
Elektrotehnički fakultet Istočno Sarajevo

**Generički Rukovalac Dokumentima
Dokument Arhitekture Softvera**

Verzija 1.0

Generički Rukovalac Dokumentima	Verzija: 1.0
Dokument Arhitekture Softvera	Datum: 21/11/2016

Istorija Revizija

Datum	Verzija	Opis	Autor
21/11/2016	1.0	Dokument Arhitekture Softvera	Nataša Pećanac

Generički Rukovalac Dokumentima	Verzija: 1.0
Dokument Arhitekture Softvera	Datum: 21/11/2016

Sadržaj

1.	Uvod	4
1.1	Svrha	4
1.2	Opseg	4
1.3	Definicije, akronimi i skraćenice	4
2.	Prikaz Arhitekture	4
3.	Ciljevi i ograničenja arhitekture	4
4.	Use-Case pogled	4
5.	Logički pogled	5
5.1	Pregled	5
5.2	Arhitektonski značajni dizajn paketi	5
6.	Pogled procesa	5
7.	Implementacioni pogled	5
7.1	Pregled	5
7.2	Slojevi arhitekture	5
8.	Veličina i performanse	6
9.	Kvalitet	6

Generički Rukovalac Dokumentima	Verzija: 1.0
Dokument Arhitekture Softvera	Datum: 21/11/2016

Dokument Arhitekture Softvera

1. Uvod

Dokumentacija Arhitekture Softvera daje pregled svih bitnih aspekata arhitekture softvera.

1.1 Svrha

Dokumentacija Arhitekture Softvera obezbjeđuje potpun pregled arhitekture sistema, koristeći nekoliko različitih arhitekturnih pogleda u cilju prikazivanja različitih aspekata sistema. Namjera je da se obuhvate i prikažu značajne arhitektonske odluke koje su donesene u sistemu.

1.2 Opseg

Dokumentacija Arhitekture Softvera daje pregled arhitekture softvera Generički Rukovalac Dokumentima. U okviru dokumenta dati su različiti pogledi arhitekture. To su pogled slučajeva upotrebe, logički pogled, implementacioni pogled, pogled podataka.

Ovaj dokument takođe obuhvata odjeljak u kome su dati veličina, performanse i kvalitet sistema.

1.3 Definicije, akronimi i skraćenice

Sve definicije, akronimi i skraćenice mogu se pronaći u Glossary dokumentaciji.

2. Prikaz Arhitekture

Arhitektura je u ovom dokumentu data kroz niz pogleda. To su:

- Use-Case view
- Logical view
- Process view
- Implementation view

Svi ovi pogledi su UML pogledi razvijeni uz pomoć SAP Sybase PowerDesigner alata.

Razvoj softvera se zasniva na MVC arhitekturi. Model – View – Controller arhitektura podrazumijeva da u okviru sistema postoje pogledi i modeli koji međusobno komuniciraju posredstvom kontrolera. Ovakva organizacija uslovljava postojanje različitih modula koji obavljaju pojedine funkcije.

Pored osnovnog MVC šablona u arhitekturu su ugrađeni i drugi šabloni kao što su: Composite, Command, State, Facrotty, Singleton,...

3. Ciljevi i ograničenja arhitekture

Neki osnovni zahtjevi koji imaju značajan uticaj na arhitekturu sistema su:

- Svi zahtjevi navedeni u requirements modelu moraju biti zadovoljeni
- Sve performanse i zahtjevi navedeni u dokumentu vizije i dokumentu dodatne specifikacije moraju biti uzeti u obzir pri implementaciji sistema.
- Sistem treba omogućiti korisnicima rad sa dokumentima na način specificiran u zahtjevima korisnika
- Zbog MVC arhitekture sistema zahtjeva se jasno razdvajanje modula i njihovih ovlaštenja, te precizno definisanje njihove saradnje
- Zbog MVC arhitekture javlja se veliki broj klasa
- Cilj je omogućiti kasnije proširenje sistema uz oslonac na MVC arhitektur.

4. Use-Case pogled

Use-Case view je bitan za selekciju skupa scenarija i/ili slučajeva upotrebe koji su od najvećeg značaja. On opisuje skup scenarija i/ili slučajeva upotrebe koji predstavljaju neku značajnu, centralnu funkcionalnost. Takođe opisuje skup scenarija i/ili slučajeva upotrebe koji su značajno pokriveni arhitekturom (koji koriste mnoge arhitektonske elemente) ili koji prikazuju specifičnu, delikatnu tačku arhitekture.

Slučajevi upotrebe su:

Generički Rukovalac Dokumentima	Verzija: 1.0
Dokument Arhitekture Softvera	Datum: 21/11/2016

- Rad sa dokumentima
- Rad sa radnim prostorom
- Rad sa projektom
- Rad sa pojedinačnim dokumentom
- Rad sa stranicom
- Rad sa slotom
- Rad sa elementom

Detaljan opis slučajeva korištenja nalazi se u:

- Use-Case dijagramu sistema
- Use-Case dokumentaciji

5. Logički pogled

Opisuje najvažnije klase, njihovu organizaciju u podsisteme i pakete, kao i organizaciju tih podsistema i paketa. Takođe daje opis najvažnijih realizacija slučajeva upotrebe.

5.1 Pregled

Sistem se zasniva na MVC arhitekturi te su i pojedinačne cjeline organizovane na takav način. Model – View – Controller arhitektura zahtjeva jasno razdvajanje prikaza od samih podataka. U okviru sistema mogu postojati različiti pogledi koji se oslanjaju na isti model i na iste podatke. Kontroler treba da sinhronizuje rad pogleda i modela.

5.2 Arhitektonski značajni dizajn paketi

Najvažniji paketi su Model, View, Controller i Command.

- Model – obuhvata klase modele. Ove klase zadužene su za rad sa podacima. Pored upravljanja korisničkim podacima u okviru ovih klasa obuhvaćeno je i upravljanje podacima bitnim za sistem.
- View – obuhvata klase pogleda. Ove klase zadužene su za prikaz podataka korisnicima na odgovarajući način. Za jedan model može postojati više pogleda.
- Controller – obuhvata klase kontrolera. Ove klase zadužene su za sinhronizaciju odgovarajućih modela i pogleda tako da se podaci iz modela na adekvatan način prikazuju u pripadajućim pogledima.
- Command – obuhvata klase pojedinačnih komandi. Kao bitan dio ovog paketa imamo implementaciju undo i redo komandi.

6. Pogled procesa

Opisuje zadatke uključene u izvršenje sistema, njihove interakcije i konfiguracije. Takođe opisuje alokaciju objekata i klasa u zadacima.

Obavljanje bilo kog procesa u okviru sistema zasniva se na saradnji klasa. Kontroler opslužuje korisničke akcije tako što upravlja i sinhronizuje rad pogleda i modela.

7. Implementacioni pogled

Ovaj dio dokumenta daje opis implementacionog modela, dekompoziciju softvera u slojeve i arhitektonski značaj pojedinačnih komponenti.

7.1 Pregled

U ovom dijelu dat je pregled slojeva arhitekture. U nastavku se definiše način razdvajanja arhitekture sistema na pojedinačne slojeve koji imaju specifične funkcije. Dat je opis tih funkcija i način saradnje različitih slojeva.

7.2 Slojevi arhitekture

Prezentacioni sloj – najviši sloj. Obezbeđuje prikaz podataka korisniku.

Generički Rukovalac Dokumentima	Verzija: 1.0
Dokument Arhitekture Softvera	Datum: 21/11/2016

Sloj veze – kontrolni sloj – obezbjeđuje razdvajanje pogleda i podataka. U okviru sistema to su klase kontrolera.

Sloj podataka – najniži sloj koji nije direktno dostupan korisniku. Sloj podataka čuva konkretne strukture podataka i zna kako da im pristupa. U okviru sistema to su klase modela.

8. Veličina i performanse

Prva verzija softvera obuhvata samo osnovne funkcionalnosti koje su implementirane kroz relativno mali broj klasa.

9. Kvalitet

Uz kvalitetnu implementaciju MVC arhitekture softvera trebao bi se postići značajan kvalitet sistema koji ima osobine proširljivosti i prenosivosti.