

# **Analytická dokumentace**

## ***Projekt volební systém***

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

## Obsah:

Balíček Registrační systém .....	3
Balíček Requirements Model .....	3
Balíček Funkční požadavky .....	3
Systém bude umožňovat editaci uživatelů či účastníků .....	3
Systém bude umožňovat evidenci přítomnosti účastníků .....	3
Systém bude umožňovat ověření uživatele pomocí jména a hesla .....	4
Systém bude umožňovat vytváření nových uživatelů .....	4
Systém bude umožňovat vytvářet a editovat volební akce .....	4
Systém bude umožňovat vytváření či importování nových účastníků .....	4
Balíček Nefunkční požadavky .....	4
Systém bude poskytovat databázi účastníků pro Mobilní hlasovací systém a pro systém EVolby .....	4
Systém bude používat souborovou databázi .....	4
Systém poběží na platformě Java SE/EE .....	4
Balíček Use Case Model .....	5
Balíček Actors .....	5
Administrátor .....	6
Registrátor .....	6
SuperAdmin .....	7
Balíček Primary Use Cases .....	7
Dokončit registraci .....	7
Editovat administrátora .....	8
Editovat registrátora .....	9
Editovat účastníka .....	9
Importovat účastníky .....	10
Odstranit voliče .....	11
Označit odchod .....	11
Označit příchod .....	12
Přidat registrátora .....	12
Vytvořit administrátora .....	13
Vytvořit akci .....	13
Vytvořit nového účastníka .....	14
Zobrazit seznam nepřítomných účastníků .....	15
Zobrazit seznam přítomných účastníků .....	15
Zobrazit seznam účastníků akce .....	16
Balíček Domain Model .....	16
Balíček Domain Objects .....	17
Admin .....	17
Akce .....	18
Přítomnost .....	18
Ucastnik .....	18

## Balíček Registrační systém:

## Balíček Requirements Model:

### Requirements Model

*Created By:* Ondra on 19.11.2005

*Last Modified:* 30.10.2011

*Popis:*

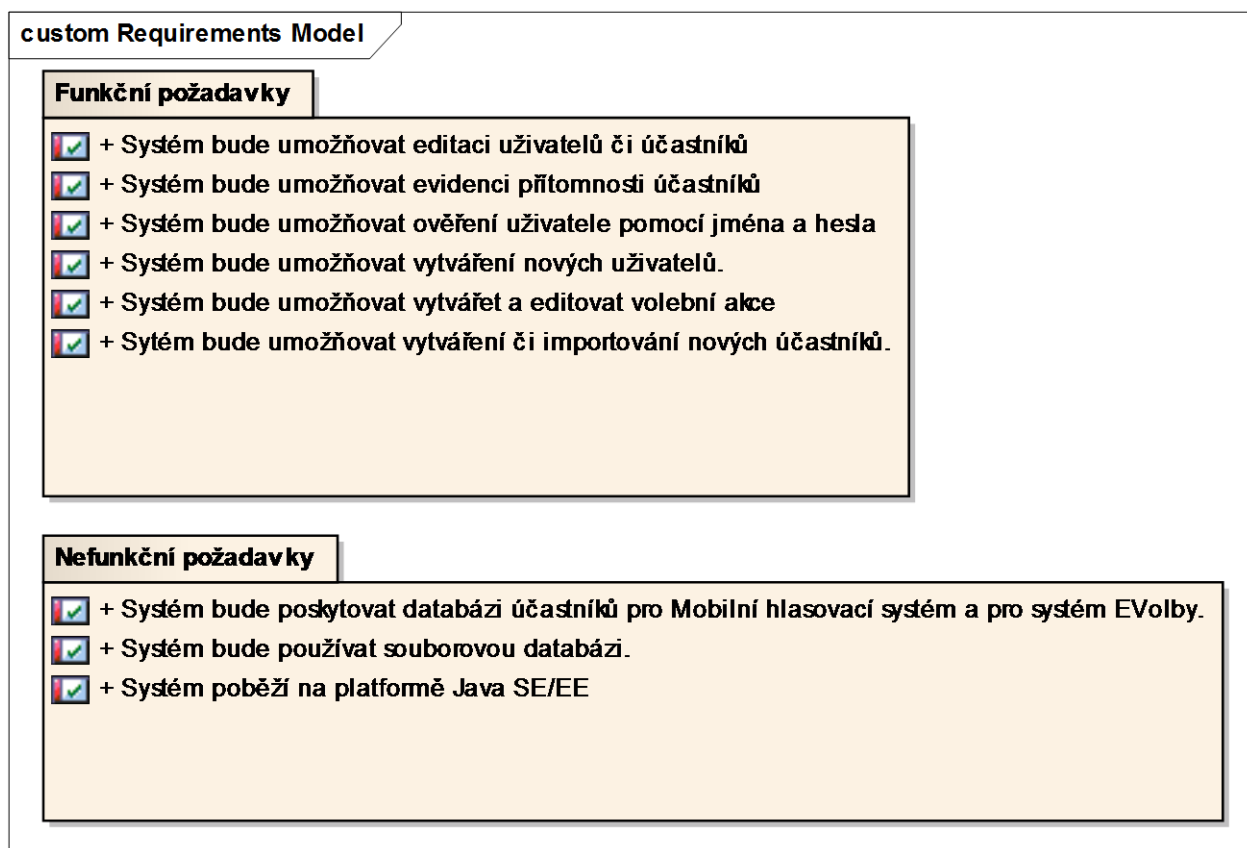


Figure: 1

## Balíček Funkční požadavky:

### Systém bude umožňovat editaci uživatelů či účastníků

*Popis:*

Systém bude umožňovat každému uživateli editovat sám sebe a podle uživatelské role i ostatní uživatele. Bude také umožňovat editovat údaje účastníků, nebo je například odstraňovat.

### ***Systém bude umožňovat evidenci přítomnosti účastníků***

#### ***Popis:***

U každého účastníků, bude možno v libovolném čase zjistit, zdali byl zrovna na akci přítomen nebo ne.

### ***Systém bude umožňovat ověření uživatele pomocí jména a hesla***

#### ***Popis:***

Každý uživatel se bude moci přihlásit do systému pomocí hesla a loginu. Tyto údaje se uživateli vygenerují při zaregistrování a poté si je bude moci sám změnit.

### ***Systém bude umožňovat vytváření nových uživatelů.***

#### ***Popis:***

Každý uživatel bude moci vytvořit nového uživatele jehož role bude mít nižší práva než on. Nejvyšší role se bude napevno zadávat do databáze.

### ***Systém bude umožňovat vytvářet a editovat volební akce***

#### ***Popis:***

Než se budou importovat či vytvářet noví uživatelé musí být vytvořena nějaká volební akce. Tu budou moci uživatelé podle své role vytvářet.

### ***Sytém bude umožňovat vytváření či importování nových účastníků.***

#### ***Popis:***

Importování bude umožněno z externího souboru .csv a vytváření nových účastníků proběhne když účastník nebude importován z externího souboru.

### **Balíček Nefunkční požadavky:**

#### ***Systém bude poskytovat databázi účastníků pro Mobilní hlasovací systém a pro systém EVolby.***

#### ***Popis:***

Oba systémy, které registrační systém spojuje si budou moci nahrát účastníky z společné databáze vytvořené pro zaregistrované účastníky reg. systémem.

### ***Systém bude používat souborovou databázi.***

#### ***Popis:***

Systém bude ukládat entity do databáze MySql.

## Systém poběží na platformě Java SE/EE

### Popis:

U registračního systému není potřeba přístupu přes webový browser, bude přístupný pouze na místě, a proto bude použita standartní edice Javy.

## Balíček Use Case Model:

### Use Case Model

*Created By:* Stopka on 19.11.2005

*Last Modified:* 30.10.2011

### Popis:

V systému budou tři aktéři kteří budou mít možnost provádět akce podle usecase diagramu níže.

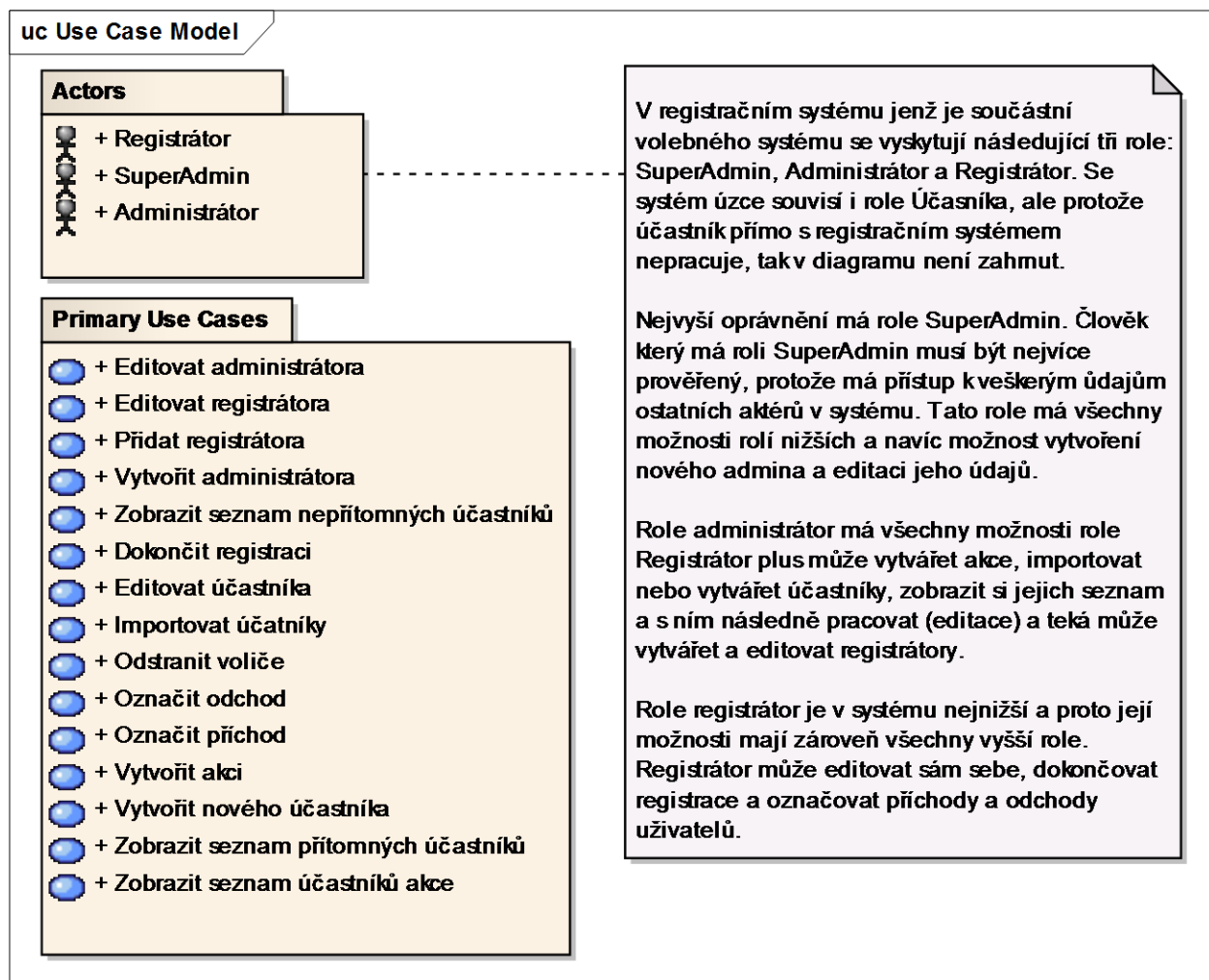


Figure: 2

## **Balíček Actors:**

### **Actors**

*Created By:* Stopka on 19.11.2005

*Last Modified:* 30.10.2011

*Popis:*

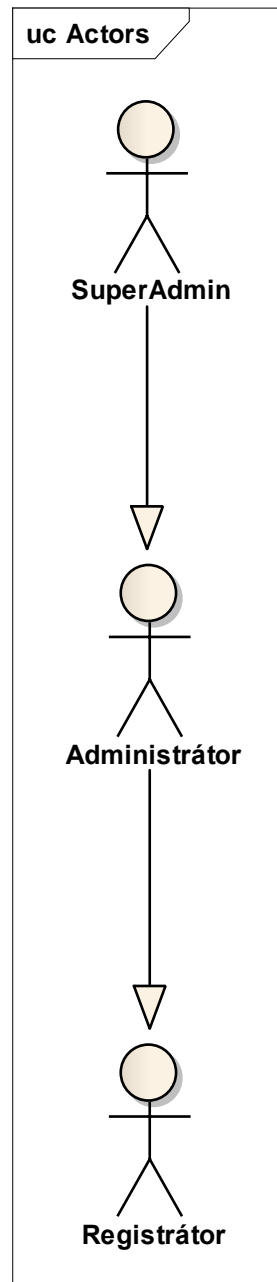


Figure: 3

## **Administrátor**

### *Popis:*

Role administrátor má všechny možnosti role Registrátor plus může vytvářet akce, importovat nebo vytvářet účastníky, zobrazit si jejich seznam a s ním následně pracovat (editace) a také může vytvářet a editovat registrátory.

## **Registrátor**

### *Popis:*

Role registrátor je v systému nejnižší a proto její možnosti mají zároveň všechny vyšší role. Registrátor může editovat sám sebe, dokončovat registrace a označovat příchody a odchody uživatelů.

## **SuperAdmin**

### *Popis:*

Role SuperAdmin má nejvyšší oprávnění. Člověk který má roli SuperAdmin musí být nejvíce prověřený, protože má přístup k veškerým údajům ostatních aktérů v systému. Tato role má všechny možnosti rolí nižších a navíc možnost vytvoření nového admina a editaci jeho údajů.

## Balíček Primary Use Cases:

### Primary Use Cases

Created By: Stopka on 19.11.2005

Last Modified: 30.10.2011

Popis:

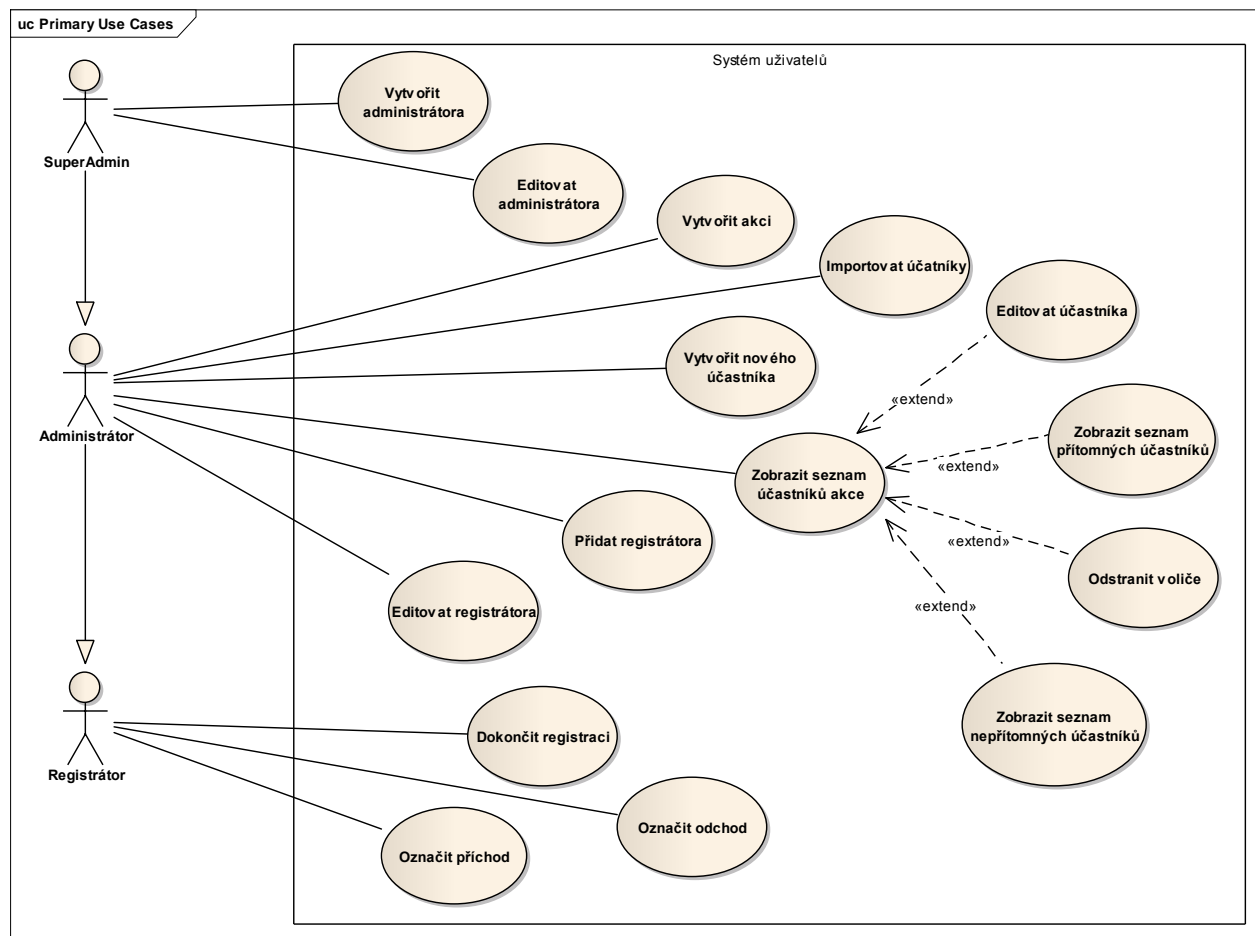


Figure: 4



## **Dokončit registraci**

### **Popis:**

Nastává když SuperAdmin, Admin nebo Registrátor bude potřebovat dokončit registraci účastníka, což zahrnuje zkontrolování údajů a vygenerování přihlašovacího jména a hesla.

### **Požadavky:**

- Předregistrování  
Účastník musí být již v databázi vytvořen.
- Přihlášení  
Uživatel musí být do systému přihlášen.

### **Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Admin, Registrátor.
  - **Vstup:** Zaregistrovaný účastník, buď pomocí importu nebo jinak, který ještě nemá vygenerované přístupové údaje.
  - **Výstup:** Zaregistrovaný účastník, který má vygenerované přístupové údaje a má přístup do systému.
    1. Aktér vyhledá účastníka v databázi
    2. Aktér požádá systém o vygenerování přihlašovacích údajů
    3. Systém vygeneruje přihlašovací údaje (ID heslo) a uloží je do databáze.

## **Editovat administrátora**

### **Popis:**

Nastává když SuperAdmin bude potřebovat editovat údaje nějakého administrátora. Například když admin ztratí přihlašovací údaje.

### **Požadavky:**

- Existence  
Editovaný administrátor musí v systému existovat.
- Oprávnění  
Uživatel musí mít k dané akci potřebné oprávnění. Musí to být SuperAdmin.
- Přihlášení  
Uživatel musí být do systému přihlášen.

### **Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin
  - **Vstup:** Vytvořený admin.
  - **Výstup:** Admin s editovanými údaji.
    1. Aktér požádá o zobrazení seznamu adminů.
    2. Systém zobrazí seznam.
    3. Aktér vybere daného admina.

**Požadavky:**

4. Systém zobrazí detail admina s možností editace údajů.
5. Aktér zedituje požadované údaje a požádá o dokončení.
6. Systém zažádá o potvrzení a uloží změny.

**Editovat registrátora****Popis:**

Nastává když SuperAdmin nebo Administrátor bude potřebovat změnit údaje Registrátora, např. když registrátor zapomene svoje přihlašovací údaje.

**Požadavky:**

- Existence  
Editovaný uživatel musí v systému existovat.
- Oprávnění  
Uživatel musí mít k akci příslušná oprávnění. Tzn. nesmí to být registrátor.
- Přihlášení  
Uživatel musí být v systému přihlášen.

**Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Existující registrátor
  - **Výstup:** Registrátor s editovanými údaji.
    1. Aktér požádá o zobrazení seznamu registrátorů
    2. Systém zobrazí registrátory
    3. Aktér vybere požadovaného registrátora.
    4. Systém zobrazí údaje registrátora s možností editace.
    5. Aktér zedituje požadované údaje a požádá o dokončení.
    6. Systém se zeptá na potvrzení a uloží změny.

**Editovat účastníka****Popis:**

Začíná když SuperAdmin nebo Administrátor budou potřebovat zeditovat údaje nějakého účastníka.

**Požadavky:**

- Existence  
Účastník musí v systému existovat.

**Požadavky:**

- Oprávnění  
Uživatel musí mít k editaci účastníka oprávnění. Tzn. nesmí to být registrátor.
- Přihlášení  
Uživatel musí být přihlášen.

**Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořený účastník.
  - **Výstup:** Účastník s novými údaji.
    1. Aktér požádá o zobrazení seznamu účastníků.
    2. Systém zobrazí účastníky na dané akci.
    3. Aktér vybere daného účastníka.
    4. Systém zobrazí údaje účastníka s možností editace.
    5. Aktér zedituje údaje účastníka a požádá o dokončení.
    6. Systém zažádá o potvrzení a uloží změny.

**Importovat účastníky****Popis:**

Začíná v momentě, kdy má administrátor seznam účastníků (csv soubor) a potřebuje jej uložit do systému

**Požadavky:**

- Existence akce  
V systému musí být již vytvořená akce ke které se účastníci budou přiřazovat.
- Oprávnění  
Uživatel musí mít k akci oprávnění. Nesmí to být registrátor.
- Přihlášení  
Uživatel musí být přihlášený do systému

**Scénář:**

- Import
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořená akce a existující externí .csv soubor s účastníky.
  - **Výstup:** Importování účastníků z csv souboru uložení v databázi.
    1. Aktér požádá o importování účastníků k dané akci.
    2. Systém požádá o zadání cesty k danému souboru.
    3. Aktér zadá cestu k .csv souboru s daty účastníků a požádá o import.
    4. Systém importuje účastníky přiřadí je k dané akci.

## Odstranit voliče

### Popis:

Začíná, když administrátor některého již importovaného voliče potřebuje odstranit

### Požadavky:

- Existence  
Odstraňovaný účastník musí již v systému existovat.
- Oprávnění  
Uživatel musí mít k odstranění účastníka oprávnění. Nesmí to být registrátor.
- Přihlášení  
Uživatel musí být přihlášený do systému

### Scénář:

- Odstranění voliče
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořený účastník.
  - **Výstup:** Odstraněný účastník.
    1. Aktér požádá o zobrazení seznamu účastníků.
    2. Systém zobrazí účastníky na dané volební akci s možností dalších akcí.
    3. Aktér vybere účastníka a zvolí akci odstranit.
    4. Systém zažádá o potvrzení a odstraní účastníka.

## Označit odchod

### Popis:

Začíná v momentě, když bude potřebovat SuperAdmin, Admin nebo Registrátor zaznamenat odchod účastníka.

### Požadavky:

- Existence  
Účastník musí být v systému registrován a musí mít vygenerovány přístupové údaje.

**Požadavky:**

- Přihlášení  
Uživatel musí být do systému přihlášený.
- Přítomný  
Účastník musí být označen jako přítomný jinak by nemohl být označen jako nepřítomný.

**Scénář:**

- Odchod
  - **Aktéři:** SuperAdmin, Admin, Registrátor.
  - **Vstup:** Zaregistrovaný účastník.
  - **Výstup:** Zaregistrovaný účastník se zaznamenanou dobou odchodu a označením jako nepřítomný..
    1. Aktér vyhledá účastníka v databázi
    2. Aktér požádá systém o zaznamenání odchodu.
    3. Systém zaznamená dobu odchodu a označí účastníka jako nepřítomného.

**Označit příchod****Popis:**

Začíná v momentě, když bude potřebovat SuperAdmin, Admin nebo Registrátor zaznamenat příchod účastníka.

**Požadavky:**

- Existence  
Účastník musí v systému existovat a musí mít dokončenou registraci.
- Nepřítomnost  
Účastník musí být v době označování příchodu označen jako nepřítomný.
- Přihlášený  
Uživatel musí být do systému přihlášený.

**Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Admin, Registrátor.
  - **Vstup:** Zaregistrovaný účastník.
  - **Výstup:** Zaregistrovaný účastník se zaznamenanou dobou příchodu a označením jako přítomný..
    1. Aktér vyhledá účastníka v databázi
    2. Aktér požádá systém o zaznamenání příchodu.
    3. Systém zaznamená dobu příchodu a označí účastníka jako přítomného.

## **Přidat registrátora**

### *Popis:*

Vytvoří uživatele typu admin, který bude obsluhovat registrační systém, ale nebude mít plná práva

### **Požadavky:**

- Oprávnění  
Uživatel musí mít k akci příslušná oprávnění tzn., nesmí to být Registrátor.
- Přihlášení  
Uživatel musí být v systému přihlášený.

### **Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** -
  - **Výstup:** Vytvořený Nový registrátor uložený v dtb.
    1. Aktér požádá systém o vytvoření nového registrátora.
    2. Systém zobrazí formulář pro vytvoření registrátora
    3. Aktér vyplní údaje a požádá o dokončení.
    4. Systém uloží nového registrátora do databáze.

## **Vytvořit administrátora**

### *Popis:*

Nastává když SuperAdmin bude potřebovat vytvořit nového administrátora.

### **Požadavky:**

- Oprávnění  
Uživatel musí mít k akci příslušná oprávnění tzn., musí to být SuperAdmin.
- Přihlášení  
Uživatel musí být v systému přihlášený.

### **Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin
  - **Vstup:** -
  - **Výstup:** Vytvořený administrátor
    1. Aktér požádá o vytvoření nového adminstrátora.
    2. Systém zobrazí formulář.
    3. Aktér vyplní údaje a požádá o dokončení.
    4. Systém uloží nového admina do databáze.

## Vytvořit akci

### Popis:

Začíná, když potřebuje administrátor založit v systému novou akci (chystá se schůze, na které se bude hlasovat a podobně)

### Požadavky:

- Oprávnění  
Uživatel musí mít k akci příslušná oprávnění. Nesmí to být registrátor.
- Přihlášení  
Uživatel musí být do systému přihlášen.

### Scénář:

- Vytvoření akce
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** -
  - **Výstup:** Vytvořená akce k volebnímu sjezdu
    1. Administrátor požádá vytvoření volební akce.
    2. Systém zobrazí formulář.
    3. Administrátor zadá jméno akce a datum konání a další údaje o akci a požádá o dokončení.
    4. Systém uloží novou akci do databáze.

## Vytvořit nového účastníka

### Popis:

Nastává když SuperAdmin nebo Administrátor bude chtít vytvořit nového účastníka, který nebyl do systému nahrán při importování z externího souboru.

### Požadavky:

- Existence akce  
V systému musí být vytvořena akce, ke které bude účastník přiřazen.
- Oprávnění  
Uživatel musí mít k akci příslušná oprávnění. (nesmí to být registrátor)
- Přihlášení  
Uživatel musí být v systému přihlášen.

### Scénář:

- Basic Path
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořená akce.
  - **Výstup:** Vytvořený účastník přiřazený k dané akci.
    1. Aktér vybere akci požádá o vytvoření nového účastníka.
    2. Systém zobrazí formulář pro vyplnění údajů o účastníkovi.
    3. Aktér zadá údaje a požádá o dokončení.
    4. Systém uloží nového účastníka.

## **Zobrazit seznam nepřítomných účastníků**

### **Popis:**

Začíná když aktér potřebuje vidět seznam nepřítomných účastníků.

### **Požadavky:**

- Existence akce  
V systému musí být vytvořená akce jejíž seznam uživatelů chceme zobrazit.
- Oprávnění  
Uživatel musí mít pro akci oprávnění (nesmí to být registrátor).
- Přihlášení  
Uživatel musí být v systému přihlášen.

### **Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořený seznam účastníků.
  - **Výstup:** Zobrazený seznam s nepřítomnými účastníky na akci .
    1. Aktér požádá o zobrazení seznamu účastníků.
    2. Systém zobrazí účastníky na dané volební akci s možností dalších akcí.
    3. Aktér vybere akci zobrazit pouze nepřítomné.
    4. Systém odfiltruje přítomné účastníky a zobrazí pouze nepřítomné.

## **Zobrazit seznam přítomných účastníků**

### **Popis:**

Začíná když aktér potřebuje vidět přítomné účastníky.

### **Požadavky:**

- Přihlášení  
Administrátor musí být přihlášen k systému

### **Scénář:**

- Basic Path
  - **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořený seznam účastníků.
  - **Výstup:** Zobrazený seznam s přítomnými účastníky na akci .
    1. Aktér požádá o zobrazení seznamu účastníků.
    2. Systém zobrazí účastníky na dané volební akci s možností dalších akcí.
    3. Aktér vybere akci zobrazit pouze přítomné.
    4. Systém odfiltruje nepřítomné účastníky a zobrazí pouze přítomné.



## **Zobrazit seznam účastníků akce**

### **Popis:**

Začíná, když administrátor nebo SuperAdmin chce shlédnout seznam účastníků na akci.

### **Požadavky:**

- Existence akce  
V systému musí existovat akce jejíž uživatele chceme zobrazit.
- Oprávnění  
Uživatel musí mít k akci příslušné oprávnění (nesmí to být registrátor).
- Přihlášení  
Uživatel musí být přihlášen k systému.

### **Scénář:**

Zobrazit seznam

- **Aktéři:** SuperAdmin, Administrátor
  - **Vstup:** Vytvořený seznam účastníků.
  - **Výstup:** Zobrazený seznam s účastníky akce plus s možností dalších voleb.
1. Aktér požádá o zobrazení seznamu účastníků.
  2. Systém zobrazí účastníky na dané akci s možností dalších akcí se seznamem (editace údajů, odstranění atd.).

## Balíček Domain Model:

### Domain Model

*Created By:* Ondra on 19.11.2005

*Last Modified:* 30.10.2011

*Popis:*

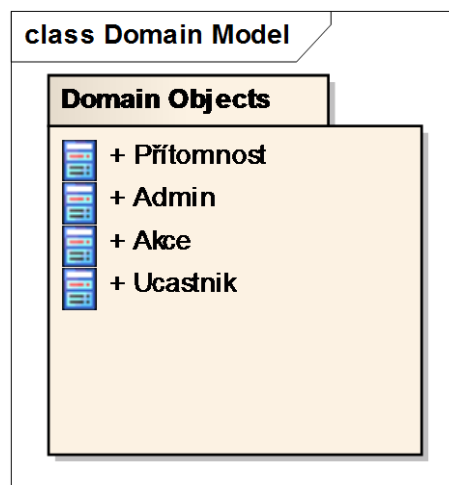


Figure: 5

## Balíček Domain Objects

### Domain Objects

*Created By:* Ondra on 11.1.2006

*Last Modified:* 30.10.2011

*Popis:*

Model popisuje fyzické entity v systému registrace. Budou tam entity admina, volební akce, účastníka a jeho přítomnosti.

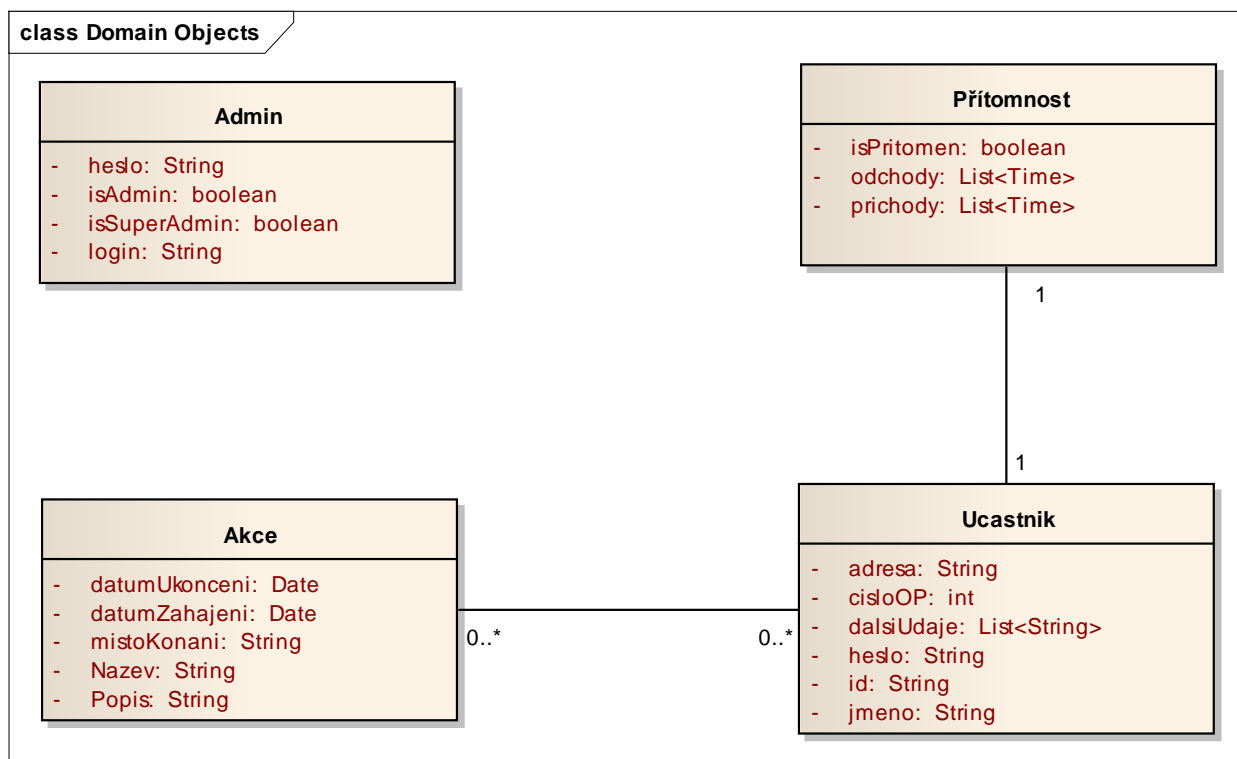


Figure: 6

## Admin

### Popis:

Administrátor je entita, která reprezentuje osobu, jenž se systémem pracuje. Jelikož jsou v systému tři aktéři entita mezi nimi rozlišuje podle nastavených příznaků isAdmin a isSuper admin.

Registrátora reprezentuje entita jenž má tyto tři příznaky nastavena na false, Administrátor má isAdmin = true a isSuperAdmin = false a SuperAdmin má oba true.

Dále už je pak v entitě pouze heslo a login, tedy údaje kterými se budou uživatelé do systému přihlašovat.

## Akce

### Popis:

Entita akce jednoduše reprezentuje volební akci. Je svázána s entitou účastník relací 0 ..\* ku 1 .. \*, to znamená že nově vytvořená akce nemusí mít nutně přiřazené účastníky a poté se jich akci může přiřadit libolný počet.

V entitě je pět jednoduchých atributů. Název akce, místo konání, datумы zahájení a ukončení a nějaká stručná poznámka.

## **Přítomnost**

### *Popis:*

Entita přítomnost je svázána s entitou účastníky relací One to One. Každý účastník tedy má právě jednu přítomnost a každá přítomnost náleží právě jednomu účastníkovi.

Tato entita má v sobě údaje o tom kdy účastník byl přítomen na dané akci a kdy na dané akci nebyl. Atribut isPřítomen udává zda je účastník přítomen právě ve chvíli kdy se účastníka sledujeme. Atributy List<Time> a List<Time> mají v sobě uloženy doby příchodů a odchodů a lze z nich vypočítat účastníkovu přítomnost v libovolné době.

## **Ucastnik**

### *Popis:*

Entita účastník reprezentuje člověka, jenž se bude účastnit dané akce. Je asociována s entitami akce a s entitou přítomnost. Každý uživatel musí být přiřazen nějaké akci a každý uživatel má právě jednu přítomnost.

Účastník má v sobě základní údaje o osobě jako Jméno, Adresa, číslo občanského průkazu a případně další libovolné akce. Dále pak heslo a id, které se uživateli vygenerují když se dostaví na akci kde je zaregistrován.

## **Platforma / framework:**

Systém bude používat platform Java SE, protože nebude potřeba vzdálenému přístupu přes webový prohlížeč, ale bude běžet jako klasická desktopová aplikace. Dále bude aplikace používat databázi MySQL.