



Školní informační portál

Stránky projektu: <https://github.com/bouskfil/SIP>

Členové týmu:

Filip Bouška -	bouskfil@fel.cvut.cz
Tom Nováček -	novacto2@fel.cvut.cz
Dominik Hons -	honsdomi@fel.cvut.cz
Tomáš Jiran -	jiranto2@fel.cvut.cz
Vojtěch Bartovský -	vojtech.bartovsky@seznam.cz

Cvičící:

Ing. Ondřej Macek

Cvičení:

A4B33SI, 3. semestr, pondělí, 12.45

Odevzdání:

28. 10. 2013

Vize projektu

Obsah

Abstrakt	3
Zainteresované instituce a osoby	3
Zadavatel a dodavatel	3
Uživatelé systému	3
FURPS	4
Funkcionalita	4
Užitečnost/ Uživatelská použitelnost.....	4
Spolehlivost (Reliability)	4
Výkon (Performance).....	4
Rozšiřitelnost (Supportability)	4
Operační systémy.....	4
Obchodní a právní aspekty	4
Implementace	4
Časový harmonogram.....	5
Finanční odhad.....	5
Přínos projektu	5

Abstrakt

Cílem ŠIP (Školní informační portál) je zrychlit a usnadnit přístup k informacím o studiu - jak pro studenty, tak pro pedagogy. U mnoha škol je získávání těchto dat velmi složité díky jejich roztržitosti na několika portálech, což neúměrně komplikuje celkovou koordinaci výuky. ŠIP bude všechny informace shromažďovat na jednom místě, čímž se přejde organizačním nepřesnostem. Bude konzistentní a pro nové uživatele bude snadné se v něm zorientovat. Usnadní vzájemnou komunikaci mezi studenty a pedagogy. Dále bude sloužit k záznamu studijních výsledků, rezervaci zkoušek a jednoduchému zadávání a odevzdávání domácích úkolů. Systém bude přístupný pouze registrovaným uživatelům, kteří budou mít jasně dané role a oprávnění. ŠIP bude navržen jako webová aplikace, bude tedy dostupný ze všech zařízení s internetovým prohlížečem.

Zainteresované instituce a osoby

Zadavatel a dodavatel

Zadavatelem je *Vysoká škola pravdy a lásky (VŠPL)*. Konkrétně její jednatel: Ing. Tomáš Novák.

Za dodavatele:
Filip Bouška

Uživatelé systému

Uživatelé systému jsou studenti a zaměstnanci školy. Uživatelé třetích stran nemají do systému přístup. Uživatele dělíme do těchto skupin:

1. Učitelé
2. Studenti
3. Administrátor
4. Studijní oddělení

Studenti mohou přistupovat k dostupnému obsahu, jako je prohlížení informací o předmětech nebo kontakty na ostatní uživatele. U zapsaných předmětů mají přístup k zadaným úkolům, přednáškám, hodnocení předmětu atd.

Učitelé mají přístup k editaci vyučovaného předmětu (změna osnov, nahrávání přednášek, zadávání úkolů, hodnocení studentů apod.). Dále mohou prohlížet obsah stejně jako studenti.

Administrátor má plné oprávnění provádět veškeré činnosti i přístup k obsahu v plném rozsahu.

Studijní oddělení bude moci spravovat agendu všech předmětů.

FURPS

Funkcionalita

- ŠIP je primárně určen pro studenty a zaměstnance školy
- ŠIP nabízí uživatelům snadný přístup k důležitým informacím o studiu
- Studentům bude umožňovat zapisovat se do předmětů (tvořit si rozvrh), odevzdávat úkoly, prohlížet si studijní materiály
- Učitelé budou moci hodnotit studenty, zadávat úkoly, vystavovat přednášky

Užitečnost/ Uživatelská použitelnost

- ŠIP bude sdružovat informace v jedné aplikaci, odpadne tedy nutnost shánění informací na několika různých místech
- Poskytne platformu pro snadnou komunikaci mezi studenty a pedagogy
- Aplikace bude mít přehledné grafické prostředí a její použití bude intuitivní i pro technicky méně zdatné uživatele

Spolehlivost (Reliability)

- ŠIP bude realizován jako webová aplikace, nebude tedy vyžadovat instalaci
- Jelikož bude aplikace nahrána na serveru, bude k ní tedy online přístup a případné výpadky se budou moci řešit okamžitě
- Systém bude nasazen na serveru školy

Výkon (Performance)

- Rychlost odezvy bude záviset na databázovém stroji
- Nenáročný datový přenos
- Systém bude navržen tak, aby k němu mohlo přistupovat až několik stovek uživatelů zároveň

Rozšiřitelnost (Supportability)

- Do systému se v budoucnu budou moci přidávat další funkcionality, jako například služba pro výběr a odevzdávání bakalářských a disertačních prací
- V případě zájmu může být ŠIP navržen také jako mobilní aplikace

Operační systémy

Naše aplikace nemá žádné speciální požadavky na operační systémy.

Obchodní a právní aspekty

Zakoupená aplikace smí být použita pro jednu školu a nesmí být dále šířena.

Implementace

Aplikace bude napsána v programovacím jazyce Java a bude použit framework Java Play.

Časový harmonogram

Termín dodání podkladů podle kterých bude provedena analýza je 1.10. 2013. Zároveň se tento den uskuteční schůzka jednatelů obou stran. Schůzka bude sloužit k vyjasnění detailů projektu.

Termín pro vypracování kompletní analýzy je stanoven na 28.10. 2013. Mezi tímto termínem se zadavatel zavazuje poskytnout nejméně dvě konzultační schůzky. Termíny schůzek budou domluveny individuálně. První verze analýzy (1.0) bude předána 2.11. 2013 panem Filipem Bouškou.

Po předání finální analýzy má zadavatel lhůtu 5 pracovních dnů, kdy musí předat písemné vyjádření, zda bude vytvořena smlouva na realizaci projektu dle přiložené dokumentace.

Finanční odhad

Odhad není nikterak závazný a slouží čistě k odhadu finančních nákladů.

Aktivita	Doba	cena	info
Shromažďování požadavků	15	4 500 Kč	2 cvičení po 1,5 h o 5 lidech vynásobeno 300 Kč
Design aplikace	82,5	24 750 Kč	11 cvičení po 1,5 h o 5 lidech vynásobeno 300 Kč
Implementace	84	25 200 Kč	Čas pro implementaci vynásobeno 300 Kč
Testování	50	15 000 Kč	Čas pro testování vynásobeno 300 Kč
Suma		69 450 Kč	Součet výdajů
Zisk		10 418 Kč	15 % zisk pro firmu
Cena		79 868 Kč	Celková cena

Přínos projektu

Jako největší přínos projektu si slibujeme usnadnění studia pro studenty, jelikož náš systém bude shromažďovat vše potřebné pro studium. Od informací o předmětech až po komunikaci s učiteli. Přínosem pro učitele bude snadná orientace v domácích úkolech, jejich vystavování a následné hodnocení.

ŠKOLNÍ INFORMAČNÍ PORTÁL

Dokumentace k 4. iteraci

Obsah dokumentu

Business Process Models	2
Přihlašování na ZK	2
Zapsání do semestru	5
Proces úkolů	10
Závěrečné hodnocení	13
Business Domain Model	16
Requirements.....	18
Requirements Model.....	18
Funkční požadavky	19
Obecné/kvalitativní požadavky	28
Use Cases.....	29
Správa účtů	29
Komunikace	32
Předměty.....	34
Rozvrh.....	35
Úkoly	37
Zkoušky	39
Use-case Mapping	42
Analytický Doménový Model	45
Architecture Model	57
Database Model	63
Component Model	64
Deployment Model	64
Sequence Model	65

Business Process Models

Přihlašování na ZK

Přihlašování na ZK - (Activity diagram)

Created By: Tomáš on 8.11.2013, Last Modified: 8.11.2013

Model zachycuje procesy spojené s vypisováním termínů zkoušek učiteli a přihlašováním na termíny studenty.

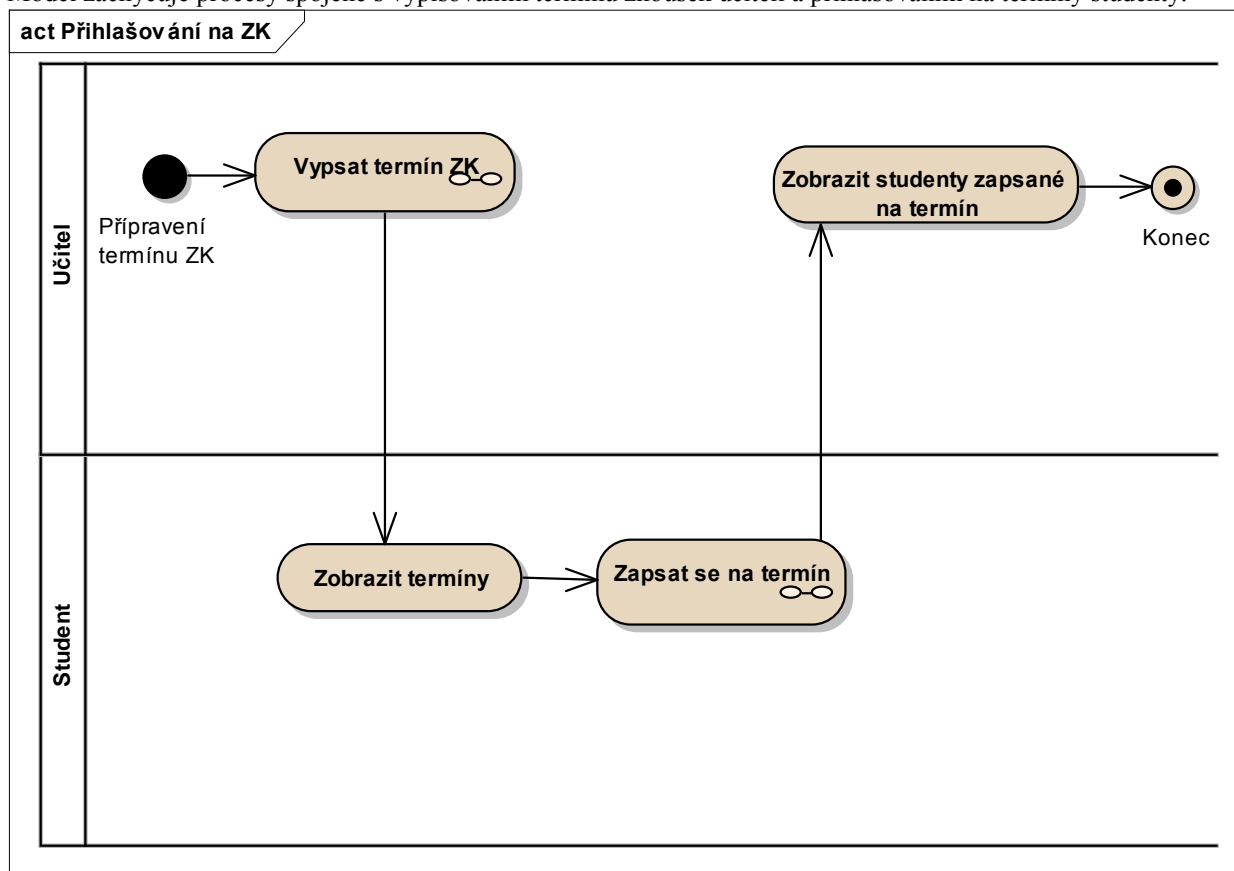


Figure: 1

Vypsat termín ZK

Vypsat termín ZK - (Activity diagram)

Created By: Tomáš on 8.11.2013, *Last Modified:* 8.11.2013

Figure: 2

Zapsat se na termín

Zapsat se na termín - (Activity diagram)

Created By: Tomáš on 8.11.2013, *Last Modified:* 8.11.2013

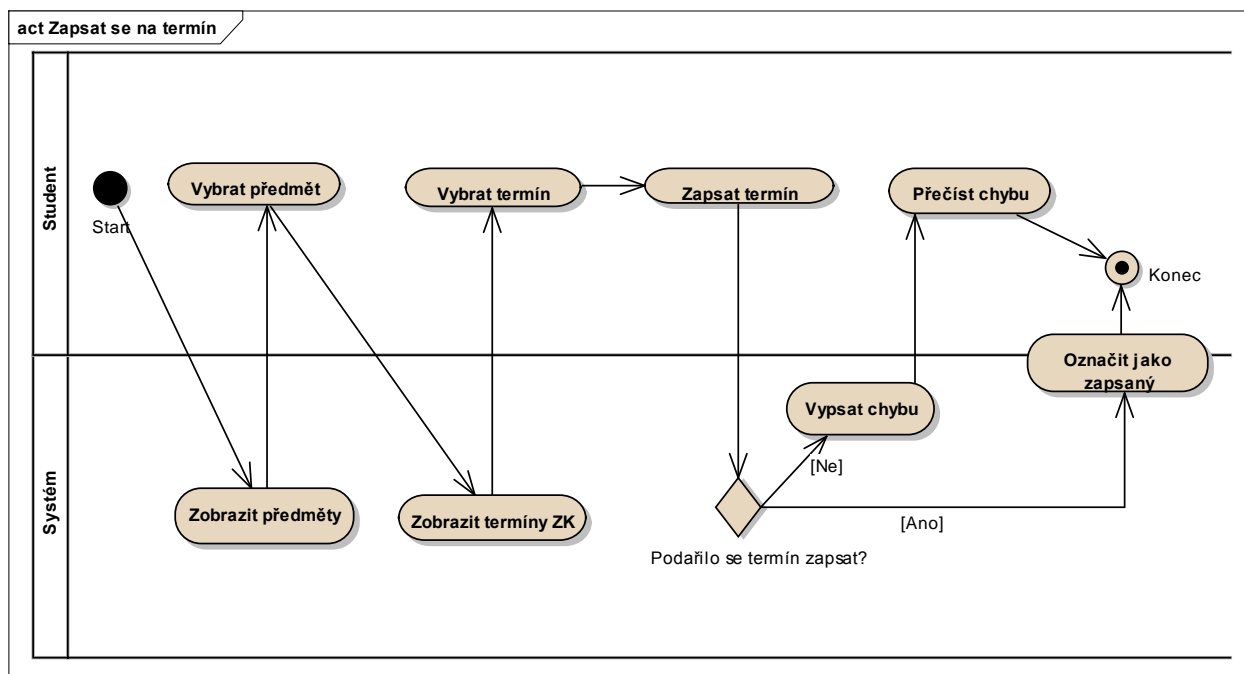


Figure: 3

Zapsání do semestru

Zapsání do semestru - (Activity diagram)

Created By: Administrator on 13.10.2013, Last Modified: 28.10.2013

Model popisuje procesy související se zapisováním předmětů.

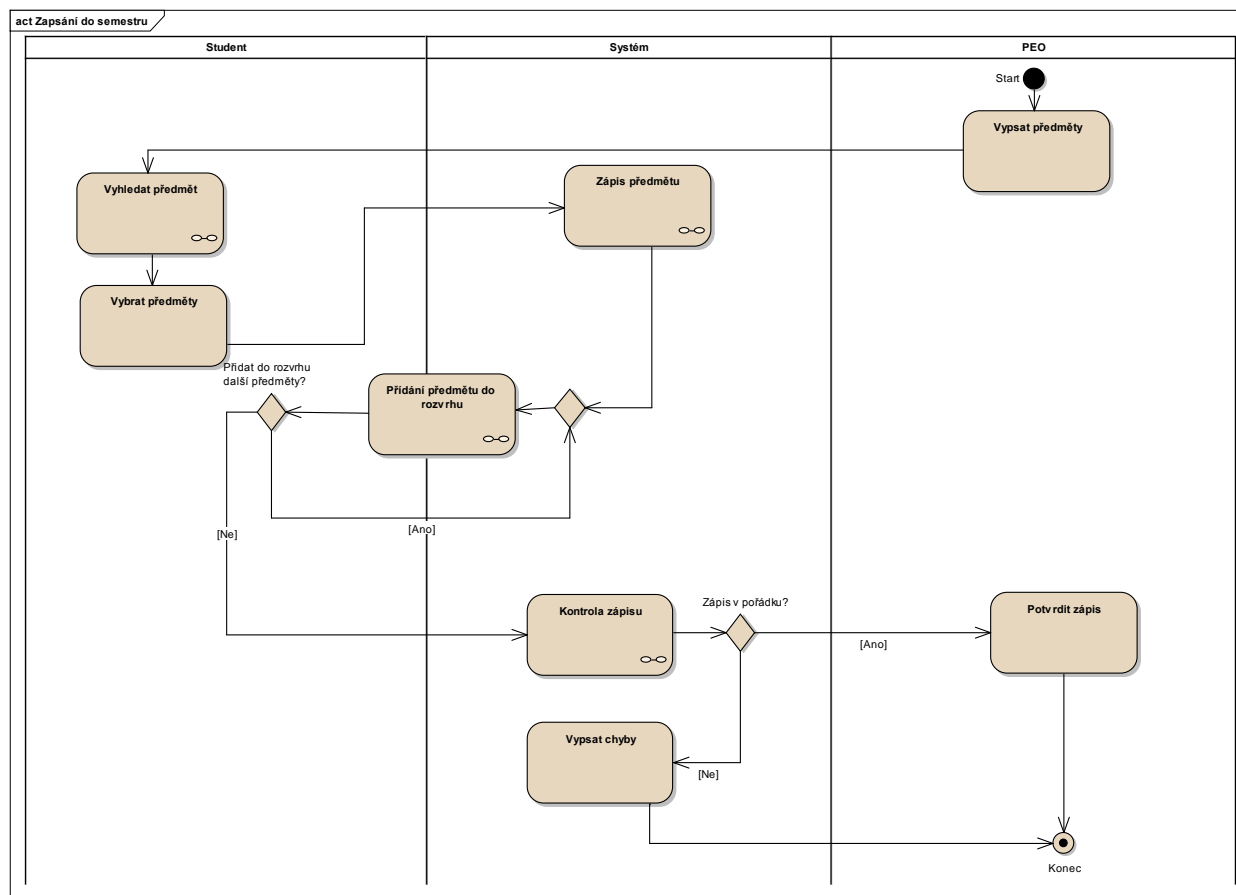


Figure: 4

Kontrola zápisu

Kontrola zápisu - (Activity diagram)

Created By: Administrator on 6.10.2013, Last Modified: 27.10.2013

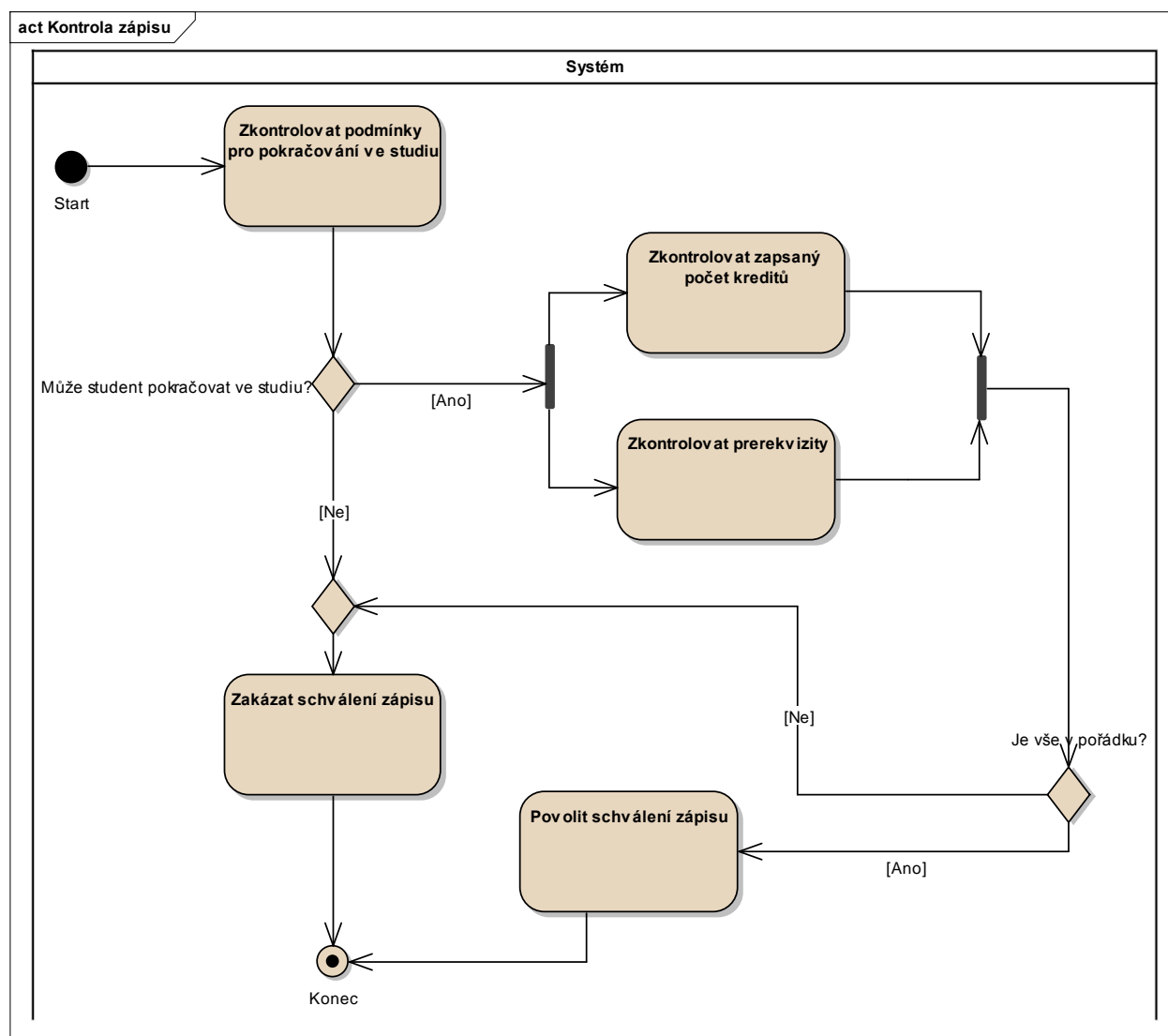


Figure: 5

Přidání předmětu do rozvrhu

Přidání předmětu do rozvrhu - (Activity diagram)

Created By: Administrator on 6.10.2013, Last Modified: 21.11.2013

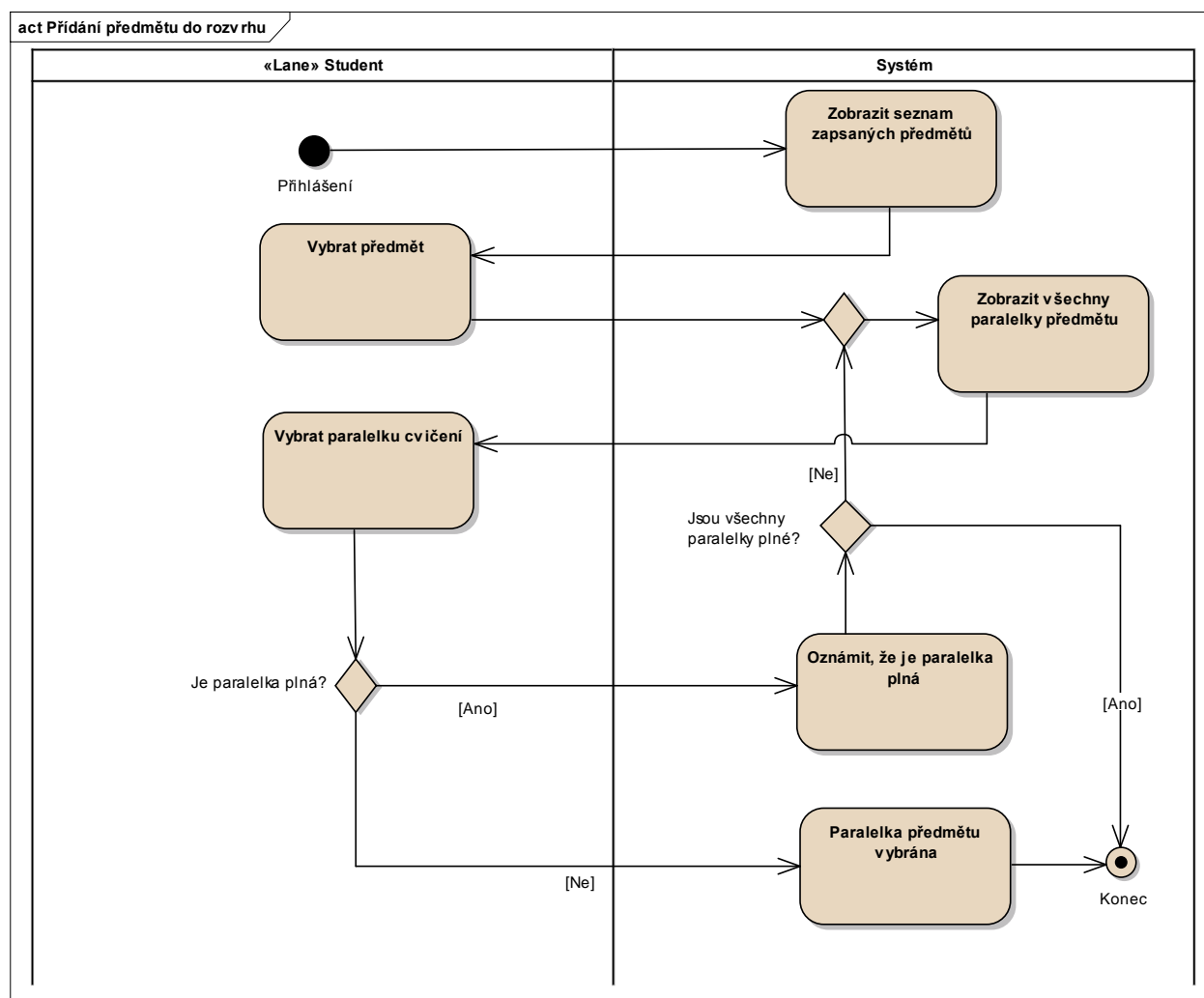


Figure: 6

Vyhledat předmět

Vyhledat předmět - (Activity diagram)

Created By: Administrator on 6.10.2013, Last Modified: 27.10.2013

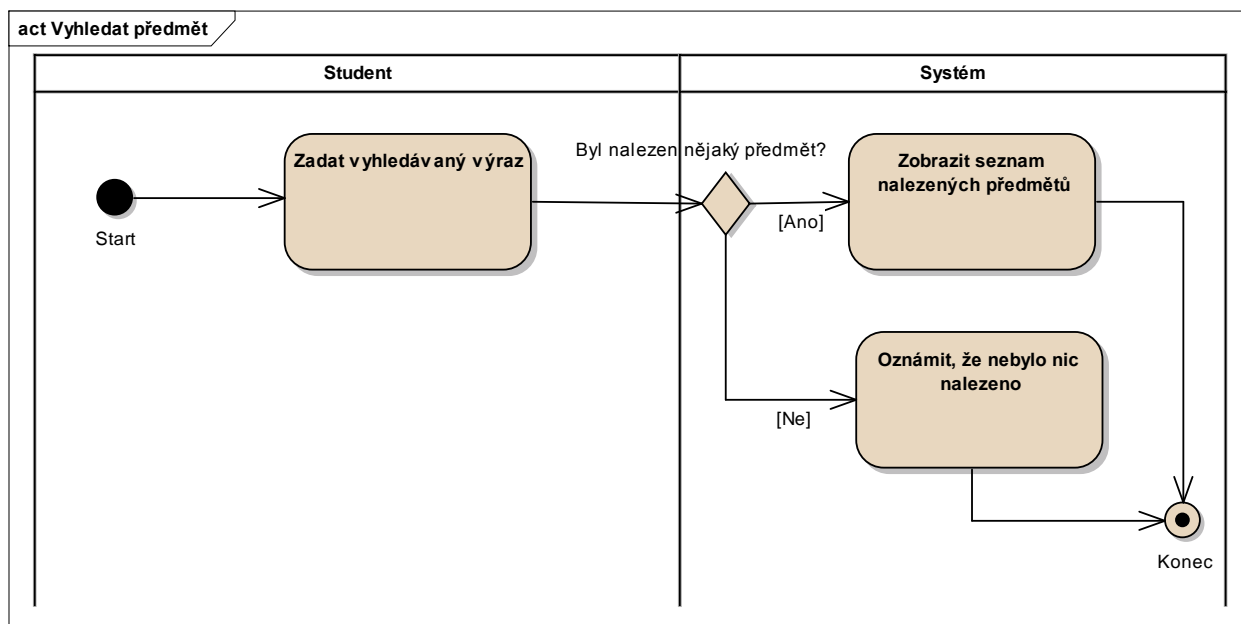


Figure: 7

Zápis předmětu

Zápis předmětu - (Activity diagram)

Created By: Administrator on 6.10.2013, *Last Modified:* 27.10.2013

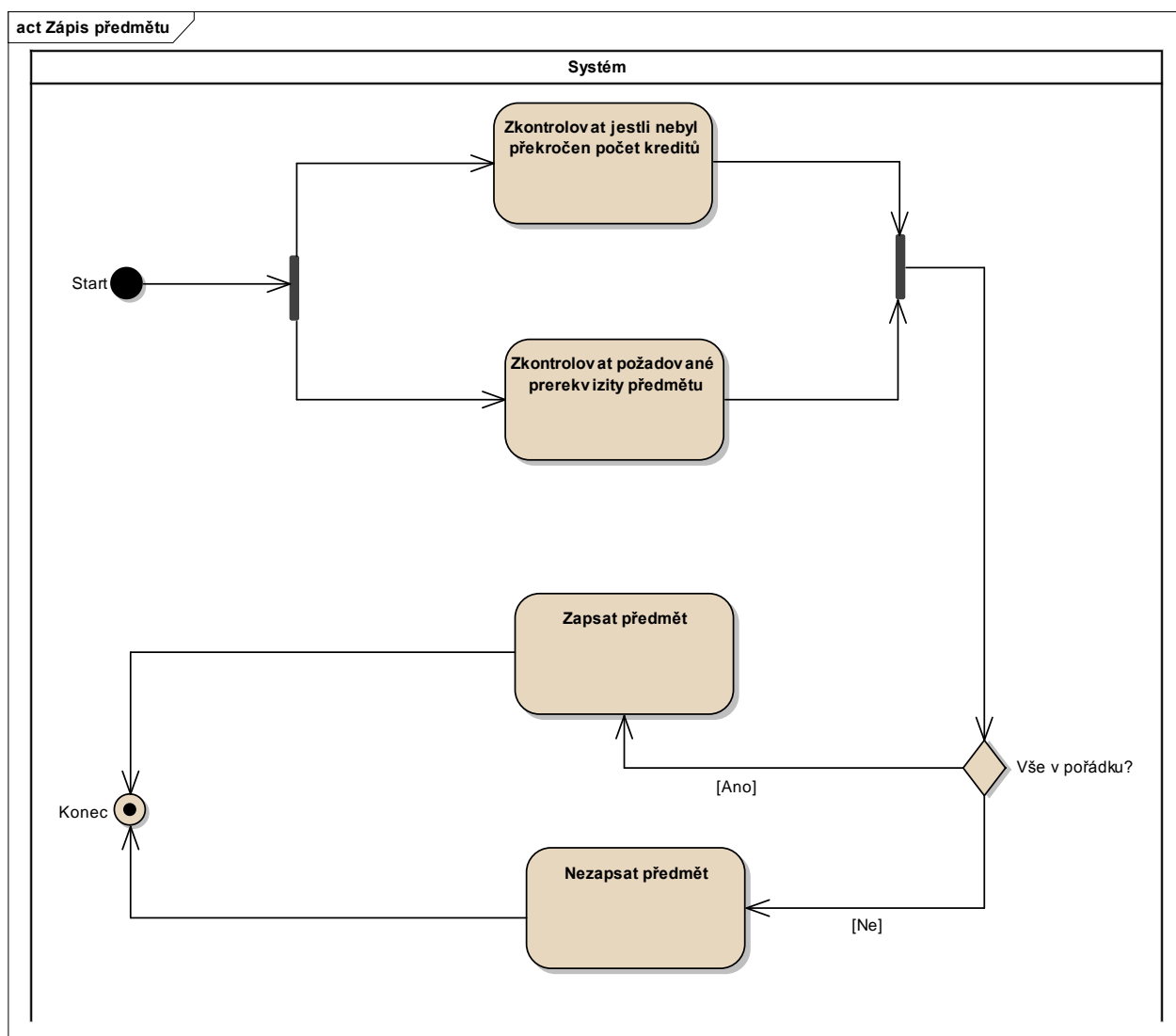


Figure: 8

Proces úkolů

Proces úkolů - (Activity diagram)

Created By: Dominik on 8.10.2013, Last Modified: 10.11.2013

Model popisuje procesy související se zadáváním a odevzdáváním domácích úkolů.

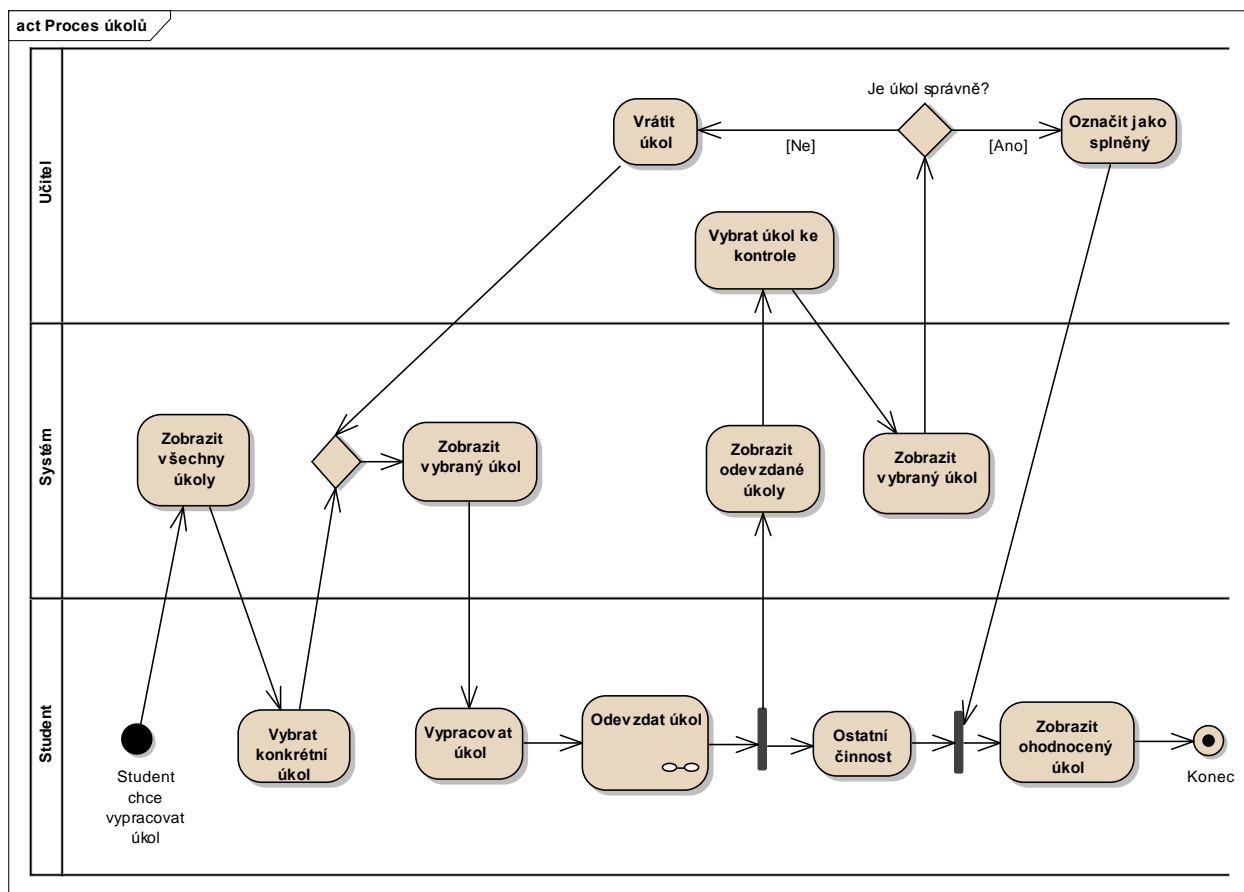


Figure: 9

Odevzdat úkol

Odevzdání úkolu - (Activity diagram)

Created By: Dominik on 15.10.2013, Last Modified: 8.11.2013

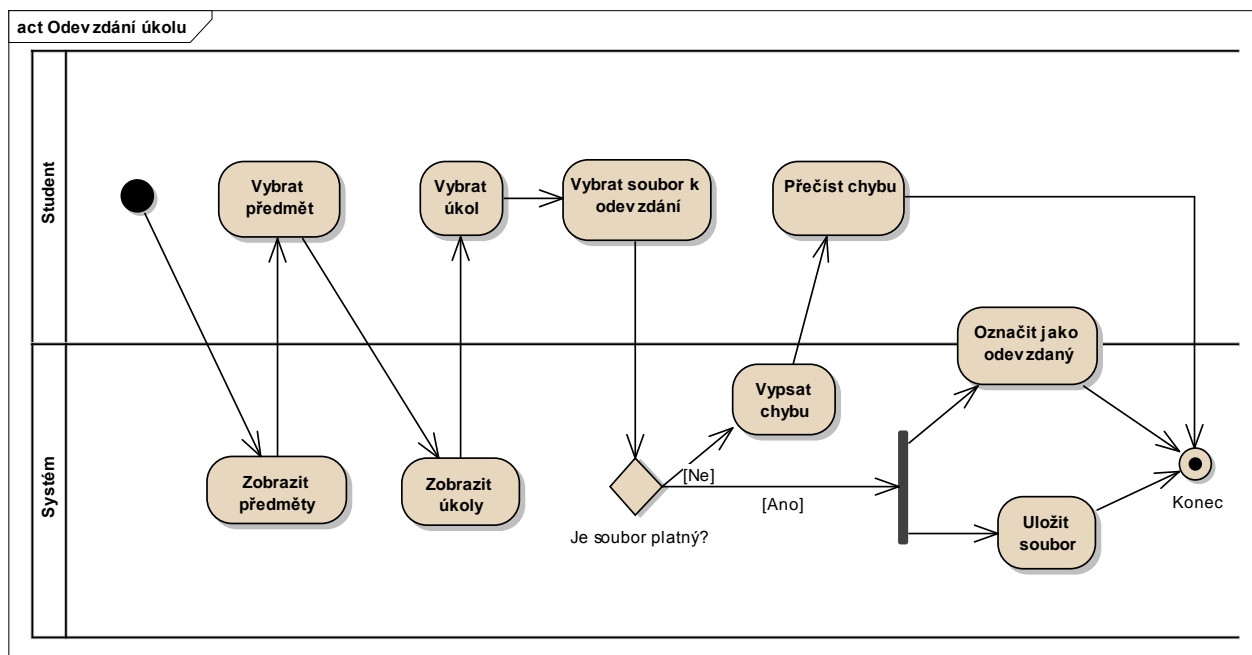


Figure: 10

Zadat úkol

Zadat úkol - (Activity diagram)

Created By: Dominik on 15.10.2013, Last Modified: 6.11.2013

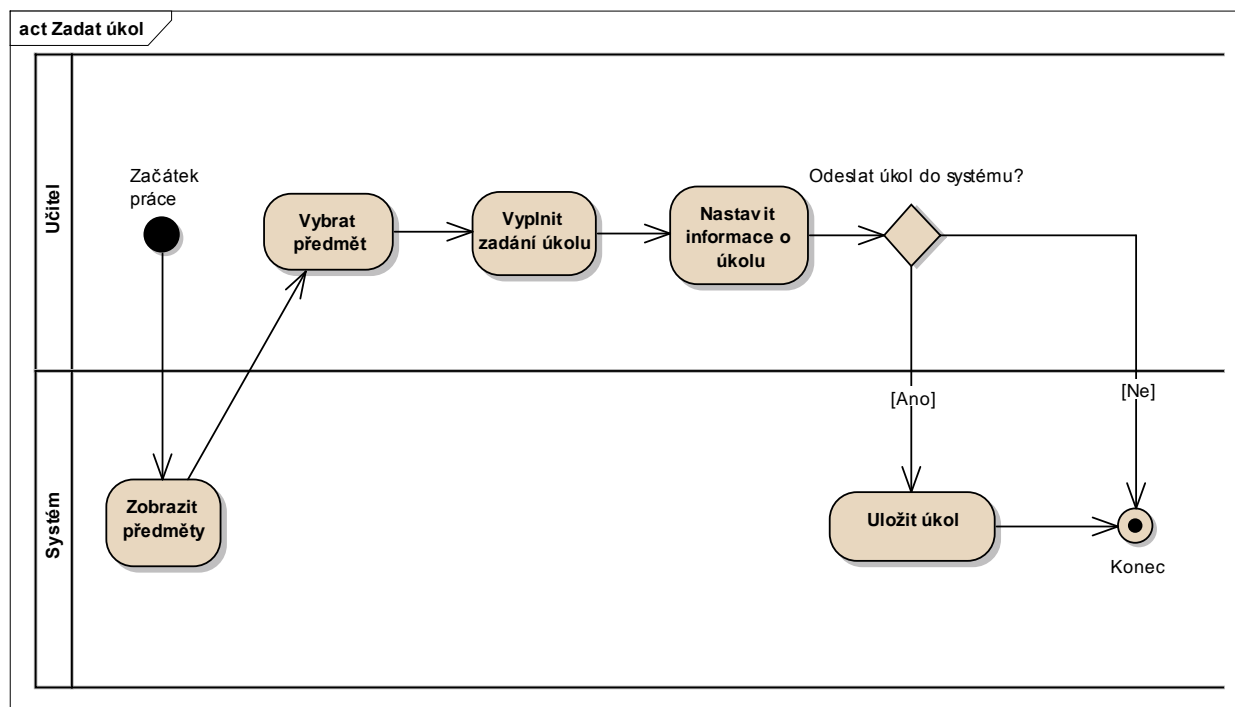


Figure: 11

Závěrečné hodnocení

Závěrečné hodnocení - (Activity diagram)

Created By: Tom on 12.10.2013, *Last Modified:* 10.11.2013

Model popisuje procesy při udělování známek studenta.

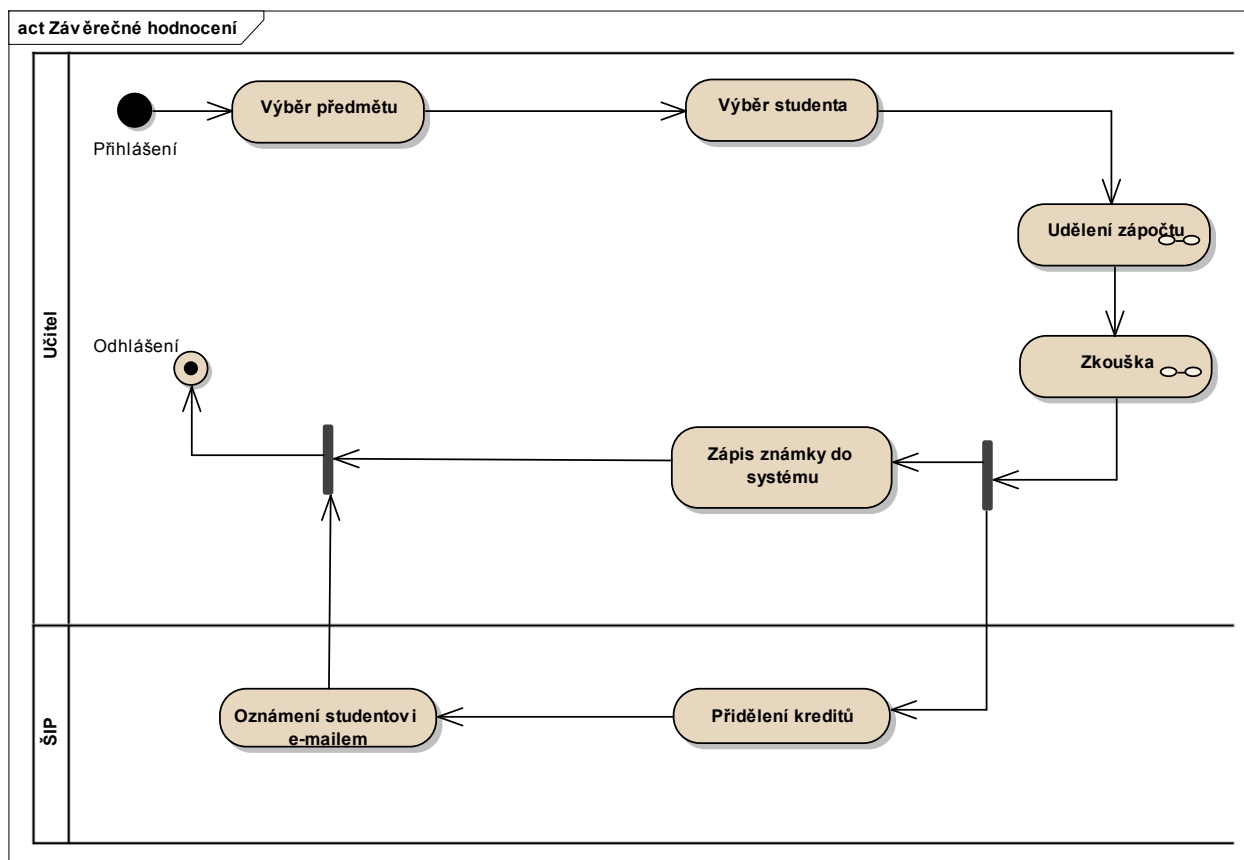


Figure: 12

Zkouška

Zkouška - (Activity diagram)

Created By: Tom on 20.10.2013, *Last Modified:* 8.11.2013

Proces zkoušení studenta a udělení známky za zkoušku.

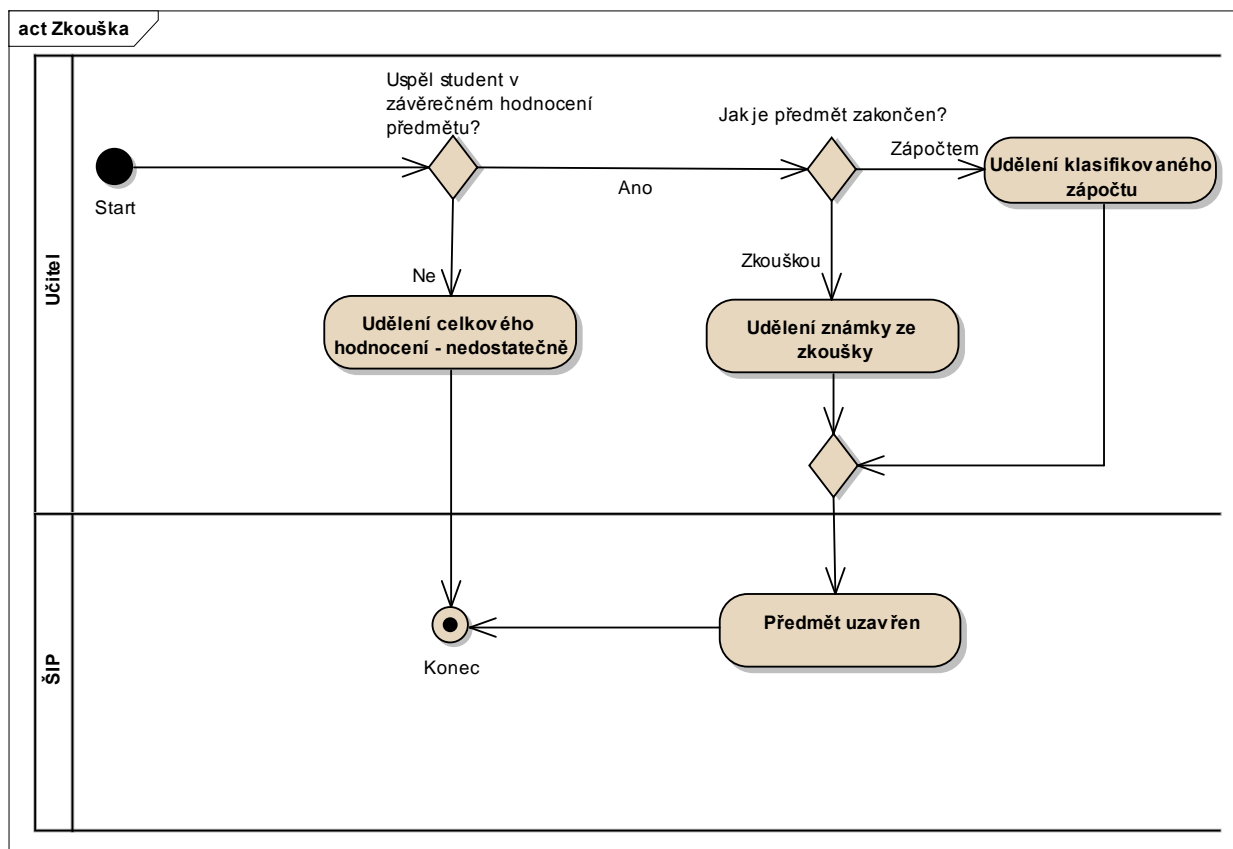


Figure: 13

Figure: 14

Udělení zápočtu

Zápočet - (Activity diagram)

Created By: Tom on 20.10.2013, Last Modified: 28.10.2013

Proces popisuje udělení zápočtu studentovi.

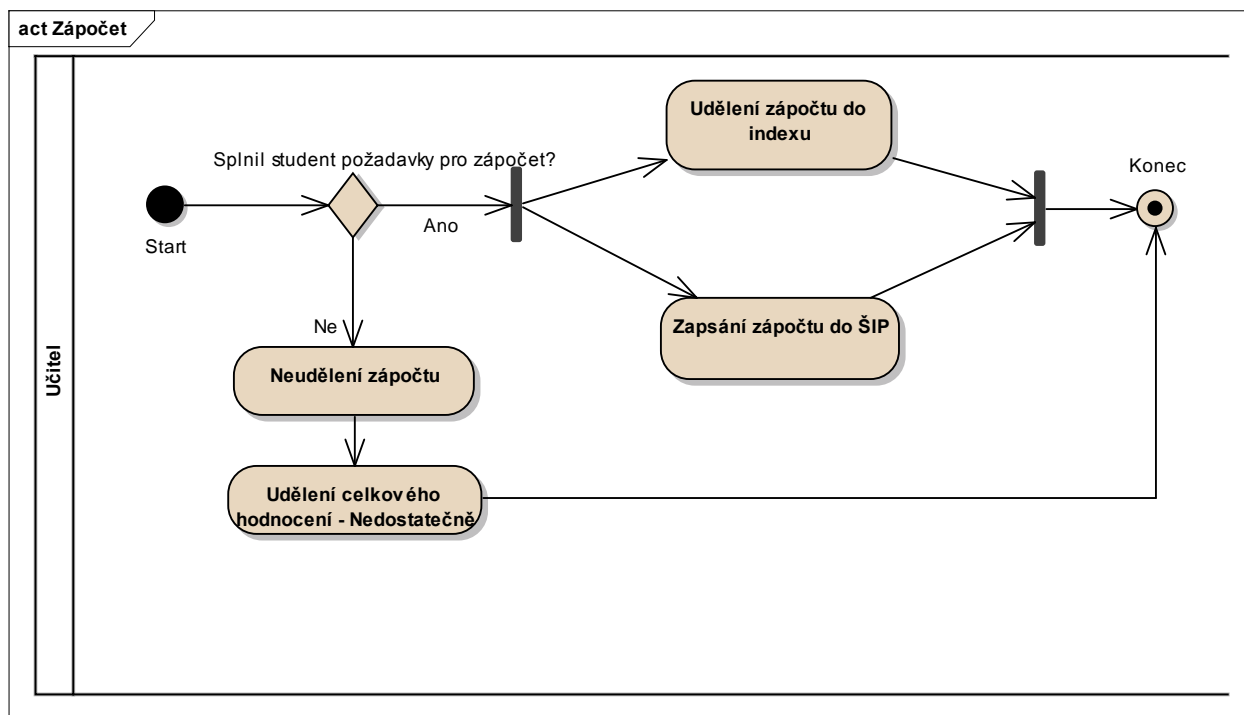


Figure: 15

Business Domain Model

Domain Model - (Class diagram)

Created By: Dominik on 13.10.2013, *Last Modified:* 22.11.2013

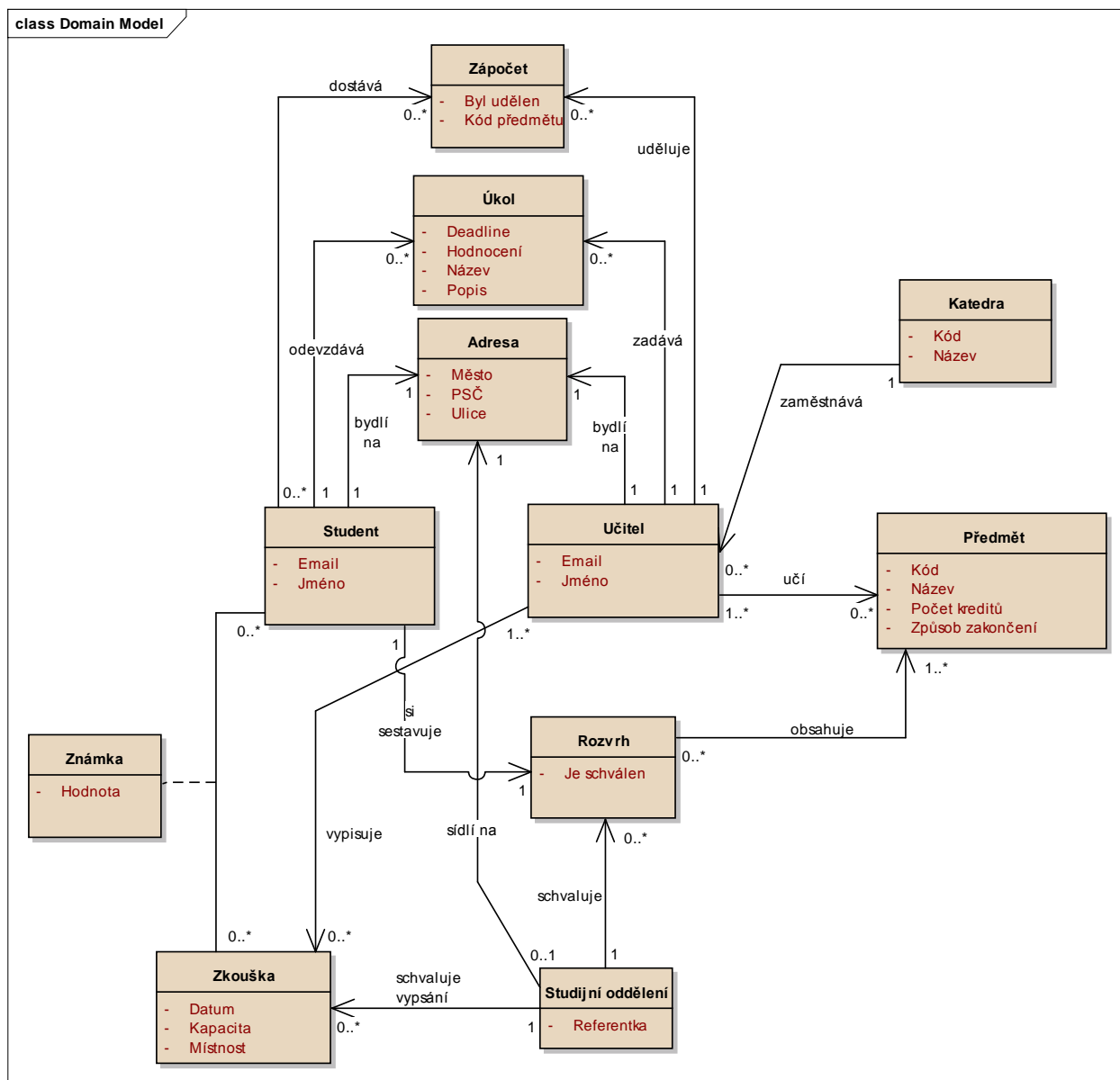


Figure: 16

Requirements

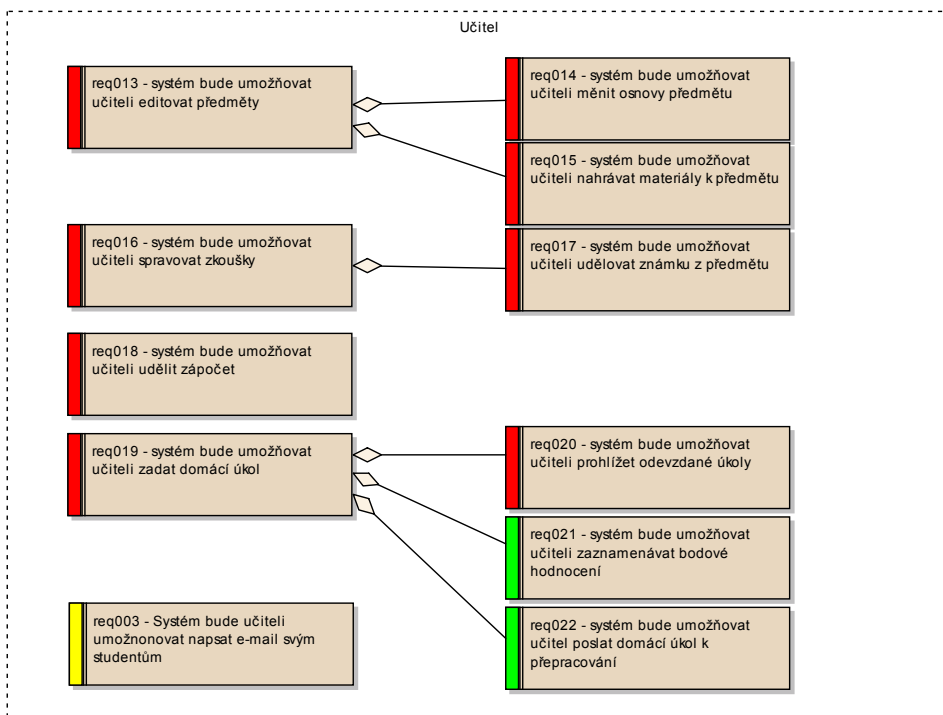
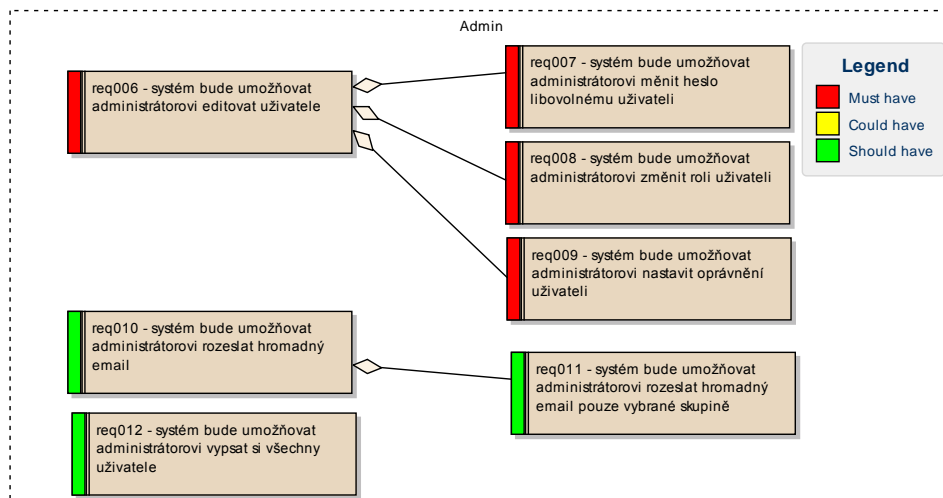
Requirements Model

req Požadavky	
Funkční požadavky	Obecné/kvalitativní požadavky
<ul style="list-style-type: none"> + req001 - systém bude umožňovat uživateli, aby se přihlásil + req002 - systém bude umožňovat uživateli změnit si heslo + req003 - Systém bude učitelům umožňovat napsat e-mail svým studentům + req004 - Systém bude umožňovat e-mailovou komunikaci mezi uživateli. + req005 - systém bude umožňovat pro editovat předměty + req006 - systém bude umožňovat administrátorovi editovat uživatele + req007 - systém bude umožňovat administrátorovi měnit heslo libovolnému uživateli + req008 - systém bude umožňovat administrátorovi změnit roli uživatele + req009 - systém bude umožňovat administrátorovi nastavit oprávnění uživatele + req010 - systém bude umožňovat administrátorovi rozeslat hromadný email + req011 - systém bude umožňovat administrátorovi rozeslat hromadný email pouze vybrané skupině + req012 - systém bude umožňovat administrátorovi vypsat si všechny uživatele + req013 - systém bude umožňovat učitelům editovat předměty + req014 - systém bude umožňovat učitelům měnit osnovy předmětů + req015 - systém bude umožňovat učitelům nahraďovat materiály k předmětům + req016 - systém bude umožňovat učitelům spravovat zkoušky + req017 - systém bude umožňovat učitelům udělovat známky z předmětů + req018 - systém bude umožňovat učitelům udělit zápočet + req019 - systém bude umožňovat učitelům zadat domácí úkol + req020 - systém bude umožňovat učitelům prohlížet odevzdané úkoly + req021 - systém bude umožňovat učitelům zaznamenávat bodové hodnocení + req022 - systém bude umožňovat učitelům poslat domácí úkol k přepracování + req023 - Systém bude umožňovat pro zadat informace o předmětu + req024 - Systém bude umožňovat pro smazat předmět + req025 - systém bude umožňovat pro měnit osnovy předmětů + req026 - systém bude umožňovat automatické rozeslání emailu s upozorněním na neschválený zápis + req027 - systém bude umožňovat kontrolu prerekvizit zapisovaných předmětů + req028 - systém bude umožňovat výpis seznamu všech předmětů k zápisu + req029 - systém bude umožňovat kontrolu dostatku kreditů pro zápis do dalšího semestru + req031 - systém bude umožňovat pro smazat libovolného studenta + req032 - systém bude umožňovat pro měnit osobní údaje u studenta + req033 - systém bude upozorňovat studenta na kolizi předmětů + req034 - systém bude umožňovat studentovi zapsat si předmět + req035 - systém bude umožňovat studentovi přidat si předmět do rozvrhu + req036 - systém bude umožňovat studentovi zapsat se na zkoušku + req037 - systém bude umožňovat studentovi zobrazit si svůj rozvrh + req038 - systém bude umožňovat studentovi vytisknout si svůj rozvrh + req039 - systém bude umožňovat studentovi odevzdat domácí úkol + req040 - systém bude umožňovat studentovi zobrazit jeho studijní výsledky + req041 - systém bude umožňovat studentovi zobrazit si hodnocení domácího úkolu + req042 - systém bude umožňovat studentovi zrušit si zapsaný předmět + req043 - systém bude umožňovat studentovi odebrat si předmět z rozvrhu + req049 - systém bude umožňovat pro přidávat předměty + req030 - systém bude umožňovat pro přidat studenta do systému 	<ul style="list-style-type: none"> + req044 - systém bude implementován v jazyce Java + req045 - systém bude implementován pomocí frameworku Java Play + req046 - systém bude realizován jako webová aplikace + req047 - systém bude optimalizován pro prohlížeč Chrome + req048 - systém bude lokalizován do angličtiny

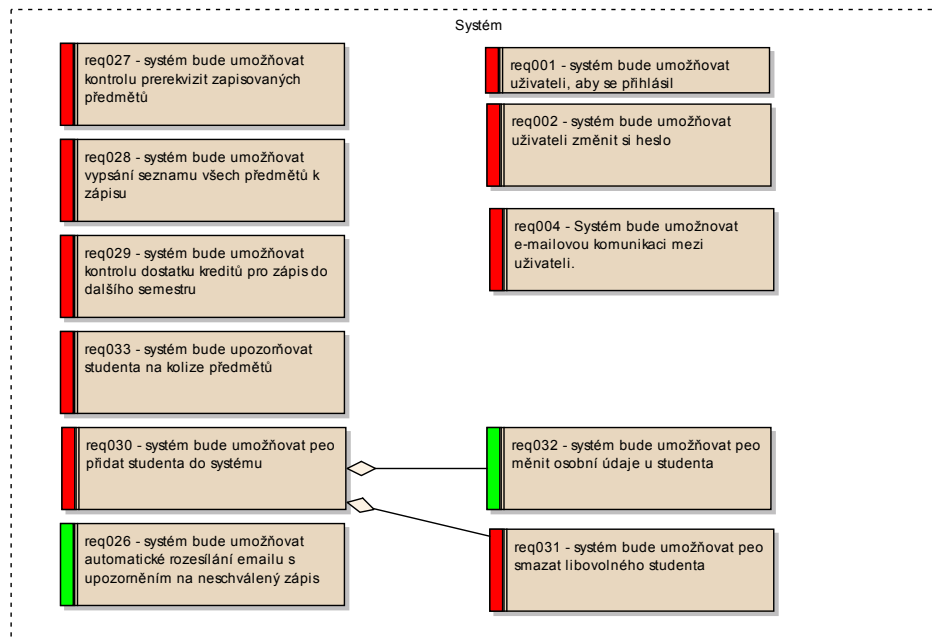
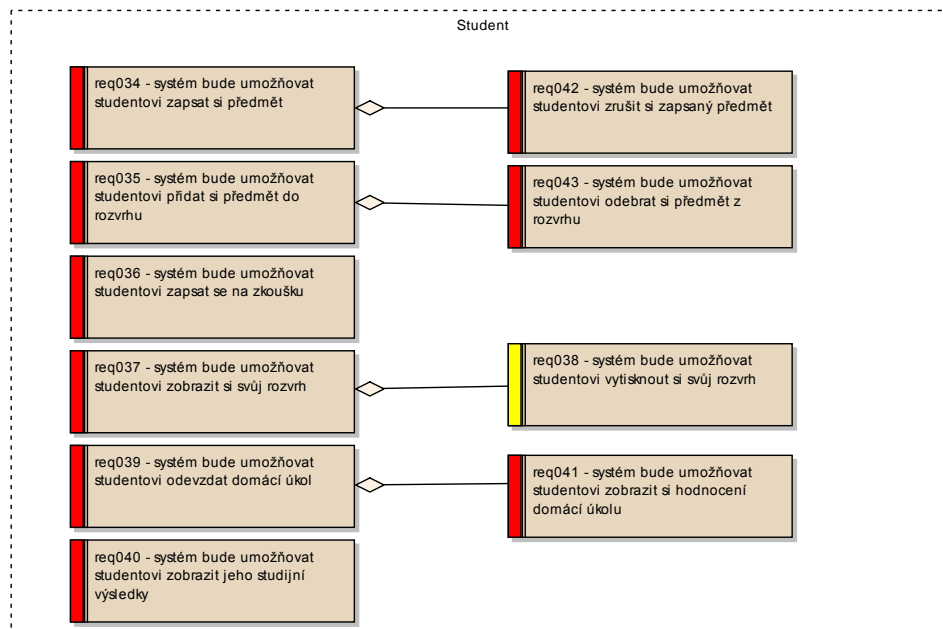
Figure: Požadavky

Funkční požadavky

custom Funkční požadavky



custom Funkční požadavky



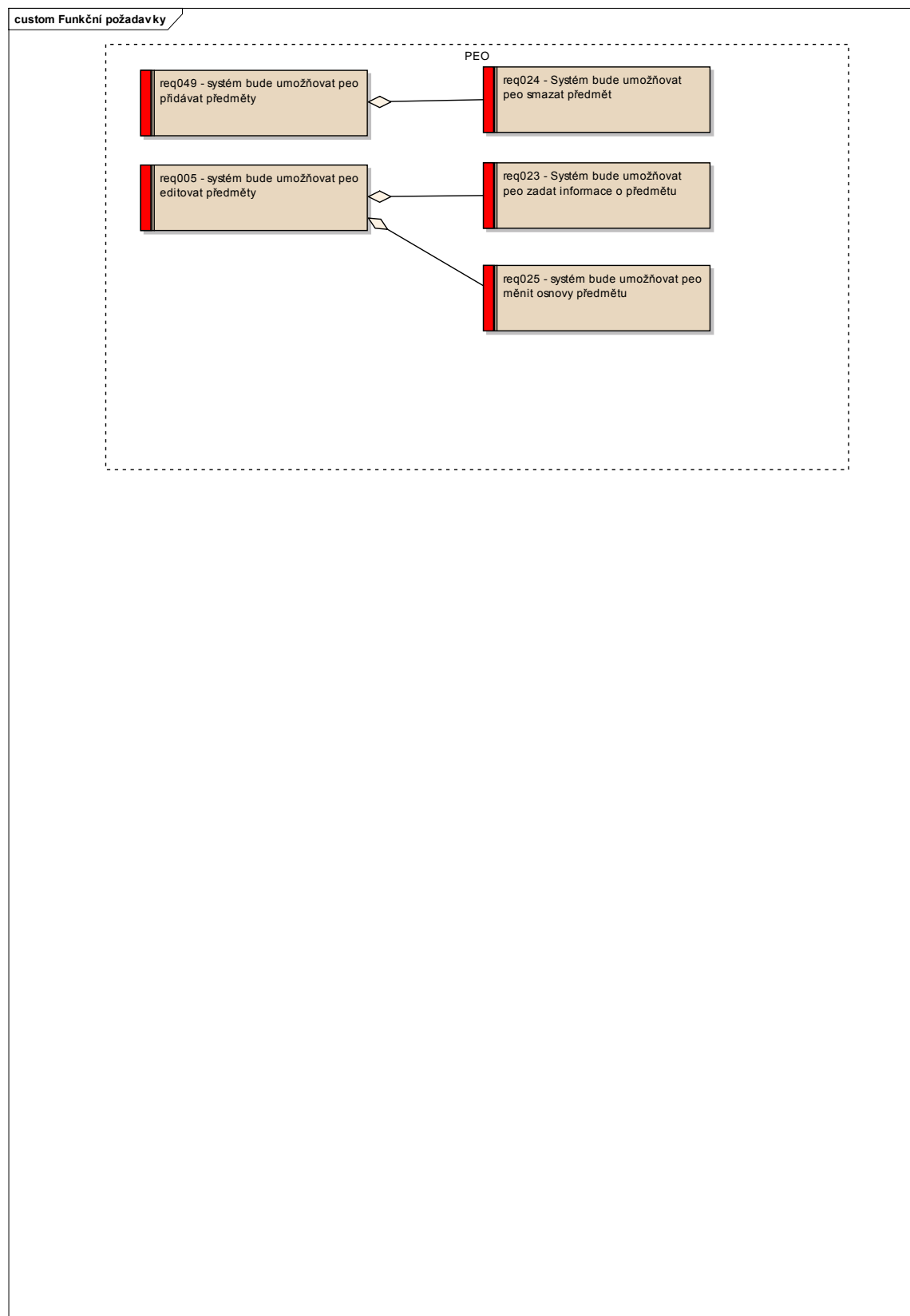


Figure: Funkční požadavky

req001 - systém bude umožňovat uživateli, aby se přihlásil

«Functional» *Status:* Must have *Priority:* High *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Po zadání uživatelského jména a hesla se bude moci každý uživatel, který je v systému registrován, přihlásit do systému.

req002 - systém bude umožňovat uživateli změnit si heslo

«Functional» *Status:* Must have *Priority:* High *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Přihlášený uživatel si může změnit heslo.

req003 - Systém bude učiteli umožňovat napsat e-mail svým studentům

«Functional» *Status:* Could have *Priority:* High *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Učitel může napsat e-mail pouze svým studentům

req004 - Systém bude umožňovat e-mailovou komunikaci mezi uživateli.

«Functional» *Status:* Must have *Priority:* High *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Každý uživatel může napsat e-mail jinému uživateli.

req005 - systém bude umožňovat peo editovat předměty

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Systém bude umožňovat pedagogickému oddělení editovat předměty.

req006 - systém bude umožňovat administrátorovi editovat uživatele

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Administrátor může přidávat nové uživatele, může je i libovolně mazat a může editovat uživatelské údaje.

req007 - systém bude umožňovat administrátorovi měnit heslo libovolnému uživateli

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Administrátor může změnit heslo libovolnému uživateli.

req008 - systém bude umožňovat administrátorovi změnit roli uživateli

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Administrátor může změnit roli libovolnému uživateli.

req009 - systém bude umožňovat administrátorovi nastavit

oprávnění uživateli

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Administrátor může nastavit oprávnění k určitým úkonům libovolnému uživateli.

req010 - systém bude umožňovat administrátorovi rozeslat hromadný email

«Functional» *Status:* Should have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Administrátor může rozeslat hromadný email všem uživatelům registrovaných v systému.

req011 - systém bude umožňovat administrátorovi rozeslat hromadný email pouze vybrané skupině

«Functional» *Status:* Should have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Administrátor může rozeslat hromadný email skupině, kterou si vybere. Např. všem studentům, všem učitelům, studentům studující daný předmět atd.

req012 - systém bude umožňovat administrátorovi vypsat si všechny uživatele

«Functional» *Status:* Should have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Administrátor si může nechat vypsat všechny uživatele, kteří jsou registrováni v systému

req013 - systém bude umožňovat učiteli editovat předměty

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Uživatel může editovat předměty, které učí.

req014 - systém bude umožňovat učiteli měnit osnovy předmětu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Učitel může měnit osnovy vyučovaného předmětu.

req015 - systém bude umožňovat učiteli nahrávat materiály k předmětu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Učitel může k předmětu, který učí, nahrávat studijní materiály, jako např. přednášky.

req016 - systém bude umožňovat učiteli spravovat zkoušky

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Učitel může vypisovat, mazat a editovat termíny zkoušky.

req017 - systém bude umožňovat učiteli udělovat známku z předmětu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Učitel může udělovat studentům známku z předmětu.

req018 - systém bude umožňovat učiteli udělit zápočet

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Učitel může studentovi udělit zápočet z předmětu.

req019 - systém bude umožňovat učiteli zadat domácí úkol

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Učitel může zadávat domácí úkoly z daného předmětu.

req020 - systém bude umožňovat učiteli prohlížet odevzdané úkoly

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Učitel může prohlížet studenty odevzdané úkoly.

req021 - systém bude umožňovat učiteli zaznamenávat bodové hodnocení

«Functional» Status: Should have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Učitel může zaznamenávat bodové hodnocení u studenta.

req022 - systém bude umožňovat učitel poslat domácí úkol k přepracování

«Functional» Status: Should have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Učitel může posílat domácí úkoly, které byly splněné chybně, k přepracování.

req023 - Systém bude umožňovat peo zadat informace o předmětu

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Systém bude umožňovat pedagogickému oddělení zadávat informace o předmětu.

req024 - Systém bude umožňovat peo smazat předmět

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Systém bude umožňovat pedagogickému oddělení smazat předmět.

req025 - systém bude umožňovat peo měnit osnovy předmětu

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Systém bude umožňovat pedagogickému oddělení měnit osnovy předmětu.

req026 - systém bude umožňovat automatické rozesílání emailu s upozorněním na neschválený zápis

«Functional» Status: Should have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Systém bude automaticky rozesílat email, ve kterém bude upozorňovat studenta na

neschválený zápis.

req027 - systém bude umožňovat kontrolu prerekvizit zapisovaných předmětů

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém bude při zápisu předmětu kontrolovat, zda má student splněny potřebné prerekvizity.

req028 - systém bude umožňovat vypsání seznamu všech předmětů k zápisu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém vypisuje seznam všech předmětů, které si student může zapsat.

req029 - systém bude umožňovat kontrolu dostatku kreditů pro zápis do dalšího semestru

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém kontroluje, zda má student dostatečný počet kreditů pro zápis do dalšího semestru.

req030 - systém bude umožňovat peo přidat studenta do systému

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Pedagogické oddělení může přidat nového studenta do systému.

req031 - systém bude umožňovat peo smazat libovolného studenta

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Pedagogické oddělení může smazat libovolného studenta.

req032 - systém bude umožňovat peo měnit osobní údaje u studenta

«Functional» *Status:* Should have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Pedagogické oddělení může provádět změny údajů u daného studenta.

req033 - systém bude upozorňovat studenta na kolize předmětů

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém upozorňuje studenta pokud se mu některé předměty kryjí.

req034 - systém bude umožňovat studentovi zapsat si předmět

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Student si může zapsat předmět ze seznamu předmětů.

req035 - systém bude umožňovat studentovi přidat si předmět do rozvrhu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Student si může přidat zapsaný předmět do svého rozvrhu.

req036 - systém bude umožňovat studentovi zapsat se na zkoušku

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Student se může zapsat na vypsaný termín zkoušky z daného předmětu.

req037 - systém bude umožňovat studentovi zobrazit si svůj rozvrh

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Student si může zobrazit svůj osobní rozvrh.

req038 - systém bude umožňovat studentovi vytisknout si svůj rozvrh

«Functional» *Status:* Could have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Student si může vytisknout svůj rozvrh.

req039 - systém bude umožňovat studentovi odevzdat domácí úkol

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Student může do systému nahrát vypracovaný domácí úkol.

req040 - systém bude umožňovat studentovi zobrazit jeho studijní výsledky

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém zobrazí známky, které v minulosti student dostal.

req041 - systém bude umožňovat studentovi zobrazit si hodnocení domácí úkolu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém umožňuje studentovi, aby si zobrazil hodnocení odevzdaného domácího úkolu, které dostal od daného učitele.
Popřípadě chyby, které v úkolu měl.

req042 - systém bude umožňovat studentovi zrušit si zapsaný předmět

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0
Systém bude umožňovat studentovi zrušení zapsaného předmětu.

req043 - systém bude umožňovat studentovi odebrat si předmět z rozvrhu

«Functional» *Status:* Must have *Priority:* Medium *Difficulty:* Medium
Phase: 1.0 *Version:* 1.0

Systém umožňuje studentovi odebrání si předmětu ze studentova rozvrhu.

req049 - systém bude umožňovat přidávat předměty

«Functional» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Systém bude umožňovat pedagogickému oddělení přidávat nové předměty do databáze.

Obecné/kvalitativní požadavky

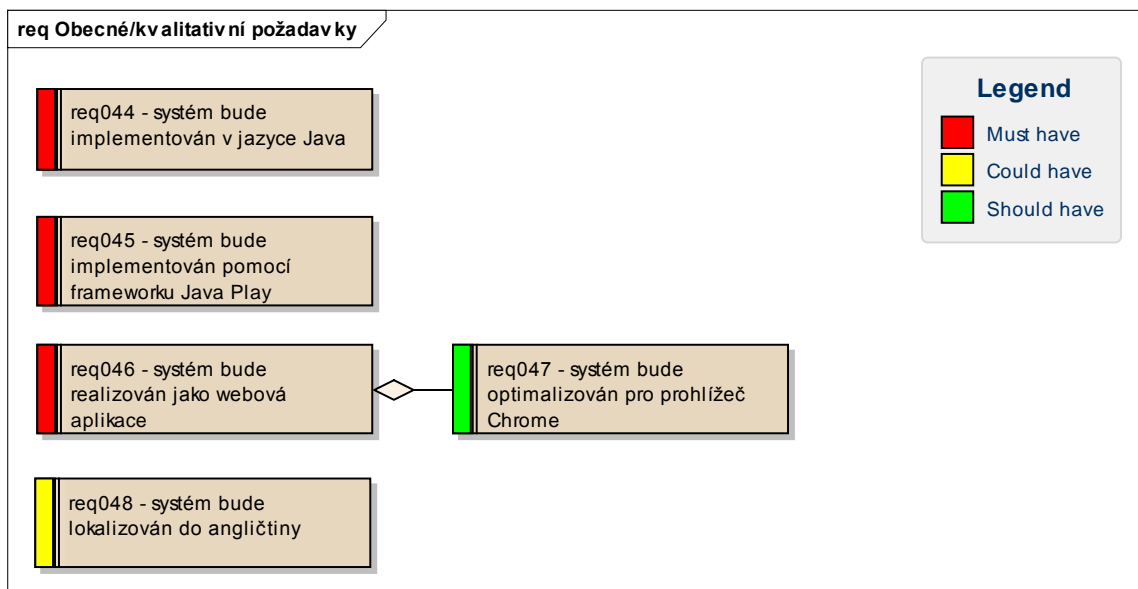


Figure: Obecné/kvalitativní požadavky

req044 - systém bude implementován v jazyce Java

«Performance» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Aplikace bude napsána v jazyce Java. Bude tedy multiplatformní.

req045 - systém bude implementován pomocí frameworku Java Play

«Performance» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Aplikace bude vytvořena pod frameworkem Java Play.

req046 - systém bude realizován jako webová aplikace

«Performance» Status: Must have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Systém bude realizován jako webová aplikace přístupná pomocí internetového prohlížeče.

req047 - systém bude optimalizován pro prohlížeč Chrome

«Performance» Status: Should have Priority: Medium Difficulty: Medium
Phase: 1.0 Version: 1.0

Aplikace bude testována a laděna v prohlížeči Google Chrome.

req048 - systém bude lokalizován do angličtiny

«Functional» Status: Could have
Phase: 1.0

Priority: Medium
Version: 1.0

Difficulty: Medium

Use Cases

1 Správa účtů

1.1 Use-cases

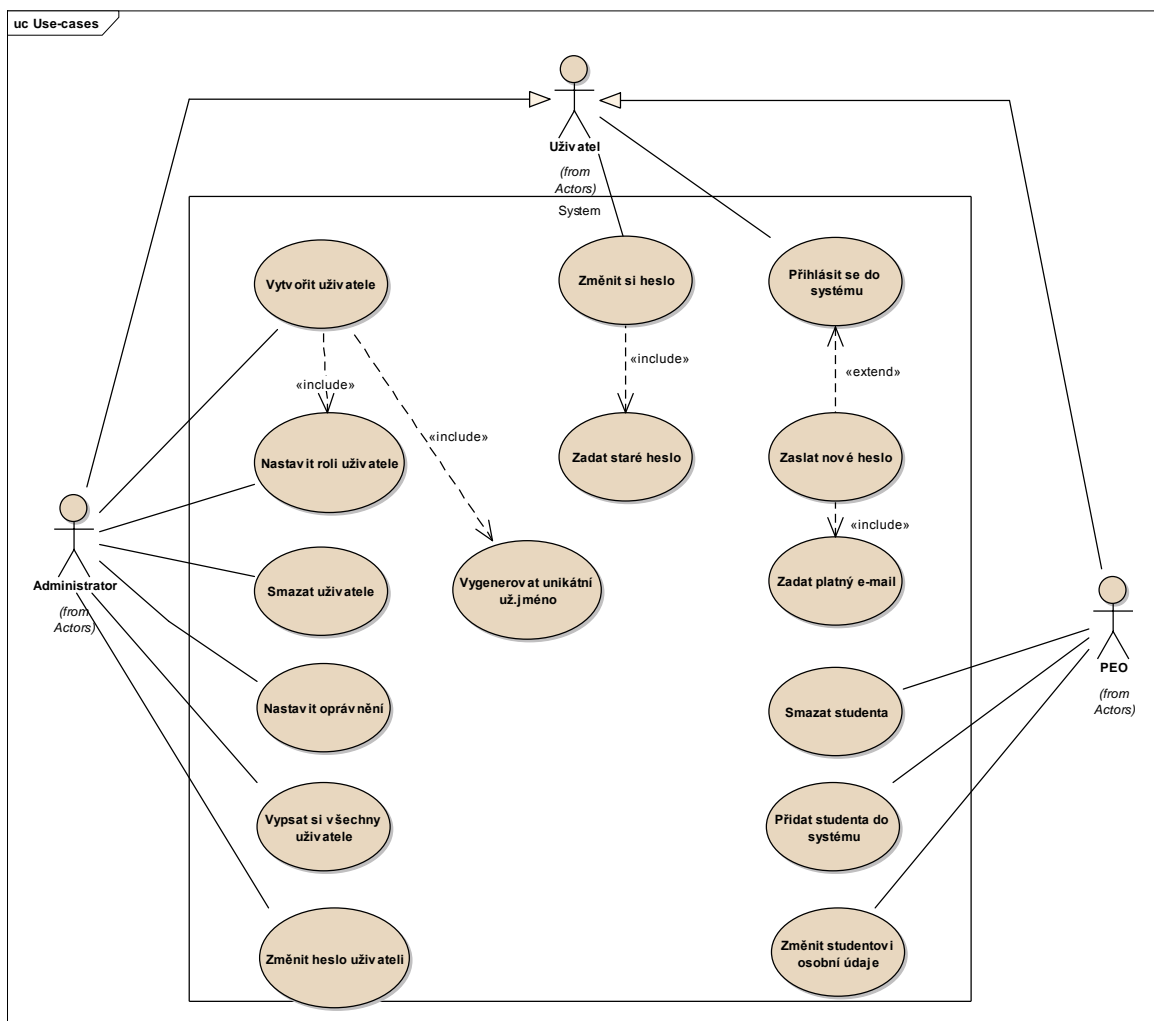


Figure 18: Use-cases

1.1.1 Přihlásit se do systému

Basic Path

1. Uživatel požaduje přihlášení do systému.
2. Systém zobrazí formulář pro zadání přihlašovacího jména a hesla.
3. Uživatel zadá přihlašovací jméno a heslo.
4. Systém ověří údaje a přihlásí osobu do systému.

Alternate Scenarios:

Uživatel nezná heslo (krok 3).

1. Uživatel požaduje zaslání uživatelského jména a hesla.
2. Systém zobrazí formulář pro vyplnění e-mailu.
3. Uživatel zadá svůj platný e-mail.
4. Systém odešle přihlašovací data na zadaný e-mail.

Exceptions:

Systém nepřihlásí osobu do systému (krok 5).

1. Systém vypíše chybové hlášení.

1.1.2 Smazat studenta

Basic Path

1. PEO požaduje smazání studenta ze systému.
2. Systém se zeptá na jméno nebo kód studenta.
3. PEO zadá kód studenta.
4. Systém vyhledá studenta.
5. PEO přikáže smazat vybraného studenta ze systému.
6. Systém smaže studenta ze systému.

Alternate Scenarios:

PEO nezná kód studenta. (krok 3)

1. PEO požaduje nalezení studenta podle jména.
2. Systém zobrazí formulář pro nalezení studenta podle jména.
3. PEO napíše jméno studenta.
4. Systém zobrazí kódy studentů daného jména.

1.1.3 Vytvořit uživatele

Basic Path

1. Administrátor požaduje vytvoření nového uživatele.
2. Systém zobrazí formulář pro vytvoření nového uživatele.
3. Administrátor zadá jméno a příjmení nového uživatele.
4. INCLUDE (Vygenerovat už.jméno)
5. Systém vygeneruje unikátní uživatelské jméno.
6. INCLUDE (Nastavit roli uživatele)
7. Administrátor nastaví uživateli role.
8. Systém přidá uživatele do databáze.

1.1.4 Změnit si heslo

Basic Path

1. Uživatel požaduje změnu současného hesla.
2. Systém zobrazí formulář pro vyplnění současného a nového hesla.
3. Uživatel vyplní své současné a nové heslo.
4. Systém ověří správnost starého hesla a nastaví nové.

Exception:

Uživatel zadá špatně současné heslo (krok 3).

1. Systém zobrazí varovnou hlášku.

2 Komunikace

2.1 Účastníci

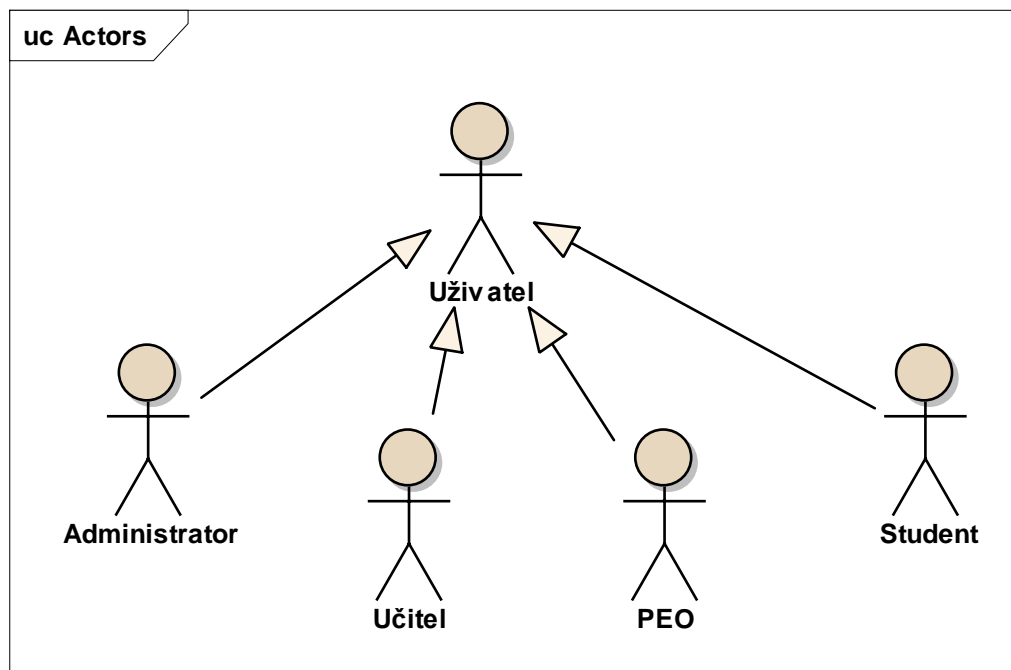


Figure 19: Actors

2.2 Use-cases

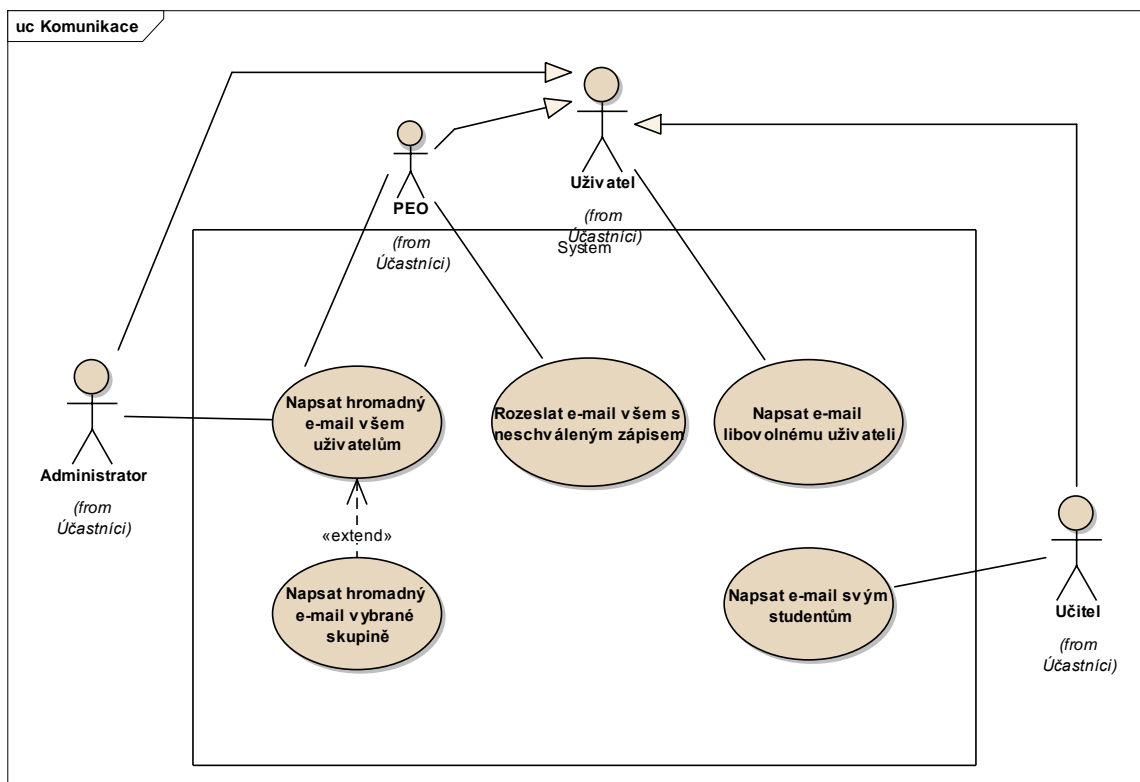


Figure 20: Komunikace

2.2.1 Napsat e-mail libovolnému uživateli

Basic Path

1. Uživatel chce napsat e-mail.
2. Systém zobrazí formulář pro psaní e-mailu.
3. Uživatel zadá e-mailovou adresu adresáta, předmět e-mailu a napíše text e-mailu.
4. Systém průběžně ukládá text e-mailu.
5. Uživatel potvrdí odeslání e-mailu.
6. Systém odešle e-mail.

Exception:

- Uživatel zadá chybný e-mail (krok 3).
1. Systém zobrazí varovnou hlášku.
- Délka předmětu je delší než 50 znaků (krok 3).
1. Systém zobrazí varovnou hlášku.

2.2.2 Napsat hromadný e-mail všem uživatelům

Basic Path

1. Uživatel chce napsat všem uživatelům hromadný e-mail.
2. Systém zobrazí formulář e-mailu.
3. Uživatel vybere za příjemce všechny osoby, vyplní e-mail a dá ho odeslat.
4. Systém odešle e-mail a vypíše, že byl e-mail úspěšně odeslán.

Exception:

System neodešle e-mail. (krok 4)
1. System vypíše chybovou hlášku.

3 Předměty

3.1 Use-cases

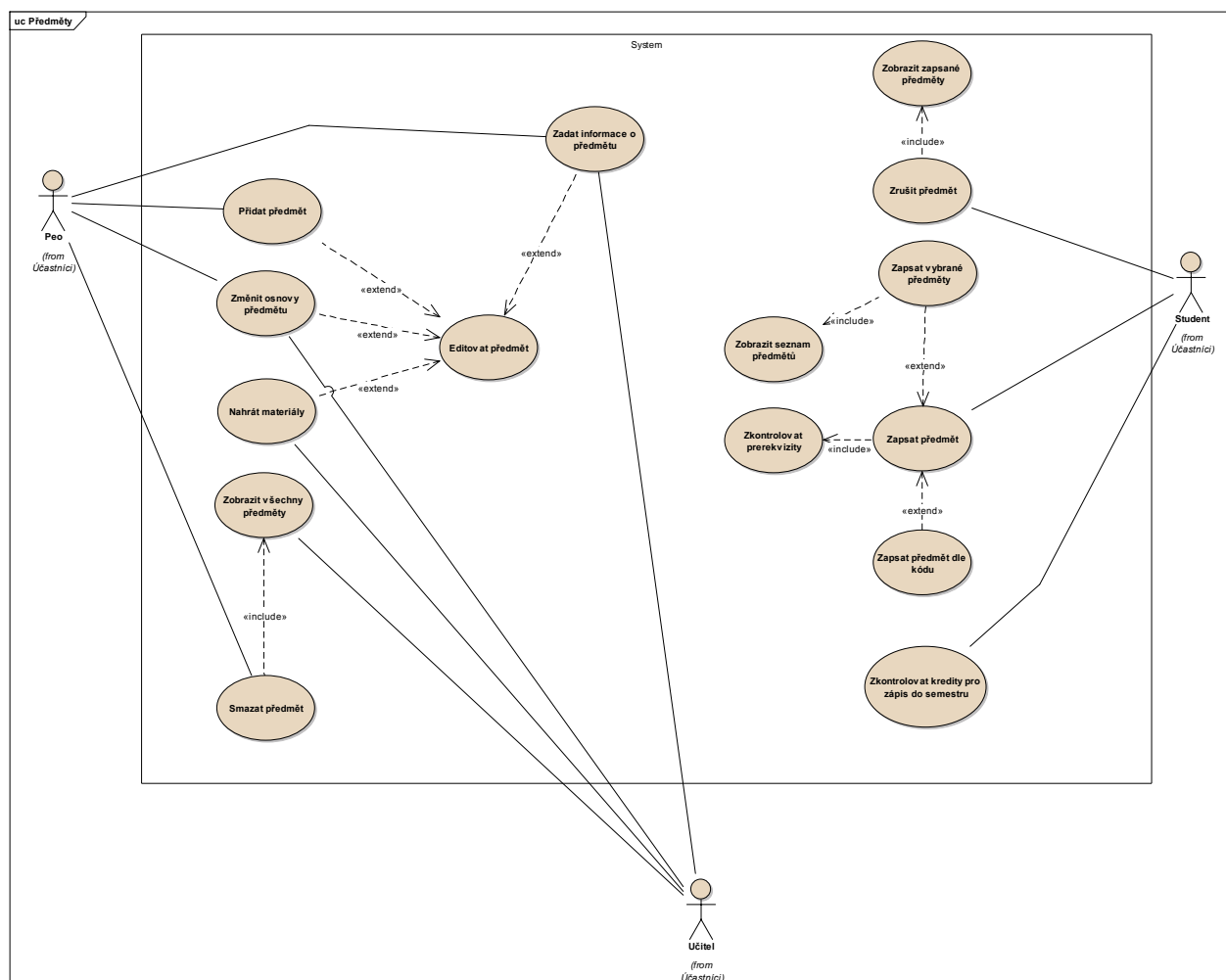


Figure 22: Předměty

3.1.1 Přidat předmět

Basic Path

1. PEO požaduje přidání předmětu.

2. Systém zobrazí formulář pro přidání nového předmětu.
3. PEO vyplní formulář předmětu a odešle k přidání.
4. Systém zkontroluje, zda lze předmět přidat . Poté předmět zapíše a zobrazí ho studentovi mezi zapsané předměty.

Alternate Scenarios:

Systém předmět nepřidá. (krok 5)

1. Systém informuje PEO, proč nešlo předmět přidat.
2. PEO upraví formulář předmětu a odešle k přidání.

3.1.2 Zapsat předmět

Basic Path

1. Student požaduje zapsání předmětu.
2. Systém zobrazí seznam všech předmětů.
3. Student vybere předmět.
4. Systém zkontroluje, zda lze předmět zapsat. Poté předmět zapíše a zobrazí ho studentovi mezi zapsané předměty.

Exceptions:

Systém předmět nezapíše. (krok 5)

1. Systém informuje studenta, proč nebyl předmět zapsán (např. nebyly splněny prerekvizity) .

3.1.3 Zrušit předmět

Basic Path

1. Student požaduje zrušení předmětu.
2. Systém zobrazí seznam všech předmětů, které má student zapsané.
3. Student vybere předmět.
4. Systém zkontroluje, zda lze předmět zrušit. Poté předmět zruší, informuje studenta a přestane předmět studentovi zobrazovat mezi zapsanými předměty.

Exceptions:

Systém nezobrazí žádné předměty. (krok 2)

1. Systém informuje uživatele, že nemá zapsané žádné předměty.

Systém předmět nezruší. (krok 5)

1. Systém informuje studenta, proč se nepodařilo předmět zrušit.

4 Rozvrh

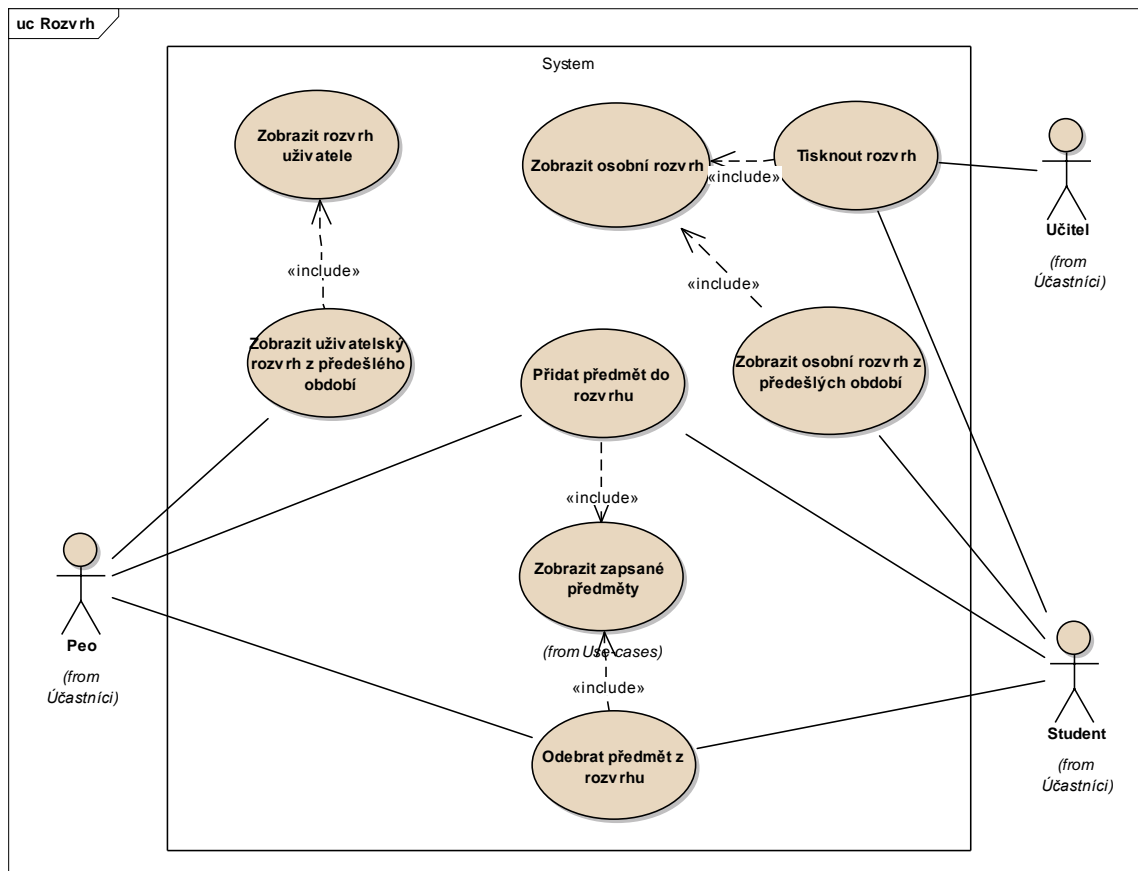


Figure 24: Rozvrh

4.1.1 Odebrat předmět z rozvrhu

Basic Path

Uživatelé: Student, PEO

1. Uživatel požaduje odebrání předmětu z rozvrhu.
2. Systém zobrazí uživateli jeho rozvrh.
3. Uživatel vybere předmět z rozvrhu.
4. Systém zobrazí paralelky předmětu.
5. Uživatel se odebere z paralelky.
6. Systém odebere uživateli předmět z rozvrhu.

4.1.2 Přidat předmět do rozvrhu

Basic Path

Uživatel: Student, PEO

1. Uživatel požaduje přidání předmětu do rozvrhu.
2. Systém zobrazí seznam všech předmětů, které si může uživatel do rozvrhu přidat.
3. Uživatel vybere předmět.
4. Systém zobrazí paralelky předmětu.

5. Uživatel si zvolí paralelku.
6. Systém zkontroluje, zda lze předmět přidat uživateli pro vybranou paralelku do rozvrhu. Poté předmět přidá a zobrazí ho uživateli v rozvrhu.

Alternate Scenarios:

Systém nepřidal uživateli předmět do rozvrhu. (krok 7)

Systém nepřidal uživateli předmět do rozvrhu.

1. Systém informuje uživatele, proč nešlo předmět pro vybranou paralelku přidat.
2. Uživatel si vybere validní paralelku.

4.1.3 Tisknout rozvrh

Basic Path

Uživatelé: Student, Učitel

1. Uživatel si chce vytisknout osobní rozvrh.
2. Systém zobrazí uživateli jeho rozvrh.
3. Uživatel přikáže rozvrh vytisknout.
4. Systém rozvrh vytiskne.

4.1.4 Zobrazit osobní rozvrh z předešlých období

Basic Path

1. Student si chce zobrazit osobní rozvrh z předešlého období.
2. Systém zobrazí studentovi jeho současný osobní rozvrh.
3. Student si zvolí z nabídky období, z kterého chce svůj rozvrh zobrazit.
4. Systém zobrazí rozvrh studentovi jeho rozvrh ze zvoleného období.

5 Úkoly

5.1 Use-cases

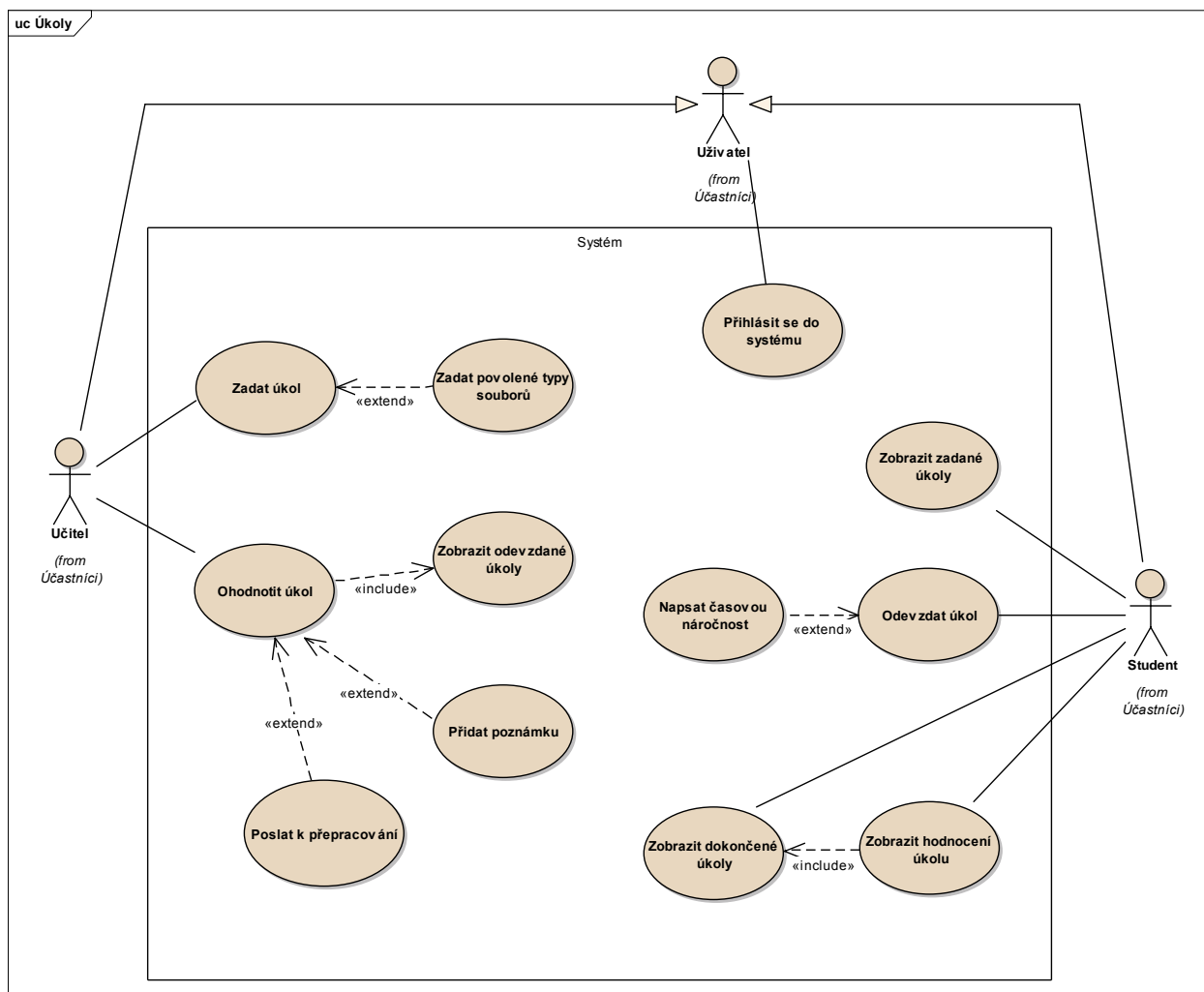


Figure 26: Úkoly

5.1.1 Odevzdat úkol

Basic Path

1. Student chce odevzdat úkol.
2. Systém studentovi zobrazí všechny předměty, pro které může odevzdat úkol.
3. Student vybere požadovaný předmět.
4. Systém zobrazí aktuálně aktivní úkoly.
5. Student vybere konkrétní úkol.
6. Systém zobrazí formulář pro výběr souboru k odevzdání.
7. Student vybere soubor a potvrdí odeslání.
8. Systém zkontroluje, zda je soubor platný.
9. Soubor je platný.
10. Systém uloží úkol a označí ho jako odevzdáný.

Exceptions:

Soubor nebyl platný. (krok 9)

1. Systém vypíše studentovi chybovou hlášku .

5.1.2 Ohodnotit úkol

Basic Path

1. Učitel chce ohodnotit úkol.
2. Systém učiteli zobrazí všechny předměty, pro které může ohodnotit úkol.
3. Učitel vybere požadovaný předmět.
4. Systém zobrazí odevzdané úkoly z předmětu.
5. Učitel vybere konkrétní úkol k ohodnocení.
6. Systém zobrazí vybraný úkol a formulář pro ohodnocení.
7. Učitel zadá hodnocení.
8. Systém uloží hodnocení.

Exceptions:

Systém nezobrazil žádné předměty. (krok 2)

1. Systém zobrazí upozornění, že učitel nemá žádný dostupný předmět.

5.1.3 Zadat úkol

Basic Path

1. Učitel chce zadat nový úkol.
2. Systém učiteli zobrazí všechny předměty, pro které může zadat úkol.
3. Učitel vybere požadovaný předmět.
4. Systém zobrazí formulář pro zadání úkolu.
5. Učitel vyplní formulář.
6. Systém uloží zadaný úkol.

Exception:

Systém neuloží úkol.(krok 6)

1. Systém zobrazí chybové hlášení.

5.1.4 Zobrazit hodnocení úkolu

Basic Path

1. Student chce zobrazit hodnocení svého úkolu.
2. Systém studentovi zobrazí všechny předměty, ke kterým mohl odevzdat úkoly.
3. Student vybere požadovaný předmět.
4. Systém zobrazí dokončené úkoly.
5. Student vybere konkrétní úkol.
6. Systém zobrazí hodnocení úkolu.

Exceptions:

Systém nezobrazí žádné úkoly. (krok 4)

1. Systém informuje studenta, že nemá žádné odevzdané úkoly .

6 Zkoušky

6.1 Use-cases

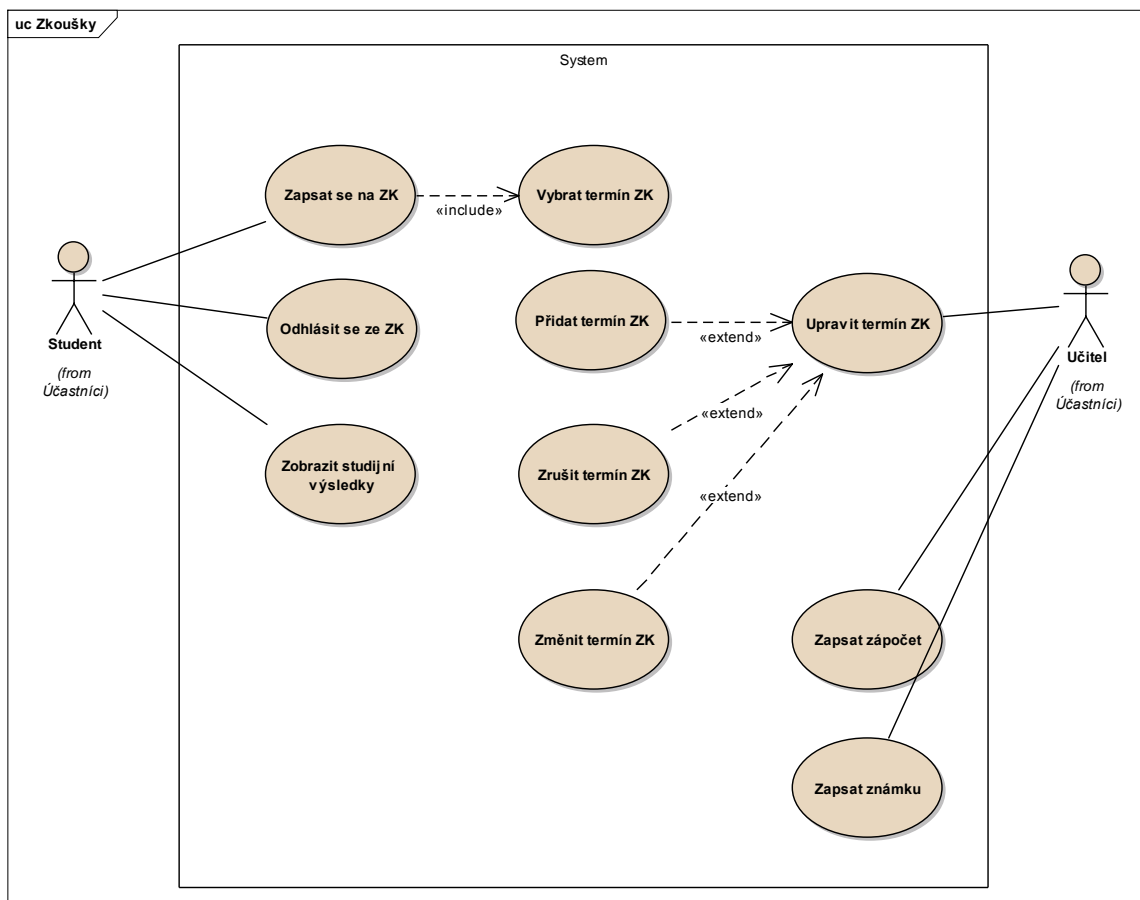


Figure 28: Zkoušky

6.1.1 Odhlásit se ze ZK

Basic Path

1. Student požaduje odhlášení z termínu zkoušky.
2. Systém studentovi zobrazí všechny jeho předměty semestru.
3. Student vybere požadovaný předmět.
4. Systém zobrazí vypsane termíny zkoušek předmětu.
5. Student přikáže požadovaný termín odhlásit.
6. Systém ověří, zda může studentovi termín odhlásit. Poté termín odhlásí a zobrazí ho jako nepřihlášený.

.

Exceptions:

Systém nezobrazil žádné předměty. (krok 2)

1. Systém informuje studenta, že nemá zapsané žádné předměty.

Systém nezobrazil žádné termíny zkoušek. (krok 4)

1. Systém informuje studenta, že termíny zkoušek nebyly vypsány .

Termín se nepodařilo odhlásit (krok 7).

1. Systém informuje studenta, že se termín nepodařilo odhlásit a z jakého důvodu.

Systém nezobrazil žádné předměty.

6.1.2 Přidat termín ZK

Basic Path

1. Učitel požaduje přidání termínu zkoušky.
2. Systém učiteli zobrazí všechny předměty.
3. Učitel vybere předmět.
4. Učitel vytvoří termín a požádá o jeho uložení do systému.
5. Systém odsouhlasí termín jako validní. Systém zkontroluje, jestli lze termín přidat. Systém termín přidá a zobrazí uložený termín učiteli a studentům.

Alternativní scénáře:

Systém neodsouhlasí termín jako validní (krok 6)

Exceptions:

Systém termín nepřidá. (krok 8)

Systém neodsouhlasí termín jako validní.

1. Systém požádá učitele o validní zadání.
 2. Učitel vytvoří validní termín a požádá o jeho uložení do systému.
 3. Systém zkontroluje, že byl termín zadán validně.
- Systém termín přidá.
1. Systém informuje učitele, že termín nelze přidat.

6.1.3 Zapsat se na ZK

Basic Path

1. Student požaduje zápis na termín zkoušky z určitého předmětu.
2. Systém studentovi zobrazí všechny jeho předměty semestru.
3. Student vybere požadovaný předmět.
4. Systém zobrazí vypsane termíny zkoušek předmětu.
5. Student přikáže termín zapsat
6. Systém ověří, zda může studentovi termín zapsat. Poté termín zapíše a zobrazí ho jako zapsaný.

Exceptions:

Systém nezobrazil žádné předměty. (krok 2)

Systém nezobrazil žádné termíny zkoušek. (krok 4)

Systém termín nezapsal. (krok 7)

Systém nezobrazil žádné předměty

1. Systém informuje studenta, že nemá žádné předměty zapsané.

Systém termín nezapsal

1. Systém informuje studenta, že se termín nepodařilo zapsat a z jakého důvodu.

Systém nezobrazil žádné termíny zkoušek.

1. Systém informuje studenta, že termíny zkoušek ještě nebyly vypsány .

6.1.4 Zapsat známku

Basic Path

1. Učitel požaduje zapsat studentovi známku z předmětu.
2. Systém učiteli zobrazí všechny předměty.
3. Učitel vybere příslušný předmět.
4. Systém zobrazí studenty předmětu, kteří mají nárok na udělení známky a ještě jim nebyla udělena.
5. Učitel vybere studenta a zapíše mu známku.
6. Systém uloží známku, přidělí studentovi odpovídající kredity a zobrazí známku studentovi a učiteli.

Poté informuje studenta e-mailem.

Exceptions:

Systém nezobrazí žádné studenty. (krok 4)

Učitel může udělení známky kdykoliv přerušit.

Systém nezobrazí žádné studenty.

1. Systém informuje učitele, že už nejsou studenti, kterým by šlo zapsat známku .

6.1.5 Zapsat zápočet

Basic Path

1. Učitel chce studentovi zapsat zápočet.

2. Systém učiteli zobrazí všechny předměty.

3. Učitel vybere příslušný předmět.

4. Systém zobrazí studenty předmětu, kteří ještě nedostali zápočet.

5. Učitel vybere studenta.

6. Učitel označí studentovi zápočet jako udělený nebo neudělený.

7. Systém uloží stav zápočtu a zobrazí ho studentovi a učiteli. Poté informuje studenta e-mailem.

Exceptions:

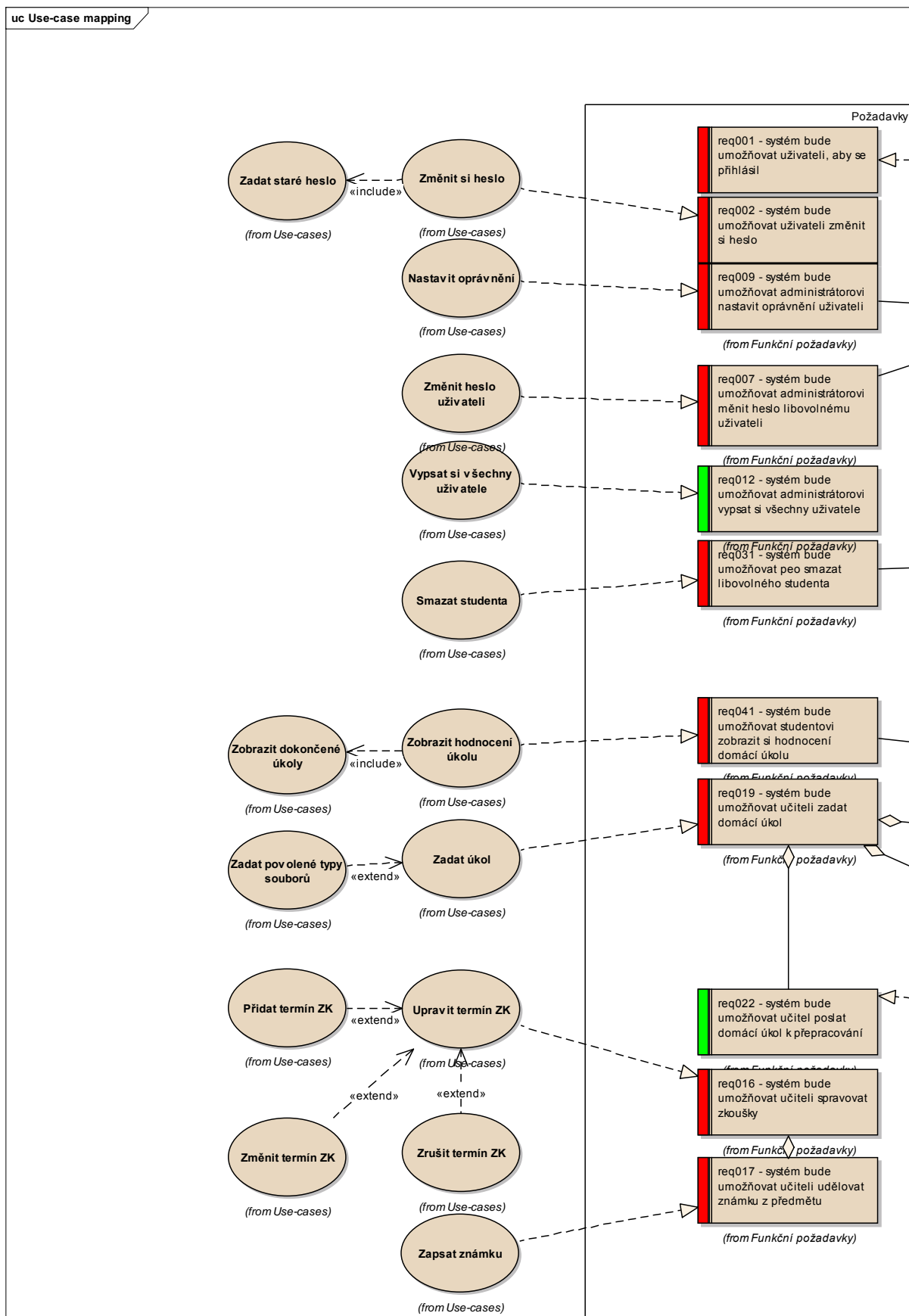
Systém nezobrazí žádné studenty. (krok 4)

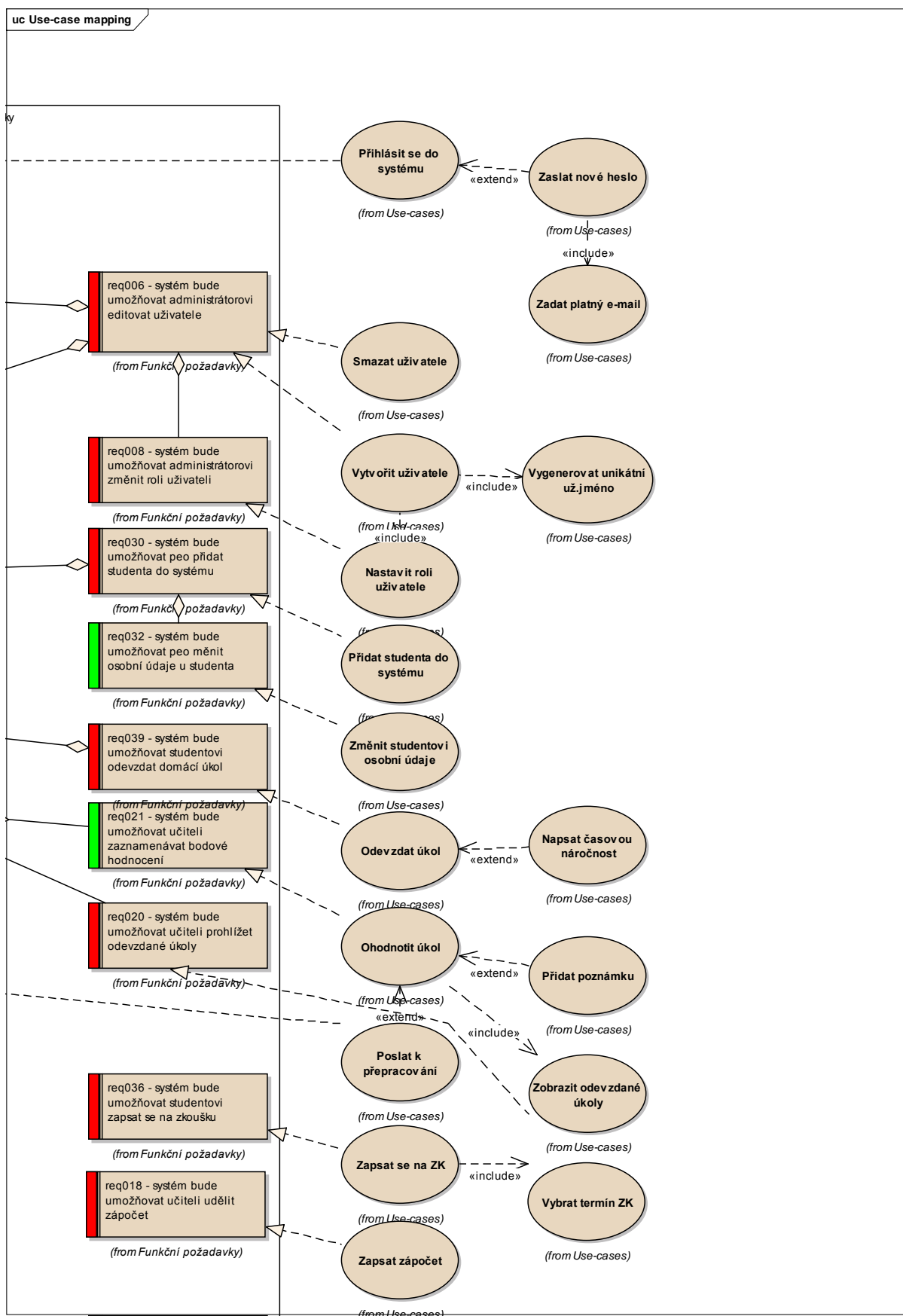
1. Systém informuje učitele, že všichni studenti předmětu už dostali zápočet.

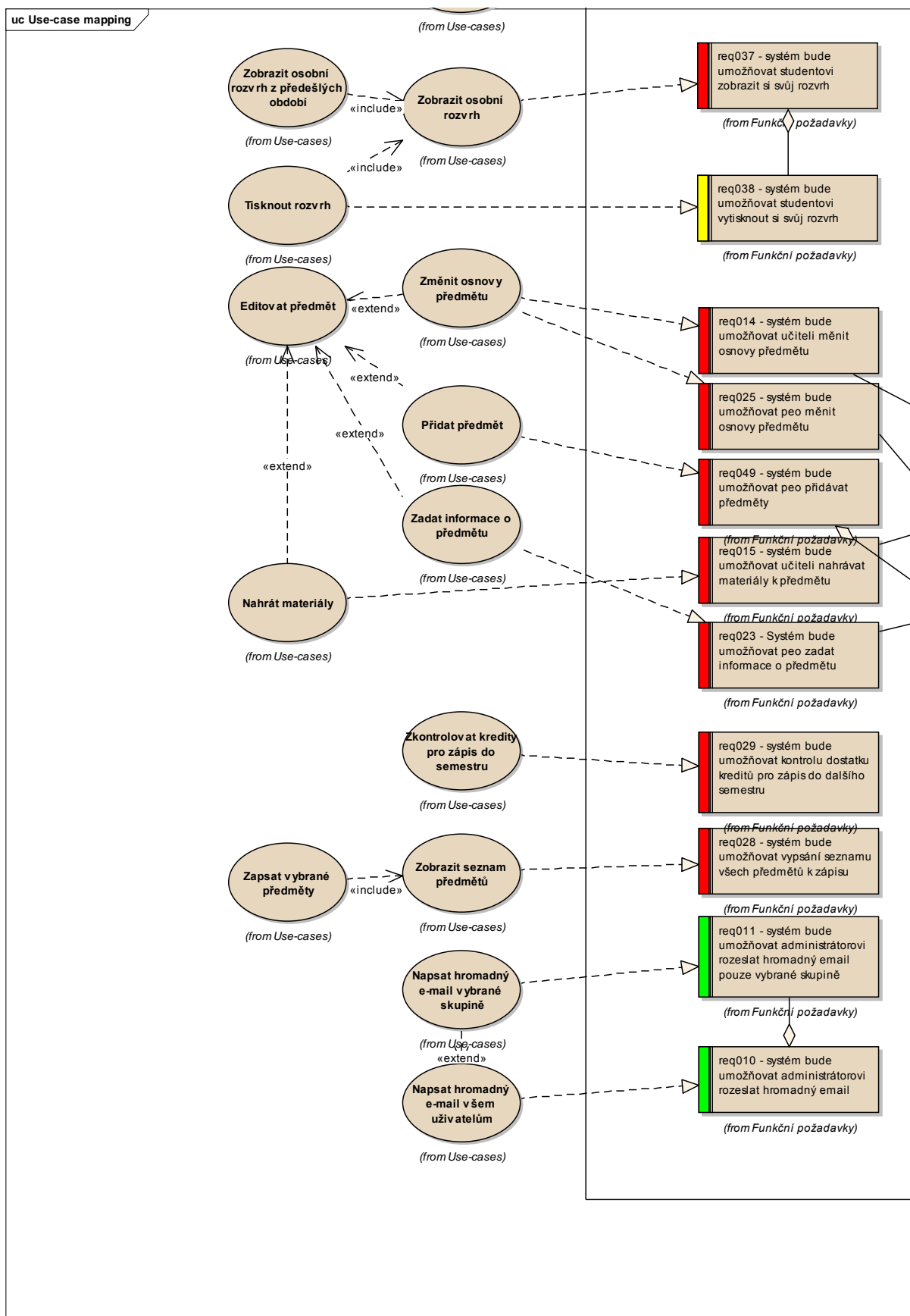
Use-case Mapping

Use-case mapping - (*Use Case diagram*)

Created By: Tom on 27.10.2013, *Last Modified:* 10.11.2013







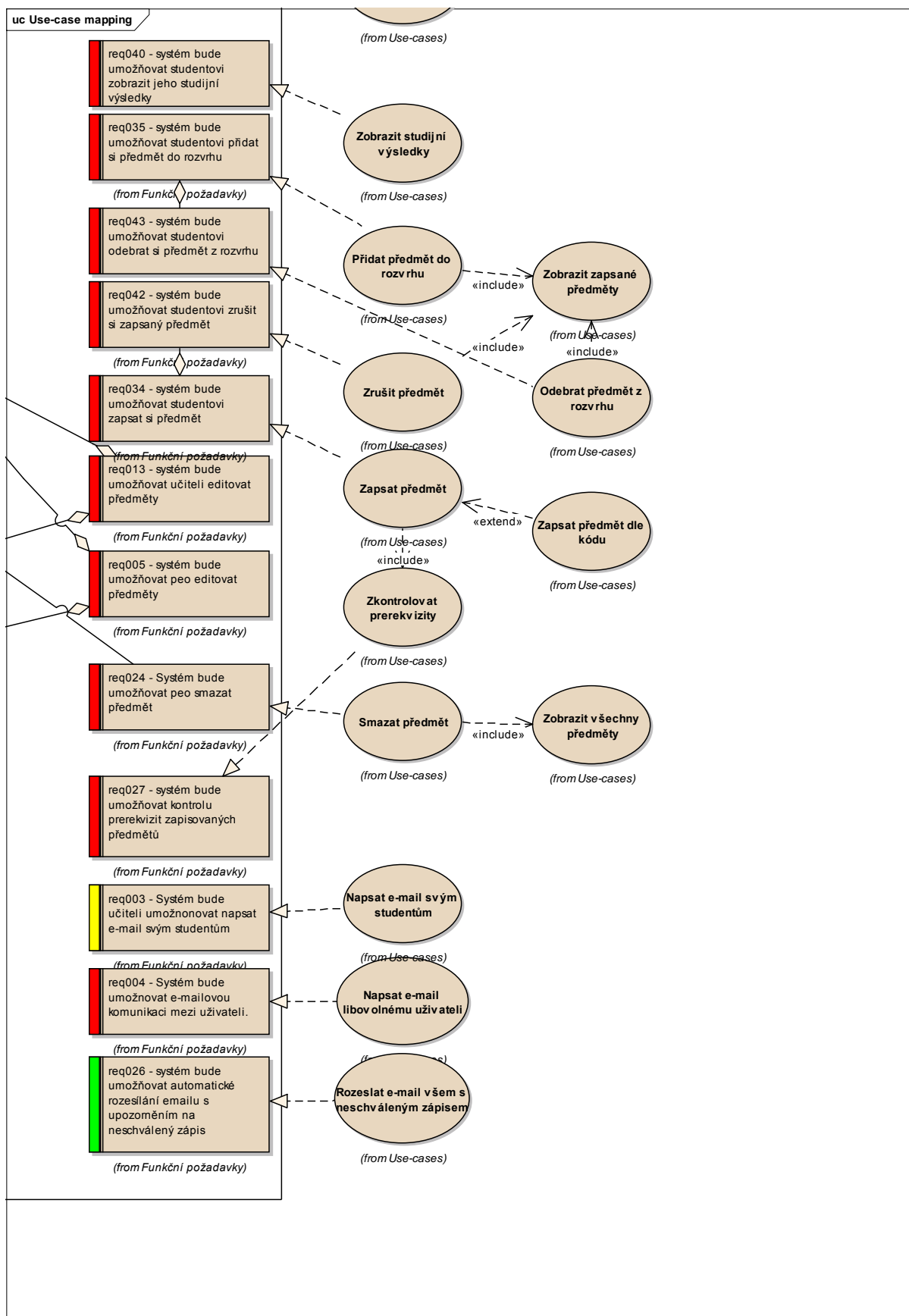


Figure: 29

Analytický Doménový Model

Analytický Doménový Model - (*Class diagram*)

Created By: Dominik on 8.11.2013, *Last Modified:* 8.11.2013

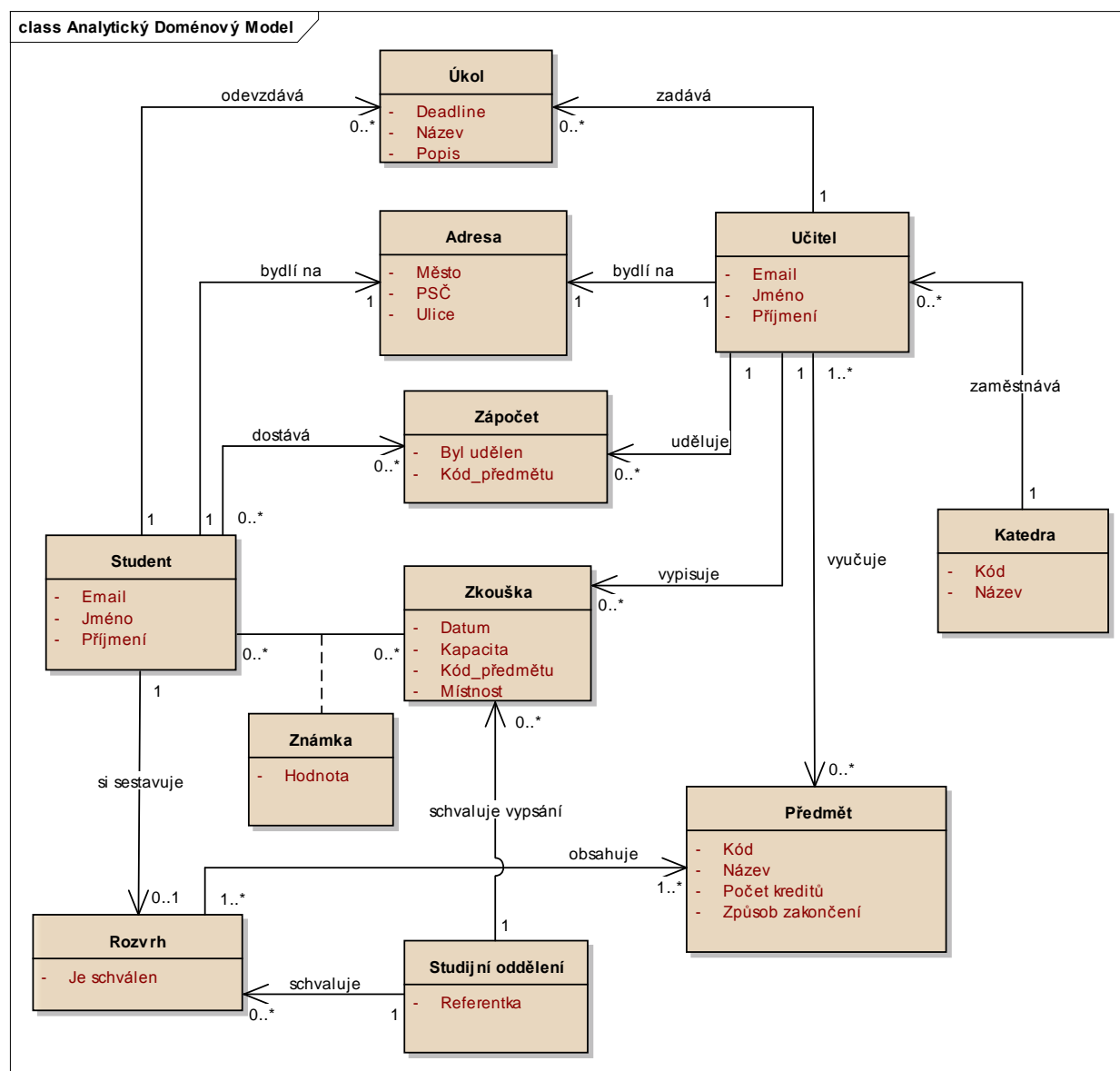


Figure: 30

Adresa

Entita Adresa uchovává informace o adresách učitele nebo studenta.

Connections

Connector	Source	Target	Notes
Association bydlí na Source -> Destination	Public Učitel	Public Adresa	Učitel má svou adresu
Association bydlí na Source -> Destination	Public Student	Public Adresa	Student má svou adresu

Attributes

Attribute	Notes	Constraints and tags
Město Private	Název města	<i>Default:</i>
PSC Private	Poštovní směrovací číslo	<i>Default:</i>
Ulice Private	Název ulice a číslo popisné	<i>Default:</i>

Katedra

Entita Katedra reprezentuje určitou katedru školy.

Connections

Connector	Source	Target	Notes
Association zaměstnává Source -> Destination	Public Katedra	Public Učitel	Katedra zaměstnává učitele

Attributes

Attribute	Notes	Constraints and tags
Kód Private	Označení katedry	<i>Default:</i>
Název Private	Název katedry	<i>Default:</i>

Předmět

Entita Předmět reprezentuje vypsany předmět.

Connections

Connector	Source	Target	Notes
Association obsahuje Source -> Destination	Public Rozvrh	Public Předmět	Rozvrh obsahuje aspoň jeden předmět.
Association vyučuje Source -> Destination	Public Učitel	Public Předmět	Učitel může vyučovat i více předmětů

Attributes

Attribute	Notes	Constraints and tags
Kód Private	Označení předmětu	<i>Default:</i>

Attribute	Notes	Constraints and tags
Název Private	Název předmětu	<i>Default:</i>
Počet kreditů Private	Počet kreditů za předmět	<i>Default:</i>
Způsob zakončení Private	Specifikace způsobu zakončení: zápočtem, zápočtem a zkouškou, nebo klasifikovaným zápočtem	<i>Default:</i>

Rozvrh

Entita Rozvrh uchovává informace o rozvrzích studentů.

Connections

Connector	Source	Target	Notes
Association obsahuje Source -> Destination	Public Rozvrh	Public Předmět	Rozvrh obsahuje aspoň jeden předmět.
Association schvaluje Source -> Destination	Public Studijní oddělení	Public Rozvrh	
Association si sestavuje Source -> Destination	Public Student	Public Rozvrh	Student si může sestavit svůj týdenní rozvrh.

Attributes

Attribute	Notes	Constraints and tags
Je schválen Private	Indikátor, zda byl rozvrh schválen	<i>Default:</i>

Attribute	Notes	Constraints and tags

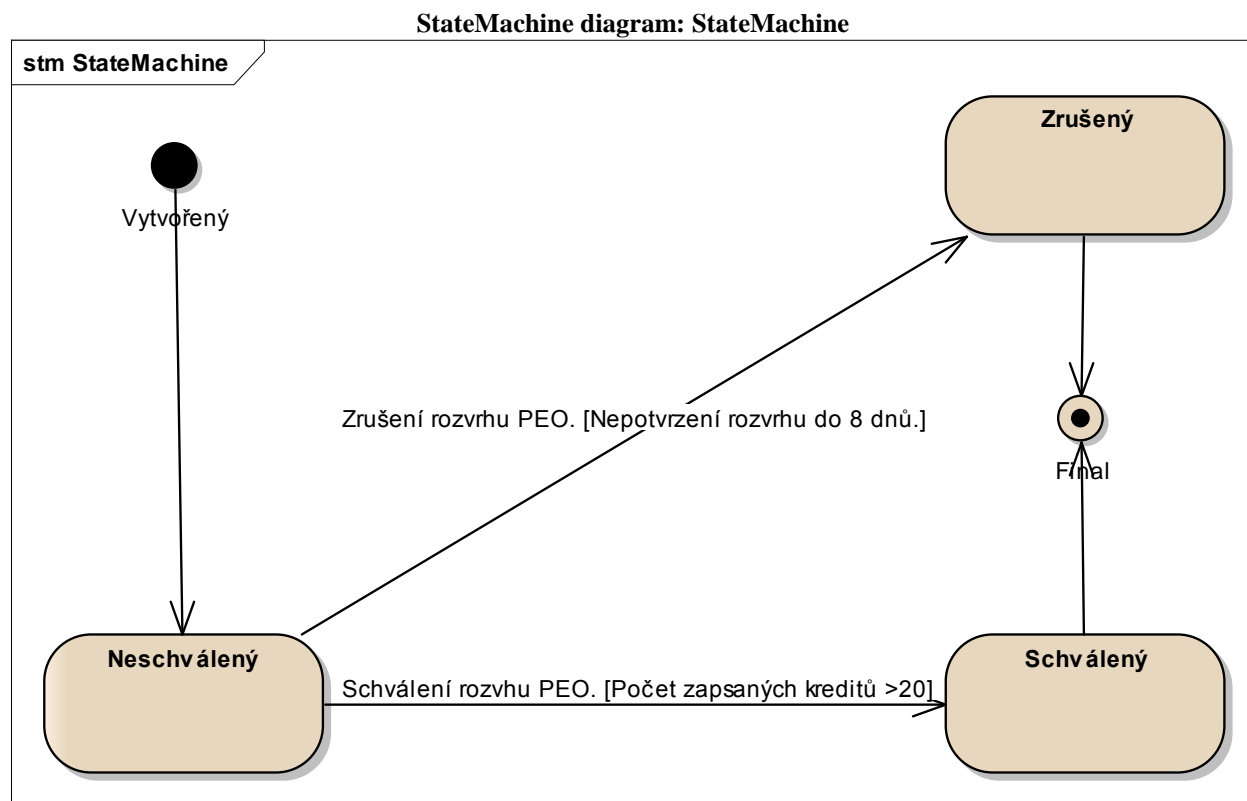


Figure: 2

Student

Entita Student reprezentuje studenta školy.

Connections

Connector	Source	Target	Notes
<u>AssociationClass</u> Unspecified	Public Student	Public Zkouška	Student může absolvovat více zkoušek. Z každé zkoušky dostává maximálně jednu konečnou známku.
<u>Association</u> bydlí na Source -> Destination	Public Student	Public Adresa	Student má svou adresu

Connector	Source	Target	Notes
Association dostává Source -> Destination	Public Student	Public Zápočet	Student dostává informaci o získání nebo nezískání zápočtů
Association odevzdává Source -> Destination	Public Student	Public Úkol	Student vypracovává a odevzdává své úkoly
Association si sestavuje Source -> Destination	Public Student	Public Rozvrh	Student si může sestavit svůj týdenní rozvrh.

Attributes

Attribute	Notes	Constraints and tags
Email Private	Emailová adresa	<i>Default:</i>
Jméno Private	Jméno studenta	<i>Default:</i>
Příjmení Private	Příjmení studenta	<i>Default:</i>

StateMachine diagram: StateMachine

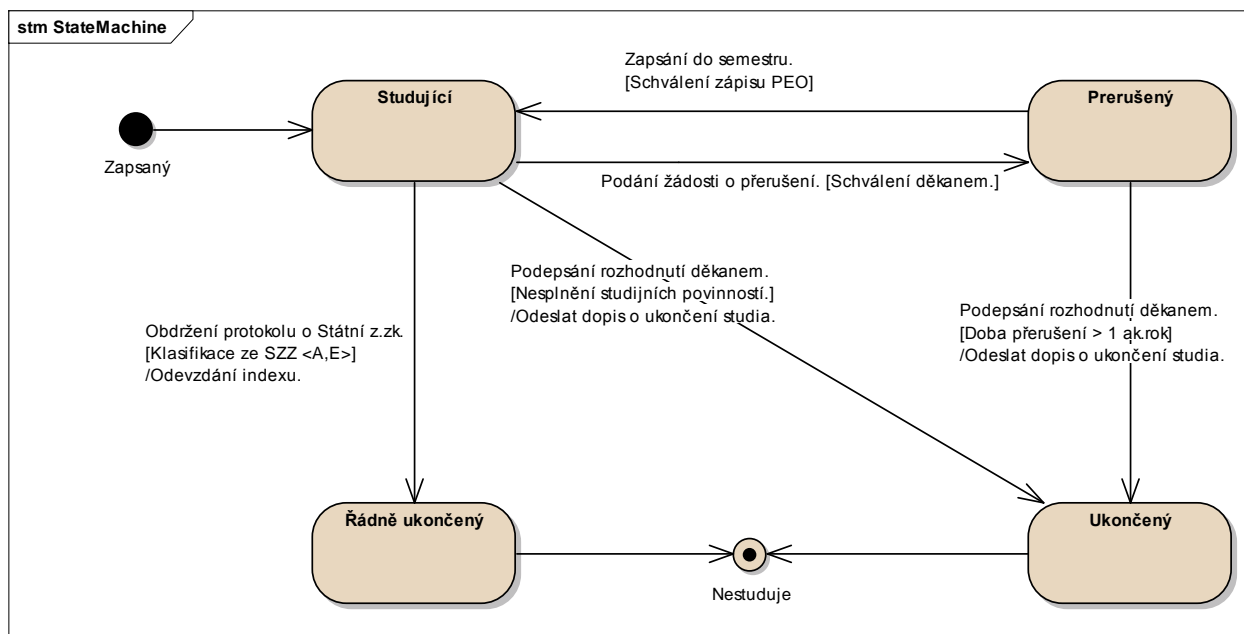


Figure: 3

Studijní oddělení

Entita Studijní oddělení uchovává informace o studijním oddělení školy.

Connections

Connector	Source	Target	Notes
Association schvaluje Source -> Destination	Public Studijní oddělení	Public Rozvrh	
Association schvaluje vypsání Source -> Destination	Public Studijní oddělení	Public Zkouška	Studijní oddělení rozhoduje o schvalování vypisovaných zkoušek

Attributes

Attribute	Notes	Constraints and tags
Referentka Private	Informace o referentce Studijního oddělení	<i>Default:</i>

Učitel

Entita učitel udržuje informace o učitelích školy.

Connections

Connector	Source	Target	Notes
Association bydlí na Source -> Destination	Public Učitel	Public Adresa	Učitel má svou adresu
Association uděluje Source -> Destination	Public Učitel	Public Zápočet	Učitel rozhoduje o udělení nebo neudělení zápočtu z předmětu studentům
Association vypisuje Source -> Destination	Public Učitel	Public Zkouška	Učitel může vypsát zkoušku
Association vyučuje Source -> Destination	Public Učitel	Public Předmět	Učitel může vyučovat i více předmětů
Association zadává Source -> Destination	Public Učitel	Public Úkol	Učitel může zadávat studentům úkoly
Association zaměstnává Source -> Destination	Public Katedra	Public Učitel	Katedra zaměstnává učitele

Attributes

Attribute	Notes	Constraints and tags
Email Private	Emailová adresa	<i>Default:</i>
Jméno Private	Jméno učitele	<i>Default:</i>

Attribute	Notes	Constraints and tags
Příjmení Private	Příjmení učitele	<i>Default:</i>

Zkouška

Entita Zkouška udržuje informace o zkouškách vypsáných z určitého předmětu.

Connections

Connector	Source	Target	Notes
<u>AssociationClass</u> Unspecified	Public Student	Public Zkouška	Student může absolvovat více zkoušek. Z každé zkoušky dostává maximálně jednu konečnou známku.
<u>Association</u> schvaluje vypsání Source -> Destination	Public Studijní oddělení	Public Zkouška	Studijní oddělení rozhoduje o schvalování vypisovaných zkoušek
<u>Association</u> vypisuje Source -> Destination	Public Učitel	Public Zkouška	Učitel může vypsát zkoušku

Attributes

Attribute	Notes	Constraints and tags
Datum Private	Datum konání zkoušky	<i>Default:</i>
Kapacita Private	Maximální počet míst na zkoušce	<i>Default:</i>

Attribute	Notes	Constraints and tags
Kód_předmětu Private	Specifikuje, kterého předmětu se zkouška týká	<i>Default:</i>
Místnost Private	Místnost konání zkoušky	<i>Default:</i>

Známka

Entita Znamka specifikuje známku, kterou byl Student ohodnocen z určitého předmětu. Ze zkoušky má student maximálně jednu konečnou známku.

Attributes

Attribute	Notes	Constraints and tags
Hodnota Private	Hodnota známky	<i>Default:</i>

Zápočet

Entita Zápočet udržuje informace o zápočtech studentů.

Connections

Connector	Source	Target	Notes
Association dostává Source -> Destination	Public Student	Public Zápočet	Student dostává informaci o získání nebo nezískání zápočtů
Association uděluje	Public	Public	Učitel rozhoduje o udělení nebo

Connector	Source	Target	Notes
Source -> Destination	Učitel	Zápočet	neudělení zápočtu z předmětu studentům

Attributes

Attribute	Notes	Constraints and tags
Byl udělen Private	Indikátor, zda byl zápočet udělen	<i>Default:</i>
Kód_předmětu Private	Specifikuje, kterého předmětu se zápočet týká	<i>Default:</i>

Úkol

Entita Úkol udržuje informace o úkolech. Úkoly jsou zadávány učiteli a vypracovávají studenty.

Connections

Connector	Source	Target	Notes
Association odevzdává Source -> Destination	Public Student	Public Úkol	Student vypracovává a odevzdává své úkoly
Association zadává Source -> Destination	Public Učitel	Public Úkol	Učitel může zadávat studentům úkoly

Attributes

Attribute	Notes	Constraints and tags
Deadline Private	Termín odevzdání	<i>Default:</i>

Attribute	Notes	Constraints and tags
Název Private	Název úkolu	<i>Default:</i>
Popis Private	Zadání úkolu, bodové hodnocení, typy souborů.	<i>Default:</i>

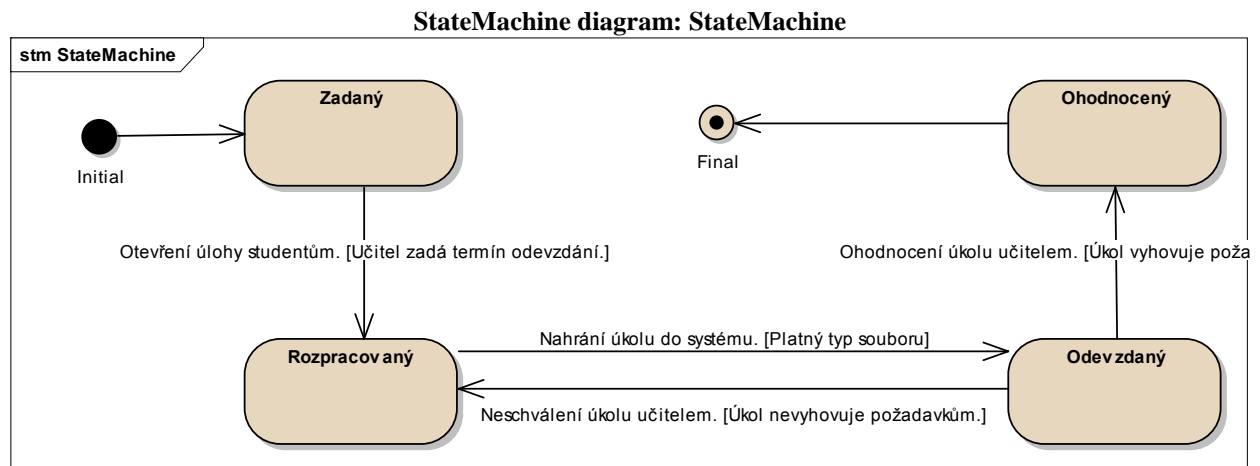


Figure: 4

Architecture Model

Architecture Model - (Package diagram)

Created By: Tomáš on 22.11.2013, *Last Modified:* 22.11.2013

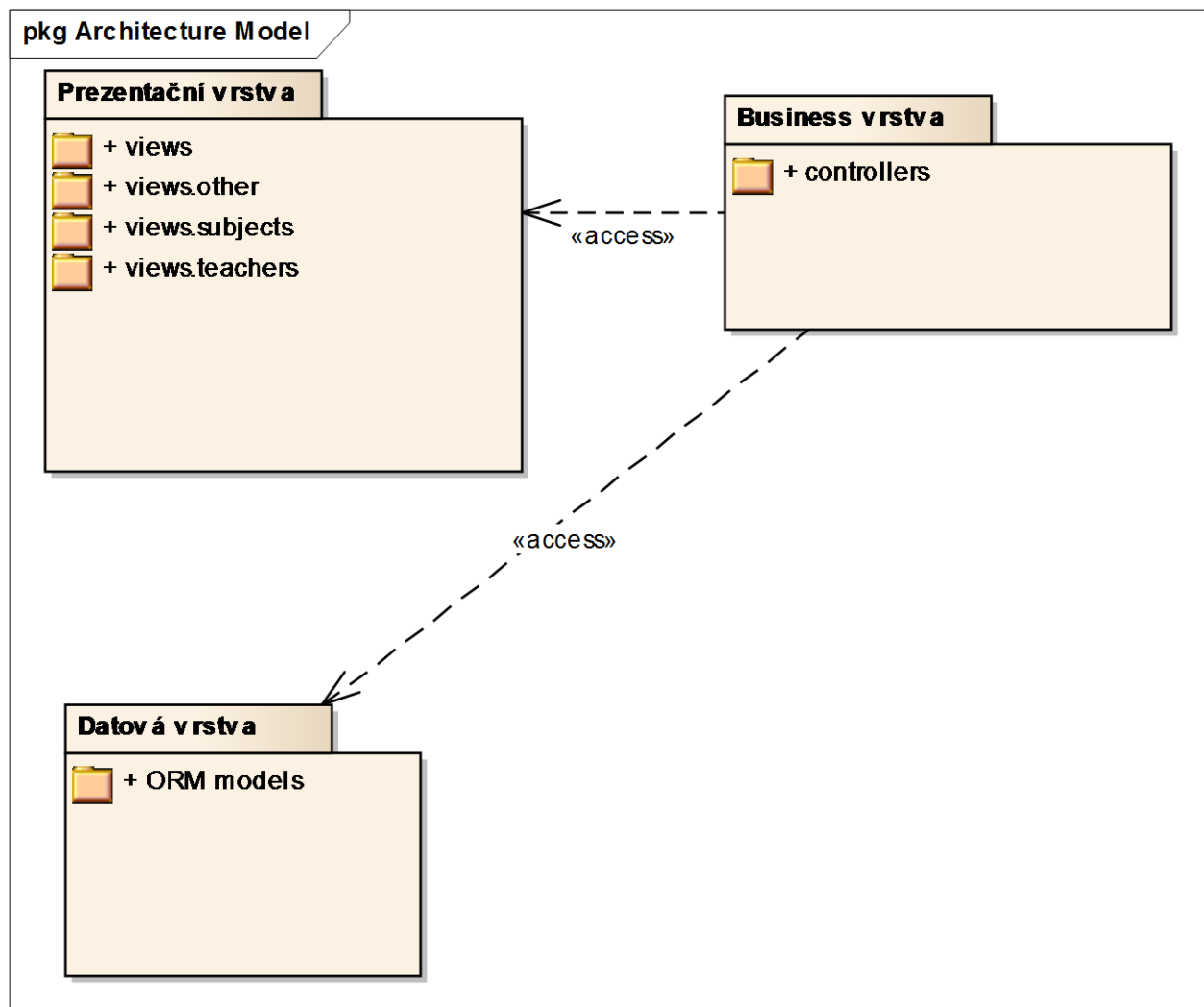


Figure: 1

Business vrstva

Business vrstva popisuje chování aplikace z pohledu obchodní logiky. Využívá ORM mapování tříd z datové vrstvy.

Business vrstva - (Package diagram)

Created By: Tomáš on 22.11.2013, *Last Modified:* 24.11.2013

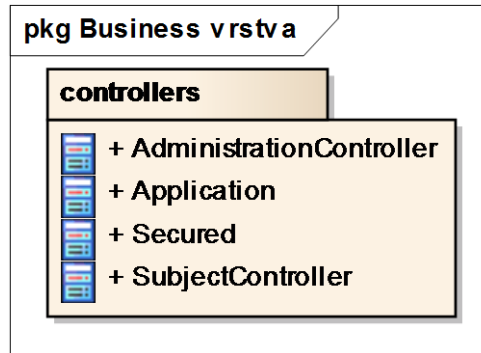


Figure: 2

controllers

Složka obsahující všechny controllers play aplikace.

controllers - (Class diagram)

Created By: Tomáš on 22.11.2013, Last Modified: 24.11.2013

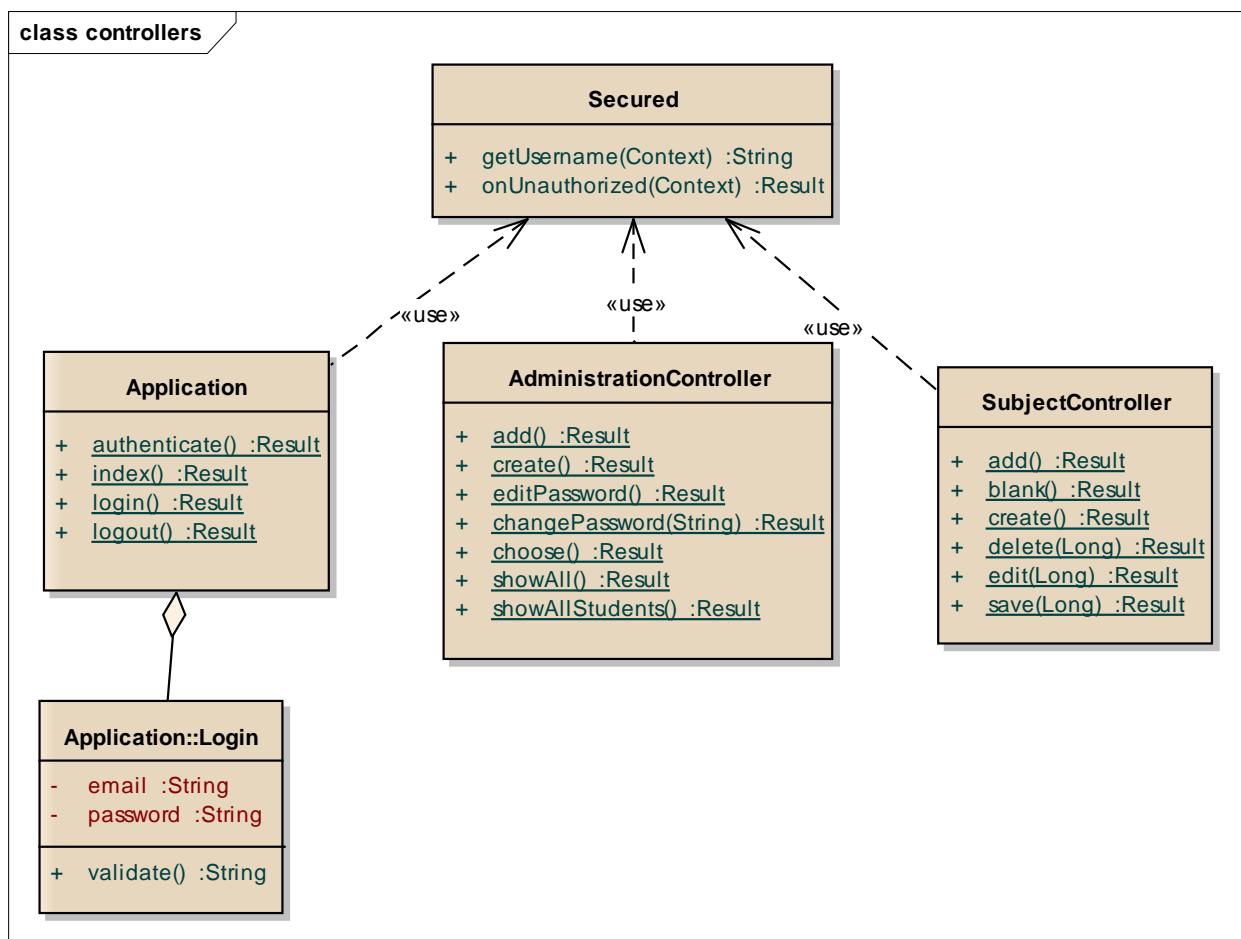


Figure: 3

Datová vrstva

Datová vrstva obsahuje ORM mapování tříd, reprezentujících jednotlivé komponenty databáze.

Datová vrstva (models) - (*Package diagram*)

Created By: Tomáš on 22.11.2013, *Last Modified:* 22.11.2013

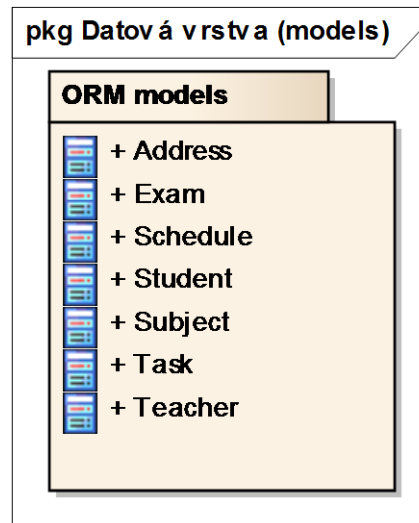


Figure: 4

ORM models

Balíček s ORM třídami.

models - (*Class diagram*)

Created By: Tomáš on 22.11.2013, *Last Modified:* 22.11.2013

class models

Address
<ul style="list-style-type: none"> - city :String - find :Finder<Long, Address> - id :Long - postcode :int - Street :String
<ul style="list-style-type: none"> + <u>create(Address) :void</u> + <u>delete(Long) :void</u> + <u>getCity() :String</u> + <u>getId() :Long</u> + <u>getPostcode() :int</u> + <u>getStreet() :String</u> + <u>setCity(String) :void</u> + <u>setId(Long) :void</u> + <u>setPostcode(int) :void</u> + <u>setStreet(String) :void</u>

Exam
<ul style="list-style-type: none"> - date :Date - id :Long - student :List<Student>
<ul style="list-style-type: none"> + <u>getDate() :Date</u> + <u>getId() :Long</u> + <u>setDate(Date) :void</u> + <u>setId(Long) :void</u>

Schedule
<ul style="list-style-type: none"> + <u>find :Finder<Long, Schedule></u> - id :Long - student :Student - subject :List<Subject>
<ul style="list-style-type: none"> + <u>getId() :Long</u> + <u>setId(Long) :void</u>

Student
<ul style="list-style-type: none"> - address :Address - email :String - exams :List<Exam> + <u>find :Finder<Long, Student></u> - id :Long - lastname :String - name :String - schedule :Schedule - subjects :List<Subject> - tasks :List<Task>
<ul style="list-style-type: none"> + <u>create(Student) :void</u> + <u>delete(Long) :void</u> + <u>getAddress() :Address</u> + <u>getEmail() :String</u> + <u>getExams() :List<Exam></u> + <u>getId() :Long</u> + <u>getLastName() :String</u> + <u>getName() :String</u> + <u>getSchedule() :Schedule</u> + <u>getSubjects() :List<Subject></u> + <u>getTasks() :List<Task></u> + <u>setAddress(Address) :void</u> + <u>setEmail(String) :void</u> + <u>setExams(List<Exam>) :void</u> + <u>setId(long) :void</u> + <u>setLastName(String) :void</u> + <u>setName(String) :void</u> + <u>setSchedule(Schedule) :void</u> + <u>setSubjects(List<Subject>) :void</u> + <u>setTasks(List<Task>) :void</u>

Subject
<ul style="list-style-type: none"> - code :String - department :String + <u>find :Finder<Long, Subject></u> - garant :String - id :Long - information :String - name :String - studentList :List<Student> - teachers :List<Teacher>
<ul style="list-style-type: none"> + <u>copySubject(Long) :void</u> + <u>create(Subject) :void</u> + <u>delete(Long) :void</u> + <u>getCode() :String</u> + <u>getDepartment() :String</u> + <u>getGarant() :String</u> + <u>getId() :Long</u> + <u>getInformation() :String</u> + <u>getName() :String</u> + <u>setCode(String) :void</u> + <u>setDepartment(String) :void</u> + <u>setGarant(String) :void</u> + <u>setId(Long) :void</u> + <u>setInformation(String) :void</u> + <u>setName(String) :void</u>

Task
<ul style="list-style-type: none"> - description :String - find :Finder<Long, Task> - id :Long - name :String - students :List<Student>
<ul style="list-style-type: none"> + <u>getId() :Long</u> + <u>getName() :String</u> + <u>setId(Long) :void</u> + <u>setName(String) :void</u>

Teacher
<ul style="list-style-type: none"> + <u>finder :Finder<Long, Teacher></u> - id :Long - name :String
<ul style="list-style-type: none"> + <u>all() :List<Teacher></u> + <u>create(Teacher) :void</u> + <u>delete(Long) :void</u> + <u>getAllNames() :List<String></u>

Figure: 5

Prezentační vrstva

Prezentační vrstva slouží k prezentování informací uživateli. K zobrazování uživateli jsme využili jsme play framework.

Prezentační vrstva (views) - (Package diagram)

Created By: Tomáš on 22.11.2013, *Last Modified:* 24.11.2013

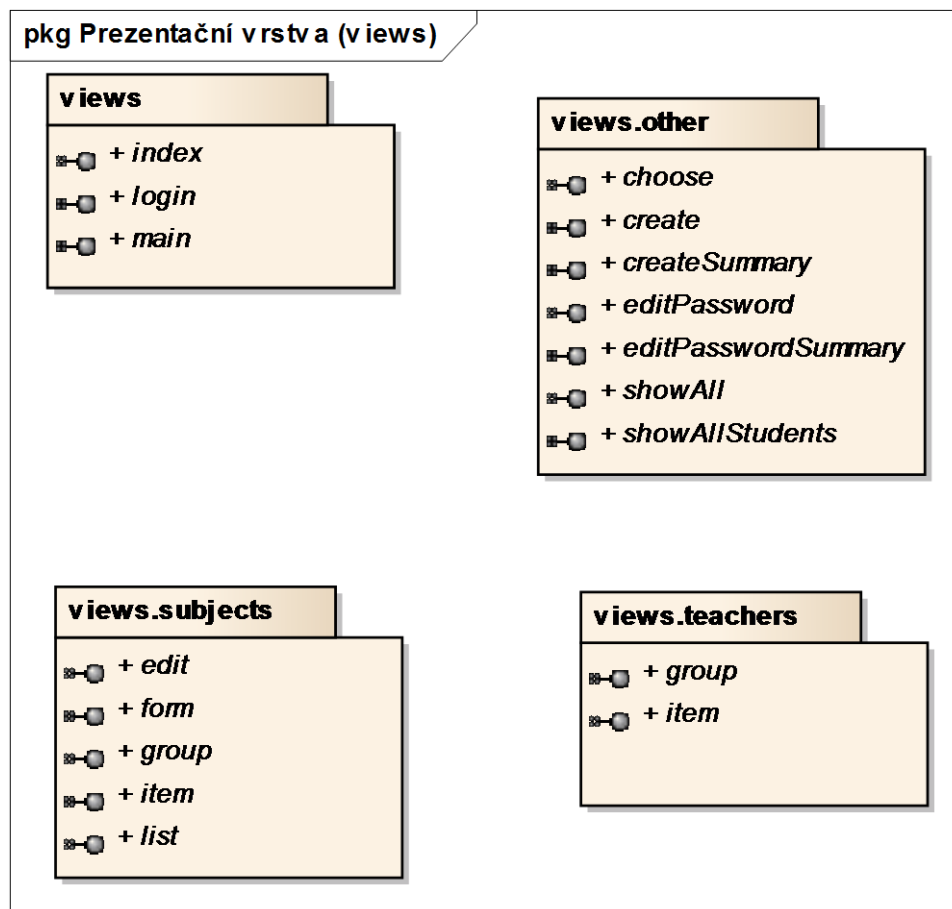


Figure: 6

views

Obsahuje základní templates aplikace.

views.other

Obsahuje templates související s přihlašováním.

views.subjects

Obsahuje templates pro use-cases s předměty.

views.teachers

Obsahuje templates pro akce učitele.

Database Model

Database Model - (*Class diagram*)

Created By: Tomáš *on* 22.11.2013, *Last Modified:* 22.11.2013

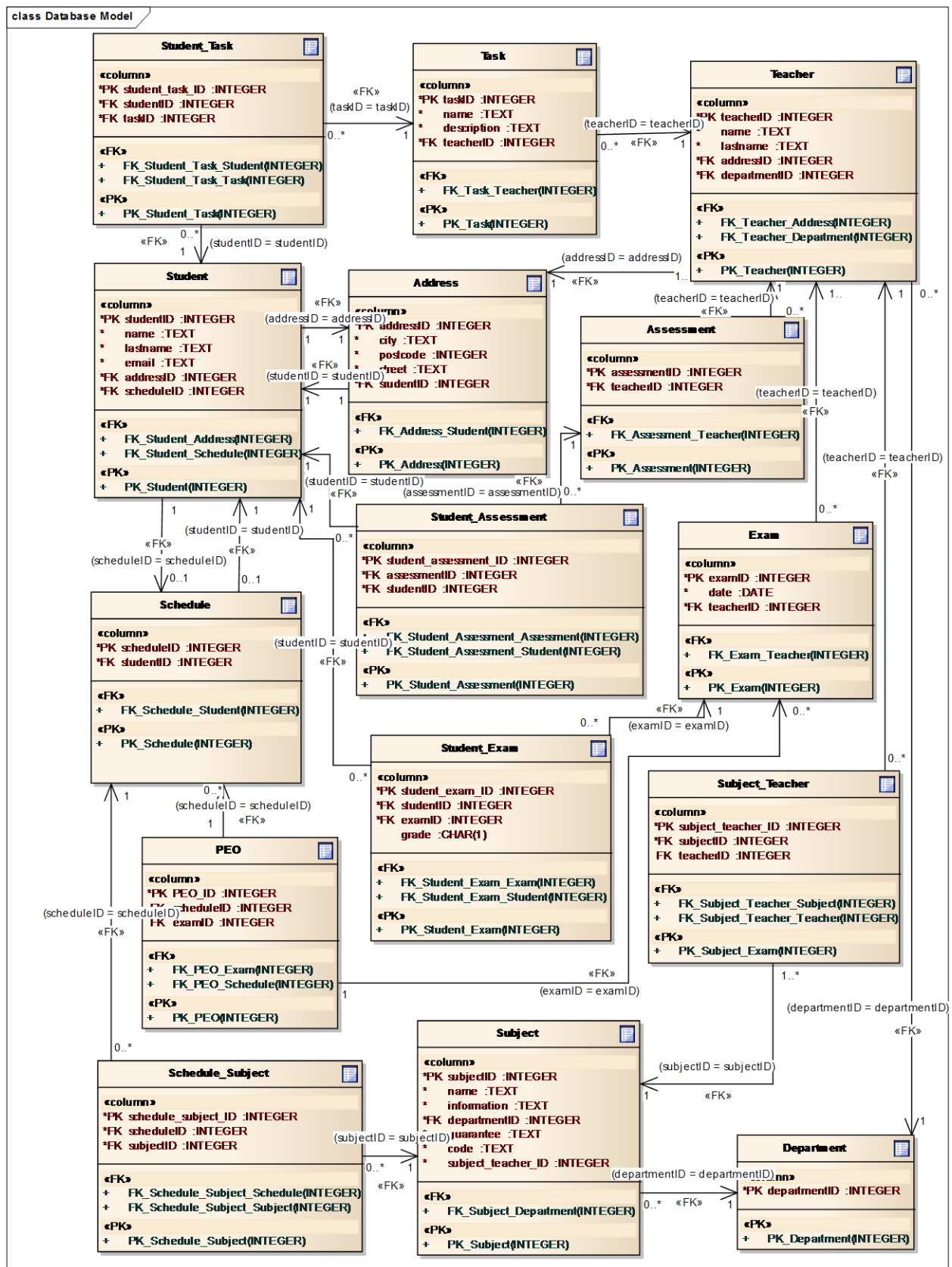


Figure: 7

Component Model

Component Model - (Component diagram)

Created By: Tom on 8.12.2013 , Last Modified: 8.12.2013

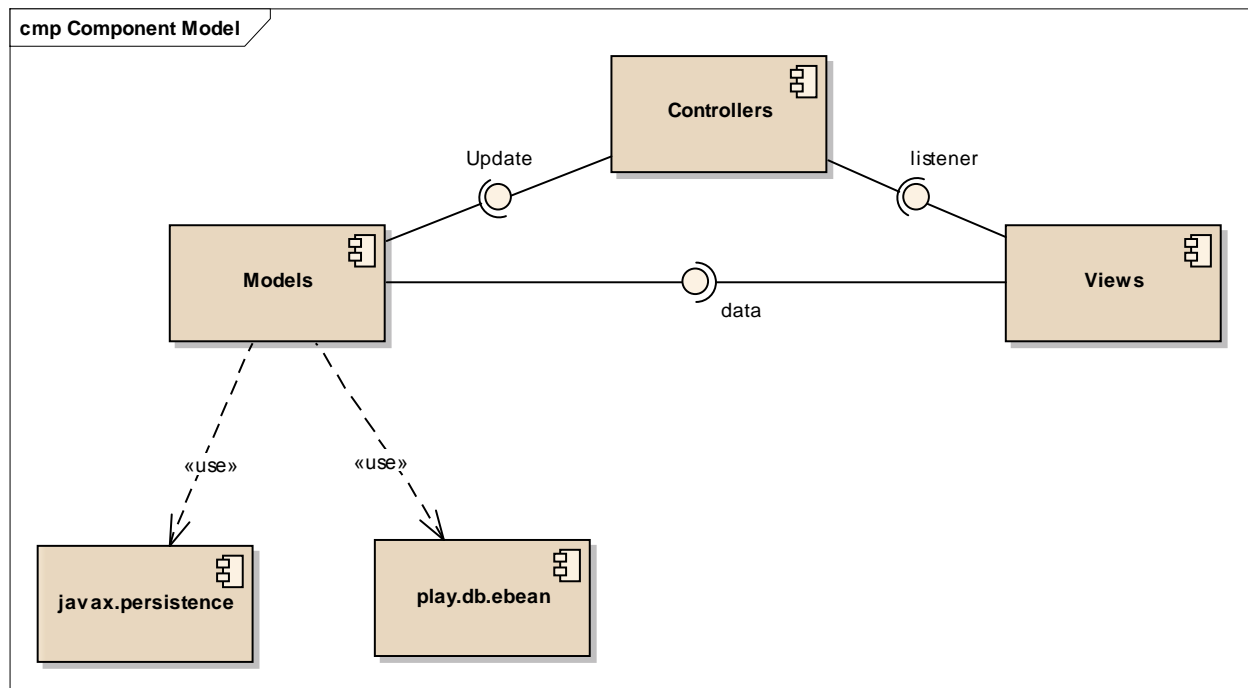


Figure: 8

Deploymenet Model

Deploymenet Model - (Deployment diagram)

Created By: Tom on 21.11.2013, Last Modified: 22.11.2013

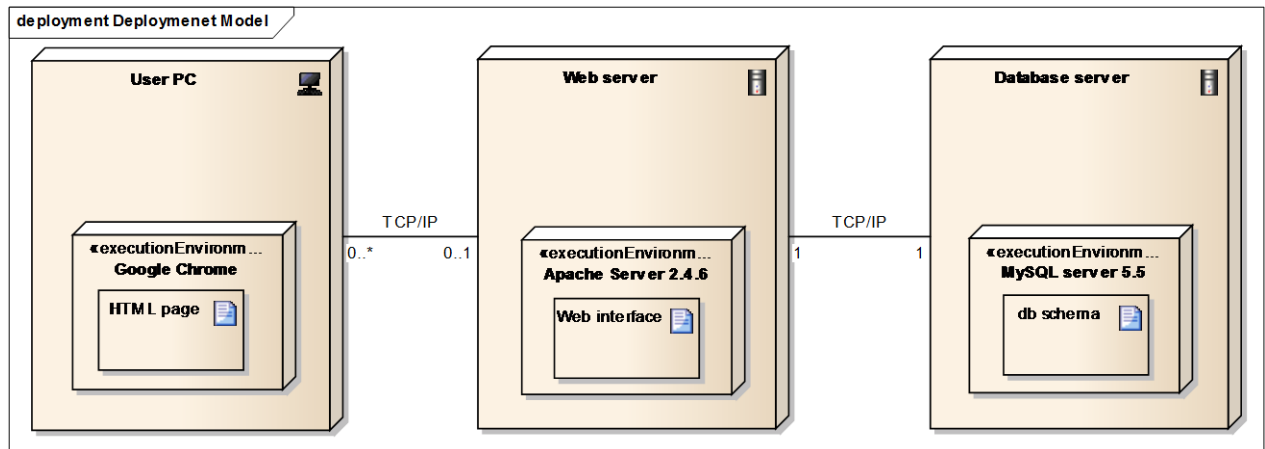


Figure: 9

Sequence Model

Vytvořit předmět - (Interaction diagram)

Created By: Tom on 24.11.2013, Last Modified: 24.11.2013

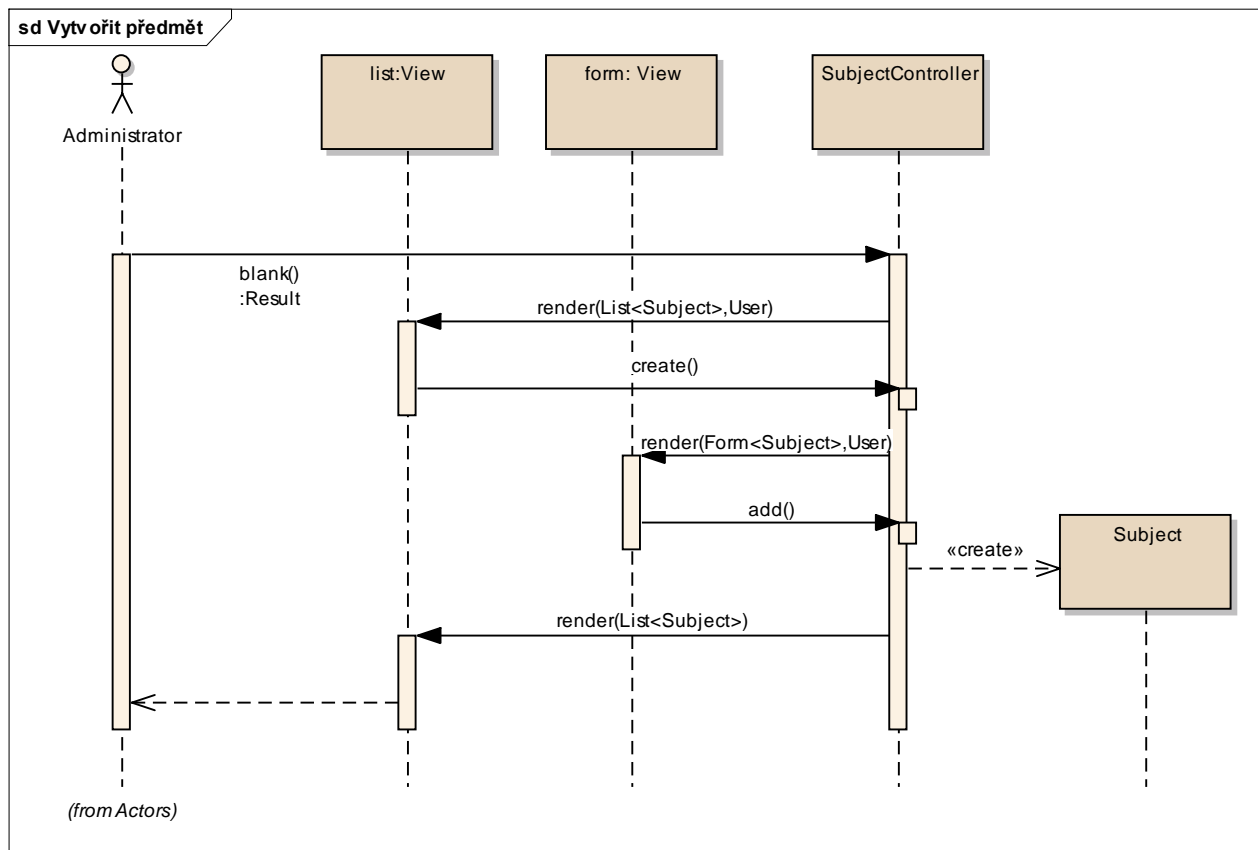


Figure: 10

Vytvořit uživatele - (Interaction diagram)

Created By: Tom on 24.11.2013, *Last Modified:* 24.11.2013

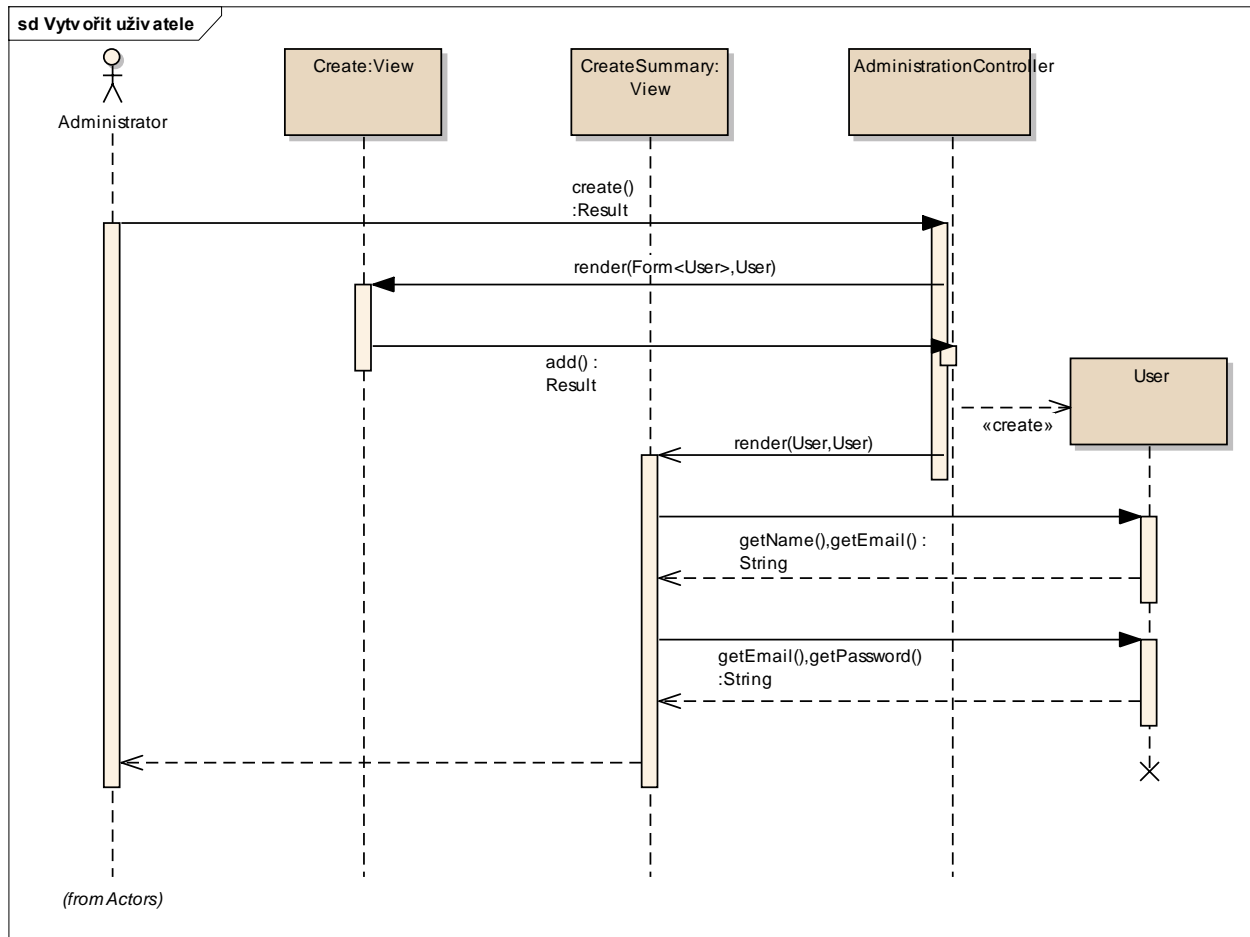


Figure: 11

Zmenit si heslo - (Interaction diagram)

Created By: Tom on 23.11.2013, *Last Modified:* 24.11.2013

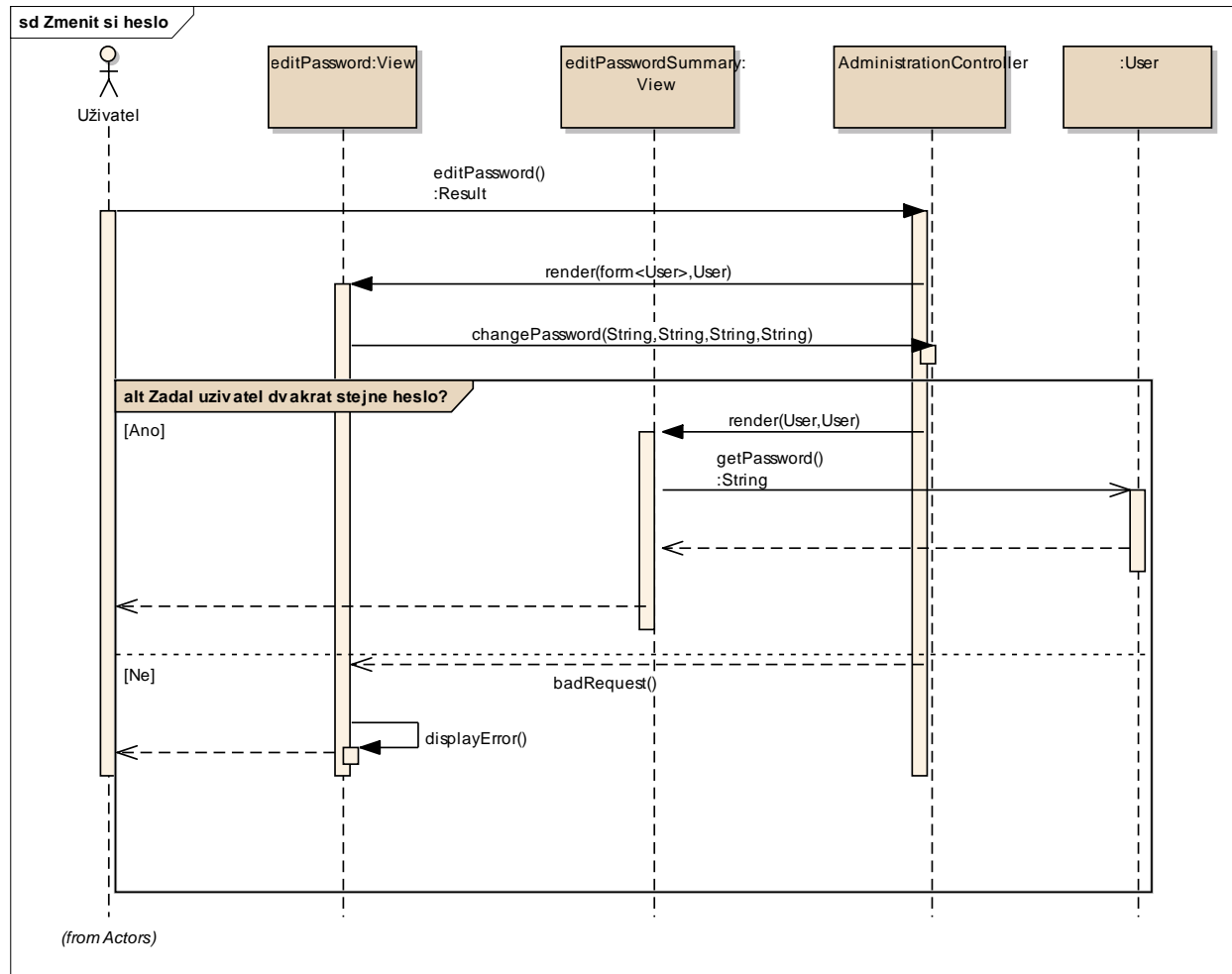


Figure: 12

Zobrazit vsechny uzivatele - (Interaction diagram)

Created By: Tom on 23.11.2013, Last Modified: 24.11.2013

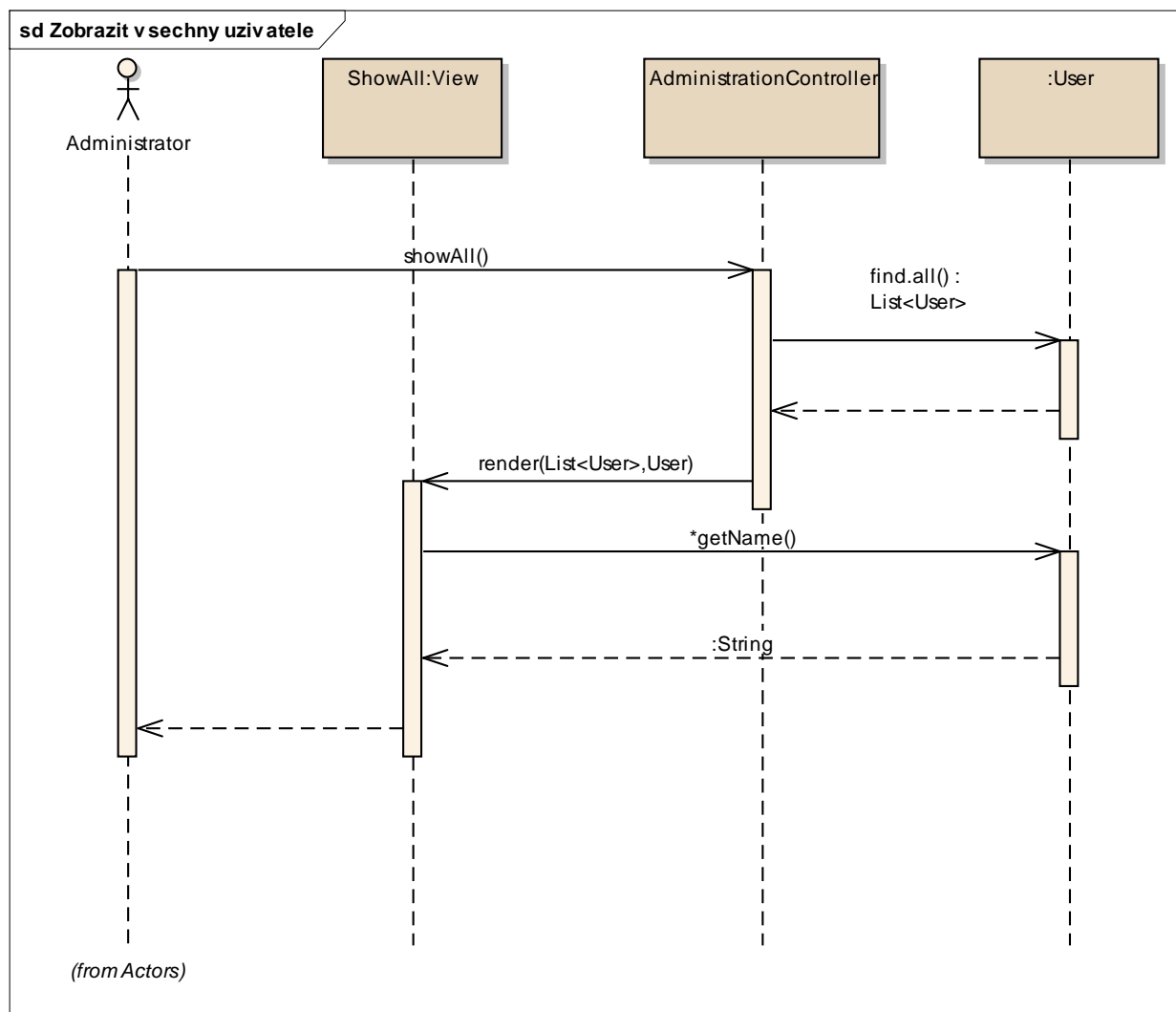


Figure: 13

Instalační manuál

Pro programátora:

Ke spuštění aplikace je potřeba:

1. Mít funkční JDK verzi 6 nebo pozdější, ke stažení na:
<http://www.oracle.com/technetwork/java/javase/downloads/index.html>
2. Výhodou je Java Runtime Environment, nejlépe Eclipse, IntelliJ případně NetBeans, s nainstalovaným *play plugin*.
3. Stáhnout si *play Framework* zde: <http://www.playframework.com/download>
4. Připsat si cestu ke stažené složce *play* na konec systémové proměnné PATH.
5. Aplikaci je pak možné spustit z terminálu příkazem *play ~run*.

Pro uživatele:

K chodu aplikace uživatel nepotřebuje instalovat žádné speciální programy. Je pouze nutné připojení k internetu a přístupové údaje k přihlášení poskytnuté Studijním oddělením.

Uživatelský manuál

V přihlašovací okně se můžete přihlásit jako admin, student nebo učitel. Dále uvedení uživatelé jsou vytvořeni pro potřeby testování a ukázkou funkčnosti.

1. Admin

email: admin@admin.com heslo: admin

Menu

1. ŠIP - Úvodní stránka s logem.
2. Předměty - Stránka předmětů. Admin může přidat předmět, dále vidí seznam všech předmětů, může je upravit, smazat nebo zobrazit podrobné informace. Přidat předmět - Stránka umožňuje přidat předmět do databáze. Kliknutím na název předmětu se zobrazí podrobné informace. Po stisknutí Upravit může admin upravit informace daného předmětu. Tlačítko Smazat smaže daný předmět.
3. Rozvrh - Zatím neimplementováno.
4. Studijní výsledky - Zatím neimplementováno.
5. Zápočty a zkoušky – Pro admina nepřístupné.
6. Domácí úkoly - Pro admina nepřístupné.
7. Ostatní - Přidat uživatele zobrazí formulář pro vytvoření nového uživatele.
 - Zobrazit všechny uživatele zobrazí seznam všech uživatelů v systému.
 - Změnit heslo. Formulář pro změnu hesla aktuálního uživatele. Je třeba zadat původní heslo.
8. Odhlásit se - Odhlásí uživatele a zobrazí login obrazovku.

2. Student

email: student@student.com heslo: student

Menu

1. ŠIP - Úvodní stránka s logem.
2. Předměty - Stránka předmětů. Student vidí seznam všech předmětů, nemůže je upravit, ale může zobrazit podrobné informace kliknutím na název předmětu. V detailu předmětu si může student zapsat předmět tlačítkem Zapsat, nebo zrušit zapsání tlačítkem Zrušit zapsání. Zobrazit zapsané předměty zobrazí seznam všech předmětů, které má student zapsané.
3. Rozvrh - Zatím neimplementováno.

4. Studijní výsledky - Zatím neimplementováno.

5. Zápočty a zkoušky - Zobrazí se seznam vypsaných zkoušek. U každé zkoušky je políčko přihlásit, jehož pomocí se může student přihlásit na zkoušku. Poté se zobrazí podrobné informace o zkoušce. Pokud má student zapsaný některý termín zkoušky, může se odhlásit tlačítkem Odhlásit.

6. Domácí úkoly – Zobrazí se seznam předmětů, které má student zapsány. Kliknutím na název předmětu se zobrazí seznam úkolů, pokud je nějaký úkol zadáný. Kliknutím na název úkolu se zobrazí jeho detaily.

7. Ostatní - Změnit heslo. Formulář pro změnu hesla aktuálního uživatele. Je třeba zadat původní heslo.

8. Odhlásit se - Odhlásí uživatele a zobrazí login obrazovku.

3. Učitel

email: teacher@teacher.com heslo: teacher

Menu

1. ŠIP - Úvodní stránka s logem.

2. Předměty - Stránka předměty. Učitel vidí seznam všech předmětů, nemůže je upravit, ale může zobrazit podrobné informace kliknutím na název předmětu.

3. Rozvrh - Zatím neimplementováno.

4. Studijní výsledky - Zatím neimplementováno.

5. Zápočty a zkoušky - Zobrazí se seznam vypsaných zkoušek. Učitel může jednotlivé termíny upravit nebo smazat. Tlačítko přidat termín spustí formulář pro vypsání nového termínu.

6. Domácí úkoly - Zobrazí se seznam předmětů, které učitel učí. Kliknutím na název předmětu se zobrazí seznam úkolů, pokud je nějaký úkol zadáný. Kliknutím na název úkolu se zobrazí jeho detaily a dále pak odevzdané úlohy pro daný úkol. Tlačítkem zadat úkol se spustí formulář pro vytvoření nového úkolu.

7. Ostatní - Změnit heslo. Formulář pro změnu hesla aktuálního uživatele. Je třeba zadat původní heslo.

- Zobrazit studenty. Zobrazí seznam všech studentů v systému.

8. Odhlásit se - Odhlásí uživatele a zobrazí login obrazovku.

Zpráva o implementaci

1. Vytvoření struktury aplikace, zvolen model-view-controller přístup
2. Vytvořen databázový model podle objektového návrhu Ebean, který je obsažen v frameworku Java Play!
3. Vytvořeno rozlišování mezi různými uživatelskými rolemi (admin, student a učitel)
4. Implementovány funkce pro práci s předměty (přidat předmět, upravit předmět, smazat předmět, zobrazit všechny předměty, zobrazení detailu předmětu, zapsat předmět, zobrazit zapsané předměty)
5. Implementovány administrátorské funkce (přidat uživatele, zobrazit všechny uživatele)
6. Implementována uživatelská funkce pro změnu hesla.
7. Implementováno zobrazení všech termínů zkoušky.
8. Implementovány možnosti studentu zapsat si předmět a zrušit jeho zapsání.
9. Implementovány funkce pro administraci zkoušek. Učitel může přidávat a upravovat termíny zkoušek. Student se může přihlásit a také odhlásit z jednotlivých termínů.
10. Implementovány funkce pro práci s domácími úkoly. Učitel může zadávat úkoly k předmětům, které učí. Dále si může zobrazit odevzdané úkoly. Student si může zobrazit všechny zadané úkoly a také jejich zadání.

Akceptační testy

Domácí úkoly

Učitel

Zobrazit odevzdané úkoly

- Když učitel neučí žádné předměty, zobrazí se hláška: „Neučíte žádný předmět.“
- Pokud má učitel nějaké předměty, zobrazí se jejich seznam.
- Když předmět nemá žádný zadaný úkol, zobrazí se hláška: „Žádný úkol.“
- Když zadaný úkol nemá žádné odevzdané řešení, zobrazí se hláška: „Žádný odevzdaný úkol.“

Zadat úkol

- Když není zadaný název, termín nebo zadání úkolu, u příslušného pole se zobrazí informační hláška.
- Když zadaný název úkolu je už v databázi, zobrazí se u pole „Název úkolu“ hláška: „Již existuje úkol s tímto názvem.“
- Pokud vše proběhlo správně, zobrazí se okno „Úkol vytvořen“ a přehled zadaného úkolu.

Student

Zobrazit zadané úkoly

- Když student nemá zapsané žádné předměty, zobrazí se hláška: „Nemáte zapsaný žádný předmět.“
- Pokud má student nějaké předměty, zobrazí se jejich seznam.
- Když předmět nemá žádný zadaný úkol, zobrazí se hláška: „Žádný úkol.“

Administrace

Admin

Vytvořit uživatele

- Když pole Jméno, Příjmení, Email, Uživatelská role a Heslo nejsou vyplněná, zobrazí se informační hláška, která upozorní na nutnost vyplnění daných polí.
- Když pole „Email“ není vyplněné běžným typem emailové adresy (např.: example@example.address), zobrazí se upozornění, že email není validní.
- Když je v poli „Email“ zadaný email, který je už použitý v databázi, zobrazí se hláška: „Tento email je již v databázi.“
- Když do pole „Heslo“ zadám řetězec kratší než 6 znaků, zobrazí se varování, že heslo musí být alespoň 6 znaků dlouhé.

- Když zadaný řetězec v poli „Zopakujte heslo“ není stejný jak v poli „Heslo“, zobrazí se hláška: „Hesla nejsou stejná“.

Uživatel

Změnit heslo

- Když do pole „Původní heslo“ zadáme jinou hodnotu, než je původní heslo, zobrazí se hláška: „Špatné heslo.“
- Když do pole „Heslo“ zadám řetězec kratší než 6 znaků, zobrazí se varování, že heslo musí být alespoň 6 znaků dlouhé.
- Když zadaný řetězec v poli „Zopakujte heslo“ není stejný jak v poli „Heslo“, zobrazí se hláška: „Hesla nejsou stejná“.

Zkoušky

Učitel

Zobrazit vypsané termíny

- Když nebyly zatím vypsaný žádné termíny budoucích zkoušek, zobrazí se učiteli hláška „Žádné termíny nejsou vypsaný“.
- Pokud byly již vypsaný nějaké termíny zkoušek, zobrazí se jejich úplný seznam.

Přidat termín

- Když si učitel zobrazí zkoušky, zobrazí se pouze jemu možnost: „Přidat termín“.
- Když učitel přidává zkoušku, nabídne se mu úplný výběr všech předmětů a všech zkoušejících v databázi.
- Pokud učitel při přidávání termínu nevyplnil všechny povinné položky zkoušky, je upozorněn příslušnou hláškou u položky či položek. Neúplný termín se neukládá do databáze.
- Pokud se učitel rozhodne přidání termínu zrušit, termín zůstane nezměněn.
- Pokud učitel nevyplnil všechny vstupní položky ve správném formátu, musí je vyplnit správně, nebo přidání termínu zrušit.
- Pokud učitel nevyplnil validní datum, musí ho vyplnit správně, nebo přidání termínu zrušit.

Změnit termín

- Pokud byly vypsaný nějaké termíny, může je učitel změnit.
- Pokud nejsou všechny povinné položky zkoušky vyplněny, je položka automaticky doplněna starou hodnotou, učitel pak může tuto úpravu potvrdit.
- Pokud se učitel rozhodne změnu termínu zrušit, termín zůstane nezměněn.
- Pokud učitel nevyplnil všechny vstupní položky ve správném formátu, musí je vyplnit správně, nebo přidání termínu zrušit či termín smazat.

- Pokud učitel nevyplnil validní datum, musí ho vyplnit správně, nebo přidání termínu zrušit či termín smazat.

Smazat termín

- Pokud byly vypsány nějaké termíny, může je učitel smazat.
- Pokud se učitel rozhodne termín smazat, bude hned smazán bez ohledu na korektnost vyplněného formuláře.

Student

Zobrazit dostupné vypsané termíny

- Když nebyly zatím vypsány žádné termíny z předmětů, které má student zapsané, zobrazí se studentovi hláška „Žádné termíny nejsou vypsány“.
- Pokud byly již vypsány nějaké termíny zkoušek, zobrazí se pouze ty, které jsou z předmětů zapsaných studentem.

Zapsat se na termín

- Pokud se podařilo studentovi přihlásit se na termín zkoušky, zobrazí se nápis „Zkouška v pořádku přihlášena“ a detaily termínu.
- Když se student přihlásí na termín, u termínu již není nápis „Přihlásit“, ale „Odhlásit“.

Odhlásit se z termínu

- Pokud se podařilo studentovi odhlásit se z termínu zkoušky, zobrazí se nápis „Zkouška v pořádku odhlášena“ a detaily termínu.
- Když se student odhlásí z termínu, u termínu již není nápis „Odhlásit“, ale „Přihlásit“.

Myslím, že práce v týmu a komunikace mezi členy se nám dařila. Pro každou iteraci jsme si obvykle vhodně úkoly rozdělili, k žádným větším nejasnostem v týmu nedocházelo. Volba využít *play framework* byla nejspíš dobrá, po krátkém zaučení může člověku skutečně usnadnit práci. K problému jsme akorát narazili u závěrečného spuštění aplikace na DVD s *virtualbox*. Příště bych asi také preferoval postup a prostředky, které jsme zvolili.

Tomáš Jiran

Práce na projektu pro mě byla dobrá zkušenost s prací v kolektivu. Myslím, že komunikace v týmu byla dobrá, na cvičení jsme si vždy rozdělili hlavní části práce a podrobnosti a dotazy jsme řešili přes skype. Zadání projektu školního portálu bylo velmi rozsáhlé, takže se nám nikdy nestalo, že by někdo neměl co dělat. Používání gitu bylo pro mě novinkou, ze začátku jsem neviděl pozitivní přínos, ale pro práci s implementací to byl nutný a šikovný nástroj. Myslím si, že se nám jako týmu dařilo, jednotlivé iterace jsme vypracovávali včas a nenastaly žádné větší komplikace.

Dominik Hons

Až doposud jsem byl zvyklý pracovat na různých projektech, ať už školních nebo i soukromých, sám nebo maximálně ve dvou. Jako přínos předmětu Softwarové inženýrství vidím získání zkušenosti s prací v týmu více lidí. V našem týmu jsme neměli žádné větší problémy. Pokud nastal nějaký problém, tak jsme ho společně rychle vyřešili. Komunikace probíhala přes Skype a všichni členové týmu na něm byli téměř vždy k dispozici. Další přínos pro mě byla zkušenost s verzovacím nástrojem GIT, jelikož si myslím, že tento nástroj budu v budoucnu často používat. Jsem rád, že jsem si ho mohl zkusit při práci na skutečném projektu.

Filip Bouška

Celkově musím zhodnotit týmovou práci na projektu kladně. Myslím, že se nám dařilo spolu velmi dobře komunikovat jak na cvičeních, tak na Skypu, kde jsme si vždy bez problémů upřesnili práci, případně rychle vyřešili některé nedostatky. Mezi jednotlivými iteracemi (až na první, kdy tým bez oznámení opustil jeden z členů) se nám ani jednou nestalo, že by někdo vypracoval nevyhovující část projektu. Osobně vidím jako dobrou zkušenost především spolupráci v týmu, seznámení s gitem a modelování v EA.

Tom Nováček

(hodnoty v hodinách)

Filip Bouška		Dominik Hons		Tomáš Jiran		Tom Nováček		
1. iterace	celkem	19,5	celkem	44	celkem	46,5	celkem	35
Vize, studium materiálů + její tvorba	5	BPM proces úkolů	3	BPM - Přihlašování na ZK - vypracování 1. verze	3	Vize	1	
Skype - porada	1	Požadavky host a PEO	2	BPM - přihlašování na ZK - předělání	4	BPM + opravy	3	
BPM	2	BDM + úpravy	4	UC - Zkoušky - vypracování	3	Požadavky	2	
Požadavky v EA	2	Use-case model úkolů	2	UC - Předměty - úpravy include / extend	0,5	Use-case	4	
Skype - porada	0,5	Úprava BPM a BDM	2	UC - Rozvrh - úpravy include / extend	0,5	Mapping	4	
Požadavky student, obecné	1	Finální úpravy	2	BDM - dořešení multiplicit	1	revize vize	1	
BPM	1	Skype porada	3	BPM, UC - finální úpravy	1	Skype porada	3	
UC	1,5			Model požadavků - drobná editace	0,5			
GIT	1,5			Požadavky - výpis těch, které chybí dodělat	0,5			
Skype - porada, finální úpravy	4			kompletování	1			
				Vize - drobná editace a korekce překlepů	0,5			
				Skype porada	3			
2. iterace	celkem	21	celkem	16	celkem	13	celkem	12
architektonický základ + javaplay	20	Oponentský posudek	3	BPM, UCS - dodatečné opravy z 1. iterace	2	Oprava dle oponentury	2	
administrativa	1	Úprava stránek	1	Analytický model - kontrola, tutoriál, úpravy	2	Oponentura	2	
		Scénáře úkolů	3	Analytický model - přidání popisů	1	Diagramy stavů	3	
		Oprava první iterace	4	UC scénáře	7	Scénáře případů užití	3	
		Analytický doménový model	3	Kontrola projektu	1	Dokumentace	2	
		Finální kontrola projektu	2					
3. iterace	celkem	10	celkem	14	celkem	24,5	celkem	12
implementace UC	8	Admin přidat uživatele	4	play zprovoznění	10	Oponentura na hospodu	1	
administrativa	1	Logged user	3	db model	5	Diagram nasazení	4	
zpráva o impl.	1	Rozlišení uživ. Rolí	3	play tutoriály	4	Sečtení diagramy	5	
		Oprava 2. iterace	1	model architektury	4	Dokumentace	1	
		Změna hesla	1	instalační manuál	0,5	Oprava chyb	1	
		Přidání uživatele vytváří studenta	2	model architektury	0,5			
		Finální kontrola	1	model	0,5			
4. iterace	celkem	15	celkem	18	celkem	14	celkem	14
Implementace	6	Úkoly základ	3	implementace Ucs	10	Oprava dle oponentury	2	
VirtualBox	6	Zadání úkolu	4	ladění, testování	3	Dokumentace	2	
Dokumentace	2	Zobrazení úkolu studentům	4	dokončení, zhodnocení	1	Implementace	4	
Ostatní práce	1	Akceptační testy úkoly	2			testování	1	
		Zpráva o implementaci	1			Úprava modelů	4	
		Akceptační testy administrace	2			Oponentura	1	
		Uživatelský manuál	2					

	6.týden		8.týden		10.týden		12.týden		
Příjmení a jméno	Důvod přerozdělení	Přerozdělené body	Důvod přerozdělení	Přerozdělené body	Důvod přerozdělení	Přerozdělené body	Důvod přerozdělení	Přerozdělené body	Celkem
Bartovský Vojtěch	Nedodání zadane práce, neúčast na cvičeníh, nezapojení do týmové komunikace(Skype)	-10		0		0		0	-10
Bouška Filip	Sepsání vize, založení týmového repozitáře, katalog požadavků	3	Rozchození aplikace, časově nejnáročnější práce oproti ostatním členům	4		-2		-2	3
Hons Dominik	Domain model, BPM, use case	2		-1		0	Implementace, testování	2	3
Jiran Tomáš	BPM, Use case, korekce Domain modelu	2		-2	Časově nejnáročnější práce oproti ostatním členům	4		-2	2
Nováček Tom	Mapování požadavky - use case, BPM, dokumentace	3		-1		-2	Práce na modelech, oprava	2	2
Celkem (musí být 0)		0		0		0		0	0
Přerozdeleno (v každé „etapě“ musí být přerozdělen minimálně takový počet bodů kolik je členů týmu)		10		4		4		4	0

Počet členů týmu: