Generički Rukovalac Dokumentima

Dokument arhitekture softvera

Verzija 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 13/11/2016 | 1.0 | Prva verzija dokumenta | Zorana Štaka |
| 15/11/2016 | 1.1 | Dodatak | Zorana Štaka |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Uvod 4

1.1 Svrha 4

1.2 Obim 4

1.3 Definicije i skraćenice 4

2. Arhitektonski prikaz 4

3. Arhitekturni ciljevi i ograničenja 4

4. Use-Case pogled 4

4.1 Use-Case realizacija 5

5. Veličina i performanse 5

6. Kvalitet 5

Dokument arhitekture softvera

# Uvod

Prva faza u procesu projektovanja softvera je projektovanje arhitekture. Arhitektura predstavnja temelj na kojem se dalje gradi softver. Arhitektura identifikuje glavne strukturne komponente u sistemu i odnose između njih. Uvod uključuje svrhu, obim, definicije i skraćenice.

## Svrha

Pruža se sveobuhvatni pregled na sistem, koristeći brojne različite arhitektonske poglede koji omogućavaju prikaz različitih aspekata sistema. Namjena je sa se uhvate i prenesu značajne arhitektonske odluke koje su napravljene u sistemu.

## Obim

Ovaj dokument opisuje arhitekturu sistema.

## Definicije i skraćenice

* MVC – Model – Viewer – Contoller

# Arhitektonski prikaz

Pri dizajniranju ovog sistema korišćena je arhitektonski šablon MVC. Ovaj šablon razdvaja podatke od načina pirikaza. To znači da jedan skup podataka može da se prikaže na više načina. Kontroler osluškuje i prihvata zahtjev od klijenata za izvršenje operacije. Nakon toga poziva operaciju koja je definisana u modelu, i ako model promjeni stanje, obavještava viwer o promjeni stanja. Ovim se dijeli i sisttem na module(manje jedinice koje obavljaju neku funkciju). Pored MVC šablona koriste se i kreacioni šabloni(Singleton, Abstract Factory, Builder, Composite, Observer).

U nastavku će biti objašnjeni i predstavljeni sljedeći pogledi: Use-Case(slučajevi korišćenja), logički, Deployment(Fizička arhitektura sistema).

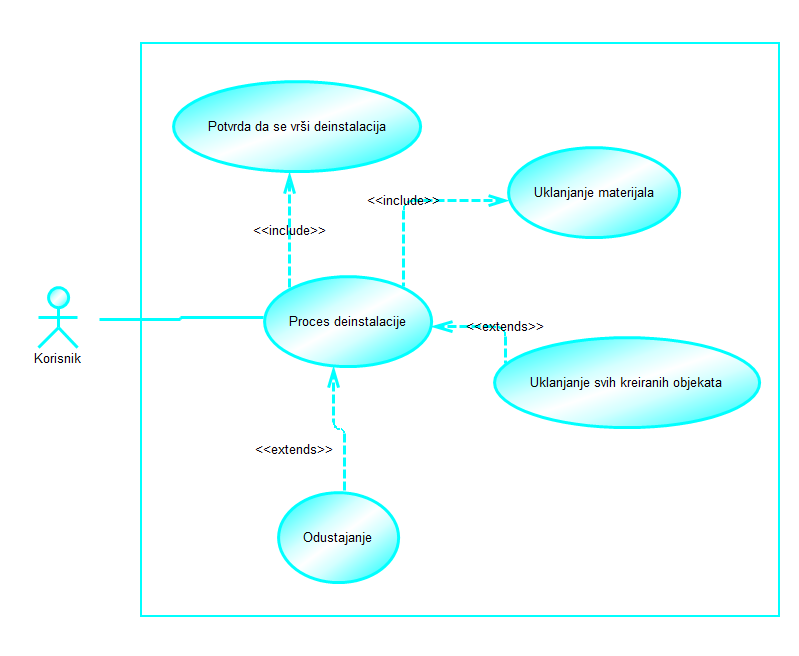
# Arhitekturni ciljevi i ograničenja

Ciljevi korišćenja MVC arhitekture su razdvajanje odgovornosti među klasma, lakše izmjene i jednostavno proširivanje. Arhitektura se projektuje tako da pronalazi kritične operacije unutar komponenti i umanji komunikciju među njima. Koristi se slojevita arhitektura gdje se kritični dijelovi nalaze u najdubljem sloju. Operacije koje su vezane za sigurnost su smještene u jednoj komponenti ili više podkomponenti. Arhitektura treba da bude projektovana tako da sadrži dodatne kompnente koje bi omogućile ažuriranje komponenti bez zaustavljanja sistema. Arhitektura treba da sadrži komponente koje se lako mogu izmjeniti.

# Use-Case pogled

Use-case pogled je veoma važan, jer opisuje interakciju korisnika i sistema. Use-case pogled opisuje skup scenarija. Služi za ocjenu kvaliteta softvera.

## Use-Case realizacija



# Veličina i performanse

Dopuniti

# Kvalitet

Softver će biti na raspolaganju 24 sata dnevno, 7 dana u sedmici.