testu ve výsledcích testu podle NetID, jména a příjmení studenta.
- Podílení se na změně identifikace učitelů z NetID na identifikátor IS/STAG.

# 10 Náměty na rozšíření do budoucna

## 10.1 Učebny s testovacím prostředím

V současné době je již implementováno omezení spouštění testů pouze z učebny. Také je implementována poměrně slabá ochrana proti kopírování otázek z běžícího testu.

Lektor má tak možnost spustit test pro studenty v učebně. Nicméně pokud chce zamezit podvodnému jednání studentů (opisování, kopírování otázek, ...) musí celou dobu běhu testu studenty sledovat. To často není v možnostech lektora.

Díky webové povaze aplikace Testy se k ní musí přistupovat pomocí internetového prohlížeče. Student si tedy při vykonávání testu může lehce (například v jiné záložce prohlížeče) otevřít jinou webovou stránku, z které může zjišťovat odpovědi na otázky položené v testu.

Těmto popsaným problémům nelze nijak účinně zamezit v rámci úprav aplikace Testy.

Řešením tohoto problému by bylo vytvoření testovacího prostředí na počítačích v učebnách. Na těchto počítačích by byl kromě běžně používaného operačního systému i operační systém nakonfigurovaný pro vykonávání testů. To znamená nastavená maximálně restriktivní politika, kdy by uživatel mohl pouze spustit internetový prohlížeč (jiné programy by nebyly přístupné) a spustit v něm test. Byla by vypnuta schránka pro kopírování, snímání obrazovky a odpojena všechna zařízení typu CD, DVD rekordér včetně USB portů pro připojení flash disku. Nejlepší by bylo zvolit takový internetový prohlížeč, který jde nakonfigurovat restriktivně s ohledem na ukládání webových stránek.

Firewall operačního systému by byl nastaven tak, aby umožňoval komunikaci pouze se serverem, kde je aplikace Testy. Aby nebylo umožněno ani otevření materiálu z aplikace Studijní materiály z portálu IS/STAG, je nutné aplikaci Testy nasadit na jiný server s jinou adresou IP.

Když by se toto udělalo, tak by studenti v učebně restartovali počítače do tohoto operačního systému s testovacím prostředím a v něm by vykonali test. U testu by mělo být také zajištěno, že jej budou psát pouze studenti v učebně (viz. kapitola 8.3). Tímto by se komplexním způsobem a spolehlivě zamezilo kopírování otázek, opisování z materiálů z internetu a dalšímu podvodnému jednání.

## 10.2 Automatizovaný zápis známek

Učitelé Fakulty Filozofické naznačili nápad mít možnost vypsat sérii testů, které se automaticky vyhodnotí a podle nějakého klíče nebo vzorce se vypočítá celkové hodnocení studenta v rámci předmětu. Na základě tohoto hodnocení bude nebo nebude studentovi automaticky zapsán zápočet do IS/STAG. Na základě získaného zápočtu by byl student připuštěn ke zkoušce. Zkouška by byla opět test, který by bylo možné absolvovat nejvýše třikrát. Každý pokus by se zaznamenal do IS/STAG. Na základě posledního výsledku studenta by se opět podle nějakého kritéria automaticky zapsala známka do IS/STAG.

V podstatě se jedná o téměř plně automatizovaný systém testování a hodnocení studenta. Je potřeba jenom ručně zapsat známku do výkazu o studiu.

Toto rozšíření by vyžadovalo poněkud hlubší analýzu. Nicméně i bez této analýzy je zřejmé, že by se jednalo o podstatnou a velkou změnu nebo rozšíření aplikace Testy. Zřejmě i z tohoto důvodu zamítlo Informační centrum pardubické Univerzity implementaci této funkcionality prakticky ihned i bez podrobnějších požadavků nebo analýzy.

# 11 Závěr

Práce se zabývá e-learningem a testováním jako jeho nedílnou součástí. Srovnává moderní přístupy za pomoci výpočetní techniky s klasickou formou vzdělávání. Je zřejmé, že pokud je správně zvolena forma e-learningového kurzu a do kurzu jsou vhodně zařazeny zpětnovazebné testy a hodnotící testy, může e-learning nabídnout velice výhodnou alternativu oproti klasické formě výuky. S výhodou lze též využívat kombinaci obou přístupů.

Testování získaných znalostí je velice užitečné jako prvek zpětné vazby studia, pro hodnocení studentů a nachází uplatnění i při rozřazení studentů do kurzů podle úrovně znalostí.

Rozšíření a zdokonalení modulu Testy na pardubické Univerzitě je věnována podstatná část práce. Provedená analýza požadavků a návrh řešení dobře posloužily pro následnou implementaci. Některé i velice užitečné požadavky a náměty na rozšíření nebo změnu bohužel nebyly implementovány. I přesto myslím, že se povedlo testovací modul posunout o velký kus blíže představám lektorů a že tyto rozšíření umožnila lepší využití jednotného univerzitního testovacího rozhraní většímu množství lektorů. Součástí změn v modulu Testy je i rozšíření uživatelské příručky a vytvoření rychle a pohodlně použitelného návodu „Jak na to“, který je přístupný na portálu Univerzity formou často kladených otázek.

V poslední kapitole práce jsou zmíněny náměty a návrhy na možná budoucí rozšíření aplikace tak, aby se ještě více zvýšila přidaná hodnota aplikace pro její uživatele.

# Seznam zkratek a pojmů

**LMS** – Learning Management System je řídící výukový systém (systém pro řízení výuky), tedy aplikace řešící administrativu a organizaci výuky v rámci e-learningu.

**LCMS** – Learning Content Management systém je systém pro tvorbu obsahu e-learningových kurzů

**Blended learning** – kombinace standardní výuky (prezenční, face-to-face) s e-learningem. Blended learning se snaží kompenzovat některé dílčí nevýhody e-learningu při plnění vzdělávacích cílů kombinací s prvky standardní výuky.

**IS/STAG** – Informační systém / STudijní AGendy.

**GNU/GPL** – GNU is Not Unix / General Public License je licence pro užívání a šíření svobodného softwaru.

**PHP** – je rekurzivní zkratka PHP Hypertext Preprocesor. Je to skriptovací jazyk, který umožňuje začleňovat dynamické prvky do HTML stránek.

**HTML** – HyperText Markup Language je značkovací jazyk pro psaní webových stránek.

**ERP** – Enterprise Resource Planning je označení informačního systému integrujícího množství procesů související s podnikovým plánováním, zejména logistikou výroby.

**JavaScript** – jazyk sloužící pro programování interaktivních prvků ve webových stránkách.

**Adobe Flash** – grafický vektorový program určený pro tvorbu interaktivních animací, často vkládaných do webových stránek.

**Business logika** – aplikační logika, tedy zachycení procesů a činností, které jsou vykonávány prostřednictvím aplikace.

**Autentizace** – zjištění a ověření totožnosti.

**Back end** – je administrační rozhraní neviditelné pro většinu uživatelů aplikace (např. správa produktů v internetovém obchodě).

**Front end** – je rozhraní aplikace určené pro běžného uživatele (např. rozhraní nakupování v internetovém obchodě).

**AD** – Active Directory je technologie Microsoftu poskytující mimo jiné autentizaci, přístup přes LDAP protokol, centrální úložiště pro data aplikací.

**LDAP** – protokol pro přístup a modifikaci dat v adresářové službě.

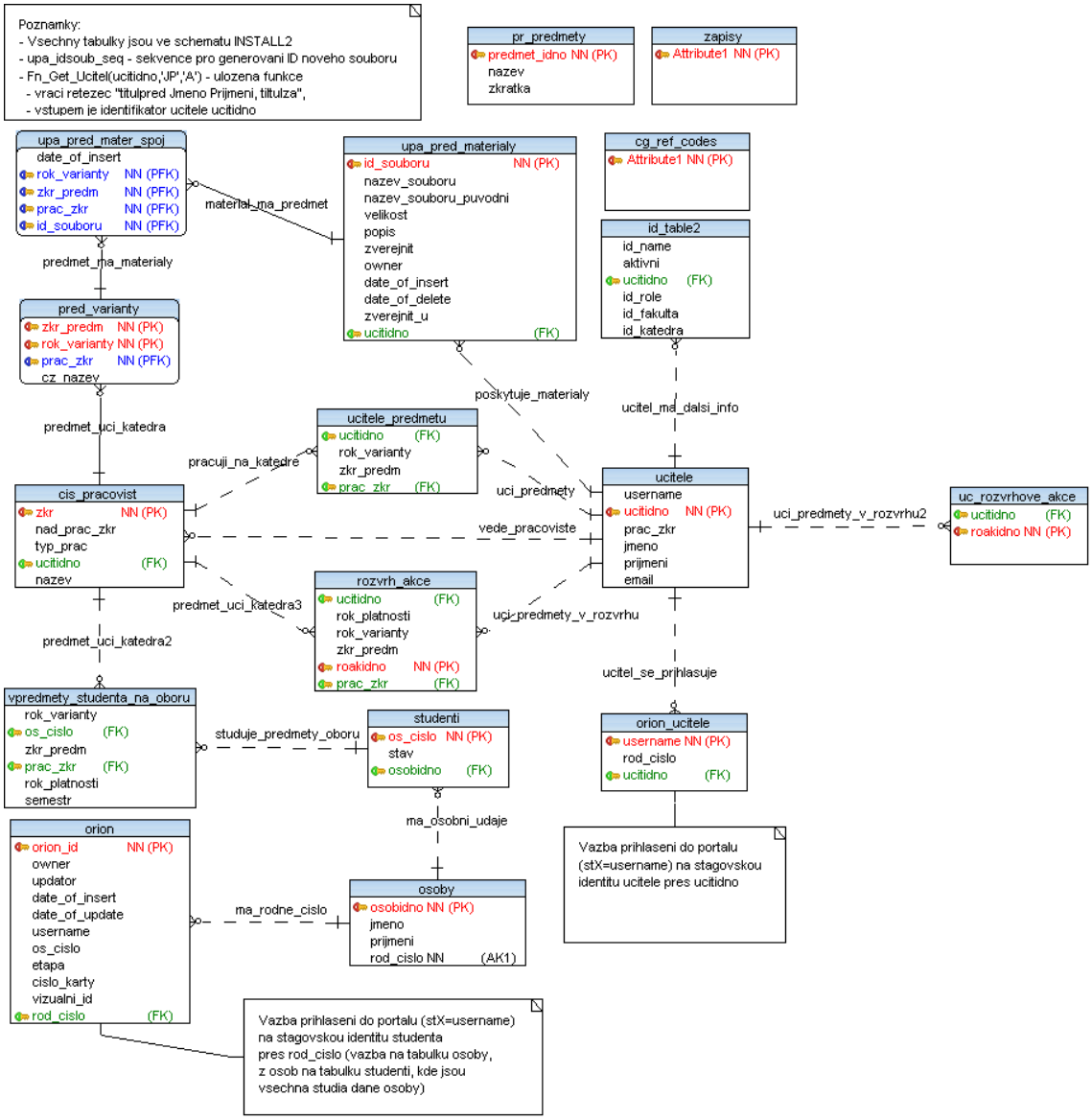
**Adresa IP** – je adresa, která jednoznačně identifikuje síťové rozhraní v počítačové síti.

**DHCP** – Dynamic Host Configuration Protocol se používá pro automatické dynamické přidělování adres IP jednotlivým počítačům v síti.

**PDF** – Portable Document Format, tedy přenosný formát dokumentů je souborový formát vyvinutý pro účel nezávislého ukládání dokumentů. Formát zajišťuje, že se dokument zobrazí na libovolné softwarové a hardwarové platformě stejně.

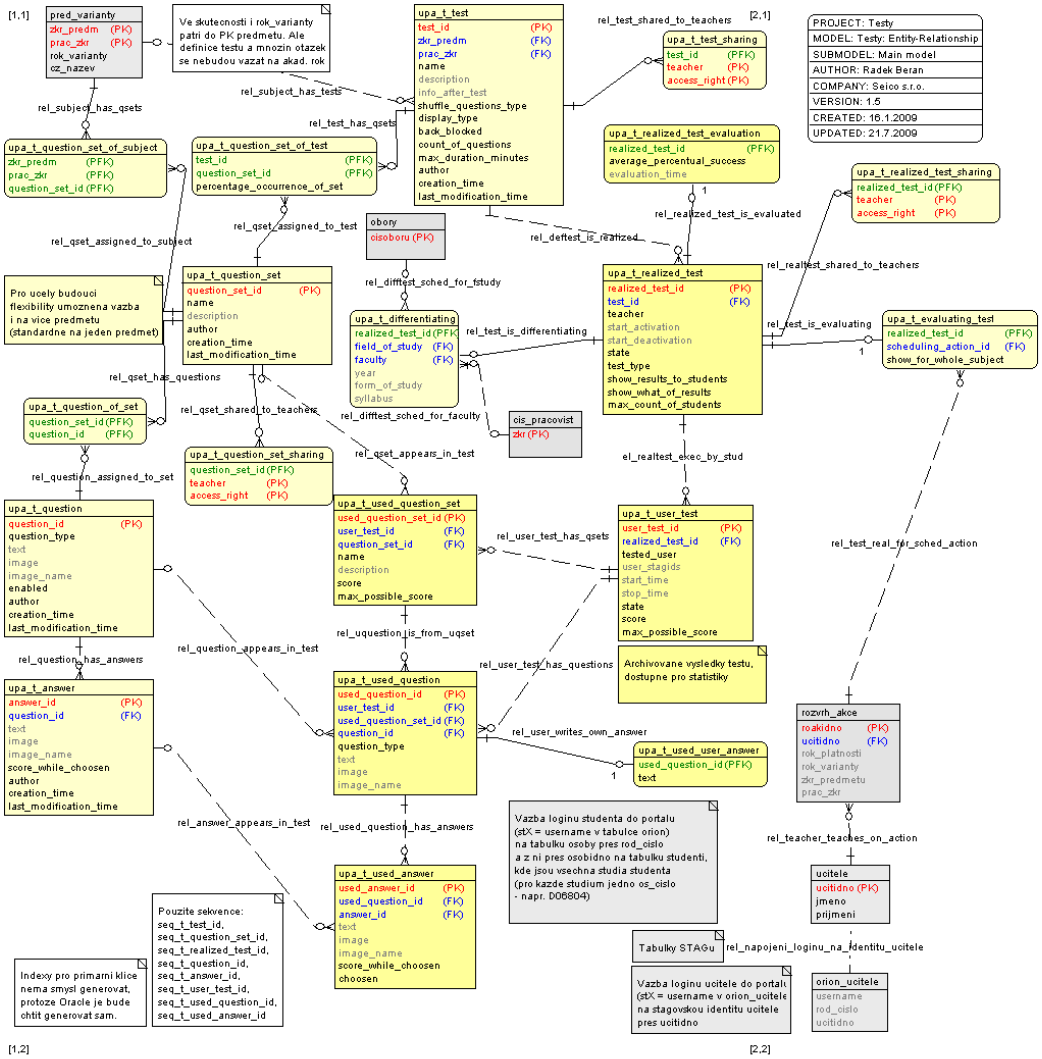
**E-R diagram** – slouží pro zobrazení datového modelu, tedy entit a vazeb mezi entitami.

# Příloha č. 1: Zjednodušené schéma systému IS/STAG pro potřeby aplikace Testy




Zdroj: firemní materiály Seico s.r.o.

# Příloha č. 2: E-R diagram databáze aplikace Testy ve výchozím stavu




Zdroj: firemní materiály Seico s.r.o.

# Použitá literatura


1. HORTON, William. *E-learning by design. 1 st edition*. [s.1.] : Pfeifer, 2006. 640 s. ISBN 0787984256.
2. HRONÍK, František. *Rozvoj a vzdělávání pracovníků*. Praha : Grada Publishing, 2007. 240 s. ISBN 978-80-247-1457-8.
3. Slovník bezpečného internetu. *Safer internet* [online]. 14.4.2008, [cit. 2010-07-21]. Dostupný z WWW: <http://www.saferinternet.cz/slovnik-bezpecneho-internetu/335-3>.
4. FOJTÍK, Rostislav Možnosti testování v e-learningu. In *Information and Communication Technology in Education*. Rožnov pod Radhoštěn : Ostravská univerzita, 2002. s. 350. ISBN 80-7042-828-7.
5. *ELearning* [online]. Wikipedie, otevřená encyklopedie, 2006 [cit. 2010-06-21]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/ELearning>.
6. HLAVÁČEK, Tomáš. *E-learning a jeho přínosy pro organizaci*. Pardubice, 2005. 85 s. Diplomová práce. Univerzita Pardubice, Dopravní fakulta Jana Pernera
7. *Learning Management systém.* Wikipedia [online]. 2010 [cit. 2010-07-18]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Learning_Management_System>.
8. *Blended learning.* Wikipedia [online]. 2010 [cit. 2010-07-21]. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Blended_learning>.
9. PEJŠA, J.; ZÍDEK, P. Výukové postupy [online]. Kontis s.r.o., 2010 [cit. 2010-06-20]. Dostupný z WWW: <http://www.e-learn.cz/soubory/výukové_postupy.pdf >.
10. MIKUŠ, Ľudovít; DROZDOVÁ, Matilda Testovací systém E-Study : Pomocka WBT cvičení. In *ELearn 2006*. prvé. Žilina : Žilinská univerzita, 2006. s. 175. ISBN 80-8070-505-4.
11. BYČKOVSKÝ, P. *Základy měření výsledků výuky. Tvorba didaktického testu.* Praha, ČVUT 1982
12. SALMON, Gilly. *E-Moderating : The Key to Teaching and Learning Online*. London : Kogan Page, 2000. 180 s. ISBN 0749431105.
13. *Computer based training* [online]. 2010 [cit. 2010-07-15]. Dostupné z WWW: <http://www.computerbasedtrainings.net/>.
14. *Web-Based Training Information Center* [online]. 2010 [cit. 2010-07-22]. What is Web-Based Training?. Dostupné z WWW: <http://www.webbasedtraining.com/primer_whatiswbt.aspx>.

15. JUNKOVÁ, Jana. Didaktické testování. *Informační a školící centrum SIPVZ* [online]. 2006, č.1, [cit. 2010-07-22]. Dostupný z WWW: <student.oapion.cz/ic_sipvz/obsah/didakticke_testovani.pdf>.

16. PEJŠA, J. *LMS a LCMS, vývoj kurzů* [online]. Kontis s.r.o., 2010 [cit. 2010-07-22]. Dostupný z WWW: <http://www.e-learn.cz/soubory/LMS_LCMS.pdf >.

17. *Virtuální studium – iluze, či nutnost?* [online]. Interval, webdesign a e-komerce denně, 2010 [cit. 2010-07-18]. Dostupný z WWW: <http://interval.cz/clanky/virtualni-studium-iluze-ci-nutnost>. ISSN 1212-8651.

18. ECKEL, Bruce. *Thinking in Java. 4th edition.* [s.1.] : Prentice Hall, 2006. 1150 s. ISBN 0131872486.

19. PEJŠA, J. *e-learning - trendy, měření efektivity, ROI, případové studie* [online]. Kontis s.r.o., 2006 [cit. 2006-08-26]. Dostupný z WWW: <http://www.e-learn.cz/soubory/e-learning_trends_ROI.pdf > .

20. *Moodle* [online]. 2010 [cit. 2010-07-24]. Dostupné z WWW: <http://moodle.org/>.

21. COLE, Jason, FOSTER, Helen. Using Moodle: Teaching with the Popular Open Source Course Management systém. 2nd edition. [s.1.] : O'Reilly Media, Inc., 2007. 282 s. ISBN 059652918X.

22. *EDoceo* [online]. 2010 [cit. 2010-07-24]. Dostupné z WWW: <http://www.edoceo.cz>.

23. *Blackboard* [online]. 2005 [cit. 2010-07-24]. Blackboard and WebCT Announce Agreement to Merge. Dostupné z WWW: <http://investor.blackboard.com/phoenix.zhtml?c=177018&p=irol-newsArticle&ID=767025>.

24. ALEXANDER, Dey. *AusWeb* [online]. 2002 [cit. 2010-07-24]. An Accessibility Audit of WebCT. Dostupné z WWW: <http://ausweb.scu.edu.au/aw02/papers/refereed/alexander/paper.html>.

25. *University of British Columbia* [online]. 2008 [cit. 2010-07-24]. UBC Computer Scientist Wins $100,000 Award for Popular Course Software. Dostupné z WWW: <http://www.e-strategy.ubc.ca/news/update0409/040929-goldberg.html>.

26. *IS/STAG* [online]. 2009 [cit. 2010-07-26]. Dostupné z WWW: <http://is-stag.zcu.cz>.

27. SIERRA, Kathy. *Head First Servlets and JSP. 2nd edition.* [s.1.] : O'Reilly Media, Inc., 2008. 911 s. ISBN 0596516681.

28. HORTON, Ivor. *Java 5*. Praha : Neokortex, spol. s.r.o., 2005. 1422 s. ISBN 80-86330-12-5.

29. *Spring Source* [online]. 2010 [cit. 2010-07-26]. Dostupné z WWW: <http://www.springsource.org/>.

30. *Rutgers* [online]. 2008 [cit. 2010-07-26]. How CAS Works. Dostupné z WWW: <http://idms.rutgers.edu/cas/how_does_it_work.shtml>.

31. *Unicon* [online]. 2010 [cit. 2010-07-26]. Central Authentication Service (CAS). Dostupné z WWW: <http://www.unicon.net/opensource/cas>.

32. PECINOVSKÝ, Rudolf. *Návrhové vzory*. Vyd. 1. Brno : Computer press, 2007. 527 s. ISBN 978-80-251-1582-4.

33. ARLOW, Jim; NEUSTADT, Ila. *UML2 a unifikovaný proces vývoje aplikací*. Brno : Computer press, 2008. 567 s. ISBN 978-80-251-1503-9.

34. BLOCH, Joshua. *Java efektivně : 57 rad softwareového experta*. Praha : Grada Publishing, 2002. 232 s. ISBN 80-247-0416-1.