

Specifikace Ročníkového projektu ([NPRGo45](#))

*Autor projektu: **Jakub Kinš***

*Vedoucí projektu: **Mgr. Miroslav Týnovský***

1. Úvod
 2. Analýza konkurence, popis odlišností
 3. HTML wireframy udávající strukturu uživatelského rozhraní
 4. Návrh struktury programu
 - a. CourseList
 - b. Course
 - c. Assignments
 - d. Results
 - e. Events
 - f. Resources
 - g. User
 - h. Forum
 - i. Messages
 - j. Settings
 - k. Mail
 5. Použité knihovny a nástroje
-

1. Úvod

Tento dokument navazuje na zadání tématu ročníkového projektu. Projekt bude mít pracovní název Course Manager (dále jen CM). V dalších bodech bude podrobněji rozepsaná architektura aplikace, analyzována konkurence a určena základní struktura programu.

2. Analýza konkurence, popis odlišností

Jedinou podobně tématicky zaměřenou aplikací dostupnou na trhu, je systém Moodle (www.moodle.com). Moodle se zaměřuje na e-learning a na podporu klasického vyučování. Jedná se o open-source aplikaci, která je dostupná zdarma. V následujících bodech jsou shrnuty odlišnosti od této aplikace.

- a. Moodle je určen zejména na e-learning, čili studium na dálku přes internet. CM má za cíl podpořit klasické studium prezenční. Moodle samozřejmě také může podporovat prezenční studium, ale není to jeho hlavní účel.

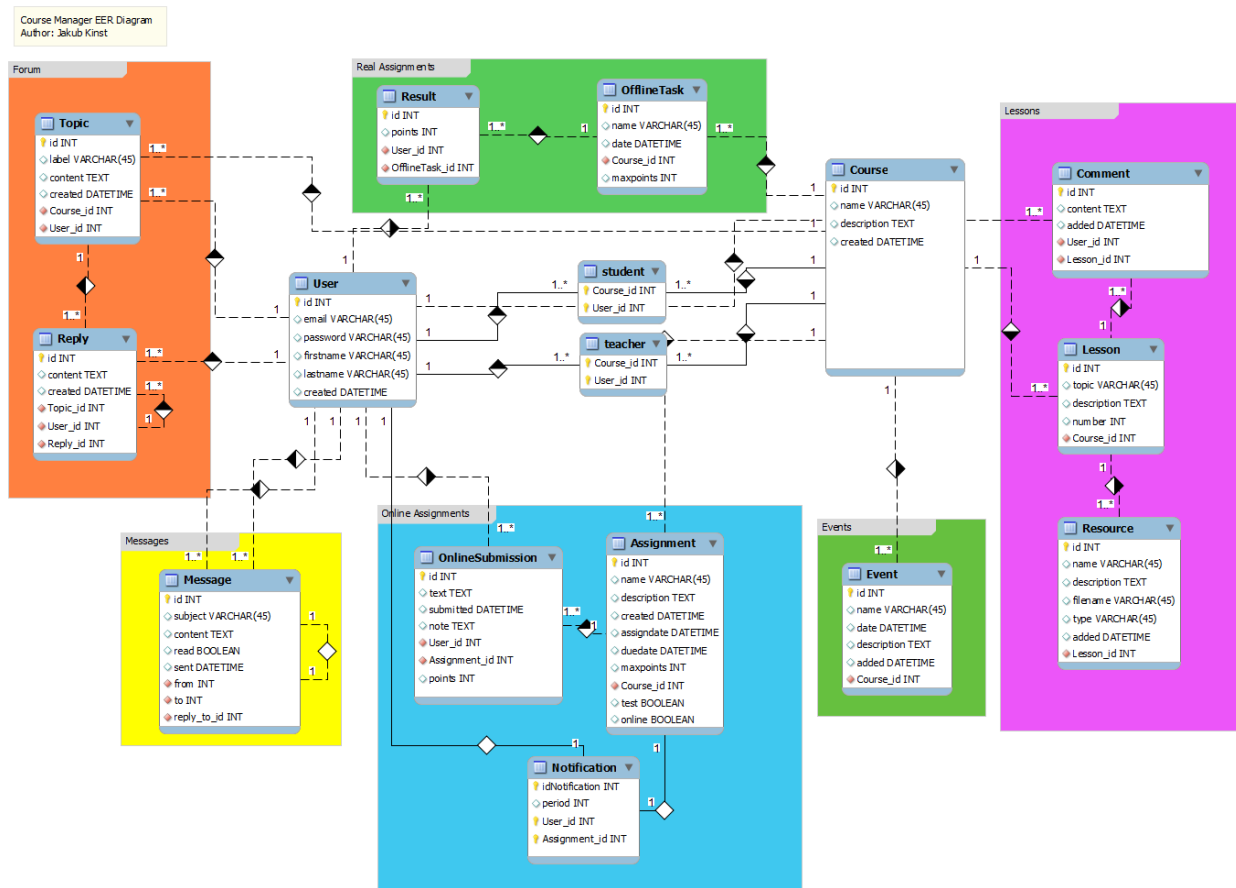
- b. CM bude od začátku vyvíjen jako internetová služba, ne balík aplikací, jako je tomu u Moodlu. Pokud chce lektor používat Moodle, musí jeho škola tuto aplikaci provozovat na svém serveru. CM bude veřejná služba, která bude na škole naprosto nezávislá.
- c. Služba bude mnohem uživatelsky jednodušší a příjemnější než Moodle. Ten je poněkud složitý a místy i chaotický. Uživatel je neustále zahrnován obrovským množstvím textu a jiných dat. Rozhraní bude v podobném jednoduchém stylu jako u portálu Facebook.
- d. Moodle ke svému běhu potřebuje administraci ze strany zaměstnanců školy. Kurzy mohou vytvářet pouze administrátoři, na žádost lektora atd. V CM si kurzy budou moci lektori založit jednoduše sami. Dále nebude k tvorbě obsahu potřeba žádná vyhrazená osoba.
- e. CM se také zaměří na to, aby lektor mohl vytvořit jednoduchý test pro studenty, tak že zadá otázky a studenti budou moci v libovolnou dobu nebo v předem stanovený čas test vyplnit a včas odevzdat. V tomto ohledu jsou vlastnosti Moodlu dost omezené. Dále bude možné, aby lektor zadal kontrolní kvíz na procvičování znalostí (např ABCD, který se sám opraví). S testy souvisí i odpovídající klasifikace, která samozřejmě pojme i hodnocení z běžné výuky.

3. HTML wireframy udávající strukturu uživatelského rozhraní

Přiložené HTML (složka */mockup/*) soubory udávají budoucí strukturu uživatelského rozhraní, přibližné rozložení ovládacích prvků a přibližné rozložení (layout) stránek. Červené odkazy jsou v návrhu aktivní.

4. Návrh struktury programu

Následující obrázek ilustruje návrh na databázový model aplikace. Tabulky nejčastěji reprezentují jednotlivé entity týkající se hlavní aplikační logiky. Další administrativní záznamy (záznamy přihlášení apod.) jsou v návrhu vynechány.



Výstup ze softwaru MySQL Workbench

Architektura aplikace bude založena na principu *MVC - Model-View-Controller* (<http://cs.wikipedia.org/wiki/Model-view-controller>), která jasně odděluje tři vrstvy aplikace - uživatelské rozhraní (View), řídicí logiku (Controller) a datový model (Model). Pro implementaci v Nette Frameworku je potřeba malé odlišnosti - využití architektury *MVP - Model-View-Presenter*, která se liší jen v drobnostech.

Každý logický celek je tedy potřeba na tyto tři vrstvy správně rozdělit. Jedna část se bude starat o samotnou logiku, výpočty apod. (Model), druhá část (Presenter) bude přijímat vstup od uživatele a podle toho, po “domluvě” s modelem, upravovat prezentovaná data a třetí část bude samotné rozhraní pro uživatele (View), které v podstatě jen “hloupě” přijme data a zobrazí je na výstup pomocí šablon.

Aplikace tedy bude mít striktně oddělené programovací technologie. Technologie HTML, PHP, JavaScript i kaskádové styly CSS budou naprosto oddělené. To znamená, že budou fyzicky v jiných zdrojových souborech.

Díky předchozímu bude v budoucnu velmi jednoduché jakoukoliv část aplikace rychle modifikovat jiným potřebám. Zároveň bude snadné program během vývoje ladit, popřípadě hledat a opravovat chyby.

Aplikace bude od začátku vyvíjena tak, aby byla v budoucnu možná snadná lokalizace do jakéhokoliv jiného jazyka. Všechny textové řetězce budou v externích

souborech. (Pravděpodobně buď XML nebo speciální třída v PHP). Výchozí jazyk bude Angličtina.

Logické moduly:

a. CourseList

Tato část má za úkol vypisovat seznam kurzů, jichž je přihlášený uživatel členem. Dalším úkolem bude zavádění nových kurzů do databáze. Po přidání kurzu převezme práci modul Course, kde teprve bude možné zvát uživatele atd.

Model bude nabízet metody na výběr kurzů z databáze. Dále vložení nového kurzu do databáze nebo jeho odebrání.

b. Course

Jak již název napovídá, tento modul bude mít za úkol vykreslovat titulní stránku kurzu, včetně seznamu lekcí. Pokud to lektor v nastavení kurzu povolí, bude zobrazen i seznam studentů, popř. jejich e-mailové adresy. Dále zde bude moci lektor upravovat vlastnosti kurzu, zvát další studenty, registrovat další lektory, zaznamenávat další lekce atp. U lekcí budou zobrazeny také komentáře studentů/lektorů.

Tento modul se bude starat i o přidávání studentů do kurzu. Přidání proběhne za pomoci pozvánky. Předpokladem je to, že lektor zná e-mailovou adresu studenta a jeho jméno. Po vytvoření kurzu přes modul CourseList zadá tvůrce jména studentů společně s jejich e-mailovými adresami. Pokud je již e-mailová adresa v systému vedena jako uživatelské jméno, objeví se uživateli v systému žádost o začlenění do kurzu. Pokud student systém ještě nepoužíval, bude na adresu zaslána pozvánka do systému. Pokud uživatel potvrdí svou identitu kliknutím na kontrolní odkaz v e-mailu, bude vytvořen nový uživatel s vygenerovaným heslem, který bude rovnou zaregistrován v kurzu. Nového uživatele však již vytváří modul User.

Model nabídne metody na načtení/upravování vlastností kurzu, metody na získávání seznamu lekcí z databáze, zanášení nových lekcí, úprava lekcí, vkládání a mazání (lektorem) komentářů u lekcí atd. Dále poskytne možnost vypsát seznam uživatelů, generování e-mailových upozornění atd.

c. Assignments

Modul bude vypisovat k danému kurzu seznam aktuálních úkolů ke zpracování. Pokud je úkol typu "online", uživatel bude moci do data odevzdání odevzdat své řešení. Buď ve formě odpovědi na otázky s možnostmi A,B,C... nebo jako text, popřípadě bude možnost odevzdat libovolný soubor. Úkol však může být i typu "offline". Potom jde o klasický úkol, bez možnosti odevzdání odpovědi online.

U každého zadání si uživatel bude moci zvolit, zda chce být o nejpozdějším datu odevzdání informován.

Lektor pak také bude mít možnost úkoly vytvářet a upravovat, popřípadě mazat. Bude mít možnost zveřejnit úkol až v nějaký předem stanovený čas a nastavit jak dlouho bude "okno" pro odevzdávání otevřeno.

Úkol také může mít příznak test. To znamená, že se nejedná až tak o domácí úkol, jako spíše o online test, který bude otevřen jen na krátký interval (např. 15 min.) a student musí v tuto dobu zadání vyřešit. Opět půjde buď o textový vstup, soubor nebo ABCD test.

Tento modul také umožní lektorovi opravovat vložená řešení úkolů/testů a následně je i obodovat.

Model bude nabízet výpis úkolů z databáze, přidávání, úpravy. Také bude do systému zanášet řešení jednotlivých studentů. Pokud se jedná o úkol typu ABCD, nabídne model i automatickou kontrolu řešení vzhledem k zadání a přidělení počtu bodů.

d. Results

Modul má za úkol zejména vypisovat tabulku výsledků studenta. Jedná se o výsledky jak klasických testů, tak i testů typu “online”, které buď opraví lektor ručně nebo je opraví systém sám (ABCD).

Dále, pokud to bude povoleno v nastavení kurzu, bude vypsána i tabulka výsledků ostatních studentů.

Model bude umět vypisovat výsledky z databáze (a to jednak výsledků online úkolů, ale také klasických “offline” úkolů, jejichž výsledky zadá lektor ručně), počítat aritmetický/vážený průměr. Dále bude moci vkládat do databáze výsledky “offline” úkolů/testů.

e. Events

Bude se jednat v podstatě o malou verzi kalendáře. Každý kurz k sobě může mít napojeno neomezené množství událostí všeho druhu. Ať už se jedná o exkurze, testy nebo jiné speciální akce. Student o těch nejbližších bude informován všude v podobě infosloupku na boku stránky. Všechny události pak bude možné shlédnout na speciální stránce. Uživatel bude mít možnost být na události upozorňován e-mailem.

Model poskytne výpis budoucích událostí a odesílání upomínek e-mailem. Dále vkládání nových událostí do databáze.

f. Resources

Tento modul nabídne lektorovi sdílení jakéhokoliv typu souboru nebo odkazu se studenty. Ke každé lekci dostane možnost připojit soubor nebo vložit odkaz na externí lokaci.

Model se postará o uložení souboru do adresářové struktury aplikace a správné nalinkování s novým záznamem v databázi. Soubory bude umět také aktualizovat, popřípadě smazat. Odkazy uloží pouze do databáze.

g. User

Modul User dokáže zobrazit profil daného uživatele. Jeho profilový obrázek, kontaktní údaje atp. Modul se také stará o zobrazení průvodce vytvořením nového uživatele (i na základě pozvánky). Uživatel pomocí modulu může i upravovat své údaje, může si změnit heslo, profilový obrázek.

Uživatel bude mít v každém kurzu jednu ze dvou rolí. Buď se bude jednat o lektora kurzu, který může kurz spravovat, nebo půjde o studenta. Díky realizaci (viz. ER diagram) bude možné, aby jeden uživatel mohl být v jednom kurzu v roli studenta a v jiném třeba v roli lektora.

Model nabídne metody k přidání uživatele, aktualizaci údajů, uložení profilového obrázku do určené složky a nalinkování s uživ. účtem v databázi, generování bezpečných hesel, kontrola bezpečnosti hesla atd.

h. Forum

Tento modul zajišťuje rychlou komunikaci mezi posluchači a lektorem pomocí

populárního stylu diskusního fóra. Fórum nabídne možnost vytvořit vlákno (téma), na které mohou ostatní vzápětí odpovídat. Uživatel bude moci své příspěvky upravit nebo smazat. Po nastavení lze také uživateli zasílat upozornění o nových tématech/odpovědích e-mailem.

Model musí nabídnout vkládání témat do databáze, následně také vkládání odpovědí a správné propojení s tématem a odpovědí na kterou odpovídá. Dále aktualizaci, popřípadě smazání příspěvků nebo celého tématu a odesílání upozornění e-mailem.

i. Messages

Modul slouží k soukromé komunikaci mezi dvěma uživateli. Uživatel může odeslat zprávu společně s připojeným souborem. Na zprávy lze jednoduše odpovídat.

Model musí zvládnout zanést zprávu do databáze a zobrazit ji u příjemce ve složce doručených zpráv. Také bude moci posílat upozornění na novou zprávu e-mailem.

j. Settings

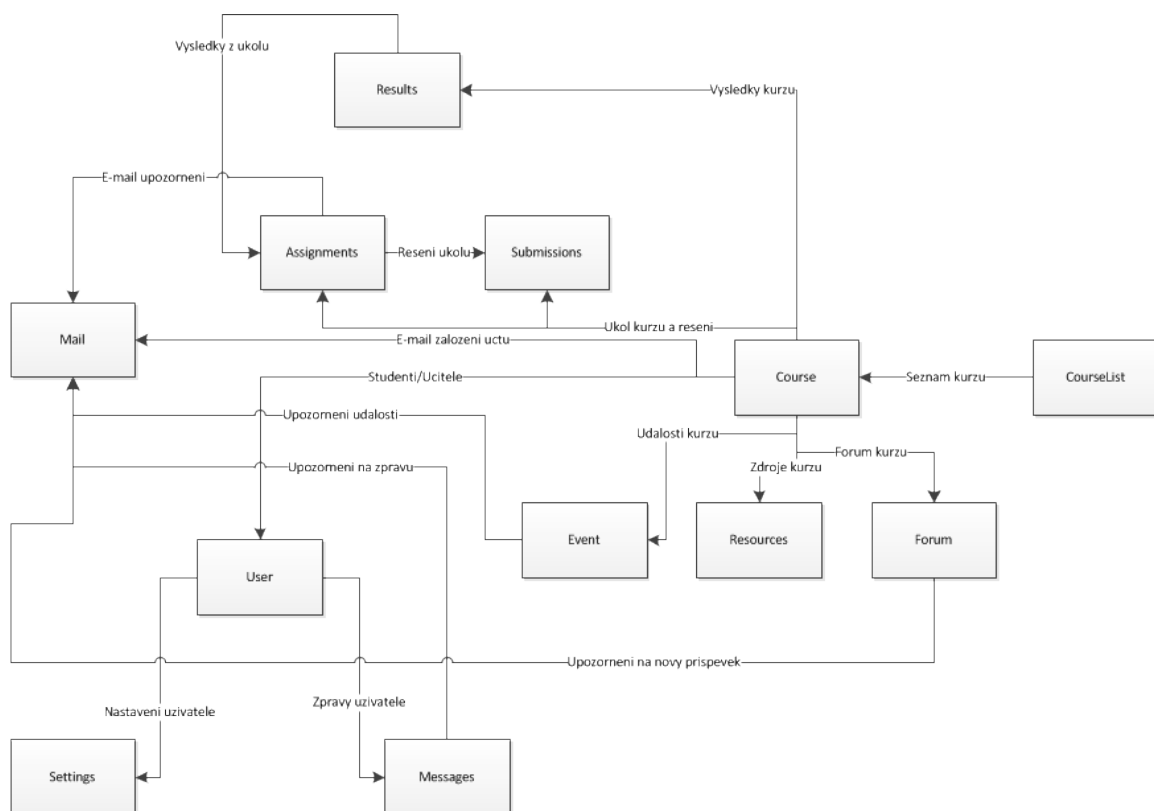
Zde bude možné nastavovat uživatelské preference systému. Nastavit možnosti zobrazení, výchozí stránky, možnosti zasílání e-mailových upozornění v rámci jednotlivých modulů.

Model poskytne potřebné metody pro aktualizaci nastavení. Dále bude nabízet výchozí profil nastavení, který bude přidělen novému účtu.

k. Mail

Modul bude používán kdykoliv, když bude potřeba odeslat mail uživateli. Model se postará o vytvoření struktury mailu s potřebnou zprávou.

Některá upozornění lze je potřeba odeslat bezprostředně po uživatelské interakci (např. vytvoření úkolu lektorem), jiná však potřebují být odeslána automaticky v určitou dobu bez spuštění skriptu ze strany uživatele. Tyto e-maily budou proto odesílány pomocí démona *cron* na straně hostingu, který v určitý interval (například každou hodinu) spustí skript, který bude mít za úkol podívat se do databáze, zkontrolovat, zda je potřeba někomu odeslat upozornění a popřípadě je odeslat.



Graf závislostí mezi moduly

5. Použité knihovny a nástroje

Použité technologie:

- HTML (<http://www.w3.org/html/>)
pravděpodobně verze 5 s využitím některých novinek
- PHP (<http://www.php.net/>)
pravděpodobně verze 5.3
- JavaScript (<http://www.javascript.com/>)
- CSS (<http://www.w3.org/Style/CSS/>)
verze 3
- MySQL (<http://www.mysql.com/>)

Jak je již známo ze zadání, Course Manager bude postaven na základech *Nette Frameworku* (<http://nette.org/cs/>). Bude se pevně držet jeho doporučeného MVP modelu a bude dodržovat i výchozí adresářovou strukturu. Jako vrstva pro práci s databází bude sloužit knihovna *dibi* (<http://dibiphp.com/cs/>). K usnadnění stylování layoutu aplikace a pro větší kompatibilitu mezi jednotlivými prohlížeči bude použit *Blueprint CSS framework* (<http://www.blueprintcss.org/>) a k pokročilým, nejen vizuálním efektům bude využita knihovna *jQuery* (<http://jquery.com/>).

K vývoji aplikace bude využita aplikace *Netbeans IDE* (<http://netbeans.org/>) včetně doplňků pro pohodlnější práci s *Nette Frameworkem*.

Jako verzovací systém bude použit systém *Git* [\(http://git-scm.com/\)](http://git-scm.com/) a jeho doplněk pro *Netbeans*.