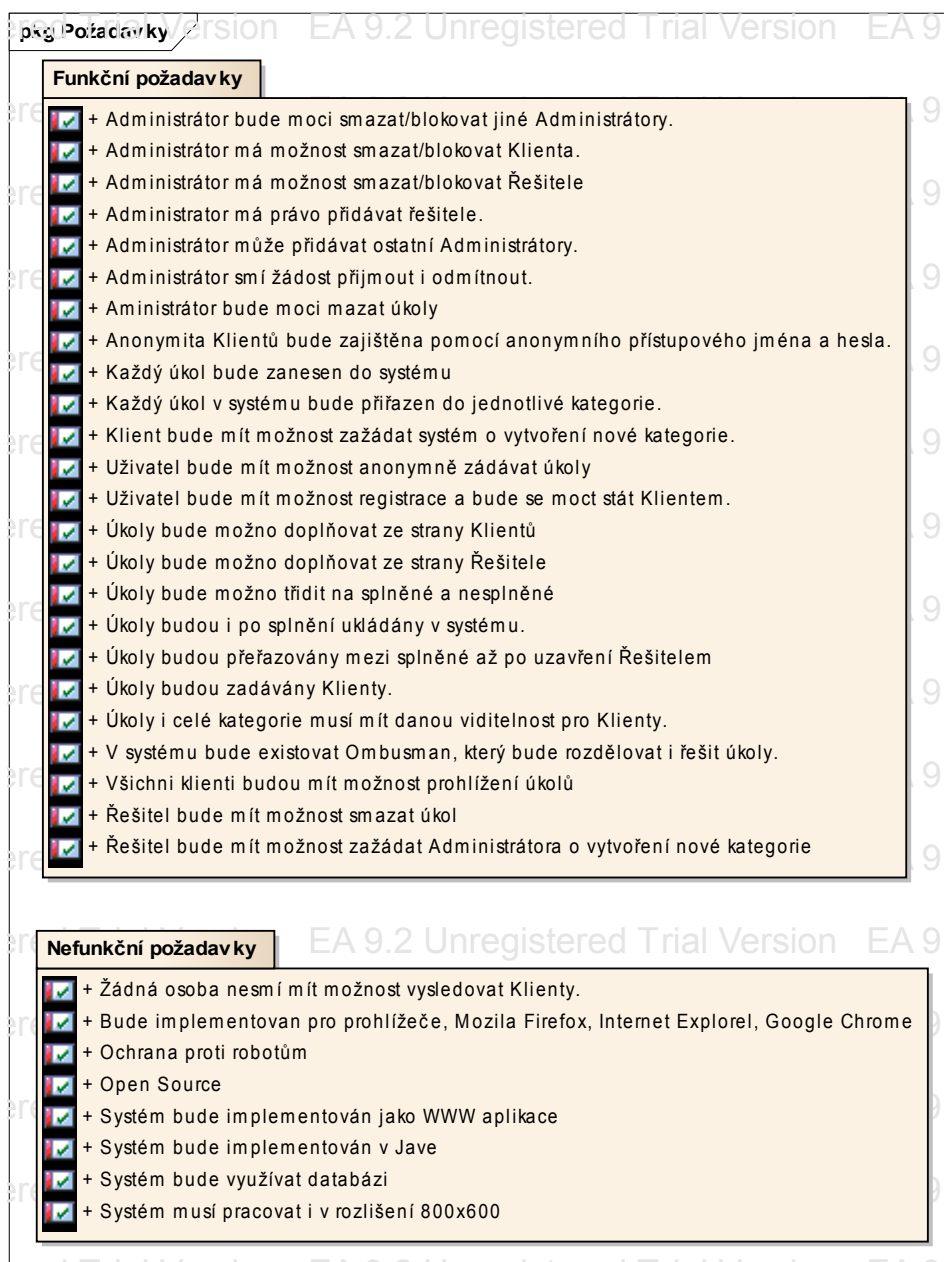


# E-Ombudsman

→ **Analýza (v. 1.3)**

# Požadavky na systém

Balíček obsahuje seznam funkčních a nefunkčních požadavků. Pro grafickou reprezentaci používá UML diagram.



*Ilustrace 1: Diagram požadavků*

# Funkční požadavky

Package obsahující funkční požadavky a jejich detaily.

## **Administrátor má právo přidávat řešitele.**

Administrátor bude moci přidávat nebo jmenovat další řešitele.

## **Administrátor bude moci smazat/blokovat jiné Administrátory.**

Pokud se Administrátor bude chovat nevhodně nebo už nevykazuje činnost, bude mít možnost Administrátor jiný účet smazat/blokovat.

Odpovídající případ užití - smazání registrovaného uživatele

## **Administrátor má možnost smazat/blokovat Klienta.**

Pokud se některý z Klientů chová nevhodně nebo pokud už nevykazuje činnost, může ho Administrátor vymazat ze systému.

Odpovídající případ užití - smazání registrovaného uživatele

## **Administrátor má možnost smazat/blokovat Řešitele**

Pokud se některý z Řešitelů už nevykazuje činnost, může ho Administrátor vymazat ze systému.

Odpovídající případ užití - smazání registrovaného uživatele

## **Administrátor může přidávat ostatní Administrátory.**

Administrátor bude mít možnost přidat dalšího Administrátora, pokud bude systém spravován více než jedním Administrátorem.

Odpovídající případ užití - Přidání nového Administrátora

## **Administrátor smí žádost přijmout i odmítnout.**

Pokud se Administrátorovi nebude zamlouvat jakákoli žádost, bude mít oprávnění tuto možnost, po zadání důvodu, zamítnout. Pokud Administrátorovi přijde žádost vhodná potvrdí ji a spracuje.

Odpovídající případ užití - Potvrzení žádosti, Nepotvrzení žádosti

## **Administrátor bude moci mazat úkoly**

Administrátor má možnost smazat zadané úkoly.

## **Anonymita Klientů bude zajištěna pomocí anonymního přístupového jména a hesla.**

Každý Klient bude povinen mít přístupové jméno a heslo. Tyto údaje jsou neveřejné a přístup k nim spadá pouze Klientovi.

Odpovídající případ užití - Přihlášení.

## **Každý úkol bude zanesen do systému**

Po zapsání úkolu Klientem je zanesen do databáze. Nemůže se stát, že kKlient zadá úkol a ten je bez záznamu vymazán nebo ztracen. I při formálním mazání úkolu je úkol přesunut do záložního úložiště.

Odpovídající případ užití - přidání úkolu

## **Každý úkol v systému bude přiřazen do jednotlivé kategorie.**

Úkoly budou spadat do kategorií, ty budou určovat způsob řešení případně vhodného Řešitele.

Odpovídající případ užití - přidání nového úkolu

### **Klient bude mít možnost požádat systém o vytvoření nové kategorie.**

Pokud Klient vytvoří úkol, který nepasuje do žádné z kategorií, bude moci požádat o vytvoření nové kategorie.

Odpovídající případ užití - požádání o vytvoření nové kategorie

### **Uživatel bude mít možnost anonymně zadávat úkoly**

Neregistrovaný uživatel bude moci přidávat anonymně úkoly.

### **Uživatel bude mít možnost registrace a bude se moci stát Klientem.**

Uživatel bude mít pouze omezená práva, pro zvýšení svých práv se bude muset registrovat a stát se Klientem.

Odpovídající případ užití - Registrace

### **V systému bude existovat Ombudsman, který bude rozdělovat i řešit úkoly.**

AIP bude kromě Řešitelů obsluhovat i Ombudsman, nadřízený Řešitelům. Bude mít práva Řešitelů, avšak navíc bude moci rozřazovat příchozí úkoly.

Odpovídající případ užití - Přidání úkolu Řešiteli

### **Všichni klienti budou mít možnost prohlížení úkolů**

Každý registrovaný uživatel bude mít možnost prohlížení úkolů, se kterými budou pracovat (jím přidáné).

Odpovídající případ užití - Zobrazení úkolů, Filtrování úkolů

### **Úkoly bude možno doplňovat ze strany Klientů**

Pokud zjistí Klient nějaké doplňující informace k úkolu, bude mít možnost úkol doplnit.

Odpovídající případ užití - Doplnění nesplněného úkolu

### **Úkoly bude možno doplňovat ze strany Řešitele**

Každý Řešitel bude mít možnost k přiděleným úkolům přidávat poznámky a postupně je upravovat a dořešovat.

Odpovídající případ užití - Doplnění nesplněného úkolu

### **Úkoly bude možno třídit na splněné a nesplněné**

AIP bude poskytovat nástroje pro třídění a zobrazování přehledných seznamů. Všechny osoby budou mít možnost třídit úkoly podle toho, které úkoly je přesně zajímají.

Use Case - Filtrování úkolů

### **Úkoly budou i po splnění ukládány v systému.**

Po splnění úkolu se úkol nesmaže, ale zapíše se do databáze splněných úkolů.

Odpovídající případ užití - splnění úkolu

### **Úkoly budou přerazovány mezi splněné až po uzavření Řešitelem**

Po splnění úkolu se úkol přeřadí do databáze splněných úkolů.

Odpovídající případ užití - Splnění úkolu

### **Úkoly budou zadávány Klienty.**

Hlavním účelem Klientů bude zadávání úkolů pro Řešitele.

Odpovídající případ užití - Přidání úkolu

### **Úkoly i celé kategorie musí mít danou viditelnost pro Klienty.**

Pokud daná kategorie nebo úkol nemá být viděn, musí být nastavená viditelnost.

### **Řešitel bude mít možnost smazat úkol**

Pokud se bude řešiteli jevit zadaný úkol jako nesplnitelný nebo chybný, tak bude mít možnost ho smazat.

### **Řešitel bude mít možnost zažádat Administrátora o vytvoření nové kategorie**

Pokud se objeví úkol, který je nezařaditelná do žádné kategorie, bude mít možnost Řešitel zažádat o vytvoření kategorie.

Odpovídající případ užití - Zažádání o vytvoření nové kategorie

## **Nefunkční požadavky**

Package obsahující nefunkční požadavky a jejich detaily.

### **Bude implementovan pro prohlížeče, Mozilla Firefox, Internet Explorel, Google Chrome**

System bude podporovat všechny běžné prohlížeče

### **Ochrana proti robotům**

AIP bude ochráněno proti robotům.

### **Open Source**

Základem AIP jsou open-sourcové programy.

### **Systém bude implementován jako WWW aplikace**

AIP bude zprostředkováno pomocí webového rozhraní.

### **Systém bude implementován v Java**

AIP je napsán v jazyce Java.

### **Systém bude využívat databázi**

System bude ukládat data do databáze.

### **Systém musí pracovat i v rozlišení 800x600**

System musí podporovat i zastaralé rozlišení.

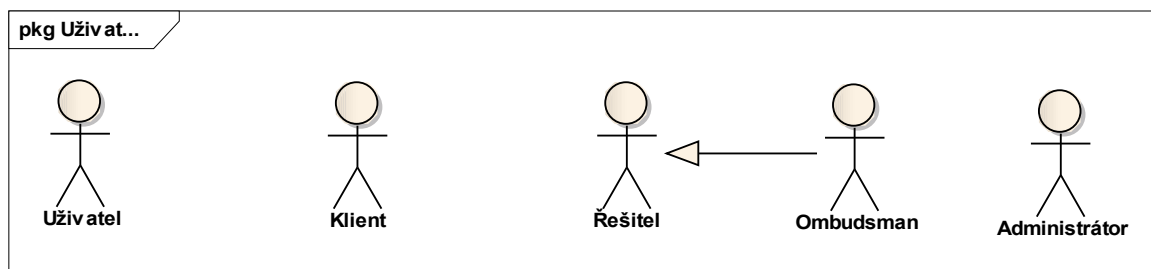
### **Žádná osoba nesmí mít možnost vysledovat Klienty.**

Klient nesmí zadávat jakékoli vysledovatelné informace a musí být bráněno, aby byl vysledovatený.

## **Případy užití**

Balíček obsahuje účastníka, seznam a popis případů užití. Pro grafickou reprezentaci používá UML Use case diagram.

## Seznam účastníků:



*Ilustrace 2: Diagram účastníků*

### **Uživatel:**

Neregistrovaný uživatel, má možnost přidávání úkolů a prohlížení záznamů.

### **Klient:**

Klient je registrovaný návštěvník AIP s zvláštním právem přidávání úkolů a prohlížením vlastních záznamů.

### **Řešitel:**

Osoba s rolí Řešitel řeší úkoly zadané Klientem.

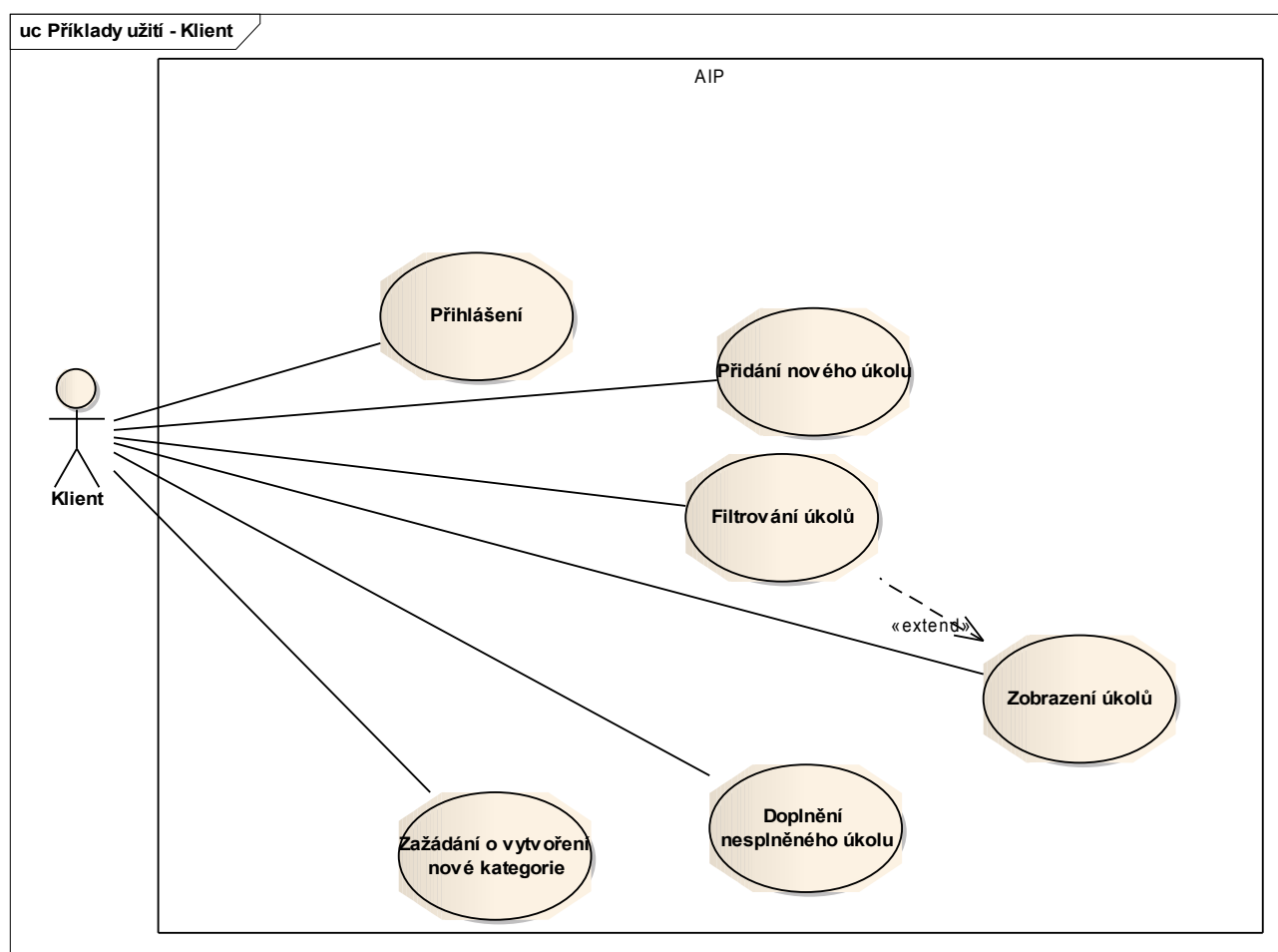
### **Ombudsman:**

Ombudsman je nadřízený všech řešitelů, má stejné povinnosti a práva. Navíc ještě musí rozřazovat úkoly pro řešitele. Ombudsman je pouze jeden.

### **Administrátor:**

Role uživatele Administrátor obsluhuje AIP, nestará se o plnění úkolů, pouze kontroluje činnost AIP a administruje uživatele, žádosti atd.

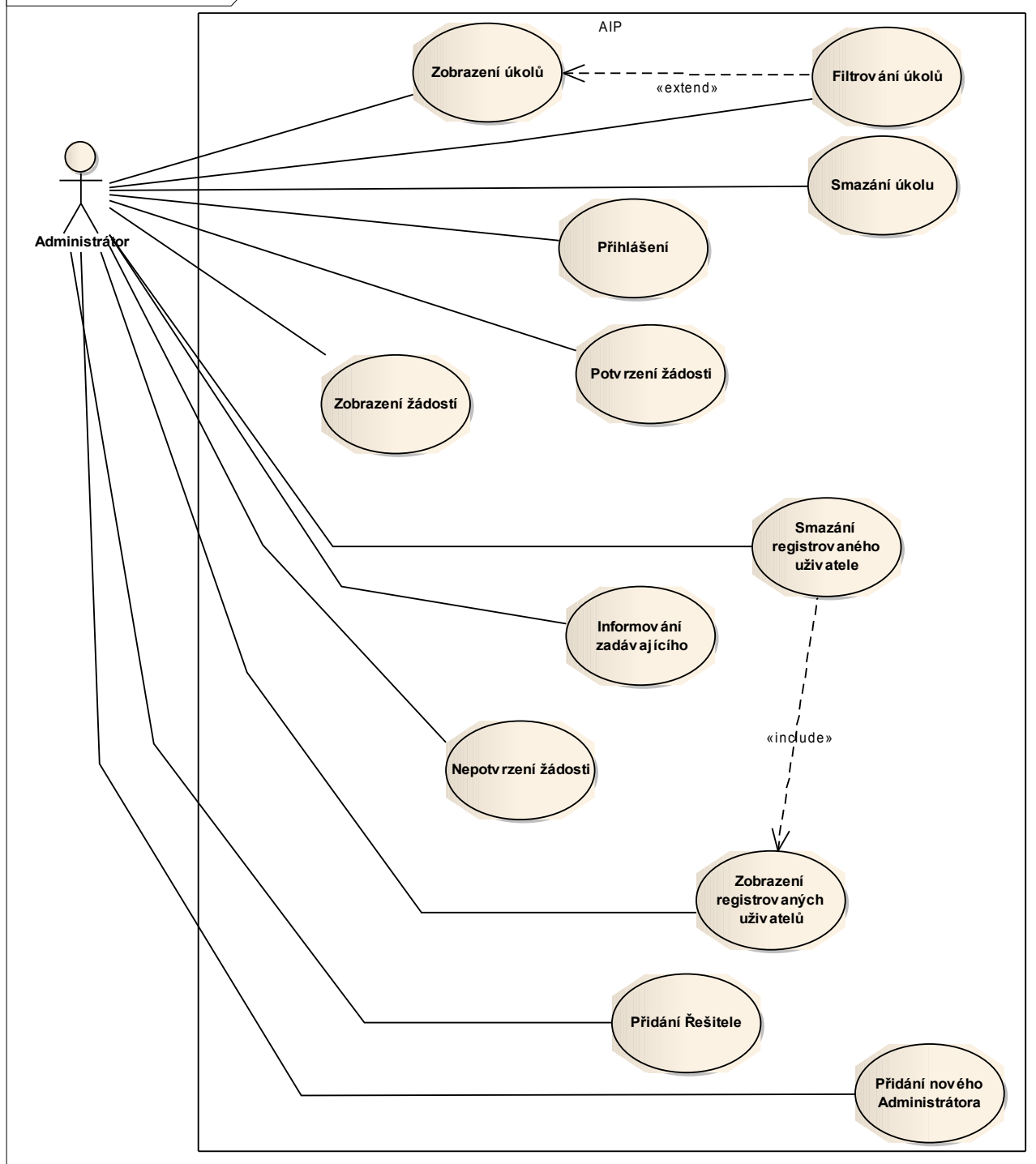
## Klient:



Ilustrace 3: Případy užití - Klient

## Administrátor:

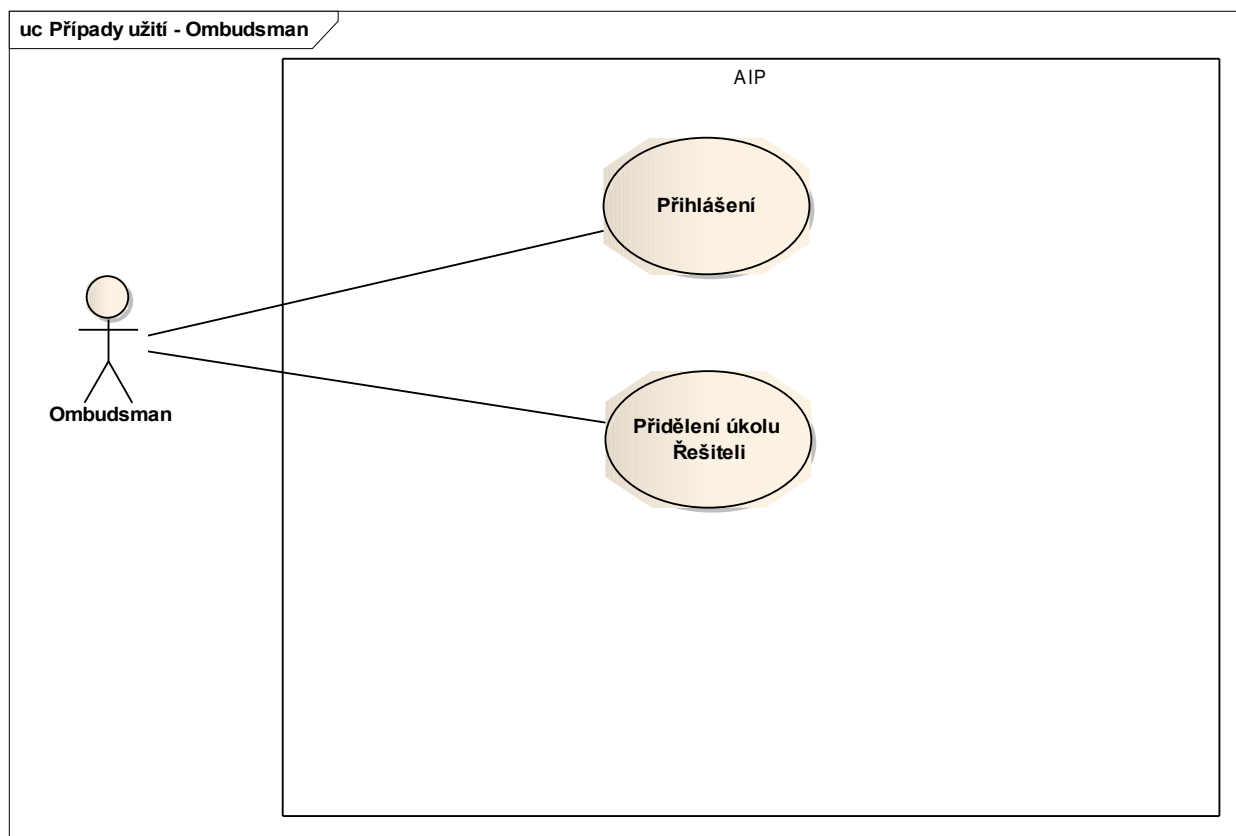
uc Případy užití - Administrátor



Ilustrace 4: Případy užití - Administrátor

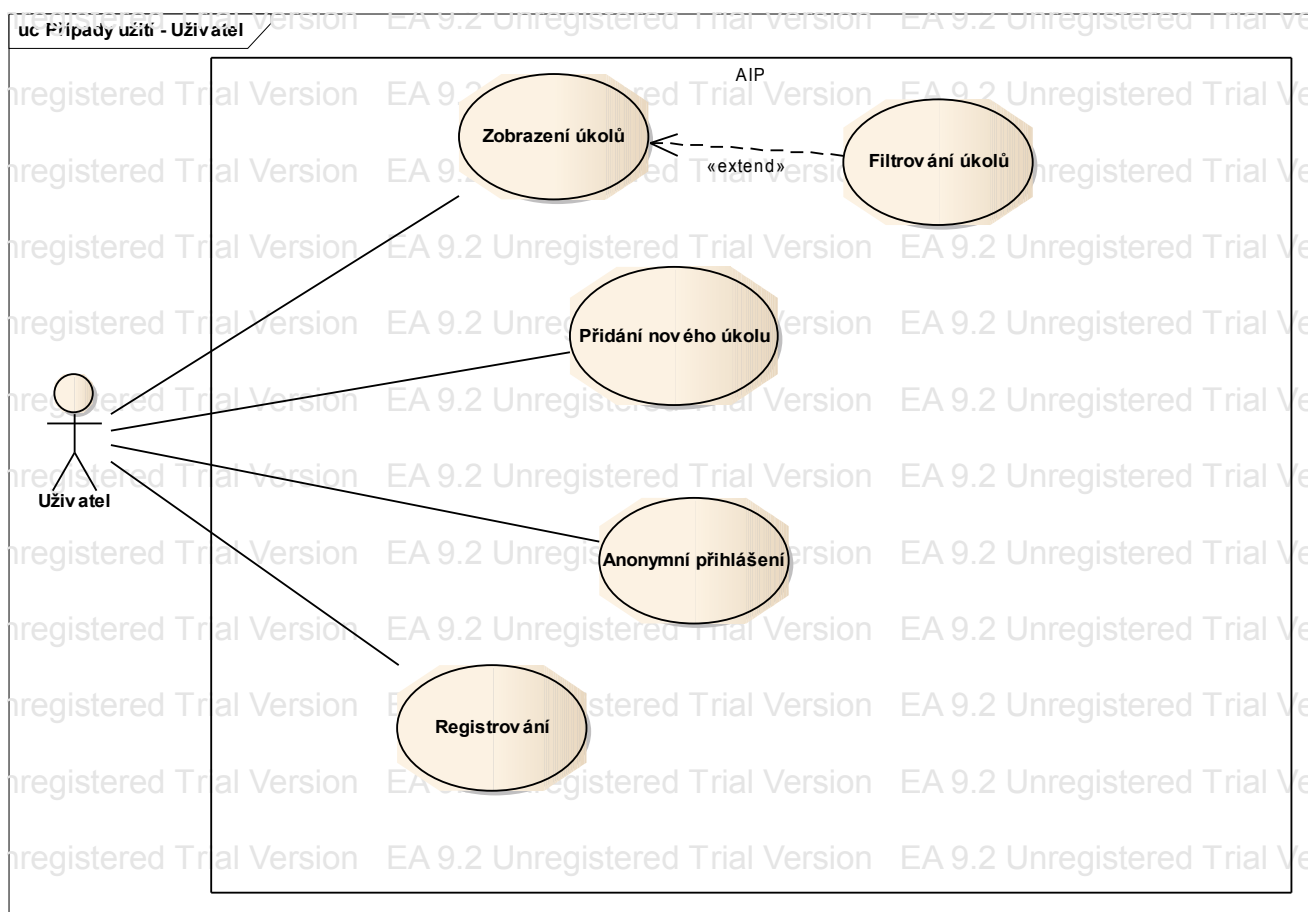


## Ombudsman:



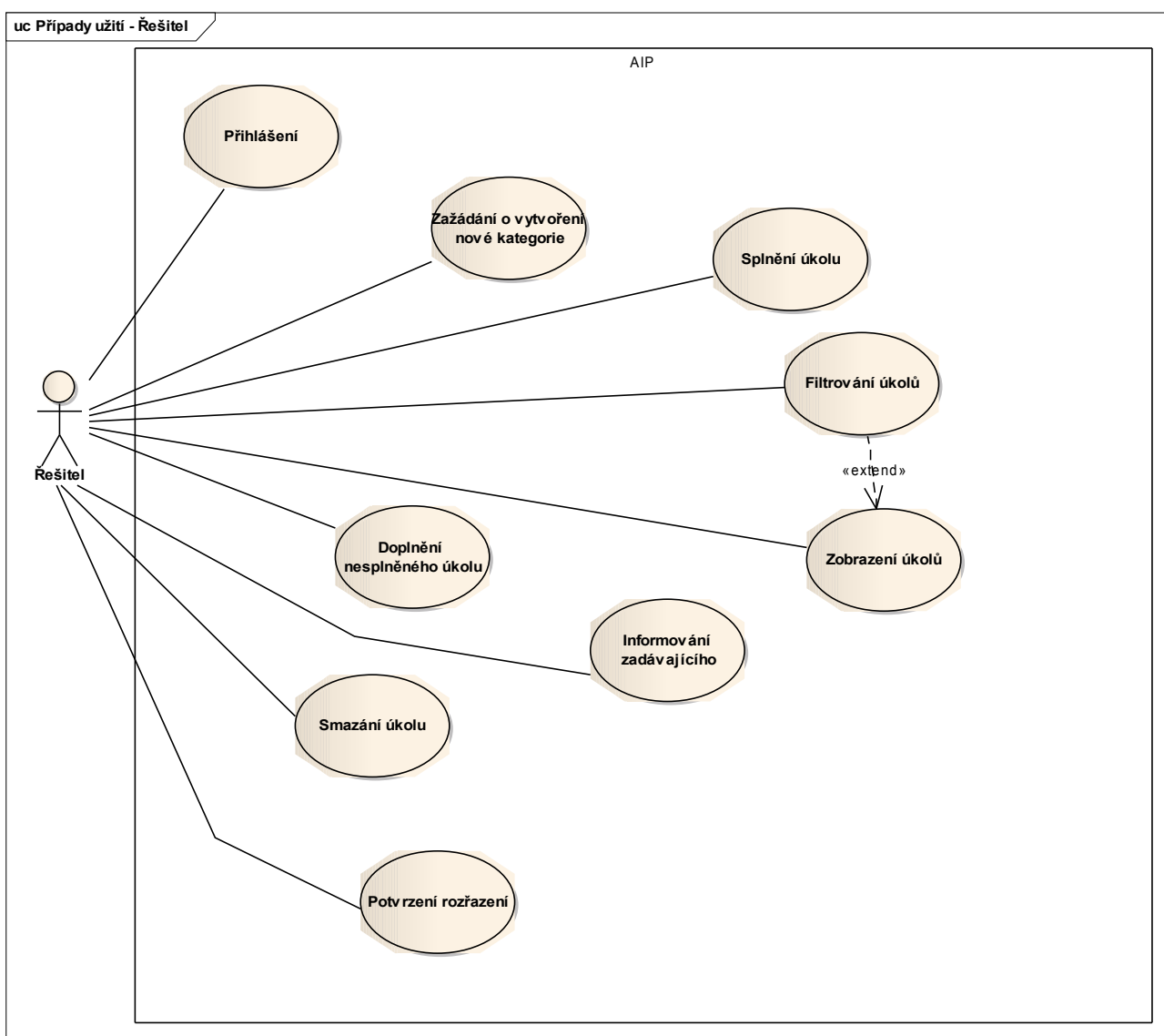
*Ilustrace 5: Případy užití - Ombudsman*

## Uživatel:



Ilustrace 6: Případy užití - Uživatel

## Řešitel:



Ilustrace 7: Případy užití - Řešitel

## Scénáře příadů užití:

### 1 Anonymní přihlášení

Standardní přihlášení neregistrovaného uživatele, v podstatě nejde o přihlášení, ale o pouhé připojení na stránky AIP.

#### Flow of Events

##### Basic Path

##### Anonymní přihlášení

1. Uživatel otevře aplikaci AIP.
2. Systém uživateli vygeneruje náhodné přihlašovací údaje a uživatel se přihlásí.
3. Systém zobrazí úvodní obrazovku uživatele

### 2 Doplnění nesplněného úkolu

Každý Klient nebo Řešitel bude doplňovat úkoly pomocí dopisování dalších informací.

#### Flow of Events

##### Basic Path

##### Doplnění nesplněného úkolu

1. Řešitel/Klient vybere nesplněný úkol.
2. Systém zobrazí podrobné informace o úkolu.
3. Řešitel/Klient zažádá o doplnění úkolu.
4. Systém zobrazí formulář pro doplnění úkolu.
5. Řešitel/Klient doplní potřebné informace o úkolu.
6. Řešitel/Klient potvrdí zadané informace.
7. Systém zobrazí potvrzení o přidání informací.

### 3 Filtrování úkolů

Při zobrazování úkolů bude možnost úkoly vyfiltrovat a rozfiltrovat podle jednotlivých podmínek.

#### Flow of Events

##### Basic Path

##### Filtrování úkolů

1. Osoba zadá požadavek o zobrazení vyfiltrovaných úkolů.
2. Systém otevře formulář pro filtrování úkolů.
3. Osoba vyplní filtrovací podmínky.
4. Systém zobrazí úkoly splňující podmínky zadané uživatelem.

## 4 Informování zadávajícího

Po provedení změn na žádosti nebo na úkolu bude zadávající informován.

### Flow of Events

#### Basic Path

##### Informování zadávajícího

1. Systém zobrazí informace o změně žádosti cílovému klientovi po jeho přihlášení na uvítací obrazovce.

## 5 Nepotvrzení žádosti

Podaná žádost od Řešitele nebo Klienta nemusí být nutně potvrzena.

### Flow of Events

#### Basic Path

##### Nepotvrzení žádosti

1. Administrátor si nechá zobrazit žádosti pomocí filtrovacího formuláře.
2. Systém zobrazí žádosti.
3. Administrátor si vybere žádost, kterou chce zamítnout.
4. Systém zobrazí podrobné informace o žádosti.
5. Administrátor vybere možnost zamítnutí žádosti.
6. Systém zobrazí formulář pro důvod zamítnutí.
7. Administrátor zadá důvod zamítnutí.
8. Administrátor potvrdí zamítnutí.
9. Systém zobrazí potvrzení zamítnutí.

## 6 Potvrzení rozřazení

Řešitel je povinen po přiřazení úkolu, tento úkol potvrdit.

### Flow of Events

#### Basic Path

##### Potvrzení rozřazení - přijetí

1. Systém zobrazí po přihlášení na uvítací obrazovce seznam přiřazených úkolů.
2. Řešitel vybere úkol k potvrzení.
3. Systém zobrazí podrobnosti o úkolu a jeho přidělenou kategorii.
4. Řešitel potvrdí úkol.
5. Systém uloží úkol mezi nesplněné.

#### Alternate

##### Potvrzení rozřazení - nepřijetí

1. Systém zobrazí po přihlášení na uvítací obrazovce seznam přiřazených úkolů.
2. Řešitel vybere úkol k potvrzení.
3. Systém zobrazí podrobnosti o úkolu a jeho přidělenou kategorii.
4. Řešitel nepotvrdí úkol a kategorii.
5. Řešitel vybere novou kategorii.
6. Systém přiřadí novou kategorii.
7. Systém uloží úkol mezi nesplněné

## 7 Potvrzení žádosti

Pokud se Administrátorovi žádost zamlouvá, může ji potvrdit.

### Flow of Events

#### Basic Path

##### Potvrzení žádosti

1. Administrátor si nechá zobrazit žádosti.
2. Systém zobrazí žádosti.
3. Administrátor s vybere žádost, kterou chce potvrdit.
4. Systém zobrazí podrobné informace o žádosti.
5. Administrátor vybere možnost potvrzení žádosti.
6. Administrátor potvrdí žádost.
7. Systém zobrazí potvrzení žádosti.

## 8 Přidání kategorie

Pokud je obsahem žádosti vytvoření nové kategorie a žádost je schválena, tak Administrátor vytvoří novou kategorii odpovídající žádosti.

### Flow of Events

#### Basic Path

##### Přidání kategorie

1. Administrátor zadá požadavek o zadání nové kategorie
2. Systém zobrazí formulář pro zadání údajů o kategorii.
3. Administrátor vyplní údaje o kategorii.
4. Administrátor potvrdí vyplněné informace.
5. Systém zobrazí potvrzení o přidání kategorie

## 9 Přidání nového Administrátora

Administrátor má možnost přidat jiného Administrátora.

### Flow of Events

#### Basic Path

##### Přidání nového Administrátora

1. Administrátor zažádá o přidání nového Administrátora.
2. Systém zobrazí formulář pro přidání nového Administrátora.
3. Administrátor vyplní údaje nového Administrátora.
4. Administrátor potvrdí vyplněné údaje.
5. Systém zobrazí potvrzení o přidání nového Administrátora.

## 10 Přidání nového úkolu

Klientův základní čin je přidávání úkolu, tj. informací od prohřešků.

### Flow of Events

#### Basic Path

##### Přidání nového úkolu

1. Klient zadá požadavek na přidání nového úkolu.
2. Systém otevře formulář s údaji o novém úkolu.
3. Klient vyplní údaje o novém úkolu.
4. Klient potvrdí vyplněné údaje.
5. Systém zobrazí potvrzení přidání úkolu.
6. Ombudsman přiřadí úkolu Řešiteli.

## 11 Přidání Řešitele

Přidání dalšího Řešitele do systému.

### Flow of Events

#### Basic Path

##### Přidání Řešitele

1. Administrátor zadá požadavek o zadání nového Řešitele
2. Systém zobrazí formulář pro zadání registračních údajů Řešitele.
3. Administrátor vyplní registrační údaje Řešitele.
4. Administrátor potvrdí zadané údaje.
5. Systém zobrazí potvrzení o přidání Řešitele

## 12 Přidělení úkolu Řešiteli

Ombudsman rozděljuje úkoly daným řešitelům

### Flow of Events

#### Basic Path

##### Přidělení úkolu Řešiteli

1. Úkol je zobrazen Ombudsmanovi.
2. Ombudsman zvolí Řešitele odpovídající kategorie.
3. Ombudsman potvrdí zadané údaje.
4. Systém zobrazí potvrzení o přiřazení úkolu danému Řešiteli.

#### Alternate

##### Přidělení úkolu Řešiteli - Řešení

1. Úkol je zobrazen Ombudsmanovi.
2. Ombudsman se rozhodne řešit úkol sám.
3. Ombudsman přiřadí úkol sám sobě.
4. Ombudsman potvrdí zadané údaje.
5. Systém zobrazí potvrzení o přiřazení úkolu Ombudsmanovi.

## 13 Přihlášení

Každý Registrovaný uživatel musí při příchodu do AIP projít přihlášením, které určí jeho roli a práva.

### Flow of Events

#### Basic Path

##### Přihlášení - přijetí

- 1.. Klient zadá požadavek na přihlášení
2. Systém otevře formulář pro zadání přihlašovacích údajů.
3. Klient zadá přihlašovací údaje.
4. Klient potvrdí údaje.
5. Systém ověřuje údaje.
6. Klient je vpuštěn do AIP, pokud je ověření platné. Systém zobrazí úvodní obrazovku pro daného klienta.

#### Alternate

##### Přihlášení - nepřijetí

1. Klient zadá požadavek na přihlášení
2. Systém otevře formulář pro zadání přihlašovacích údajů.
3. Klient zadá přihlašovací údaje.
4. Klient potvrdí údaje.
5. Systém ověřuje údaje.
6. Klient není vpuštěn do AIP, protože je ověření neplatné. Systém zobrazí informaci o neplatném přihlášení.

## 14 Registrování

Pokud chce Uživatel zvětšit své práva, zvláště přidávat úkoly, musí projít registrací.

### Flow of Events

#### Basic Path

##### Registrování

1. Uživatel zadá požadavek o registraci.
2. Systém otevře formulář pro registraci.
3. Uživatel vyplní potřebné údaje pro registraci.
4. Uživatel ukončí a odešle registrační žádost.
5. Systém odešle tuto žádost Administrátorovi.



## 15 Smazání registrovaného uživatele

Pokud Administrátor uváží, že některý reg. uživatel by měl být smazán , může ho označit blokovat/smazat.

### Flow of Events

#### Basic Path

##### Smazání reg. uživatele

1. Administrátor vybere reg. uživatele k smazání.
2. Systém vypíše podrobné informace o uživateli.
3. Administrátor označí uživatele jako smazaného.
4. Systém otevře formulář k podání vysvětlení o smazání.
5. Administrátor vyplní důvod smazání.
6. Administrátor potvrdí smazání.
7. Systém zobrazí informaci o smazání reg. uživatele.

## 16 Smazání úkolu

Pokud je úkol nesmyslný nebo nesplnitelný, může být označen jako smazaný/blokovaný.

### Flow of Events

#### Basic Path

##### Smazání úkolu

1. Administrátor/Řešitel vybere nesplněný úkol, který chce smazat.
2. Systém zobrazí podrobné údaje pro daný úkol.
3. Administrátor/Řešitel vybere možnost smazání úkolu.
4. Systém se zeptá na potvrzení smazání úkolu.
5. Administrátor/Řešitel potvrdí smazání.
6. Systém zobrazí potvrzení o smazání.

## 17 Splnění úkolu

Po vyřešení úkolu zaneše Řešitel do systému potřebné informace a "úkol splní".

### Flow of Events

#### Basic Path

##### Splnění úkolu

1. Řešitel vybere nesplněný úkol, který chce řešit.
2. Systém zobrazí podrobné údaje pro daný úkol.
3. Řešitel vybere možnost uzavření úkolu.
4. Systém otevře formulář pro doplnění úkolu.
5. Řešitel doplní informace, případné dokumenty, fotografie.
6. Systém se zeptá na potvrzení uzavření úkolu.
7. Řešitel potvrdí uzavření.
8. Systém uzavře úkol a přesune jej do uzavřených.
9. Systém zobrazí potvrzení splnění úkolu.

## 18 Zažádání o vytvoření nové kategorie

Pokud se úkol nedá zařadit do žádné kategorie, je třeba zažádat o vytvoření nové kategorie.

### Flow of Events

#### Basic Path

##### Zažádání o vytvoření nové kategorie

1. Reg. uživatel/Klient zažádá o vytvoření nové kategorie.
2. Systém otevře formulář pro vytvoření nové kategorie.
3. Reg. uživatel/Klient vyplní údaje nové kategorie.
4. Reg. uživatel/Klient potvrdí žádanku.
5. Žádost se přesune Administrátorovi.
6. Systém zobrazí potvrzení o vytvoření žádosti.

## 19 Zobrazení registrovaných uživatelů

Administrátor potřebuje k administraci uživatelů zobrazení všech reg. uživatelů.

### Flow of Events

#### Basic Path

##### Zobrazení registrovaných uživatelů

1. Administrátor zadá požadavek o zobrazení registrovaných uživatelů.
2. Systém zobrazí formulář pro vyplnění podmínek uživatelů.
3. Administrátor vyplní podmínky filtrování uživatelů
4. Systém zobrazí pouze ty uživatele, kteří splňují dané podmínky.

## 20 Zobrazení úkolů

Každá osoba má možnost zobrazení úkolu bez filtrace.

### Flow of Events

#### Basic Path

##### Zobrazení úkolů

1. Osoba zadá požadavek na zobrazení úkolů.
  2. Osoba potvrdí zadané podmínky.
  3. Systém zobrazí úkoly.
- <extend> filtrování úkolů

## 21 Zobrazení žádostí

Administrátor k administraci potřebuje, aby mohl zobrazovat všechny žádosti.

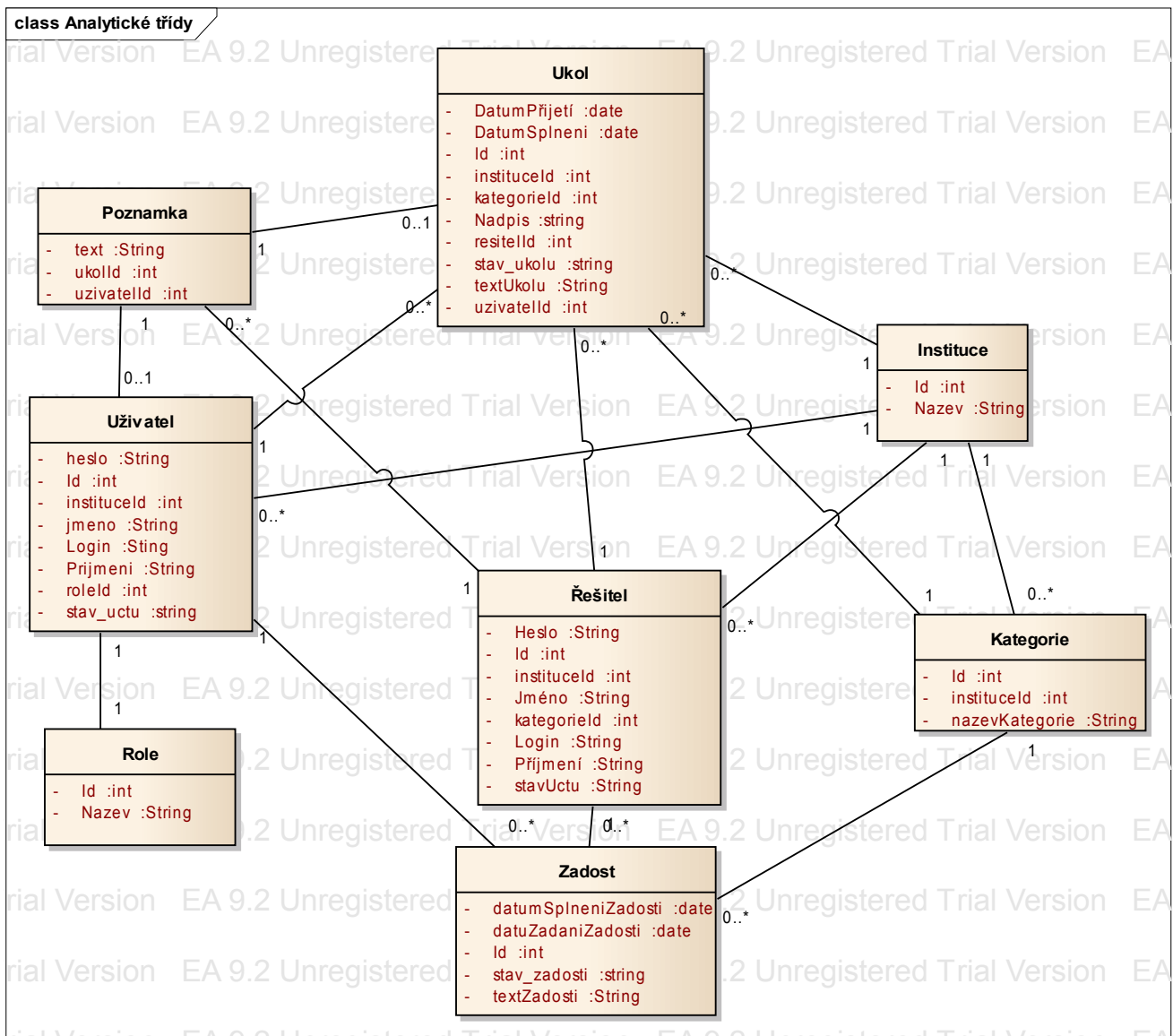
### Flow of Events

#### Basic Path

##### Zobrazení žádosti

1. Administrátor zadá požadavek na zobrazení žádostí.
2. Administrátor potvrdí zadané podmínky.
3. Systém zobrazí žádosti.

## Doménový model



Ilustrace 8: Diagram analytických tříd

### Úkol:

Úkol je základní informační jednotka celého systému, je zadávána Klienty a řešena Řešiteli.

### Řešitel:

Představuje řešitele, který řeší úkoly.

### Uživatel:

Představuje běžného uživatele systému.

### Poznámka:

Poznámka je způsob vazby Řešitele s úkolem. Řešitel produkuje doplnění při řešení úkolů. Doplnění může vytvořit i Klient pokud doplní úkol o další informace.

**Žádost:**

Třída označující požadavky od Klientů a Řešitelů

**Kategorie:**

Kategorie určuje oblast problému.

**Institute:**

Zaobaluje celý system aby mohl být použit vícekrát bez nutnosti opakované instalace.

## Platforma/Frameworky

Elektronický ombudsman bude vytvořen na platformě Java Standart Edition, deployment tedy odpovídá této platformě. Pro fungování aplikace je potřebný aplikační server. Aby mohl aplikační server. Pro práci s webovým rozhraním používá aplikace framework Java Spring a pro práci s databází to je Hibernate.

**Příklady aplikačních serverů:**

Pro fungování aplikace je tedy třeba aplikační server, na výběr je mnoha open source i placených možností.

- Open Source aplikační servery
- GlassFish
- JBoss
- Sun Java System Application Server
- Oracle AS

**Software potřebný k otevření projektu:**

Prostředí podporující programování v Java SE a Java Sping a Hibernate (Netbeans 6.9.1 a novější)  
Jeden z výše zmíněných aplikačních serverů.

## Instalace MySQL 5.x

Databáze MySQL je natolik základní databází, že je velmi obtížné nalézt hosting který ji nepodporuje. Ideální je se na toto informovat přímo u poskytovatele Vašeho hostingu. Pokud databáze MySQL nainstalována je, je třeba v ní vytvořit požadovanou strukturu tabulek. To lze udělat pomocí aplikace phpMyAdmin či v ideálním případě spuštěním skriptu createDatabaseTables.sql, který naleznete jako součást aplikace. Ten jednoduše vytvoří celou strukturu databáze za Vás. Pokud hostujete aplikaci na Vašem vlastním serveru, bude třeba MySQL nainstalovat. Ideální je pro tyto účely instalační balíček MySQL community server, který je zdarma a poskytuje dostatečnou rychlost a komfort práce. Stáhnout si jej můžete z webu <http://dev.mysql.com/downloads/mysql/>. Při tvorbě samotné databázové struktury postupujte stejně jako v předchozím případě. Nyní máte databázi MySQL nainstalovanou a připravenou k použití.

## Konfigurace Hibernate

Knihovny potřebné k běhu perzistentní vrstvy Hibernate jsou zahrnuty ve zdrojových souborech aplikace, není tedy nutné stahovat a nikam umisťovat jakékoli soubory. Je ovšem třeba editovat konfigurační soubor Hibernate a uvést v něm údaje potřebné pro připojení k databázi: adresu databáze, jméno databáze, uživatelské jméno a heslo pro připojení k databázi. Tyto informace je třeba zanesť do souboru hibernate.cfg.xml. Tento soubor je možno editovat jakýmkoli textovým editorem, jen je třeba dát si pozor aby byl uložen zas jako xml soubor. Od tohoto okamžiku bude aplikace schopna se připojit k databázi a pracovat s uloženými daty. Pokud bude aplikace vykazovat chyby, zkontrolujte, zda jste správně nainportovali strukturu databáze z předchozího kroku.

## Připojení k aplikaci

Abychom zkontrolovali, zda proběhla instalace aplikace v pořádku, je třeba se k ní připojit pomocí webového rozhraní. Pokud aplikace neposkytuje nějakou chybovou hlášku, je vše v pořádku.

Odkazy na předešlou analýzu:

[Předešlá analýza](#) (Enterprise architect dokument)

[Stránka projektu](#) (Google code)

Oblasti vyznačené **modrou** barvou označují změny od původní analýzy (odkaz výše).