UNIVERZITA PARDUBICE  
Fakulta elektrotechniky a informatiky

*Název projektu*

Seminární práce z předmětu Databázové systémy II

Pracovní tým:

Baláček D., Nyč R.

V Pardubicích dne 9. 12. 2018

Obsah

[Úvod 10](#_Toc532949096)

[Zadání 11](#_Toc532949097)

[Uživatelská dokumentace 18](#_Toc532949098)

[Základní popis používané aplikace 19](#_Toc532949099)

[Instalace 19](#_Toc532949100)

[Přístupová oprávnění 19](#_Toc532949101)

[Použití aplikace 19](#_Toc532949102)

[Programová dokumentace 20](#_Toc532949103)

[Datová část 21](#_Toc532949104)

[Fyzický model dat 21](#_Toc532949105)

[Číselníky 21](#_Toc532949106)

[Pohledy 23](#_Toc532949107)

[Funkce 25](#_Toc532949108)

[Uložené procedury 28](#_Toc532949109)

[Spouště 31](#_Toc532949110)

[Indexy 31](#_Toc532949111)

[Sekvence 31](#_Toc532949112)

[Aplikace 32](#_Toc532949113)

[Použité prostředí 32](#_Toc532949114)

[Řízení uživatelských účtů 32](#_Toc532949115)

[Moduly 32](#_Toc532949116)

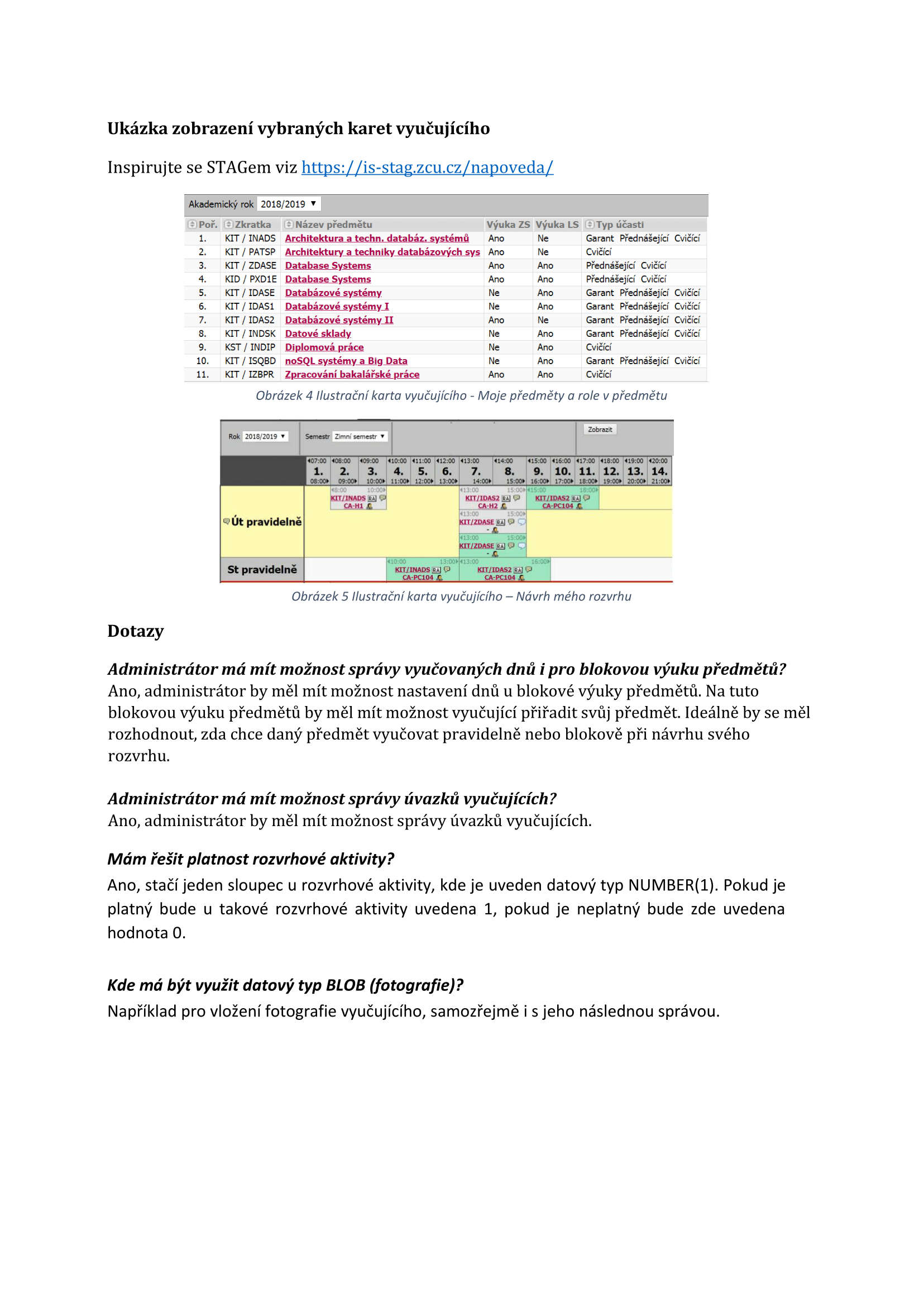
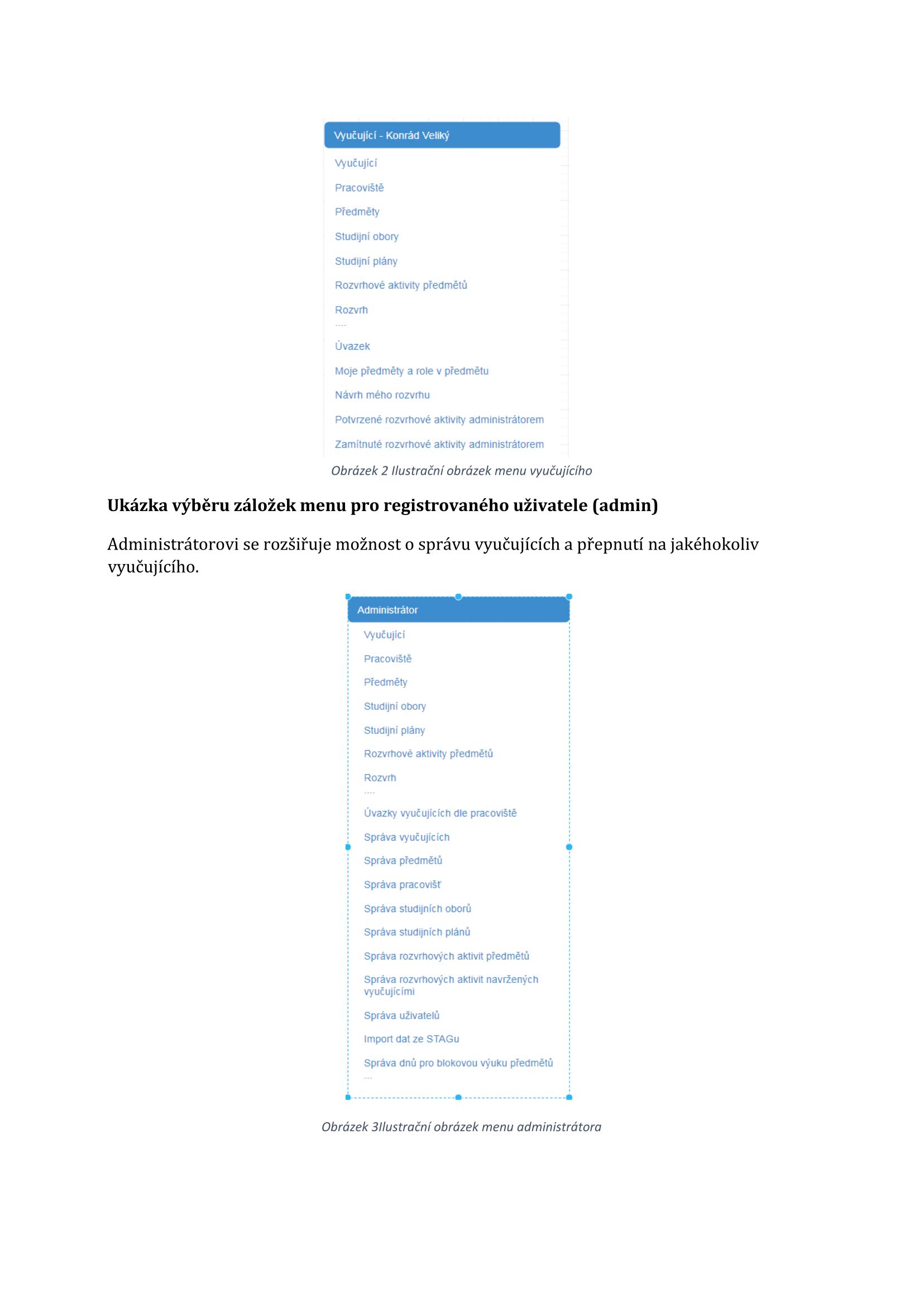
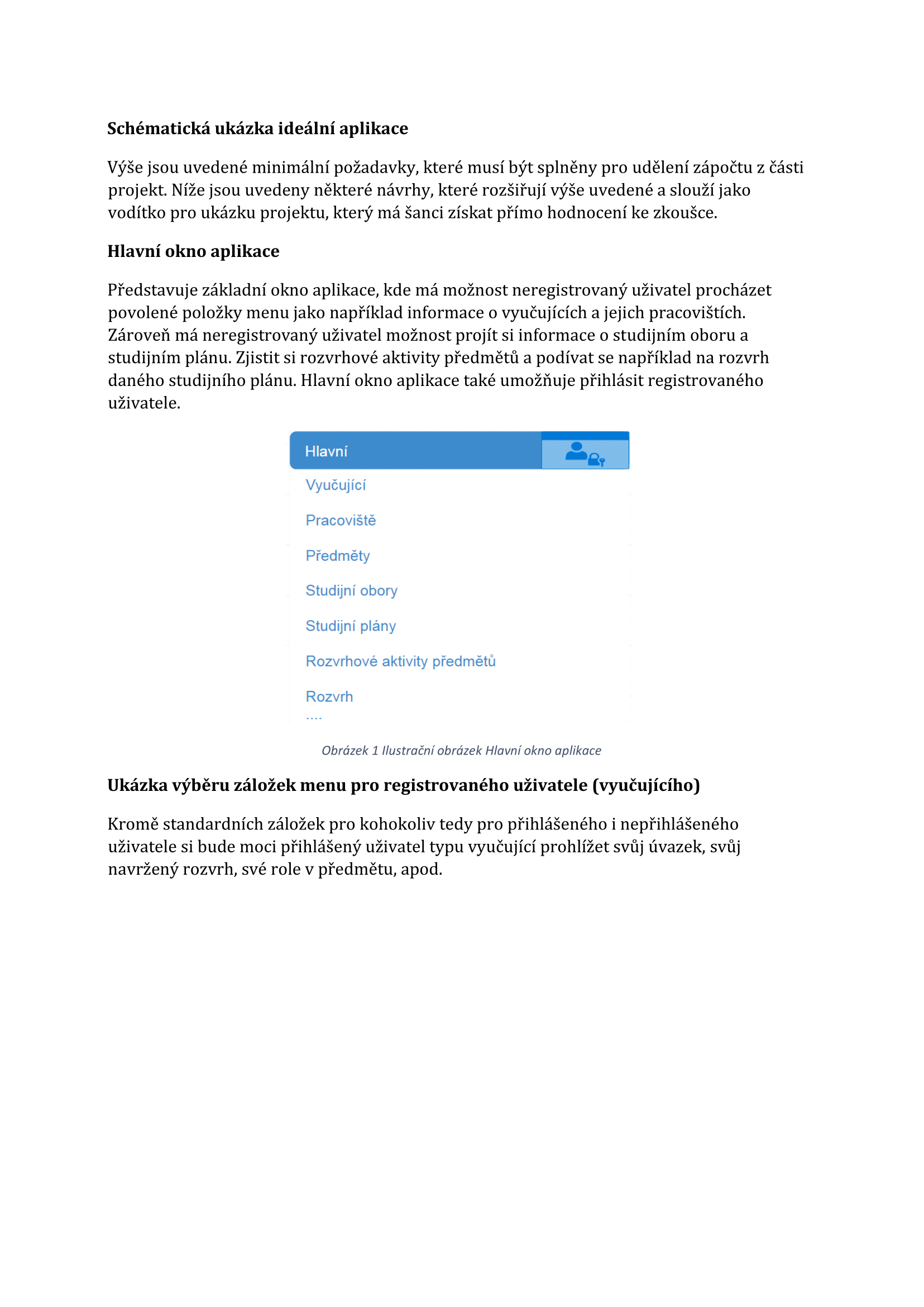
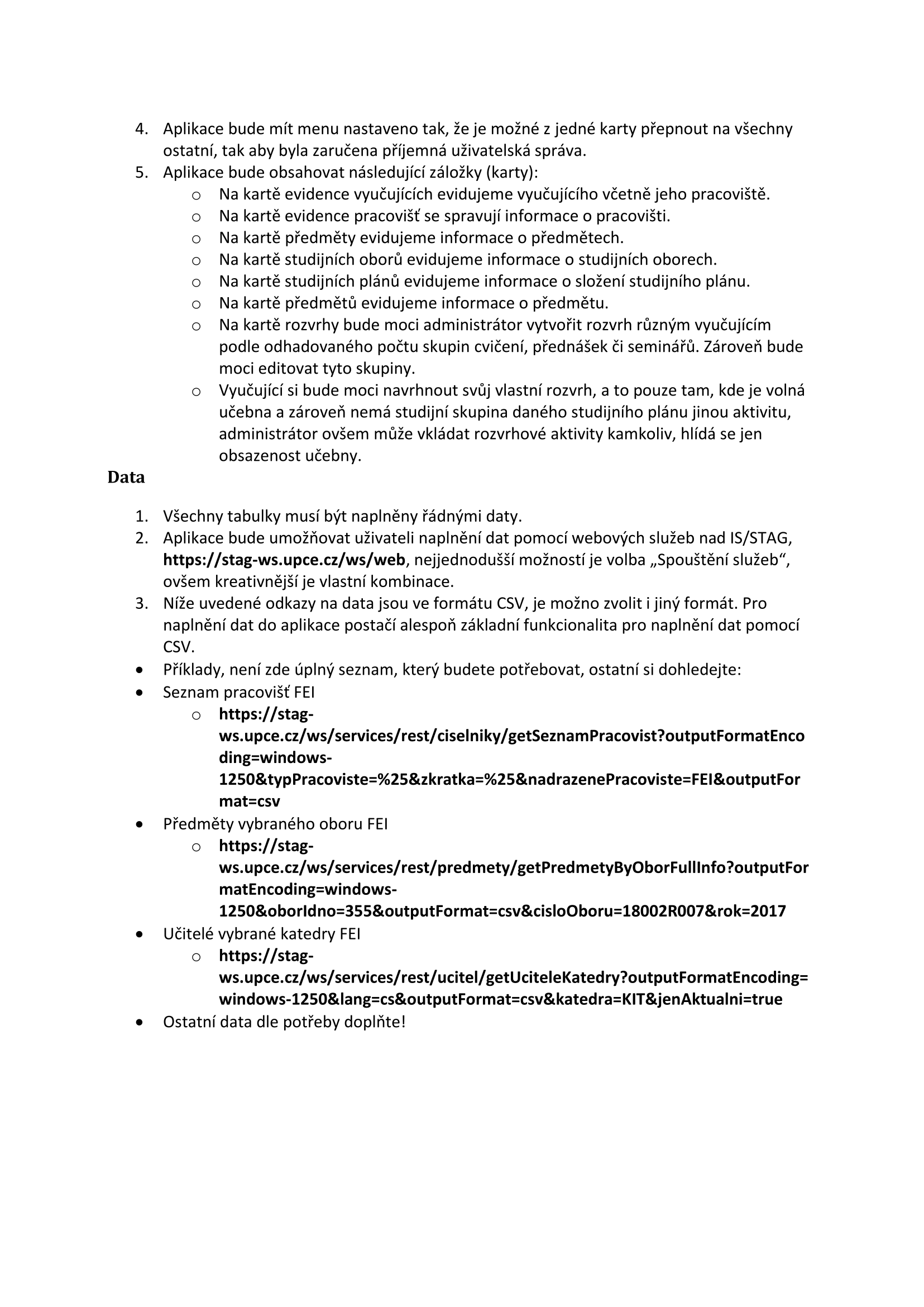
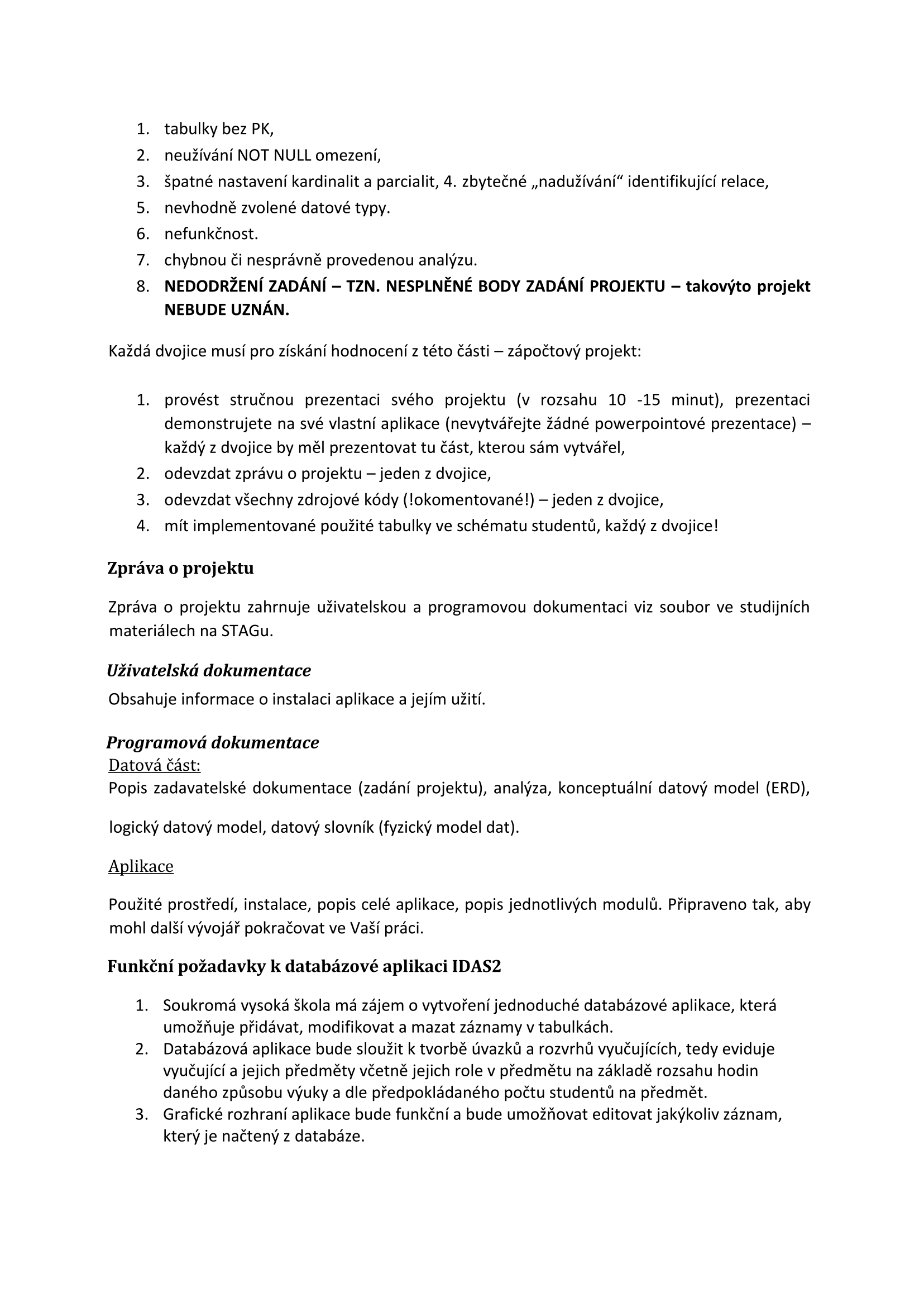
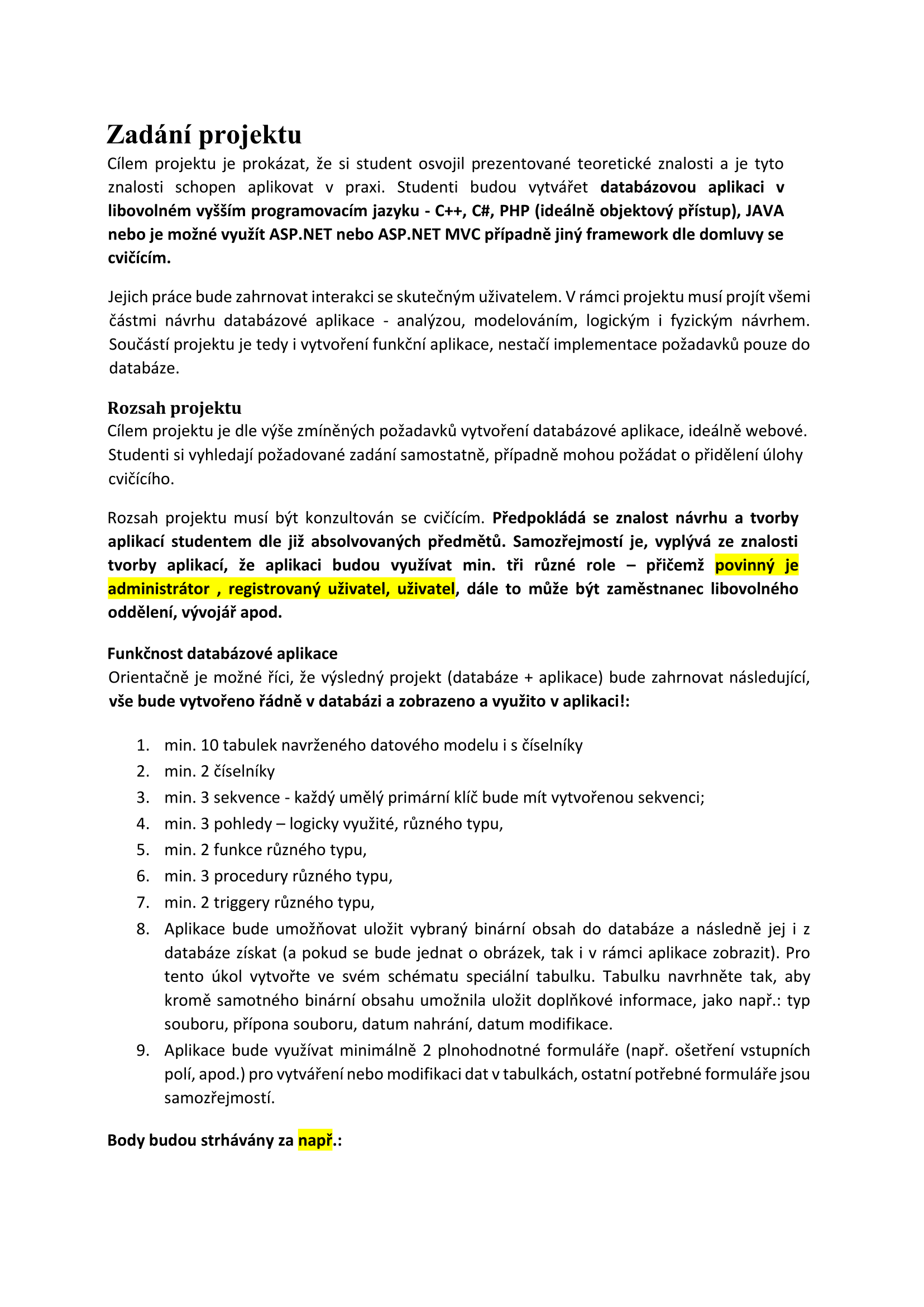
[Formuláře 33](#_Toc532949117)

[Orientace ve zdrojovém kódu 35](#_Toc532949118)

Úvod

Toto je dokumentace pro správu databáze pro vysokou školu. Dosavadní model a program uměl spravovat předměty, rozvrhové akce, vyučující, katedry, obory a vazební tabulku obor-předmět. Dokumentace obsahuje jak tuto část, tak část která je přidána vzhledem k novému zadání.

# Zadání



Uživatelská dokumentace

## Základní popis používané aplikace

Aplikace slouží pro úpravu tabulek s uloženými daty v databázi. Samostatně tak slouží pro správu systému pro vysoké školy. Pokud je uživatel administrátorem, může měnit jednotlivé tabulky, schvalovat rozvrhové akce vyučujících a importovat data ze souborů. Vyučující mohou přidávat, upravovat a odebírat rozvrhové akce. Ty po každé změně musí schválit admin. Každý registrovaný uživatel může měnit některé své informace.

Aplikace z důvodu změny zadaní po dokončení a konzultaci datového modelu neobsahuje:

1. studijní plány

## Instalace

Pro spuštění aplikace je potřeba mít nainstalovaný Java virtual machine. Pak stačí spustit soubor .jar.

## Přístupová oprávnění

Role jsou popsány dále v dokumentaci. Pro testovaní byly vytvořeny účty:

* Administrátor (a zároveň ne-vyučující) – jméno: admin heslo: admin
* Vyučující – jméno: reg heslo: reg

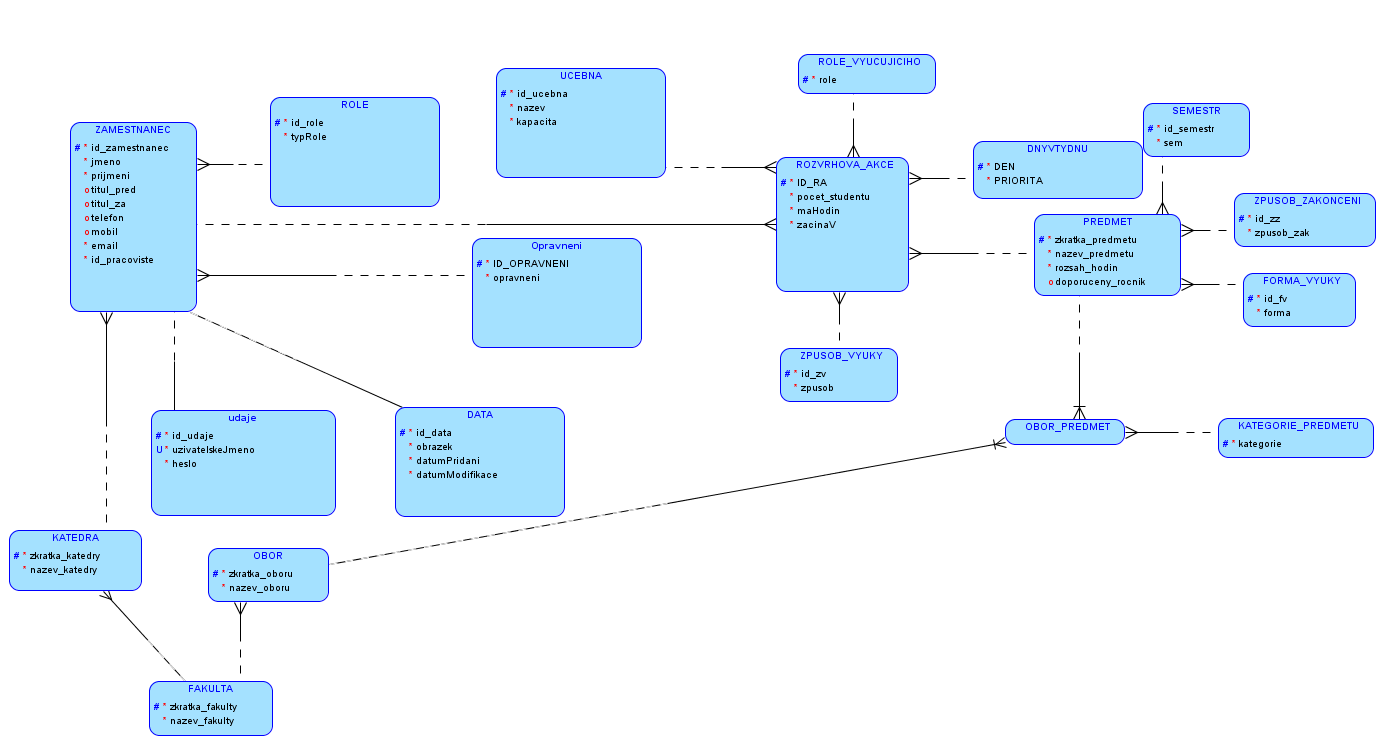
## Použití aplikace

Formuláře

Programová dokumentace

# Datová část

## Fyzický model dat



Model zachycuje databázové řešení systému pro vysokou školu.

## Číselníky

* Role
  + 1; Vyučující
  + 2; Správce systému

Role zaměstnance v instituci. Navrhované další role by byly uklízeč, sekretář, …. Pojí se k zaměstnanci.

* Opravneni
  + 1; Administrátor
  + 2; Registrovaný
  + 3; Neregistrovaný

Určuje pravomoce uživatele pro aplikaci. Pojí se k zaměstnanci.

* Zpusob\_vyuky
  + 1; Cvičení
  + 2; Seminář
  + 3; Přednáška
  + 21; Jiné

Způsob výuky dané akce. Pojí se s rozvrhovou akcí.

* Role\_vyucujiciho
  + Cvičící
  + Přednášející

Role vyučujícího v dané rozvrhové akci. Pojí se s rozvrhovou akcí.

* Dnyvtydnu
  + Pondělí; 1
  + Úterý; 2
  + Středa; 3
  + Čtvrtek; 4
  + Pátek; 5
  + Sobota; 6
  + Neděle; 7

Dny týdne rozvrhové akce s prioritou (podle které se pak řadí v aplikaci pro přehlednost). Pojí se s rozvrhovou akcí.

* Semestr
  + 1; Letní
  + 2; Zimní
  + 3; Oba

Ve kterém semestru je předmět vyučován, pojí se s předmětem.

* Zpusob\_zakonceni
  + 1; Zkouškou
  + 2; Zápočtem

Jakým způsobem je předmět zakončen. Pojí se s předmětem.

* Forma\_vyuky
  + 1; Prezenční
  + 2; Kombinovaná
  + 3; Distanční

Jakou formou je předmět vyučován. Pojí se s předmětem.

* Kategorie\_predmetu
  + A
  + B
  + C

Povinnost předmětu. Pojí se s vazební tabulkou obor\_predmet.

## Pohledy

1. Pracoviste

select \* from KATEDRA left join FAKULTA on FAKULTA.ZKRATKA\_FAKULTY = KATEDRA.FAKULTA\_ZKRATKA\_FAKULTY;

1. Rozvrhove\_akce\_ext\_view

create or replace view ROZVRHOVE\_AKCE\_EXT\_VIEW as

select r.ID\_RA as ID\_ROZVRHOVE\_AKCE, r.POCET\_STUDENTU, r.DENVTYDNU, r.SCHVALENO, r.MAHODIN, r.ZACINAV, r.ROLE\_VYUCUJICIHO\_ROLE, r.PREDMET\_ZKRATKA\_PREDMETU as ZKRATKA\_PREDMETU, r.ID\_ZAMESTNANEC as ID\_VYUCUJICIHO, p.NAZEV\_PREDMETU, p.DOPORUCENY\_ROCNIK, v.ZPUSOB, v.ID\_ZV as ID\_ZPUSOBU, z.JMENO as JMENO\_VYUCUJICIHO, z.PRIJMENI as PRIJMENI\_VYUCUJICIHO, z.TITUL\_PRED, z.TITUL\_ZA, z.TELEFON, z.MOBIL, z.EMAIL, z.KATEDRA\_ZKRATKA\_KATEDRY, u.nazev as NAZEV\_UCEBNY, u.kapacita as KAPACITA\_UCEBNY, u.ID\_UCEBNA as ID\_UCEBNY, s.sem, s.id\_semestr from ROZVRHOVA\_AKCE r

inner join PREDMET p on r.PREDMET\_ZKRATKA\_PREDMETU = p.ZKRATKA\_PREDMETU

left join SEMESTR s on p.semestr\_id\_semestr = s.id\_semestr

left join ZPUSOB\_VYUKY v on r.ZPUSOB\_VYUKY\_ID\_ZV = v.ID\_ZV

left join ZAMESTNANEC z on r.ID\_ZAMESTNANEC = z.ID\_ZAMESTNANEC

left join UCEBNA u on r.ID\_UCEBNA = u.ID\_UCEBNA;

1. Spojeni\_oboru\_predmetu\_view

select ZKRATKA\_OBORU, NAZEV\_OBORU, ZKRATKA\_PREDMETU, NAZEV\_PREDMETU, KATEGORIE\_PREDMETU\_KATEGORIE, DOPORUCENY\_ROCNIK, SEM, ZPUSOB\_ZAK, FORMA from OBOR\_PREDMET

inner join PREDMET on OBOR\_PREDMET.PREDMET\_ZKRATKA\_PREDMETU = PREDMET.ZKRATKA\_PREDMETU

inner join OBOR on OBOR\_PREDMET.OBOR\_ZKRATKA\_OBORU = OBOR.ZKRATKA\_OBORU

left join SEMESTR on PREDMET.SEMESTR\_ID\_SEMESTR = SEMESTR.ID\_SEMESTR

left join ZPUSOB\_ZAKONCENI ON PREDMET.ZPUSOB\_ZAKONCENI\_ID\_ZZ = ZPUSOB\_ZAKONCENI.ID\_ZZ

left join FORMA\_VYUKY on PREDMET.FORMA\_VYUKY\_ID\_FV = FORMA\_VYUKY.ID\_FV;

1. Vyuc\_view

select z.ID\_ZAMESTNANEC as ID\_VYUCUJICIHO,z.JMENO,z.PRIJMENI,z.TITUL\_PRED,z.TITUL\_ZA,z.TELEFON,z.MOBIL,z.EMAIL,z.KATEDRA\_ZKRATKA\_KATEDRY, o.OPRAVNENI, d.OBRAZEK, d.DATUMPRIDANI, d.DATUMMODIFIKACE from zamestnanec z

left join OPRAVNENI o on z.ID\_OPRAVNENI = o.ID\_OPRAVNENI

left join ROLE r on z.ID\_ROLE = r.ID\_ROLE

left join DATA d on z.ID\_ZAMESTNANEC = d.ID\_ZAMESTNANEC

where r.typrole = 'Vyučující';

1. Zam\_view

select z.ID\_ZAMESTNANEC,z.JMENO,z.PRIJMENI,z.TITUL\_PRED,z.TITUL\_ZA,z.TELEFON,z.MOBIL,z.EMAIL,z.KATEDRA\_ZKRATKA\_KATEDRY, o.OPRAVNENI, r.TYPROLE, d.OBRAZEK, d.DATUMPRIDANI, d.DATUMMODIFIKACE from zamestnanec z

left join OPRAVNENI o on z.ID\_OPRAVNENI = o.ID\_OPRAVNENI

left join ROLE r on z.ID\_ROLE = r.ID\_ROLE

left join DATA d on z.ID\_ZAMESTNANEC = d.ID\_ZAMESTNANEC;

1. Predmet\_ext\_view

select ZKRATKA\_PREDMETU, NAZEV\_PREDMETU, DOPORUCENY\_ROCNIK, SEMESTR\_ID\_SEMESTR, ZPUSOB\_ZAKONCENI\_ID\_ZZ, FORMA\_VYUKY\_ID\_FV, SEM, ZPUSOB\_ZAK, FORMA, SUM(POCET\_STUDENTU) as POCET\_STUDENTU from PREDMET

left join SEMESTR on PREDMET.SEMESTR\_ID\_SEMESTR = SEMESTR.ID\_SEMESTR

left join ZPUSOB\_ZAKONCENI ON PREDMET.ZPUSOB\_ZAKONCENI\_ID\_ZZ = ZPUSOB\_ZAKONCENI.ID\_ZZ

left join FORMA\_VYUKY on PREDMET.FORMA\_VYUKY\_ID\_FV = FORMA\_VYUKY.ID\_FV

left join ROZVRHOVA\_AKCE on PREDMET.ZKRATKA\_PREDMETU = ROZVRHOVA\_AKCE.PREDMET\_ZKRATKA\_PREDMETU group by ZKRATKA\_PREDMETU, NAZEV\_PREDMETU, DOPORUCENY\_ROCNIK, SEMESTR\_ID\_SEMESTR, ZPUSOB\_ZAKONCENI\_ID\_ZZ,

FORMA\_VYUKY\_ID\_FV, SEM, ZPUSOB\_ZAK, FORMA;

## Funkce

1. Dostacujemisto

Vrací boolean. True pokud se studenti dané RA do učebny vejdou, jinak false.

CREATE OR REPLACE FUNCTION dostacujeMisto

(p\_pocetStudentu ROZVRHOVA\_AKCE.POCET\_STUDENTU%TYPE, p\_idUcebny UCEBNA.ID\_UCEBNA%TYPE)

RETURN BOOLEAN IS

v\_jeVolno BOOLEAN := false;

v\_kapacita UCEBNA.KAPACITA%TYPE;

BEGIN

SELECT KAPACITA INTO v\_kapacita FROM UCEBNA WHERE ID\_UCEBNA = p\_idUcebny;

v\_jeVolno := v\_kapacita>=p\_pocetStudentu;

RETURN v\_jeVolno;

END;

1. Konechodiny

Vrací číslem, kdy končí hodina dané RA.

create or replace FUNCTION konecHodiny (p\_idRA ROZVRHOVA\_AKCE.ID\_RA%TYPE)

RETURN number IS

v\_konec ROZVRHOVA\_AKCE.ZACINAV%TYPE;

BEGIN

SELECT (MAHODIN + ZACINAV) into v\_konec

FROM ROZVRHOVA\_AKCE

WHERE ID\_RA = p\_idRA;

RETURN v\_konec;

END;

1. Mavolno

Vrací v boolean zda má vyučující v danou dobu volno či nikoliv.

create or replace FUNCTION maVolno

(p\_idZam ZAMESTNANEC.ID\_ZAMESTNANEC%TYPE, p\_zacatek ROZVRHOVA\_AKCE.ZACINAV%TYPE, p\_delka ROZVRHOVA\_AKCE.MAHODIN%TYPE, p\_den ROZVRHOVA\_AKCE.DENVTYDNU%TYPE,

p\_semestr SEMESTR.SEM%TYPE, p\_idUcebny UCEBNA.ID\_UCEBNA%TYPE, p\_idRA ROZVRHOVA\_AKCE.ID\_RA%TYPE)

RETURN BOOLEAN IS

v\_jeVolno BOOLEAN := true;

v\_zacatekDruhe ROZVRHOVA\_AKCE.ZACINAV%TYPE;

v\_delkaDruhe ROZVRHOVA\_AKCE.MAHODIN%TYPE;

v\_denDruha ROZVRHOVA\_AKCE.DENVTYDNU%TYPE;

v\_semestrDruha Semestr.sem%TYPE;

v\_jedenVObou BOOLEAN := false;

CURSOR c1 (p\_iz in NUMBER, p\_iu in NUMBER, p\_ira in NUMBER)

IS SELECT ZACINAV, MAHODIN, DENVTYDNU, SEM FROM ROZVRHOVE\_AKCE\_EXT\_VIEW WHERE (ID\_VYUCUJICIHO = p\_iz OR ID\_UCEBNY = p\_iu) AND NOT(ID\_ROZVRHOVE\_AKCE=p\_ira);

BEGIN

open c1 (p\_idZam, p\_idUcebny, p\_idRA);

LOOP

FETCH c1 INTO v\_delkaDruhe, v\_zacatekDruhe, v\_denDruha, v\_semestrDruha;

EXIT WHEN c1%NOTFOUND;

if((p\_semestr = 'Oba') or (v\_semestrDruha = 'Oba')) then

v\_jedenVObou := true;

else

v\_jedenVObou := false;

end if;

if (p\_den = v\_denDruha) then

if ((p\_zacatek BETWEEN v\_zacatekDruhe AND (v\_zacatekDruhe + v\_delkaDruhe - 0.017)) or

((p\_zacatek+p\_delka - 0.017) BETWEEN v\_zacatekDruhe AND (v\_zacatekDruhe+v\_delkaDruhe - 0.017))) then

if(v\_jedenVObou)then

v\_jeVolno := false;

elsif(p\_semestr = v\_semestrDruha)then

v\_jeVolno := false;

else

v\_jeVolno := true;

end if;

end if;

end if;

END LOOP;

CLOSE c1;

RETURN v\_jeVolno;

END;

1. Vratidprihlaseni

Vrátí ID přihlášeného zaměstnance podle vloženého jména a hesla.

create or replace FUNCTION vratIdPrihlaseni (p\_us UDAJE.UZIVATELSKEJMENO%TYPE, p\_ps UDAJE.HESLO%TYPE)

RETURN ZAMESTNANEC.ID\_ZAMESTNANEC%TYPE IS

v\_id UDAJE.ID\_ZAMESTNANEC%TYPE;

v\_ps UDAJE.HESLO%TYPE;

v\_psHash UDAJE.HESLO%TYPE;

ex\_spatneHeslo EXCEPTION;

BEGIN

SELECT ID\_ZAMESTNANEC, HESLO into v\_id, v\_ps

FROM UDAJE

WHERE UZIVATELSKEJMENO = p\_us;

select standard\_hash(p\_ps, 'MD5') into v\_psHash from dual;

IF (v\_ps = v\_psHash) THEN

RETURN v\_id;

ELSE

RAISE ex\_spatneHeslo;

END IF;

EXCEPTION

WHEN ex\_spatneHeslo THEN

raise\_application\_error(-20001, 'Zadané heslo není správné');

WHEN NO\_DATA\_FOUND THEN

raise\_application\_error(-20002, 'Takové přihlašovací jméno neexistuje');

END;

## Uložené procedury

1. Procedury pro vklad, úpravu a odstranění všech z aplikace upravitelných tabulek – vypíši jen zajímavé
2. Vlozobrazek

Zavede zaměstnanci obrázek. Pokud již fotku zaměstnanec má, upraví údaje, pokud ne, založí nový řádek.

create or replace PROCEDURE vlozObrazek(p\_obrazek DATA.OBRAZEK%TYPE, p\_idZamestnanec NUMBER)

IS

BEGIN

INSERT INTO

DATA (OBRAZEK, DATUMPRIDANI, DATUMMODIFIKACE, ID\_ZAMESTNANEC)

VALUES (p\_obrazek, SYSDATE, SYSDATE, p\_idZamestnanec);

exception

when DUP\_VAL\_ON\_INDEX THEN

update DATA

set OBRAZEK = p\_obrazek, DATUMMODIFIKACE = SYSDATE

where ID\_ZAMESTNANEC = p\_idZamestnanec;

END;

1. Smazfotku

Smaže řádek s obrázkem.

create or replace PROCEDURE smazFotku

(p\_id ZAMESTNANEC.ID\_ZAMESTNANEC%TYPE)

IS

BEGIN

delete from DATA WHERE ID\_ZAMESTNANEC = p\_id;

END;

1. Vlozzamestnance

Vloží uživatele a jeho přihlašovací údaje do databáze. Pokud některý z přihlašovacích údajů chybí, změní oprávnění na „Neregistrovaný“ a nezaloží nový řádek v tabulce přihlašovacích údajů.

create or replace PROCEDURE vlozZamestnance

(p\_jmeno VARCHAR2, p\_prijmeni VARCHAR2, p\_titulPred VARCHAR2, p\_titulZa VARCHAR2, p\_email VARCHAR2,

p\_zkratkaKatedry VARCHAR2, p\_opravneni NUMBER, p\_idRole NUMBER, p\_mobil NUMBER, p\_telefon NUMBER,

p\_uzivJmeno VARCHAR2, p\_heslo VARCHAR2)

IS

v\_id ZAMESTNANEC.ID\_ZAMESTNANEC%TYPE;

v\_idOpr NUMBER;

v\_pass VARCHAR2(50);

BEGIN

INSERT INTO ZAMESTNANEC (jmeno, prijmeni,titul\_pred,titul\_za,

email,katedra\_zkratka\_katedry, id\_opravneni, id\_role, mobil,telefon)

VALUES (p\_jmeno, p\_prijmeni, p\_titulPred, p\_titulZa, p\_email, p\_zkratkaKatedry, p\_opravneni, p\_idRole, p\_mobil, p\_telefon);

SELECT ID\_ZAMESTNANEC INTO v\_id FROM ZAMESTNANEC WHERE JMENO = p\_jmeno AND PRIJMENI = p\_prijmeni AND EMAIL = p\_email;

IF (NOT(P\_HESLO is NULL) AND NOT(P\_UZIVJMENO is NULL)) THEN

select standard\_hash(p\_heslo, 'MD5') into v\_pass from dual;

INSERT INTO UDAJE (UZIVATELSKEJMENO, HESLO, ID\_ZAMESTNANEC)

VALUES (p\_uzivJmeno, v\_pass, v\_id);

ELSE

select id\_opravneni into v\_idOpr from opravneni where opravneni = 'Neregistrovaný';

update zamestnanec

set id\_opravneni = v\_idOpr

where id\_zamestnanec = v\_id;

END IF;

END;

1. Upravzamestnance

Upraví uživatele. Pokud jsou uživatelské jméno a heslo jiné než null, znamená to změnu těchto informací a ty se pokusí změnit pokud existují, nebo vytvořit pokud neexistují. Touto metodou nelze heslo nebo jméno smazat.

create or replace PROCEDURE upravZamestnance

(p\_id NUMBER, p\_jmeno VARCHAR2, p\_prijmeni VARCHAR2, p\_titulPred VARCHAR2, p\_titulZa VARCHAR2, p\_email VARCHAR2,

p\_zkratkaKatedry VARCHAR2, p\_opravneni NUMBER, p\_idRole NUMBER, p\_mobil NUMBER, p\_telefon NUMBER,

p\_uzivJmeno VARCHAR2, p\_heslo VARCHAR2)

IS

v\_pass VARCHAR2(50);

v\_idUdaje NUMBER;

BEGIN

UPDATE ZAMESTNANEC

SET JMENO = p\_jmeno, PRIJMENI = p\_prijmeni, TITUL\_PRED = p\_titulPred, TITUL\_ZA = p\_titulZa, TELEFON = p\_telefon, MOBIL = p\_mobil,

EMAIL = p\_email, KATEDRA\_ZKRATKA\_KATEDRY = p\_zkratkaKatedry, ID\_OPRAVNENI = p\_opravneni, ID\_ROLE = p\_idRole

WHERE ID\_ZAMESTNANEC = p\_id;

IF (NOT(P\_HESLO is NULL) AND NOT(P\_UZIVJMENO is NULL)) THEN

begin

select standard\_hash(p\_heslo, 'MD5') into v\_pass from dual;

select ID\_UDAJE into v\_idUdaje from udaje where ID\_ZAMESTNANEC = p\_id;

UPDATE UDAJE

SET UZIVATELSKEJMENO = p\_uzivJmeno, HESLO = v\_pass

WHERE ID\_ZAMESTNANEC = p\_id;

exception --pokud udaje existují změní je, pokud ne (uživatel byl původně neregistrovaný) přidá je

when no\_data\_found then

INSERT INTO UDAJE (UZIVATELSKEJMENO, HESLO, ID\_ZAMESTNANEC)

VALUES (p\_uzivJmeno, v\_pass, p\_id);

end;

END IF;

END;

## Spouště

1. Triggery pro kaskádové mazání informací pro každou tabulku vyskytující se u jiné tabulky jako FK

## Indexy

1. DATA\_\_IDX
2. PRIHLASOVACIJMENO\_UNIQUE
3. UCEBNANAZEV\_UNIQUE
4. UDAJE\_\_IDX
5. ZAMESTNANECEMAIL\_UNIQUE

## Sekvence

1. Sekvence automaticky vytvořené pro IDENTIFYING sloupce umělých klíčů.

# Aplikace

## Použité prostředí

K realizaci aplikace je využito prostředí NetBeans, jazyk Java. Pro grafické uživatelské prostředí (dále GUI) je využita technologie JavaFX.

## Řízení uživatelských účtů

Aplikace rozlišuje 3 úrovně oprávnění.

1. Prohlížení

V tomto režimu může uživatel prohlížet hodnoty v databázi, rozvrhy atp.

1. Uživatel

V tomto režimu může uživatel prohlížet data v databázi a upravovat své údaje.  
Pokud je vyučující, tak své rozvrhové akce, každý uživatel pak své osobní informace.

1. Administrátor

Může prohlížet data z aplikace, měnit data ve všech tabulkách, schvalovat rozvrhové akce a importovat z XLM a CSV.

## Moduly

* ISkolniDB.java

Interface pro SkolniDB.

* SkolniDB.java

Implementuje rozhraní ISkolniDB. Slouží pro komunikaci GUI s databází. Nachází se zde metody pro vkládání, úpravu a odebírání dat z DB. Naprostá většina komunikace s DB je řešena zde.

* CsvReader.java

Třída pro import dat z CSV. Po vytvoření instance třídy lze použít metodu na import, která vrací list hodnot z CSV.

* XMLLoader.java

Třídní procedurou na import (kde vstup je cesta k souboru) parsuje vybraný XML soubor a ukládá hned do DB.

* Formuláře

Popsané níže

## Formuláře

Ve všech formulářích je tlačítko „Ok“ commit a tlačítko „Zrušit“ rollback.

* Hlavní menu

GuiFXML.fxml, GuiFXMLController.java

Slouží k základní orientaci po programu a otevření většiny hlavních formulářů.

Registrovaný uživatel má přímo z menu možnost editovat (některé) své informace v DB.

* Vyučující

FXMLVyucujici.fxml, FXMLVyucujiciController.java

Vypisuje vyučující, nebo zaměstnance (podle toho pod jakým tlačítkem se přistupovalo) do tabulky včetně fotky má-li nějakou přiřazenu. Admin má možnost přidat, upravit nebo smazat zaměstnance, přidat anebo smazat zaměstnanci fotku. Pokud formulář ukazuje jen vyučující, může jakýkoliv uživatel zobrazit rozvrhové akce vyučujícího, nebo jaké předměty vyučující vyučuje.

* Předměty

Vypisuje předměty. Admin může předměty přidat, upravit, odebrat nebo importovat (XML). Uživatel může přejít tlačítkem „Obory“ do formuláře zobrazujícího ve kterých oborech a s jakou prioritou je předmět zasazen. Tlačítko Rozvrhové akce zobrazí všechny rozvrhové akce daného předmětu.

* Obory

Vypisuje obory. Admin může obory přidávat, upravovat a odebírat. Uživatel může tlačítkem „Předměty oboru“ vyvolat formulář zobrazující jaké předměty jsou oboru přiřazeny a s jakou prioritou.

* Předměty oboru (obory předmětu)

Tento formulář není přístupný z menu. Podle toho z jakého formuláře je spuštěn ukazuje buď do jakých oborů spadá daný předmět, nebo jaké předměty má daný obor. Admin může tato spojení přidávat, upravovat nebo odebírat.

* Pracoviště

Tento formulář vypisuje katedry a fakulty pod níž spadají. Admin může katedry přidat, upravit, odebrat a importovat (CSV). Uživatel může přejít na formulář vypisující pouze zaměstnance katedry.

* Rozvrhové akce

Tento formulář není zcela přístupný z menu. Podle toho, zda je přistupován z formuláře vyučujících, předmětů či učeben vypisuje rozvrhové akce vyučujícího, předmětu či učeben. Administrátor vidí i neschválené akce, uživatel jen schválené. Admin může rozvrhové akce přidávat, upravovat, odebírat a schvalovat. Pokud jsou rozvrhové akce filtrovány dle vyučujícího nebo učebny, lze zobrazit grafický rozvrh vyučujícího nebo učebny.

Speciálně lze na tento formulář přistoupit z menu, pokud je přihlášený uživatel vyučující. Pak je mu tento formulář zpřístupněn s možností přidávání, úpravy a odebírání rozvrhových akcí, schvalovat akce však musí admin. Vyučující zde vidí i své neschválené rozvrhové akce.

* Učebny

Výpis učeben. Admin může učebny přidávat, upravovat a odebírat. Uživatel může přejít na okno rozvrhových akcí a podívat se, jaké akce v učebně jsou.

* Přihlášení

Tento formulář se zobrazí po připojení do databáze a může být znovu vyvolán z menu (pro změnu uživatele). Slouží k identifikaci uživatele a jeho oprávnění. Pro přístup bez oprávnění je potřeba kliknout na tlačítko „Prohlížet“.

* Import

Formulář pro import ze souboru. Umožňuje zvolit příponu a pak kde se soubor nachází (pak automaticky provede import). Na formuláři jsou tlačítka pro commit a rollback.

* Dialogy
  + DialogChyba.java

Tímto dialogem je ošetřena většina výjimek, ke kterým v programu dochází (dialogem nejsou odchytávány výjimky SQL při importu, neboť by mohl dialog vyskakovat pro každý importovaný řádek, nebo by první neúspěšný řádek přerušil celý import. Při importu se dá očekávat SQL výjimka porušení jedinečnosti).

Dialog je tvořen pouze textovým polem kde je výjimka vypsána a tlačítkem pro zavření dialogu.

* + DialogCommitRollback.java

Při pokusu opustit formulář jiným způsobem než pomocí tlačítek „Ok“, „Zrušit“ nebo ukončením aplikace se otevře tento dialog, který se ptá, co se změnami (či zůstat na stránce, commitovat změny nebo změny rollbackovat). Tento dialog se nezobrazí, pokud v aktivním formuláři nebyly vykonané změny.

* + DialogZobrazRAGraficky.java

Dialog je přístupný pomocí tlačítka z formuláře rozvrhových akcí, pokud je filtrován, aby ukazoval akce vyučujícího či učebny. V dialogu jsou zobrazeny jednotlivé akce, v jaké jsou učebně/kdo je vyučuje, o jaký předmět se jedná a od kdy do kdy akce probíhá.

* + DialogPripojeni.java

Jednoduchý dialog pro připojení do Oracle databáze.

* + Dialogy pro přidání/úpravu

Všechny dialogy pro přidávání se zároveň používají pro úpravu a jsou ve svém principu stejné. Pro editaci textových polí je použit TextField (vyjma telefonních čísel, která jsou čísla), pro editaci číselných hodnot je použit Spinner a pro úpravu hodnot sloužících jako FK je použit ComboBox.

## Orientace ve zdrojovém kódu

Po přihlášení uloží ID uživatele a dle toho nastaví režim programu (proměnná druhProhlizeni defaultně nastavená na „NEREGISTROVANY“ nacházející se v třídě IDAS22018.java). Po přihlášení se aktivuje hlavní menu (dle typu prohlížení a toho zda je uživatel vyučující), odkud se přechází do jednotlivých formulářů, kde jsou připraveny tradiční metody pro přidání, úpravu, odebrání a filtraci.

**Přílohy**

1. Skripty SQL
2. Zdrojové kódy aplikace, grafika, apod. vše v jednom zabaleném ZIPU!
3. Samostatně spustitelná aplikace.
4. Ostatní (libovolné soubory doplňující Vaši tvorbu databázové aplikace či nutné nestandardní doplňky nutné pro spuštění Vaší aplikaci)