

Návrhová dokumentace

Projekt volební systém

Vypracoval Petr Hlaváček, Ondřej Kulatý, Štěpán Škorpil a Václav Tarantík

Obsah:

Balíček Class Model	3
Balíček IntegrationTier	3
ActionDAO	4
JDBCManager.....	4
ParticipantDAO	4
UserDAO	4
Balíček PresentationTier	4
ChangeListener.....	5
ChangeSource	5
Globlas	7
Balíček Controller.....	8
Balíček actions	8
Balíček dialogs	9
Balíček view	9
LoginSreen	10
MainFrame	10
Balíček AbstractComponents	10
Balíček admin	11
Balíček components	11
Balíček registrator.....	12
Balíček structure model.....	12
Balíček core	13
SystemRegException	13
Balíček Entities.....	13
Action	14
Additional.....	16
Participant.....	16
Presence	18
User.....	20
Balíček Enums.....	21
Role	21
Balíček business_tier	22
IActionFacade	22
IParticipantFacade	24
IUserFacade.....	28
Balíček Utils	31
CalendarUtil.....	31
Generator	32
ImportUtil	32

Balíček Class Model

Balíček s modely balíčků a tříd, které se nacházejí v aplikaci registrační systém. V systému se zatím nacházejí 2 vrstvy a to datová v balíčku data_tier a business logika v balíčku business_tier.

Diagram: Class Model

Created By: Lahvi on 19.11.2005

Last Modified: 20.12.2011

Popis:

Model všech balíčků a jejich tříd dohromady. Můžeme zde vidět, že balíček [core](#), který obsahuje třídy obecné výjimky a balíček s entitami používá celá aplikace (DAO třídy, business třídy a i view s controllerem) Business vrstva používá DAO třídy z integrační vrstvy.

Prezentční vrstva používá business vrstvu, view používá globální třídu a konroler a kontroler používá globální třídu.

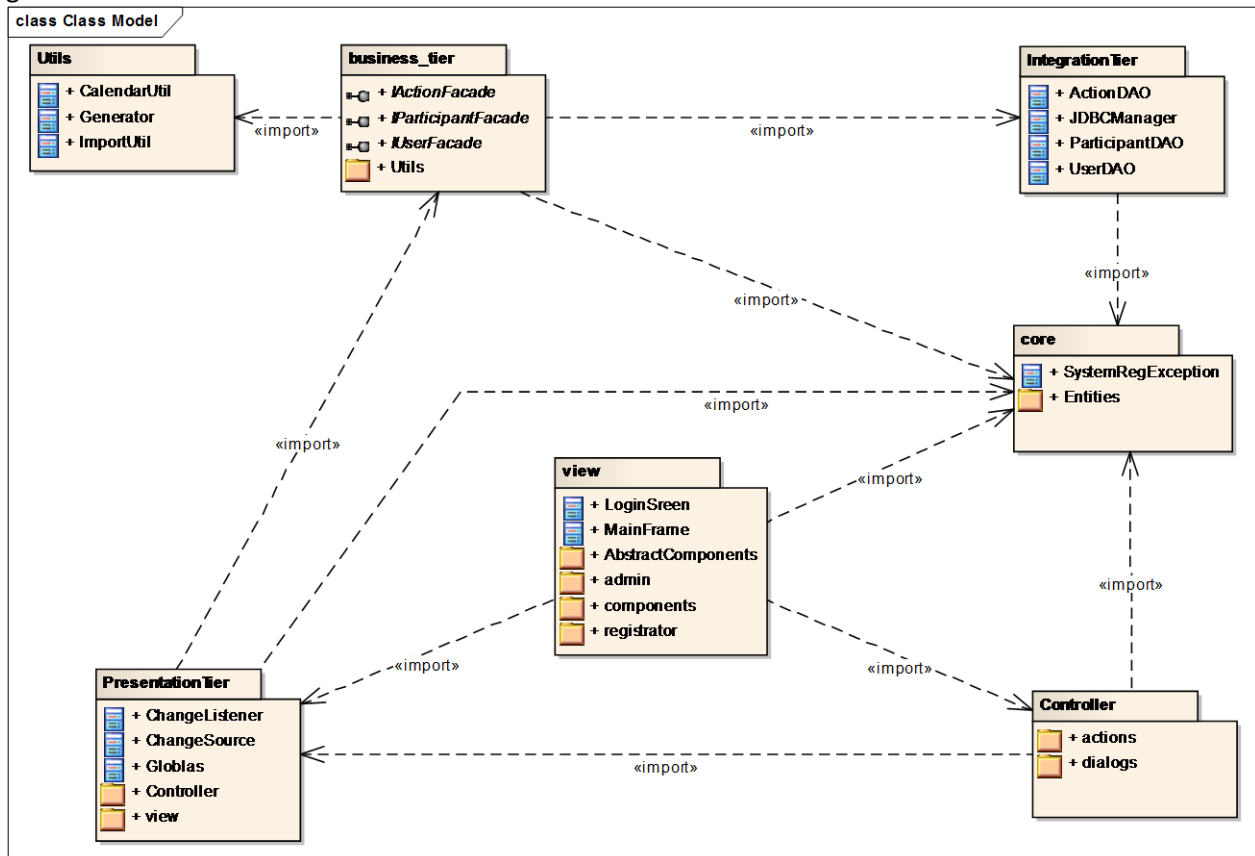


Figure: Diagram Class Model

Balíček IntegrationTier

Balíček obsahuje vrstvu pro práci databází. Nachází se zde třída spravující obecnou práci s databází [JDBCManager](#) a DAO třídy pro práci nad databází s jednotlivými entitami.

Diagram: IntegrationTier

Created By: Lahvi on 20.12.2011

Last Modified: 20.12.2011

Popis:

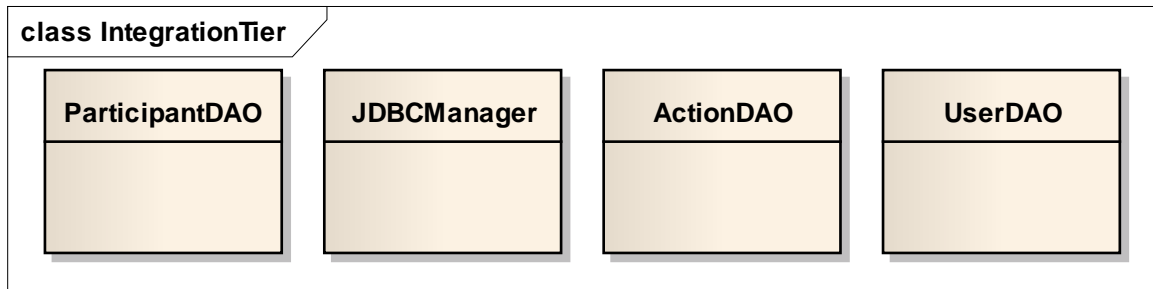


Figure: Diagram IntegrationTier

ActionDAO

Popis:

Třída pro práci s akcemi nad databází. Obsahuje metody, které jsou volány business třídami, pro různé dotazy nad tabulkami.

JDBCManager

Popis:

Třída poskytuje různé metody pro práci s databází, jako vytváření a uzavírání spojení, či resultSetů. Poskytuje funkce pro zápis do logu a funguje jako factory pro DAO třídy.

ParticipantDAO

Popis:

Třída pro práci s účastníky nad databází. Obsahuje metody, které jsou volány business třídami, pro různé dotazy nad tabulkami.

UserDAO

Popis:

Třída pro práci s uživateli nad databází. Obsahuje metody, které jsou volány business třídami, pro různé dotazy nad tabulkami.

Balíček PresentationTier

Diagram: PresentationTier

Created By: Lahvi on 20.12.2011

Last Modified: 20.12.2011

Popis:

Diagram zobrazuje vazbu posluchačů na globální třídu a také balíčky s view a controllerem.

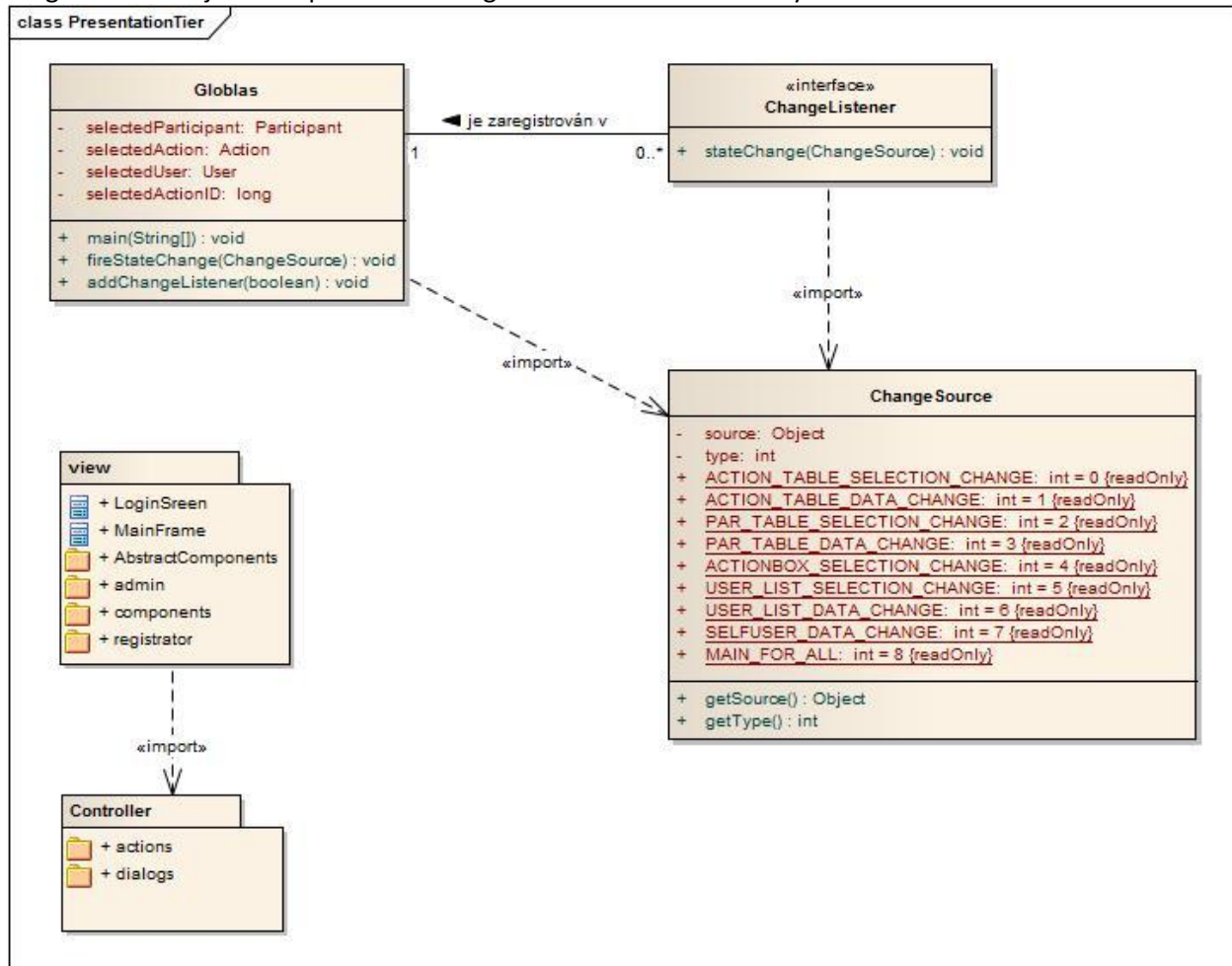


Figure: Diagram PresentationTier

ChangeListener

«interface»

Popis:

Rozhraní poskytující jedinou metodu, ve které třídy, které rozhraní budou implementovat, budou mít reakci na vyvolanou aktualizaci.

Metoda: stateChange Public

Popis: V metodě bude naimplementována reakce na danou aktualizaci. Podle parametru, který udává zdroj a typ aktualizace se může reakce lišit.

Parametr: s

Typ: ChangeSource

Popis: Zdroj vyvolané aktualizace.

ChangeSource

Popis:

Třída která obsahuje informaci o typu a zdroji vyvolané aktualizace. Typy se dělí na změnu vybrání, změnu dat či globální akci.

Atribut: source Private

Typ: Object

Popis: Objekt, který aktualizaci vyvolal.

Atribut: type Private

Typ: int

Popis: Druh aktualizace.

Atribut: ACTION_TABLE_SELECTION_CHANGE Public Static

Typ: int

Popis: Příznak pokud se na tabulce akcí změnila vybraná akce.

Atribut: ACTION_TABLE_DATA_CHANGE Private Static

Typ: int

Popis: Příznak pokud se na tabulce akcí změnil model nějaké akce.

Atribut: PAR_TABLE_SELECTION_CHANGE Private Static

Typ: int

Popis: Příznak pokud se na tabulce účastníků změnilo vybrání

Atribut: PAR_TABLE_DATA_CHANGE Private Static

Typ: int

Popis: Příznak pokud se na tabulce účastníků změnil model.

Atribut: ACTIONBOX_SELECTION_CHANGE Private Static

Typ: int

Popis: Příznak pokud se změnilo vybrané ID akce v comboBoxu dostupných akcí.

Atribut: USER_LIST_SELECTION_CHANGE Private Static

Typ: int

Popis: Příznak pokud se změnil vybraný uživatel v seznamu uživatelů.

Atribut: USER_LIST_DATA_CHANGE Private Static

Typ: int

Popis: Příznak pokud se změnila data uživatele v seznamu uživatelů.

Atribut: SELFUSER_DATA_CHANGE Private Static

Typ: int

Popis: Příznak pokud uživatel editoval vlastní údaje.

Atribut: MAIN_FOR_ALL Private Static

Typ: int

Popis: Příznak změny, na kterou by měli reagovat všichni posluchači.

Metoda: getSource Public

Popis: Vrací objekt, který změnu vyvolal.

Metoda: getType Public

Popis: Vrací typ změny.

Globlas

Popis:

Vstupní bod aplikace. Odtud se aplikace spustí (je zde metoda main). Třída také poskytuje Globální informace pro celou aplikaci (informace o vybrané akci, uživateli, či účastníkovi v tabulkách atd.). K této třídě se také registrují posluchači ze strany view, které zajímají změny datového modelu. Zároveň také poskytuje view třídy s business logikou z balíčku [business tier](#).

Jelikož ke každé globální proměnné je getr a setr ve výpisu funkcí je vynechám.

Atribut: selectedParticipant Private

Typ: Participant

Popis: Účastník, který je vybrán v tabulce účastníků, Pokud není vybrán žádný nebo tabulka není zobrazena je null.

Atribut: selectedAction Private

Typ: Action

Popis: Akce, která je vybrána v tabulce akcí, Pokud není vybrána žádná nebo tabulka není zobrazena je null.

Atribut: selectedUser Private

Typ: User

Popis: Uživatel, který je vybrán v seznamu uživatelů. Pokud není vybrán žádný nebo pokud není seznam dostupný tak je null.

Atribut: selectedActionID Private

Typ: long

Popis: Vybrané ID akce, v combo boxu u tabulky účastníků. Podle tohoto ID se načítají účastníci v jednotlivých tabulkách.

Metoda: main Public

Popis: Metoda, která spouští aplikaci.

Parametr: args

Typ: String[]

Popis: Pole s argumenty pro spuštění.

Metoda: fireStateChange Public

Popis: Metoda vyvolá aktualizace u všech zaregistrovaných posluchačů.

Parametr: s*Typ:* ChangeSource*Popis:* Zdroj vyvolané aktualizace.**Metoda: addChangeListener** Public*Popis:* Přidává nového posluchače.**Parametr: ChangeListener***Typ:* boolean*Popis:* Nově registrovaný posluchač aktualizací.

Balíček Controller

Balíček obsahuje akce a dialogy, které plní funkce jakéhosi kontroleru aplikace. Kontroler získává od globální třídy business funkce, pomocí kterých pak mění datový model aplikace.

Diagram: Controller

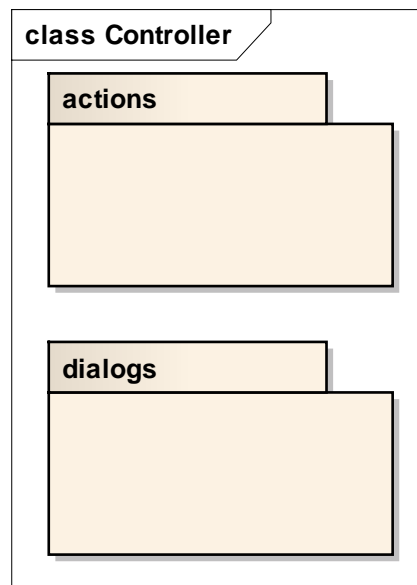
Created By: Lahvi on 20.12.2011*Last Modified:* 20.12.2011*Popis:*

Figure: Diagram Controller

Balíček actions

Zde se nachází akce, použité v menu, či v tlačítkách po celé aplikaci. Některé akce registrují u globální třídy.

Balíček dialogs

Různé dotazovací, vytvářecí upravovací dialogy.

Balíček view

Balíček obsahuje grafické komponenty k aplikaci. Jsou zde hlavní komponenty jako přihlašovací obrazovka nebo hlavní okno. Dále jsou zde abstraktní komponenty, komponenty které implementují ty grafické a komponenty specifické pro danou roli.

Diagram: view

Created By: Lahvi on 20.12.2011

Last Modified: 20.12.2011

Popis:

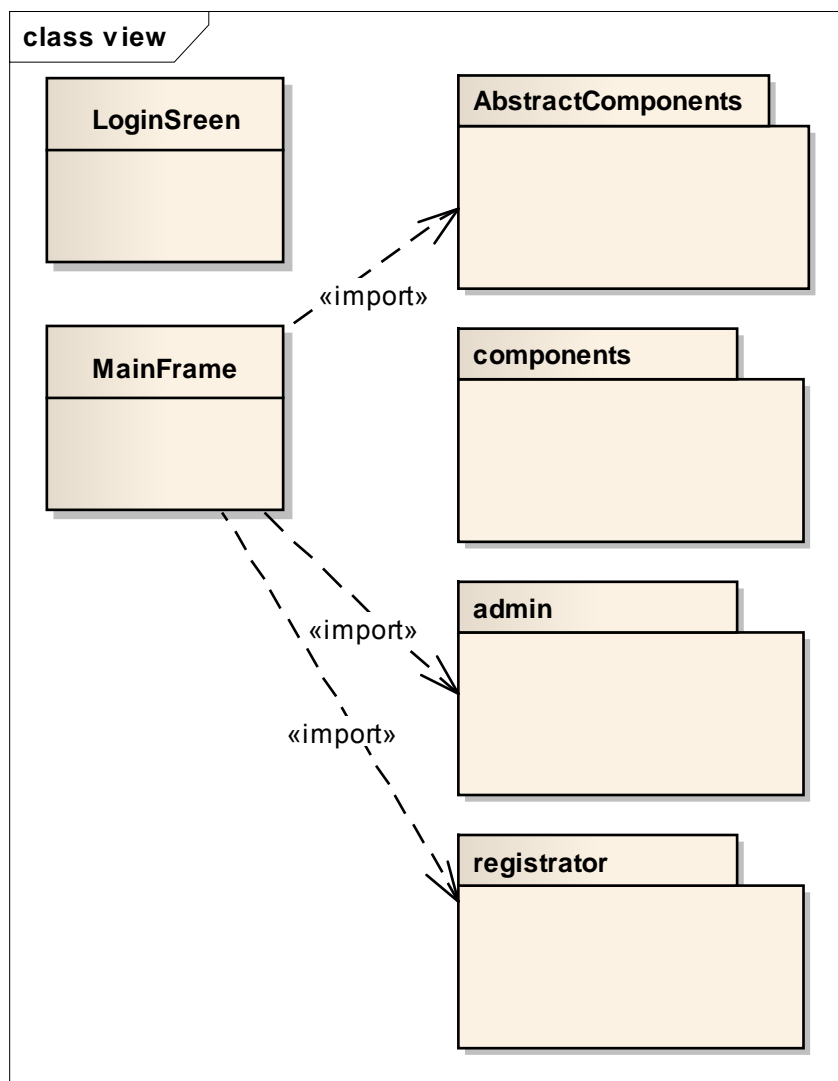


Figure: Diagram view

LoginScreen

Popis:

Přihlašovací obrazovka

MainFrame

Popis:

Hlavní panel aplikace.

Balíček AbstractComponents

Zde jsou abstraktní komponenty, jako abstraktní tabulka či panel pro tabulky.

Balíček admin

Balíček obsahuje komponenty pro administrátorské rozhraní.

Balíček components

Balíček obsahuje komponenty, které jsou například zděděny od abstraktních komponent či grafických prvků swing.

Balíček registrator

Balíček obsahuje grafické komponenty pro registrátorskou část aplikace.

Balíček core

Balíček core, obsahuje třídu pro zabalení výjimky pro celou aplikaci do jednoho typu a také balíček s entitami.

Diagram: core

Created By: Lahvi on 2.11.2011

Last Modified: 20.12.2011

Popis:

Primitivní diagram balíčku core. Je zde pouze třída výjimky a balíček entit.

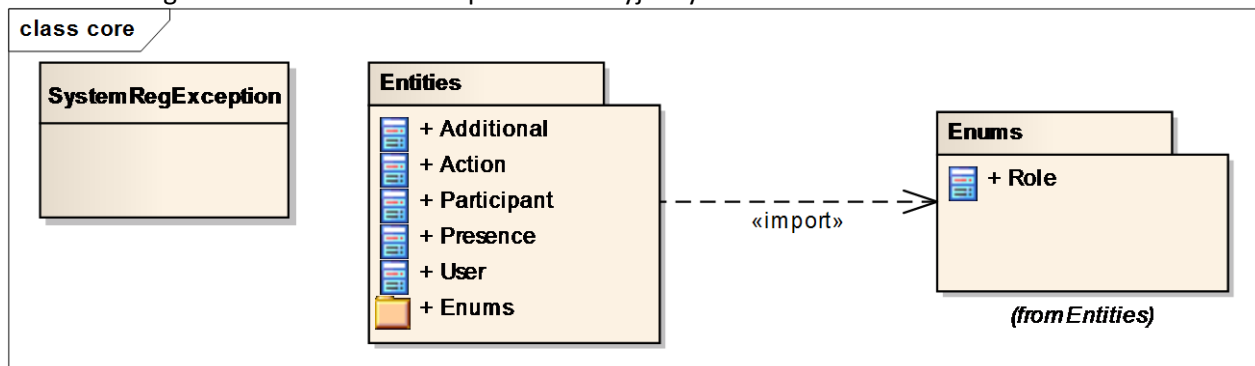


Figure: Diagram core

SystemRegException

Popis:

Balíček Entities

Balíček entites obsahuje základní entity, které se vyskytují v systému.

Diagram: Entities

Created By: Lahvi on 2.11.2011

Last Modified: 20.12.2011

Popis:

Diagram balíčku Entities. Zde se nacházejí základní entity, které se v našem systému vyskytují, a balíček kde je výčet Role.

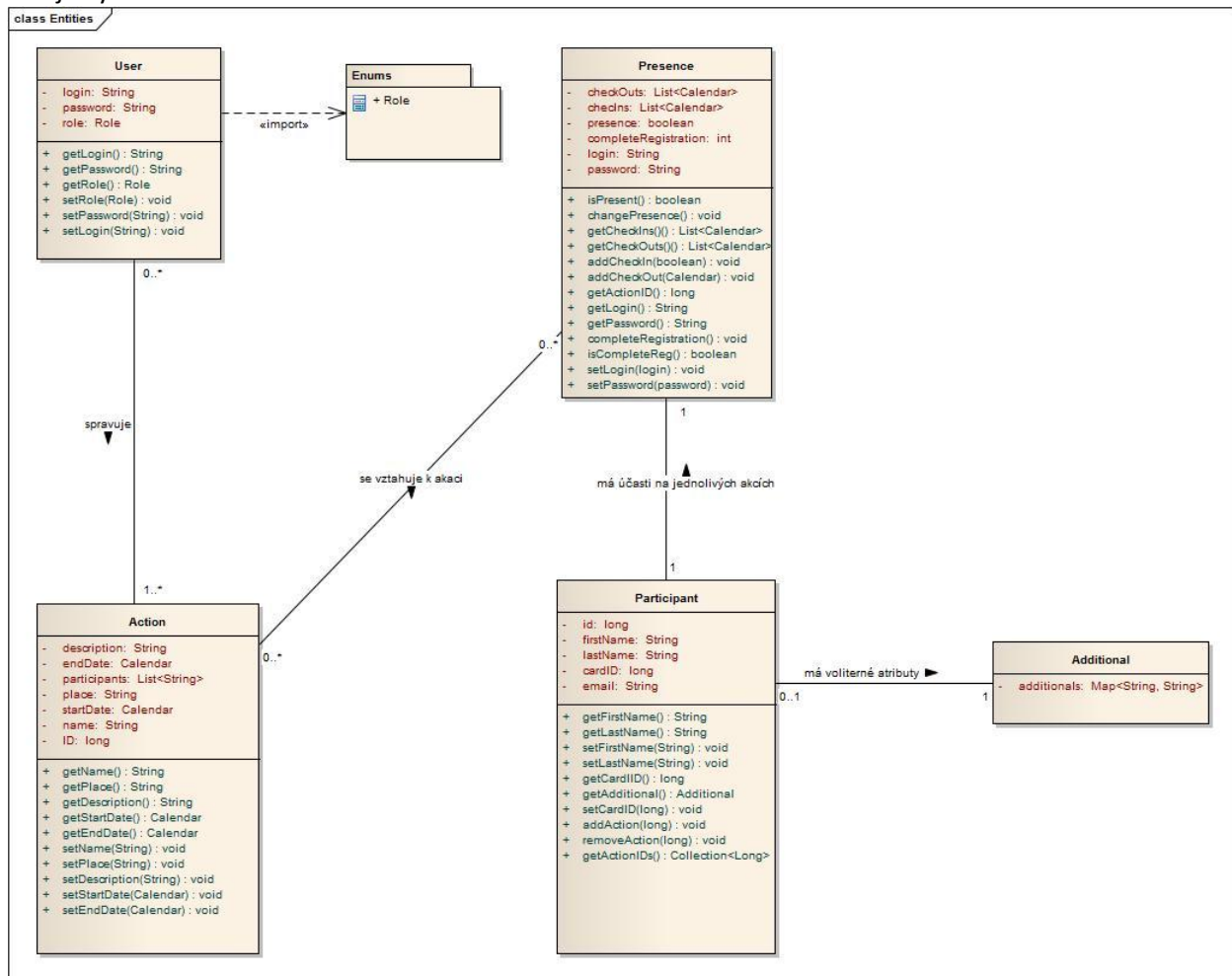


Figure: Diagram Entities

Action

Popis:

Entita reprezentuje jednotlivou volební akci. K těmto akcím pak mají účastníci nastavenou účast a uživatelé je mohou spravovat, tzn. že mají právo spravovat účastníky, kteří se dané akce účastní atd. S entitou jsou tedy asociovány entity [Participant](#) a [User](#). Akce musí být povinně asociována alespoň s jedním uživatelem (minimálně s tím, který akci vytvořil), ale s účastníky je vazba 0..*.

Akce má jako parametry název, místo konání, popis (nepovinný) a datum zahájení a ukončení dané akce.

Atribut: description Private

Typ: String

Popis: Stručný popis akce.

Atribut: endDate Private

Typ: Calendar

Popis: Datum a čas ukončení akce.

Atribut: participants Private

Typ: List<String>

Popis: Seznam loginů účastníků, kteří se na akci přihlásili.

Atribut: place Private

Typ: String

Popis: Místo konání akce.

Atribut: startDate Private

Typ: Calendar

Popis: Datum a čas zahájení akce.

Atribut: name Private

Typ: String

Popis: Název akce.

Atribut: ID Private

Typ: long

Popis: ID dané akce.

Metoda: getName Public

Popis: Vrací název akce.

Metoda: getPlace Public

Popis: Vrací místo konání dané akce.

Metoda: getDescription Public

Popis: Vrací stručný popis akce.

Metoda: getStartDate Public

Popis: Vrací datum a čas zahájení akce.

Metoda: getEndDate Public

Popis: Vrací datum a čas ukončení akce.

Metoda: setName Public

Popis: Metoda nastavuje dané akci nové jméno, které obdrží v parametru name.

Parametr: name*Typ:* String*Popis:* Nové jméno akce.**Metoda: setPlace** Public*Popis:* Metoda nastavuje nové místo konání akce, Nová hodnota je předána parametrem place.**Parametr: place***Typ:* String*Popis:* Nové místo konání akce.**Metoda: setDescription** Public*Popis:* Nastavuje nový popis akce, který se získá z parametru desc.**Parametr: desc***Typ:* String*Popis:* Nový popis akce.**Metoda: setStartDate** Public*Popis:* Nastavuje nový datum a čas zahájení dané akce. Nové hodnotu se získají parametrem startDate.**Parametr: startDate***Typ:* Calendar*Popis:* Nový datum a čas zahájení akce.**Metoda: setEndDate** Public*Popis:* Nastavuje nový datum a čas ukončení dané akce. Nové hodnoty jsou předány parametrem endDate.**Parametr: endDate***Typ:* Calendar*Popis:* Nový datum a čas ukončení akce**Additional***Popis:*

Tato entita reprezentuje volitelné parametry jednotlivých účastníků. Má v sobě mapu kde klíč je název volitelné položky a hodnota je hodnota volitelného atributu.

Entita je vázána na účastníka. Účastník má jednu nebo žádnou instanci entity additional, entita může patřit právě jednomu účastníkovi.

Atribut: additional Private*Typ:* Map<String, String>*Popis:* Mapa, kde klíč je název atributu a hodnota je hodnota volitelného atributu. Jak klíč tak hodnota jsou řetězce.**Participant**

Popis:

Účastník má v sobě základní údaje jako křestní jméno, příjmení, email a číslo rodné nebo OP. Účastník je identifikován svým jedinečným ID.

Účastník je asociován s entitou [Presence](#) a to konkrétně tak, že pro každou akci na které je účastník asociován má také jednu instanci entity presence.

Atribut: id Private

Typ: long

Popis:

Atribut: firstName Private

Typ: String

Popis: Křestní jméno účastníka.

Atribut: lastName Private

Typ: String

Popis: příjmení účastníka

Atribut: cardID Private

Typ: long

Popis: Rodné číslo nebo číslo občanského průkazu.

Atribut: email Private

Typ: String

Popis: Email účastníka.

Metoda: getFirstName Public

Popis: Vrací křestní jméno daného účastníka.

Metoda: getLastName Public

Popis: Vrací příjmení účastníka.

Metoda: setFirstName Public

Popis: Metoda nastavuje uživateli nové křestní jméno, které získá z parametru name.

Parametr: name

Typ: String

Popis: Nové křestní jméno účastníka.

Metoda: setLastName Public

Popis: Nastaví nové příjmení účastníka.

Parametr: lastName

Typ: String

Popis: Nové příjmení účastníka.

Metoda: getCardIID Public

Popis: Vrací číslo rodné/OP daného účastníka.

Metoda: getAdditional Public

Popis: Vrací objekt s nepovinnými parametry. Pokud účastník žádné volitelné parametry, vrátí null.

Metoda: setCardID Public

Popis: Nastavuje nové číslo OP / rodné číslo.

Parametr: cardID

Typ: long

Popis: Nové číslo OP, nebo rodné číslo.

Metoda: addAction Public

Popis: Přidá účastníkovi účast na nové akci.

Parametr: actionID

Typ: long

Popis: ID akce, pro kterou se vytváří nová účast.

Metoda: removeAction Public

Popis: Odebere účastníkovi účast na dané akci.

Parametr: actionID

Typ: long

Popis: ID odebírané akce.

Metoda: getActionIDs Public

Popis: Vrátí seznam s ID všech akcí, kterých se účastník účastní.

Presence

Popis:

Entita přítomnost reprezentuje zaregistrování účastníka na jednotlivé akci. Jelikož účastník může mít účast na více akcí, je s ním entita přítomnost asociována poměrem OneToMany, tzn každá přítomnost patří právě jednomu účastníkovi, který ovšem může mít 1..* účastí.

K účasti na dané akci, se vztahují přihlašovací údaje heslo a login, příznaky o přítomnosti a dokončení registrace a seznamy příchodů a odchodů.

Atribut: checkOuts Private

Typ: List<Calendar>

Popis: Seznam odchodů daného účastníka. Při každém odchodu se sem zaznamená čas a datum odchodu.

Atribut: checIns Private

Typ: List<Calendar>

Popis: Seznam s příchody daného účastníka. Při každém příchodu na akci, se do něj účastníkovi zaznamená datum a čas příchodu.

Atribut: presence Private

Typ: boolean

Popis: Příznak jestli je účastník právě přítomen.

Atribut: completeRegistration Private

Typ: int

Popis: Příznak jestli má účastník dokončenou registraci.

Atribut: login Private

Typ: String

Popis: Přihlašovací jméno. Dokud nemá účastník dokončenou registraci je null.

Atribut: password Private

Typ: String

Popis: Heslo. Dokud nemá účastník dokončenou registraci je null.

Metoda: isPresent Public

Popis: Vrací true nebo false, podle toho jestli je uživatel na akci přítomen právě teď.

Metoda: changePresence Public

Popis: Změní příznak přítomnosti. Pokud je teď účastník přítomen nastaví se na false a naopak.

Metoda: getCheckIns() Public

Popis: Vrací seznam příchodů daného uživatele.

Metoda: getCheckOuts() Public

Popis: Vrací seznam odchodů.

Metoda: addCheckIn Public

Popis: Přidá nový datum a čas příchodu.

Parametr: checkIn

Typ: boolean

Popis: Datum a čas příchodu.

Metoda: addCheckOut Public

Popis: Přidá nový datum a čas odchodu.

Parametr: checkOut

Typ: Calendar

Popis: Čas odchodu

Metoda: getActionID Public

Popis: Vrací ID akce, ke které se daná účast vztahuje.

Metoda: getLogin Public

Popis: Vratí login účastníka.

Metoda: getPassword Public

Popis: Vratí heslo účastníka.

Metoda: completeRegistration Public

Popis: Nastaví příznak o kompletnosti registrace na true.

Metoda: isCompleteReg Public

Popis: Vrací příznak o tom, zda má účastník dokončenou registraci k dané akci, ke které se účast vztahuje.

Metoda: setLogin Public

Popis: Nastaví login daného účastníka, pro akci, ke které se účast vztahuje.

Parametr: String

Typ: login

Popis: Login účastníka.

Metoda: setPassword Public

Popis: Nastaví heslo daného účastníka, pro akci, ke které se účast vztahuje.

Parametr: String

Typ: password

Popis: Heslo účastníka.

User

Popis:

Entita User, reprezentuje uživatele Registračního systému. Tato entita má tři atributy: Uživatelské jméno, heslo a uživatelskou roli. Role je z výčtu Role, z balíčku Enums. V systému tedy mohou být 3 role: SUPER_ADMIN, ADMIN a REGISTRATOR (seřazeno od nejvyšších práv po nejnižší).

User je asociován s entitou Akce a tím je řečeno jakou akci má daný uživatel právo spravovat. Násobnost je ManyToMany, protože uživatel nemusí právo ani na jednu akci a nebo taky klidně na všechny.

Atribut: login Private

Typ: String

Popis:

Atribut: password Private

Typ: String

Popis: Heslo účastníka.

Atribut: role Private

Typ: Role

Popis: Atribut, který reprezentuje roli uživatele v systému.

Metoda: getLogin Public

Popis: Vrací login usera.

Metoda: getPassword Public

Popis: Vrací heslo usera.

Metoda: getRole Public

Popis: Vrací uživatelskou roli daného uživatele.

Metoda: setRole Public

Popis: Nastaví danému uživateli novou roli.

Parametr: role

Typ: Role

Popis: Nová role uživatele.

Metoda: setPassword Public

Popis: Nastaví danému uživateli nové heslo.

Parametr: password

Typ: String

Popis: Nové heslo.

Metoda: setLogin Public

Popis: Nastaví nový login daného usera.

Parametr: login

Typ: String

Popis: Nové přihlašovací jméno daného uživatele.

Balíček Enums

V balíčku je výčet Role, který používá entita User pro identifikování uživatelské role.

Diagram: Enums

Created By: Lahvi on 2.11.2011

Last Modified: 3.11.2011

Popis:

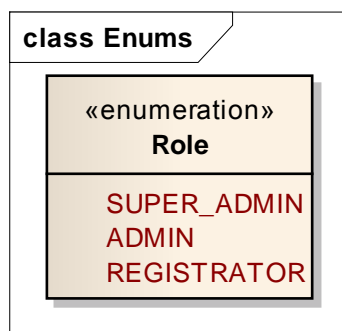


Figure: Diagram Enums

Role

«enumeration»

Popis:

Výčet Role má tři hodnoty, které reprezentují role v registračním systému.

Pro roli Registrátor má výčet hodnotu REGISTRATOR.

Pro roli Administrátor má výčet hodnotu ADMIN.

Pro roli SuperAdmin má výčet hodnotu SUPER_ADMIN.

Atribut: SUPER_ADMIN Public

Typ:

Popis: Hodnota pro roli reprezentující SuperAdmina.

Atribut: ADMIN Public

Typ:

Popis: Hodnota pro roli reprezentující Admina.

Atribut: REGISTRATOR Public

Typ:

Popis: Hodnota pro roli reprezentující Registrátora.

Balíček business tier

V balíčku business_tier bude implementována business logika aplikace. Je to jakási fasáda pro práci s nižší, datovou, vrstvou.

Diagram: business tier

Created By: Lahvi on 2.11.2011

Last Modified: 19.12.2011

Popis:

Diagram business balíčku. Zde se nacházejí interfaci, které používá view pro práci s databází. Je zde interface pro práci s účastníky, uživateli i akcemi. V hotové aplikaci se zde budou ještě nacházet třídy, které budou dané interfaci implementovat.

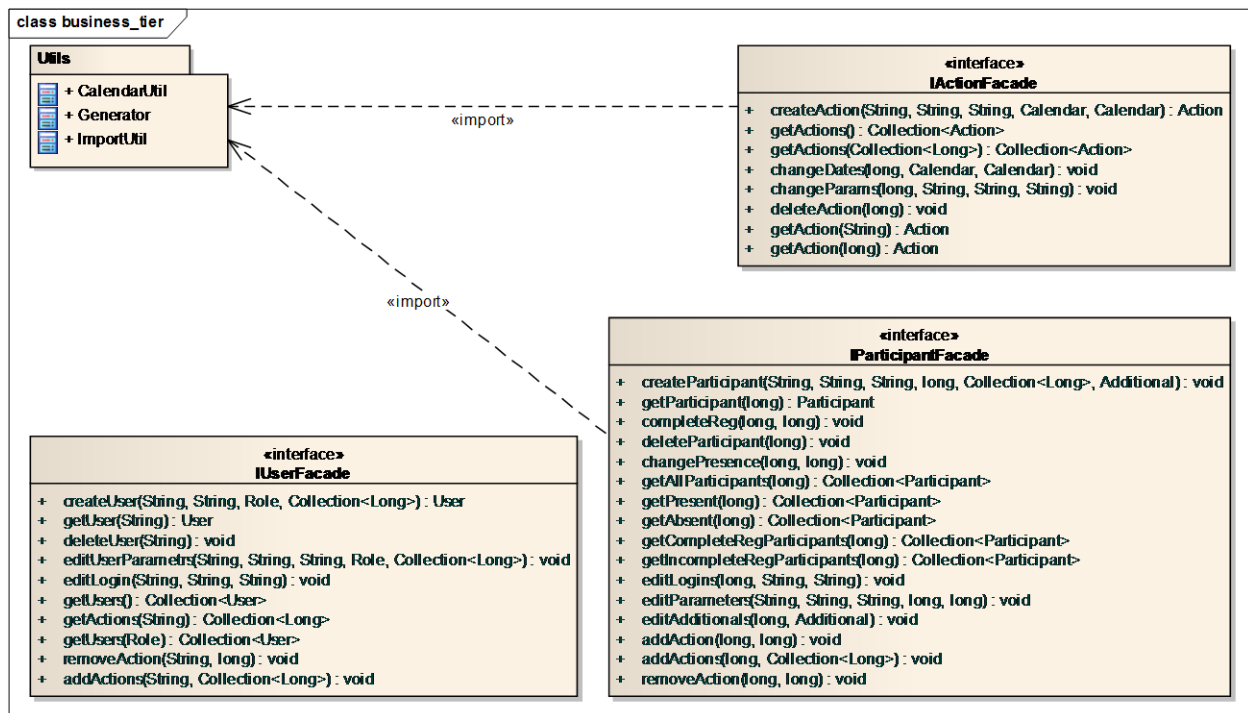


Figure: Diagram business_tier

IActionFacade

«interface»

Popis:

Interface k fasádě pracující s akcemi.

Metoda: createAction Public

Popis: Metoda, která vytváří novu akci. Potřebuje vyžaduje název, místo konání a data zahájení a ukončení. Popis akce může být null. Pro případ potřeby vrátí vytvořenou akci.

Parametr: name

Typ: String

Popis: Název akce.

Parametr: place

Typ: String

Popis: Místo konání akce.

Parametr: desc

Typ: String

Popis: Popis akce. Nepovinné.

Parametr: startDate

Typ: Calendar

Popis: Datum zahájení- akce.

Parametr: endDate

Typ: Calendar

Popis: Datum ukončení- akce.

Metoda: getActions Public

Popis: Vrací všechny vytvořené akce.

Metoda: getActions Public

Popis: Vyhledá a vrátí seznam akcí podle jejich ID, předaných parametrem.

Parametr: actionIDs

Typ: Collection<Long>

Popis: Kolekce IDček vyhledávaných akcí.

Metoda: changeDates Public

Popis: Změní data zahájení a ukončení dané akce, vyhledané podle ID předaném parametrem ID.

Parametr: ID

Typ: long

Popis: ID vyhledávané akce.

Parametr: startDate

Typ: Calendar

Popis: Parametr s novým datem zahájení.

Parametr: endDate

Typ: Calendar

Popis: Parametr se starým datem zahájení.

Metoda: changeParams Public

Popis: Změní parametry akce vyhledané podle jejího ID. Měnit se dá popis, název a místo konání. Popis není povinný parametr.

Parametr: ID

Typ: long

Popis: ID vyhledávané akce.

Parametr: name

Typ: String

Popis: Nový název akce.

Parametr: place

Typ: String

Popis: Nové místo konání dané akce.

Parametr: desc

Typ: String

Popis: Nový popis akce. Nepovinné.

Metoda: deleteAction Public

Popis: Odstraní danou akci, vyhledanou podle jejího ID. Pokud se akce účastní nějakí účastníci tak jim musí být odebrána účast na mazané akci. To samé platí i pro uživatele.

Parametr: ID*Typ:* long*Popis:* ID odstraňované akce.**Metoda: `getAction`** `Public`*Popis:* Vyhledá a vrátí akci podle jejího jména.**Parametr: name***Typ:* String*Popis:* Jméno vyhledávané akce.**Metoda: `getAction`** `Public`*Popis:* Vyhledá a vrátí akci podle jejího ID.**Parametr: ID***Typ:* long*Popis:* ID vyhledávané akce.***IParticipantFacade***

«interface»

Popis:

Interface k fasádě pracující s operacemi nad účastníkem.

Metoda: `createParticipant` `Public`*Popis:* Vytvoří nového účastníka. Povinnými parametry jsou křestní jméno a příjmení, email a číslo OP či rodné číslo. Dále je potřeba kolekce akcí, kterých se bude nový účastník účastnit. Nepovinný parametr je typu [Additional](#) a jsou v něm uloženy nepovinné atributy. Tento parametr tedy může být null.**Parametr: fName***Typ:* String*Popis:* Křestní jméno.**Parametr: IName***Typ:* String*Popis:* Příjmení.**Parametr: email***Typ:* String*Popis:* Email účastníka.**Parametr: personalID***Typ:* long*Popis:* Rodné číslo / číslo OP / jiný identifikátor účastníka.**Parametr: actionIDs***Typ:* Collection<Long>

Popis: Kolekce p^lí^lazen^lch akc^l.

Parametr: adInfo

Typ: Additional

Popis: Nepovíná parametry. M^lž^lí^le b^lí^lt {@code null}.

Metoda: getParticipant Public

Popis: Vyhledá a vrátí účastníka podle jeho ID.

Parametr: id

Typ: long

Popis: ID vyhled^lvan^lho ^lš^lř^lastn^l-ka.

Metoda: completeReg Public

Popis: Zkompletuje registraci pro daného účastníka, vyhledaného podle ID. Kompletace registrace obnáší vygenerování přihlašovacího jména a hesla, zaznamenání příchodu na akci a tudíž také nastavení příznaku přítomnosti na true. Jelikož se tyto údaje vztahují k jednotlivé akci, je potřeba předat i ID akce, pro kterou se registrace kompletuje.

Parametr: id

Typ: long

Popis: ID vyhled^lvan^lho ^lš^lř^lastn^l-ka.

Parametr: actionID

Typ: long

Popis: ID akce pro kterou se registrace kompletuje.

Metoda: deleteParticipant Public

Popis: Odstraní účastníka s daným ID. Tato metoda odstraní účastníka ze všech akcí.

Parametr: id

Typ: long

Popis: ID dan^lho ^lš^lř^lastn^l-ka.

Metoda: changePresence Public

Popis: Změní příznak přítomnosti účastníka vyhledaného podle jeho ID. Jelikož se přítomnost vztahuje k jednotlivé akci je nutné předat i ID dané akce. Pokud je na dané akci účastník zrovna přítomen zaznamená se mu odchod a nastaví příznak o přítomnosti na false a naopak.

Parametr: id

Typ: long

Popis: ID vyhled^lvan^lho ^lš^lř^lastn^l-ka.

Parametr: actionID

Typ: long

Popis: ID akce pro kterou se p^lř^lchod/odchod vztahuje.

Metoda: getAllParticipants Public

Popis: Vrátí všechny účastníky, které se účastní dané akce s daným ID.

Parametr: actionID*Typ:* long*Popis:* ID dané akce, pro kterou chceme vyhledat účastníky.**Metoda: getPresent** Public*Popis:* Vrátí všechny přítomné účastníky na akci s daným ID.**Parametr: actionID***Typ:* long*Popis:* ID akce, pro kterou chceme vyhledat účastníky.**Metoda: getAbsent** Public*Popis:* Vrátí všechny účastníky, kteří nejsou na akci s daným ID zrovna přítomni.**Parametr: actionID***Typ:* long*Popis:* ID akce, pro kterou chceme vyhledat účastníky.**Metoda: getCompleteRegParticipants** Public*Popis:* Vrátí všechny účastníky, kteří mají pro akci s daným ID kompletní registraci.**Parametr: actionID***Typ:* long*Popis:* ID akce, pro kterou chceme vyhledat účastníky.**Metoda: getIncompleteRegParticipants** Public*Popis:* Vrátí všechny účastníky, kteří ještě nemají pro akci s daným ID dokončenou registraci.**Parametr: actionID***Typ:* long*Popis:* ID akce, pro kterou chceme vyhledat účastníky.**Metoda: editLogins** Public*Popis:* Upraví heslo a přihlašovací jméno účastníkovi vyhledanému podle jeho ID. Jelikož se login a heslo vztahují k jednotlivé akci, je nutné předat i ID dané akce.**Parametr: id***Typ:* long*Popis:* ID vyhledávaného účastníka.**Parametr: newLogin***Typ:* String*Popis:* Nový login účastníka.**Parametr: newPassword***Typ:* String*Popis:* Nové heslo účastníka.**Metoda: editParameters** Public*Popis:* Upraví povinné parametry daného účastníka vyhledaného podle jeho ID. Povinné parametry jsou

křestní jméno a příjmení, email a číslo rodné / OP.

Parametr: email

Typ: String

Popis: Nová email.

Parametr: firstName

Typ: String

Popis: Nová křestní jméno.

Parametr: lastName

Typ: String

Popis: Nová příjmení.

Parametr: cardID

Typ: long

Popis:

Parametr: id

Typ: long

Popis: ID vyhledávaného účastníka.

Metoda: editAdditional Public

Popis: Upraví nepovinné parametry účastníka vyhledaného podle jeho ID.

Parametr: id

Typ: long

Popis: ID vyhledávaného účastníka.

Parametr: adParams

Typ: Additional

Popis: Tvoří se s nepovinnými parametry.

Metoda: addAction Public

Popis: Účastníkovi s daným ID přidá účast na akci jejíž ID je ve druhém parametru.

Parametr: id

Typ: long

Popis: ID vyhledávaného účastníka. Pravděpodobně použito jen v metodě {

Parametr: actionID

Typ: long

Popis: ID přidávaného akce.

Metoda: addActions Public

Popis: Aktualizuje danému účastníkovi seznam akcí, kterých se účastní. Pro odebrané se musí zrušit přítomnost a pro přidání vytvořit nová.

Parametr: id

Typ: long

Popis: ID vyhledávaného účastníka.

Parametr: actions*Typ:* Collection<Long>*Popis:* Kolekce aktualizovaných akcí.**Metoda: removeAction** Public*Popis:* Danému účastníkovi se odebere účast na dané akci.**Parametr: id***Typ:* long*Popis:* ID vyhledávaného účastníka.**Parametr: actionID***Typ:* long*Popis:* ID odebrané akce.**IUserFacade**

«interface»

Popis:

Interface k fasádě pracující s operacemi nad uživatelem.

Metoda: createUser Public*Popis:* Vytvoří a vrátí nového uživatele. K vytvoření je potřeba login, heslo a seznam akcí, které bude moci uživatel spravovat.**Parametr: login***Typ:* String*Popis:* Login nového účastníka.**Parametr: password***Typ:* String*Popis:* Heslo nového účastníka.**Parametr: role***Typ:* Role*Popis:* Role nového účastníka.**Parametr: actionIDs***Typ:* Collection<Long>*Popis:* Kolekce akcí, které bude mít uživatel právo ovlivňovat.**Metoda: getUser** Public*Popis:* Vyhledá a vrátí uživatele podle jeho přihlašovacího jména.**Parametr: login***Typ:* String*Popis:* Login podle, kterého se účastník vyhledá.**Metoda: deleteUser** Public*Popis:* Vyhledá a odstraní daného uživatele.

Parametr: login*Typ:* String*Popis:* Login vyhledávaného uživatele.**Metoda: editUserParams** Public*Popis:* Edituje parametry uživateli vyhledaného podle jeho starého loginu. Aktualizuje se jeho login, heslo i seznam akcí, které může editovat.**Parametr: login***Typ:* String*Popis:* Login, podle kterého se uživatel bude vyhledávat.**Parametr: newLogin***Typ:* String*Popis:* Nový login.**Parametr: newPassword***Typ:* String*Popis:* Nové heslo.**Parametr: newRole***Typ:* Role*Popis:* Nová role.**Parametr: newActionIDs***Typ:* Collection<Long>*Popis:* Nový seznam akcí.**Metoda: editLogin** Public*Popis:* Edituje login a heslo uživatele, vyhledaného podle jeho současného loginu. Tato metoda se používá, když si uživatel mění login a heslo sám (pro editaci práv na akce nemá oprávnění).**Parametr: login***Typ:* String*Popis:***Parametr: newLogin***Typ:* String*Popis:***Parametr: newPassword***Typ:* String*Popis:***Metoda: getUsers** Public*Popis:* Vrací všechny uživatele.**Metoda: getActions** Public*Popis:* Vrací kolekci akcí, které má právo uživatel editovat

Parametr: login*Typ:* String*Popis:* Login vyhledávaného uživatele.**Metoda: getUsers** Public*Popis:* Vrací všechny uživatele, které může uživatel s rolí v parametru role editovat. Pro registrátora to není nikdo, pro admina registrátoři a pro superadmina to jsou všichni uživatelé.**Parametr: role***Typ:* Role*Popis:* Role přihlášeného uživatele.**Metoda: removeAction** Public*Popis:* Odstraní uživateli s daným loginem právo na správu akce s daným ID.**Parametr: login***Typ:* String*Popis:* Login vyhledávaného uživatele.**Parametr: removeID***Typ:* long*Popis:* odstraňovaná akce.**Metoda: addAction** Public*Popis:* Přidá uživateli s daným loginem právo na správu akce s daným ID.**Parametr: login***Typ:* String*Popis:***Parametr: actionIDs***Typ:* Collection<Long>*Popis:*

Balíček Utils

Diagram: Utils

Created By: Lahvi on 2.11.2011*Last Modified:* 14.12.2011*Popis:*

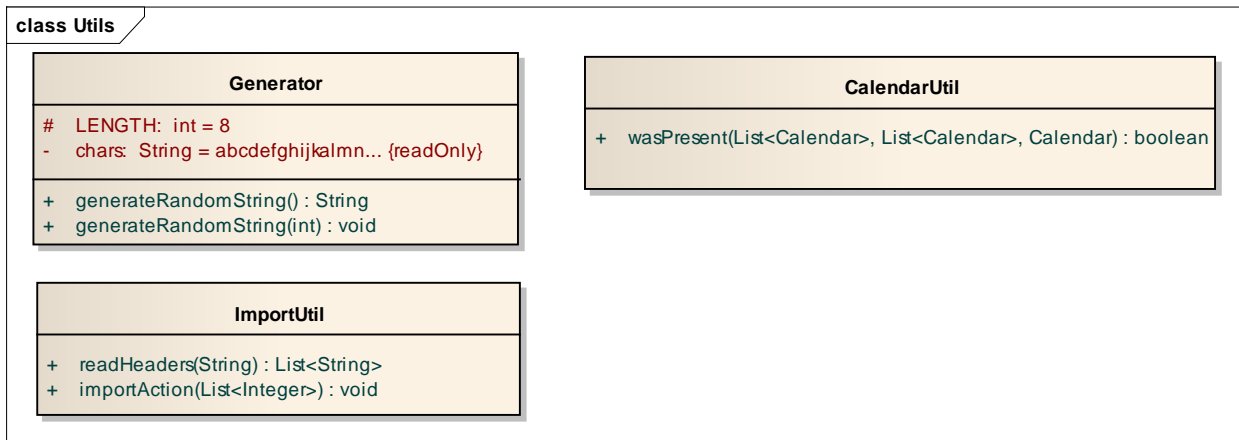


Figure: Diagram Utils

CalendarUtil

Popis:

Třída má metodu, které počítá s datумы umí vypočítat zda byl v tu kterou dobu zákazník přítomen či nepřítomen.

Metoda: wasPresent `Public Static`

Popis: Metoda podle seznamu příchodů, odchodů a podle daného času pro daného účastníka zjistí, zda byl na akci přítomen nebo ne.

Parametr: arrivals

Typ: `List<Calendar>`

Popis: List s datумы a časy příchodů.

Parametr: departures

Typ: `List<Calendar>`

Popis: List s datумы a časy odchodů

Parametr: currentTime

Typ: `Calendar`

Popis: Datum a čas pro který chceme zjistit zda byl účastník přítomen či nepřítomen

Generator

Popis:

Třída pro generování náhodných řetězců. Její metody se budou používat při potřebě vygenerovat login nebo heslo nějakému účastníkovi či uživateli.

Atribut: LENGTH `Protected Static`

Typ: `int`

Popis: Implicitní délka generovaných řetězců. Je protected, lze tedy nastavit z nějaké business třídy.

Atribut: chars `Private Static`

Typ: `String`

Popis: Řetězec znaků, ze kterého metody `generateRandomString` generují náhodné znaky.

Metoda: generateRandomString `Public Static`

Popis: Vrátí řetězec náhodný alfanumerický řetězec o defaultní délce například 10 znaků. Záleží na tom co je nastaveno v atributu LENGTH.

Metoda: generateRandomString `Public Static`

Popis: Stejně jako verze bez parametru, ale délku řetězce nastavíme paremetrem len.

Parametr: len

Typ: `int`

Popis: Délka náhodného řetězce,

ImportUtil

Popis:

Třída, která obsahuje funkce, k importu uživatelů z csv souborů

Metoda: readHeaders `Public`

Popis: Metoda načítá hlavičky sloupců z .csv souboru, ze kterého chceme importovat účastníky.

Parametr: file

Typ: `String`

Popis: Cesta k .csv souboru, který načítáme.

Metoda: importAction `Public`

Popis: Funkce vyparsuje z .csv souborů jednotlivé položky a přiřadí je k parametrům nově vytvářeného účastníka podle indexů předaných v parametru indexes.

Parametr: indexes

Typ: `List<Integer>`

Popis: List obsahující čísla sloupečků, k jednotlivým povinným parametrům. Čísla musí být v pořadí Křestní jméno, Příjmení, Email, číslo OP / rodné číslo.

