

# Vysoké učení technické v Brně

## Fakulta informačních technologií

IDS - Databázové systémy 2019

### **Projektová dokumentace**

### **Zadání číslo 49 – Bug tracker**

Dominik Harmim – xharmi00  
František Horázný – xhoraz02

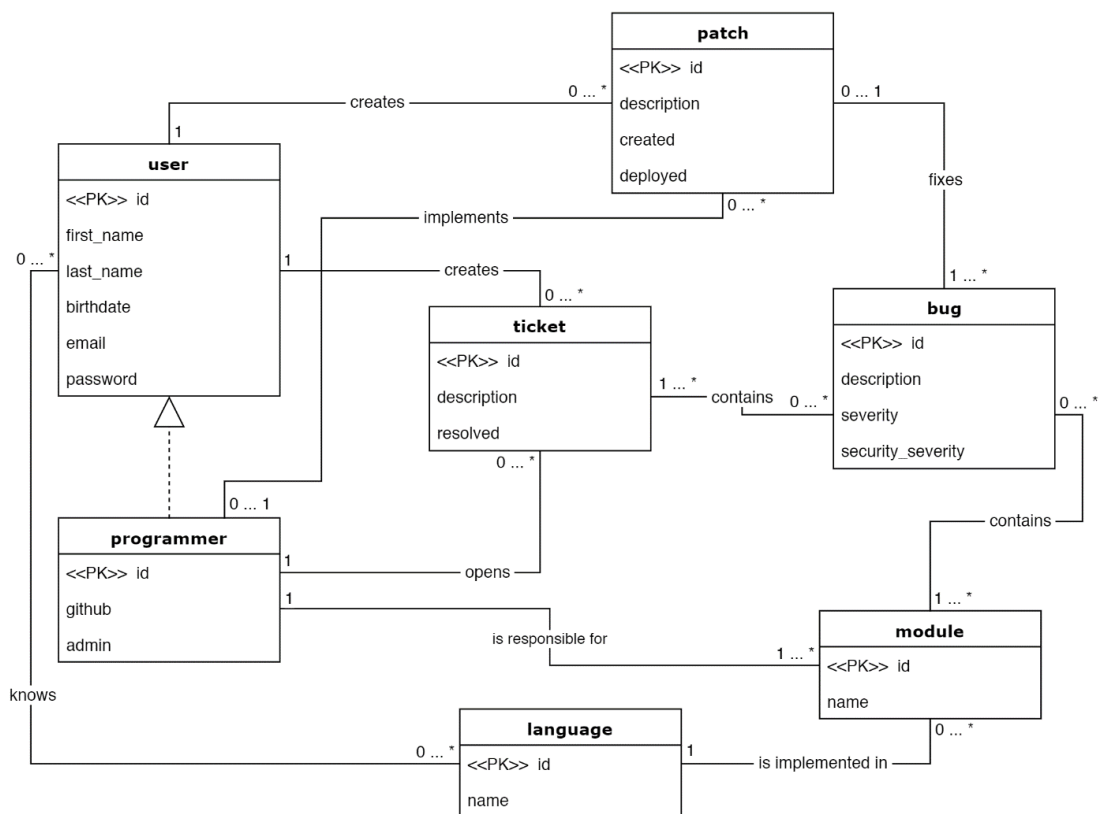
## OBSAH

1. Zadání: .....	3
2. Entity relationship diagram .....	3
3. Use case diagram.....	4
4. Implementace.....	5
a. DROP.....	5
b. CREATE .....	5
c. INSERT .....	5
d. TRIGGER.....	5
e. EXPLAIN PLAN.....	5
f. PROCEDURE .....	5
g. PRIVILEGES.....	5
h. MATERIALIZED VIEW .....	5

## 1. Zadání:

Vytvořte informační systém pro hlášení a správu chyb a zranitelností systému. Systém umožňuje uživatelům hlásit bugy, jejich závažnosti a moduly, ve kterých se vyskytly, ve formě tiketů. Tikety mohou obsahovat hlášení o více než jednom bugu a stejný bug může být zhlášen více uživateli. Bug může (ale nemusí) být zranitelností a v tomto případě zaevidujeme i potenciální míru nebezpečí zneužití této zranitelnosti. V případě zhlášení bugů, odešle systém upozornění programátorovi, který zodpovídá za daný modul, přičemž může odpovídat za více modulů. Programátor pak daný tiket zabere, přepne jeho stav na "V řešení" a začne pracovat na opravě ve formě Patche. Patch je charakterizován datem vydání a musí být schválen programátorem zodpovědným za modul, které mohou být v různých programovacích jazycích. Jeden Patch může řešit více bugů a současně řešit více tiketů a vztahuje se na několik modulů. Samotní uživatelé mohou rovněž tvořit patche. Takové patche však musí projít silnější kontrolou, než jsou zavedeny do systému. Kromě data vytvoření patche rovněž evidujte datum zavedení patche do ostrého provozu. Každý uživatel a programátor je charakterizován základními informacemi (jméno, věk, apod.), ale současně i jazyky, kterými disponuje, apod. V případě opravení bugů, mohou být uživatele upozorněni na danou opravu a případně být odměněni peněžní hodnotou (podle závažnosti bugu či zranitelnosti).

## 2. Entity relationship diagram



### 3. Use case diagram



## 4. Implementace

### a. DROP

V první řadě je potřeba případné již vytvořené tabulky zrušit, aby nedocházelo ke kolizím.

### b. CREATE

Jako další krok, je vytvoření všech tabulek včetně jejich primárních a cizích klíčů a pomocných tabulek.

### c. INSERT

Následně vložíme příkladové nebo základní data.

### d. TRIGGER

Zadáno bylo vytvořit 2 triggery,

přičemž první slouží k vytvoření primárního klíče uživatele, které se dá použít i jako pořadí registrace uživatelů.

Druhý trigger se vyvolá při vytvoření nového uživatele. Vytvoří hash zadaného hesla a uloží ho.

### e. EXPLAIN PLAN

Explain plan ukazuje detail určitého příkazu. U nás je tímto select přes 2 tabulky, který ukáže seznam uživatelů, kteří jsou programátoři a k nim počet jazyků, které ovládají. Po vytvoření indexu by se mělo zrychlit vyhledávání, neboť systém nebude muset procházet všechny usery sekvenčně.

### f. PROCEDURE

První procedura slouží k výpisu statistik ohledně počtu ticketů, bugů a useru a poměru mezi nimi.

Druhá procedura vypočítá na základě argumentu, kolik uživatelů ovládá daný jazyk.

### g. PRIVILEGES

Přidělení práv druhému členovi týmu.

### h. MATERIALIZED VIEW

Materializovaný pohled, který obsahuje souhrn uživatelů a počet jejich podaných ticketů. Tento pohled se dá využít například pro určení nejaktivnějších nebo naopak nečinných uživatelů.