

Dokument softverske arhitekture

Verzija 1.0

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 21.04.2024. | 1.0 | Opis arhitekture sistema, pogled na njene dijelove sa različitih stanovišta | Helena Ćupina |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Uvod 4

1.1 Svrha 4

1.2 Opseg 4

1.3 Pojmovi, skraćenice i akronimi 4

1.4 Reference 4

1.5 Pogled 4

2. Predstavljanje arhitekture 4

3. Arhitektonski ciljevi i ograničenja 5

4. Pregled slučajeva korištenja 5

5. Logički pogled 5

5.1 Arhitektonski značajni dizajn paketi 5

6. Pogled na isporuku 5

7. Pogled na implementaciju 6

7.1 Slojevi 6

7.1.1 Model 6

7.1.2 Pogled 6

7.1.3 Kontroler 6

8. Pogled na skladištenje podataka 6

9. Veličina i performanse 7

10. Kvalitet 7

softverske arhitekture

# Uvod

Dokument softverske arhitekture obuhvata sve bitne detalje o strukturi i organizaciji softvera u razvoju, pružajući detaljan uvid u njegovu arhitekturu. Arhitektura softvera je ključni faktor u procesu dizajniranja i projektovanja softvera, često zahtjevajući pažljivo planiranje i definisanje kako bi se osigurala funkcionalnost, skalabilnost i održivost proizvoda. Kroz dokumentaciju arhitekture, timovi mogu jasno razumeti komponente, interakcije i zahtjeve sistema, što olakšava razvoj, održavanje i nadogradnju softvera tokom njegovog životnog ciklusa.

## Svrha

Dokument softverske arhitekture detaljno opisuje arhitekturu softvera u razvoju kroz različite perspektive, omogućavajući jasan uvid u različite segmente sistema. Svojom raznolikošću pogleda, dokument pruža sveobuhvatan prikaz arhitektonskih odluka koje su donesene tokom implementacije informacionog sistema. Dokument služi kao ključni resurs za razvojni tim, pružajući im osnovu za dalje planiranje, implementaciju i održavanje softverskog rješenja.

## Opseg

Dokument softverske arhitekture pruža sveobuhvatan uvid u arhitekturu sistema Law Bridge kroz različite perspektive i aspekte. Dokument predstavlja ključan resurs za članove grupe 2 koji rade na razvoju informacionog sistema, omogućavajući im da jasno razumiju strukturu, komponente i interakcije sistema. Kroz konsultaciju dokumenta, članovi grupe 2 mogu bolje koordinisati svoje napore, implementirati arhitektonske principe i donositi odluke koje će doprinijeti efikasnom i održivom razvoju Law Bridge sistema.

## Pojmovi, skraćenice i akronimi

Definicije svih nepoznatih pojmova, svi akronimi i skraćenice koje su korišćene u Dokumentu softverske arhitekture opisane su u dokumentu Rječnik, koji se nalazi u sklopu dokumentacije sistema.

## Reference

[1] Specifikacija slučajeva korištenja

[2] Plan isporuke

[3] Specifikacija softverskih zahtjeva

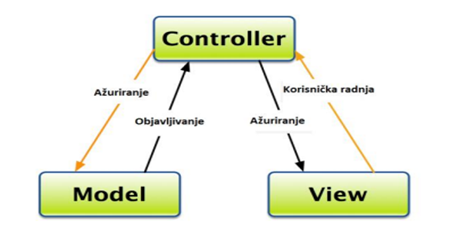
[4] Smjernice za dizajn.

## Pogled

U nastavku dokumenta se prvenstveno pruža pregled arhitekture informacionog sistema Law Bridge, obuhvatajući opis arhitekture sistema iz perspektive implementacije, slučajeva korištenja i logičkog pogleda. Osim toga, dokument sadrži smjernice koje se odnose na kvalitet softverskog proizvoda, kao i osvrtanje na veličinu i performanse sistema. Sekcije pružaju detaljan uvid u različite aspekte arhitekture, pružajući članovima tima korisne smjernice i informacije tokom razvojnog procesa..

# Predstavljanje arhitekture

U procesu razvoja aplikacije, primjenjuje se MVC arhitektura kako bi se postigla dobro strukturisana osnova. Ova arhitektura omogućava laku nadogradnju i proširenje sistema, dok istovremeno pruža jasno razdvajanje između različitih slojeva aplikacije. MVC, što označava Model-View-Controller, pruža organizovan pristup dizajnu softvera kroz jasno definisane komponente, omogućavajući efikasno programiranje i održavanje softverskih rješenja. Model se fokusira na obradu podataka i njihovo upravljanje, pogled na prezentaciju podataka korisnicima, dok kontroler omogućava komunikaciju između modela i pogleda, osiguravajući glatko funkcionisanje aplikacije.

*Slika 1.* MVC okvir

# Arhitektonski ciljevi i ograničenja

Prilikom dizajniranja informacionog sistema, ključni aspekti uključuju lakoću implementacije i efikasnu detekciju grešaka. Da bi se postigla visoka organizacija i modularnost, primjenjuju se različiti dizajnerski obrasci koji omogućavaju jasno razgraničenje odgovornosti unutar pojedinih komponenti sistema. Pristup ne samo da poboljšava kvalitet strukture sistema, već i olakšava buduće proširenje i održavanje.

Osim toga, jedan od primarnih ciljeva je i zaštita integriteta te privatnosti podataka koji se obrađuju unutar sistema. Implementira se niz bezbjednosnih mijera koje štite podatke od neovlašćenog pristupa i zloupotrebe. Takođe, sistem prati i čuva istoriju svih promjena na podacima, što je jako bitno.

# Pregled slučajeva korištenja

Pregled slučajeva korištenja daje listu scenarija iz modela koji se odnose na slučajeve korištenja softverskog proizvoda koji se razvija. Detaljan pregled svih dijagrama i scenarija za sve slučajeve korištenja su priloženi u dokumentu Specifikacija slučajeva korištenja.

# Logički pogled

U sekciji opisani su arhitektonski ključni segmenti modela dizajna koji su podjeljeni u različite pakete. Za informacioni sistem "Law Bridge", ova dekompozicija obuhvata tri paketa: model, pogled i kontroler. U sljedećem dijelu dokumenta nalazi se detaljan pregled i opis ovih paketa.

## Arhitektonski značajni dizajn paketi

* Observer je dizajnerski šablon ponašanja koji uspostavlja mehanizam pretplate, omogućavajući više objekata da budu automatski obaviješteni o promjenama koje se dešavaju u objektu koji prate.
* Command je šablon ponašanja koji zahtjev transformiše u objekat koji sadrži sve potrebne informacije o tom zahtjevu. Ova karakteristika omogućava da se zahtjevi prosljeđuju kao argumenti metodama, odlažu ili stave u red za izvršavanje, a takođe podržava i operacije koje zahtjevaju mogućnost poništavanja.
* Model obuhvata klase koje predstavljaju stanje sistema i kako su te informacije predstavljene u operativnoj memoriji računara. Ove klase su ključne za čuvanje i upravljanje podacima koji su neophodni za funkcionalnost softvera.
* View paket sastoji se od klasa koje određuju kako se podaci iz Modela prikazuju korisnicima. Ovaj paket je odgovoran za vizuelni prikaz podataka u korisničkom interfejsu.
* Controller paket djeluje kao veza između Modela i Viewa, upravljajući interakcijama između njih i osiguravajući pravilnu komunikaciju i ažuriranje podataka.Pogled na isporuku

# Pogled na isporuku

U sekciji dokumenta je dat opis mrežnih konfiguracija na kojima se softver dostavlja i pokreće. Svaki računar mora da ima sljedeće specifikacije da bi se softverski proizvod mogao instalirati i pokretati bez problema: procesor- Intel Core i5 ili ekvivalentni AMD Processor, memorija- 8GB RAM ili više, hard disk- 500GB, operativni sistem - Microsoft Windows 10 i mlađi, rezolucija ekrana - 1280x800. Detaljni pregled isporuke biće dat u dokumentu Plan isporuke gdje su navedene sve pojedinosti koje su potrebne da bi se softverski proizvod dostavio kupcu i koje su obaveze isporučioca softvera tokom tog procesa.

# Pogled na implementaciju

Implementacija softverskog proizvoda "Law Bridge" zasniva se na MVC (Model-View-Controller) arhitektonskom šablonu, koji predstavlja osnovu cjelokupnog sistema. Svaki od slojeva – model, pogled i kontroler – razvijen je primjenom odgovarajućih dizajnerskih šablona, što doprinosi efikasnijoj implementaciji, fleksibilnosti pri proširenjima i pojednostavljenju održavanja softvera. U dokumentu su detaljno opisani slojevi i njihove komponente kako bi se olakšalo razumijevanje strukture i funkcionalnosti sistema.

Dodatno, korišćenje dizajn šablona omogućava bolju organizaciju koda i smanjuje mogućnost grešaka tokom razvoja i nadogradnje. Takođe, šabloni podržavaju reusability, što znači da se komponente mogu ponovo koristiti u različitim dijelovima aplikacije ili čak u drugim projektima. Kroz naredne sekcije dokumenta, korisnici i razvijači mogu dobiti dublji uvid u specifičnosti implementacije svakog sloja, uz praktične primjere i preporuke za njihovu primjenu.

## Slojevi

### Model

Komponenta Model u MVC arhitekturi predstavlja centralnu logiku za upravljanje podacima sa kojima korisnik komunicira. Komponenta ne samo da posreduje u prenosu podataka između View i Controller komponenti, već takođe upravlja svim podacima relevantnim za poslovne procese. Klase unutar Modela zadužene su za preuzimanje podataka iz baze, njihovu obradu i ažuriranje, kao i za pripremu podataka koji će biti prikazani korisnicima.

Klase igraju ključnu ulogu u osiguranju integriteta i dosljednosti podataka unutar aplikacije, implementirajući različite poslovne pravilnike i validacije. Na ovaj način, Model osigurava da su svi podaci koji se koriste i prikazuju ažurirani i tačni, što direktno utiče na pouzdanost i efikasnost cijelokupnog softverskog rješenja.

### Pogled

Komponenta Pogled (View) u arhitekturi aplikacije odgovorna je za svu logiku vezanu za korisnički interfejs. Na primjer, korisnički interfejs obuhvata sve elemente s kojima korisnik direktno interaguje, poput tekstualnih polja, padajućih menija, dugmadi i drugih grafičkih komponenti.

Komponenta je ključna za pružanje intuitivnog i efikasnog korisničkog iskustva, omogućavajući korisnicima da lako prolaze kroz aplikaciju i efektivno izvršavaju željene zadatke. Pored toga, Pogled je zadužen za estetski izgled aplikacije, što uključuje dizajn, raspored i stil svih interfejsnih elemenata, čime se osigurava da aplikacija ne samo da funkcioniše dobro, već i izgleda privlačno.

### Kontroler

Kontroleri funkcionišu kao ključni posrednici između Modela i Pogleda, obrađujući poslovnu logiku i zahtjeve koji stižu, manipulišući podacima preko Modela i komunicirajući s Pogledom radi prikazivanja konačnog rezultata. Na primjer, korisnički kontroler upravlja interakcijama i unosima iz korisničkog interfejsa te ažurira bazu podataka koristeći model korisnika. Kontroler se takođe koristi za pregledavanje korisničkih podataka.

# Pogled na skladištenje podataka

Baza podataka koja se koristi za razvoj informacionog sistema "Law Bridge" je SQL baza podataka, a za skladištenje podataka koristi se Microsoft SQL Server. Server omogućava izvođenje različitih vrsta transakcija, integraciju poslovne logike, kao i analizu potrebnih podataka. Modelovanje baze podataka obavlja se pomoću alata SAP PowerDesigner, što omogućava efikasno i precizno dizajniranje struktura podataka. U projektu je takođe uključen detaljan dokument pod nazivom "Smjernice za dizajn", koji pruža upute i standarde za dizajniranje baze podataka. Smjernice pomažu u osiguranju da je baza podataka dobro organizovana za performanse i sigurna, što je ključno za uspješno funkcionisanje cijelokupnog sistema.

# Veličina i performanse

Izabrana softverska arhitektura podržava ključne zahtjeve kada govorimo o veličini i performansama. Ključni zahtjevi su detaljno opisani u dokumetu Specifikacija softverskih zahtjeva koji je priložen uz prijektnu dokumentaciju.

# Kvalitet

Kvalitet softvera obuhvata dvije ključne dimenzije: funkcionalni kvalitet, koji se određuje prema tome koliko softver zadovoljava postavljene funkcionalne zahtjeve, i strukturni kvalitet, koji se tiče usklađenosti sa nefunkcionalnim zahtjevima. Softverski proizvod "Law Bridge" dizajniran je s ciljem potpune realizacije kako funkcionalnih, tako i nefunkcionalnih zahtjeva definisanih tokom faze prikupljanja zahtjeva. Veći stepen ispunjenja ovih zahtjeva direktno rezultuje boljim kvalitetom softvera.