**Specifikacija zahtjeva**

**Assosciation Rule Mining – Seattle Police Department Incident Response**

**PA 1**

24. 03. 2019.

**Sažetak**

Tema ovog projekta je Association Rule Mining. To je metoda za otkrivanje relacija između vrijednosti atributa u velikim bazama podataka. Pronalaze se pravila o vezama između vrijednosti atributa pomoću kojih se može predvidjeti vrijednosti atributa na temelju vrijednosti drugog atributa. Dobili smo bazu podataka u kojoj su evidentirane reakcije policije na pozive građana u Seattleu. Nadalje, trebamo otkriti vezu između različitih događaja te njihovih atributa.

**Ciljana publika**

U našem slučaju ciljana publika su profesor i asistent koji pregledavaju naše rješenje te predanu dokumentaciju. U stvarnom svijetu to bi bili potencijalni kupci koji su zainteresirani za naš proizvod.

**Članovi projektnog tima**

1. Projektni manager – Luka Babić

2. Stručnjak za specifikaciju – Borna Gilja

3. Stručnjak za oblikovanje – Kristijan Knežić

4. Stručnjak za testiranje – Luka Šarlija

5. Stručnjak za integraciju – Marin Markanjević

**Verzija kontrole dokumenta**

|  |  |  |  |
| --- | --- | --- | --- |
| **Verzija** | **Primarni autor(i)** | **Opis verzije** | **Datum završetka** |
| Radna | Borna Gilja | Početak pisanja plana | 24.3.2019 |

1. **UVOD**

(...)

1. **SPECIFIČNI ZAHTJEVI**
   1. **Zahtjevi vanjskog sučelja**

(...)

* + 1. **Korisničko sučelje**

(...)

* + 1. **Hardwaresko sučelje**

(...)

* + 1. **Softwaresko sučelje**

(...)

* + 1. **Komunikacijski protokoli**

(...)

* 1. **Svojstva softwareskog produkta**
  2. **Atributi softwareskog sustava**
     1. **Pouzdanost**

Softver smatramo pouzdanim ukoliko obavlja svoju zadaću u skladu sa specifikacijom bez grešaka koje mogu dovesti do neželjenih rezultata ili potpunog prestanka rada sustava. U praksi softversku pouzdanost češće izražavamo kao postotak ili očekivano vrijeme rada bez grešaka.

Jedan od načina poboljšavanja pouzdanosti programskog proizvoda je korištenje defenzivnim programiranjem tijekom razvoja proizvoda. U defenzivno programiranje ubrajamo tehnike za predviđanje potencijalnih pogrešaka i njihovo ispravljanje (na primjer, ako želimo da korisnik unese broj, trebamo biti oprezni u slučaju da je korisnik pogrešno unio znak druge vrste).

(...)

* + 1. **Dostupnost**

Programski proizvod može se koristiti (i biti će spreman za rad) odmah nakon preuzimanja te kako je to prenosiv proizvod, ne zahtijeva instalaciju prije korištenja.

* + 1. **Sigurnost**

(...)

* + 1. **Održivost**

(...)

* + 1. **Prenosivost**

Programski proizvod je moguće pokretati sa različitih operativnih sustava (Windows, Linux, Mac, ...). Kako je aplikacija izrađena u Java programskoj okolini, tako i njezino pokretanje zahtijeva prijašnju instalaciju programskog proizvoda JRE (**Java Runtime Environment**). JRE postavlja nekoliko hardwareskih uvjeta, ali uvjeti su takvi da veliki broj uređaja ispunjava te uvjete.

(...)

* + 1. **Preostale performanse**

(...)

* 1. **Zahtjevi baze podataka (ukoliko postoje)**

1. **PRILOZI**