

Smjernice za dizajn

Verzija 1.0

Istorija revizija

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 01.04.2024. | 1.0 | Inicijalizacija fajla. Uvod. Opšte smjernice za projektovanje i implementaciju. | Ivana Elez |
| 01.04.2024. | 1.0 | Smjernice za dizajn baze podataka. Smjernice za arhitektonsko projektovanje. Smjernice za mehanizme. | Helena Ćupina |
| 02.04.2024. | 1.0 | Dodat logo projekta. | Helena Ćupina |
| 02.04.2024. | 1.0 | Izmjena dokumenta, dodat naziv projekta. | Ivana Elez |

Sadržaj

[1. Uvod 4](#_Toc162899327)

[1.1 Svrha 4](#_Toc162899328)

[1.2 Opseg 4](#_Toc162899329)

[1.3 Definicije, akronimi i skraćenice 4](#_Toc162899330)

[1.4 Reference 4](#_Toc162899331)

[1.5 Pregled 4](#_Toc162899332)

[2. Opšte smjernice za projektovanje i implementaciju 4](#_Toc162899333)

[2.1 Stil programiranja 4](#_Toc162899334)

[2.2 Rukovanje greškama i izuzecima 4](#_Toc162899335)

[2.3 Upravljanje memorijom 5](#_Toc162899336)

[2.4 Dokumentacija koda 5](#_Toc162899337)

[2.5 Dokumentovanje operacija 5](#_Toc162899338)

[2.6 Upotreba gotovih biblioteka 5](#_Toc162899339)

[2.7 Mapiranje od dizajna do implementacije 5](#_Toc162899340)

[2.8 Specifikacija interfejsa za podsisteme 5](#_Toc162899341)

[2.9 Struktura projekta 5](#_Toc162899342)

[3. Smjernice za dizajn baze podataka 5](#_Toc162899343)

[4. Smjernice za arhitektonsko projektovanje 6](#_Toc162899344)

[4.1 Model 7](#_Toc162899345)

[4.2 Pogled 7](#_Toc162899346)

[4.3 Kontroler 7](#_Toc162899347)

[5. Smjernice za mehanizme 7](#_Toc162899348)

Smjernice za dizajn

# Uvod

Dokument smjernica za dizajn obuhvata detaljan opis standarda, principa i konvencija koje se primjenjuju u procesu dizajniranja sistema. Ove smjernice su ključne jer pružaju jasan putokaz dizajnerima kako da kreiraju korisničko iskustvo koje je funkcionalno, intuitivno i estetski prihvatljivo. Kroz ovaj dokument, timovi za dizajn dobijaju smjernice o tome kako da postignu koherentnost, dosljednost i efikasnost u dizajniranju sistema, što olakšava implementaciju i održavanje istih.

## Svrha

Cilj dokumenta je da pruži sveobuhvatan pregled standarda dizajna, konvencija i idioma koji će biti primijenjeni u procesu dizajniranja sistema. Takođe, dokument sadrži detaljan opis pravila za standarde, upravljanje podacima, i listu konvencija koje će biti korišćene. Ovaj pristup omogućava jasno definisanje kako će se podaci konstruisati i organizovati unutar baze podataka, čime se osigurava dosljednost i efikasnost u razvoju sistema.

## Opseg

Kao zadatak imamo da realizujemo softver koji je pristupačan, jasan i lagan za klijente, a koji se temelji na preciznim smjernicama za razvoj dizajn sistema. Dokument ne samo da obuhvata smjernice za arhitekturu softvera, već i detaljno definiše sve zahtjeve klijenata kako bi se osiguralo da softver ispuni njihove potrebe. Takođe, arhitektura softvera postavlja temelje za dalji razvoj sistema, osiguravajući njegovu skalabilnost i održivost u budućnosti.

## Definicije, akronimi i skraćenice

Svi korišćeni nepoznati pojmovi, definicije, akronimi I skraćenice su opisane u dokumentu Rječnik.

## Reference

Software Design Principles and Guidelines, <https://www.dre.vanderbilt.edu/~schmidt/PDF/dp4.pdf>, posjećeno 01.04.2023.

## Pregled

Opis svih elemenata dizajna sistema je organizovan u odgovarajuće dijelove radi poboljšanja preglednosti, što olakšava pronalaženje željenog aspekta dizajna sistema.

# Opšte smjernice za projektovanje i implementaciju

Dio opštih smjernica za projektovanje i implementaciju opisuje principe koji se koriste prilikom dizajna i implementacije sistema. Kategorije su podijeljene u više tačaka:

## Stil programiranja

U strukturi projekta, paketi se nazivaju malim slovima, dok imena klasa počinju velikim početnim slovom. U slučaju višerječnih imena klasa, svaka nova riječ počinje velikim slovom. Metode započinju malim slovom, ali svaka nova riječ u imenu metode počinje velikim slovom. Svi nazivi paketa, klasa, metoda i atributa moraju biti na engleskom jeziku. Blokovi koda se označavaju vitičastim zagradama koje se otvaraju u istom redu kao i deklaracija metode, klase ili konstruktora. Konstante koje sadrže ključnu riječ "final" pišu se velikim slovima.

## Rukovanje greškama i izuzecima

Try-catch je osnovni mehanizam za rukovanje izuzecima. Kod koji može generisati izuzetak se stavlja unutar bloka try, dok se svi Java izuzeci koji se mogu javiti u tom kodu hvataju i obrađuju u blokovima catch. Metoda omogućava hvatanje različitih vrsta Java izuzetaka.

## Upravljanje memorijom

Java programski jezik uključuje "Garbage Collector" koji automatski upravlja memorijom, oslobađajući programera od brige o ručnom upravljanju memorijom. Ukoliko se desi da objekat nije obrađen od strane Garbage Collector-a, programer može ručno uništiti objekat korišćenjem destruktora.

## Dokumentacija koda

JavaDoc je fleksibilan sistem za generisanje dokumentacije koji analizira posebno formatirane komentare u Java izvornom kodu i generiše detaljnu dokumentaciju. Obično se koristi za kreiranje API dokumentacije u obliku HTML web stranica. Svaka metoda i klasa u kodu moraju biti dokumentovane, što uključuje opis klase/metode, povratnu vrijednost (ako postoji), parametre metode i ime autora. Tokom razvoja aplikacije, dozvoljeni su i inline komentari kako bi olakšali implementaciju programera, a za njih ne postoje posebna ograničenja.

## Dokumentovanje operacija

Dokumentacija se pravi u skladu s konvencijama za Java programski jezik. Pri implementaciji operacija na interfejsu, koriste se konvencionalni simboli za standardne softverske operacije. Takođe, jasno grafičko prikazivanje elemenata u editoru ima veliki značaj. Konačno, tekstualni opisi poput naziva elemenata i pomoćnih poruka (eng. tooltips) se koriste radi dodatne jasnoće i razumljivosti.

## Upotreba gotovih biblioteka

Za izradu softvera koriste se gotove biblioteke:

* Localization – biblioteka koja omogućava lokalizaciju softvera. Lokalizacija je prilagođavanje proizvoda ili usluge kako bi se zadovoljile potrebe određenog jezika, odnosno regiona u kome se koristi aplikacija. Omogućen prikaz na srpskom jeziku (latinični i ćirilično pismo) i engleskom jeziku.
* JTDS – 100% čist Java JDBC 3.0 drajver otvorenog koda za Microsoft SQL Server (6.5, 7, 2000, 2005, 2008 i 2012) i Sybase Adaptive Server Enterprise (10, 11, 12 i 15). jTDS je baziran na FreeTDS-u i trenutno je najbrži JDBC drajver spreman za proizvodnju za SQL Server i Sybase ASE. Kompatibilan je sa JDBC 3.0.

## Mapiranje od dizajna do implementacije

MVC arhitektura se primjenjuje kako bi se postigla dobro strukturirana aplikacija, omogućavajući jednostavnu proširivost sa programerske perspektive. Za modelovanje se uglavnom koristi softverski alat Power Designer, koji olakšava kreiranje osnovnih paketa MVC modela, detaljno specifikovanje klasa, interfejsa i modela, kao i definisanje veza među njima.

## Specifikacija interfejsa za podsisteme

U paketu Pogled (View) se definira izgled aplikacije, uključujući interfejse različitih podsistema koji čine standardizovan izgled grafičkog editora za algoritme. Objekti poput alatne trake, meni trake i statusne trake oblikuju vizuelni identitet softvera.

## Struktura projekta

Projekat treba da bude organizovan u logičke cjeline koje su reprezentovane kao paketi, sa jasnom definicijom koje datoteke pripadaju kojem paketu. Svi paketi bi trebalo da budu smješteni unutar paketa "src" ili "source" kao njegovi podpaketi. Klase koje se odnose na model treba da budu u paketu "models", pogledi u paketu "views", a kontroleri u paketu "controllers". Pored toga, treba obezbijediti direktorijum za smještaj ostalih resursa aplikacije, kao što su ikonice, slike i slično.

# Smjernice za dizajn baze podataka

Microsoft SQL Server je sistem za upravljanje relacionim bazama podataka (RDBMS) koji podržava širok spektar aplikacija za obradu transakcija, poslovnu inteligenciju i analitiku u korporativnim IT okruženjima. Uz Oracle Database i IBM-ov DB2, SQL Server je jedna od vodećih tehnologija baza podataka na tržištu.

Kao i ostali RDBMS softveri, Microsoft SQL Server koristi SQL kao standardizirani programski jezik za upravljanje bazama podataka i ispitivanje podataka. SQL Server je specifičan po Transact-SQL (T-SQL) implementaciji, koja proširuje standardni SQL sa vlastitim programskim ekstenzijama.

SQL Server se temelji na strukturi tabele zasnovanoj na redovima, koja povezuje podatke iz različitih tabela i izbjegava redundanciju podataka u bazi. Osim toga, relationalni model pruža referentni integritet i druge integritetske provjere kako bi se očuvala tačnost podataka i osiguralo pouzdano upravljanje transakcijama.

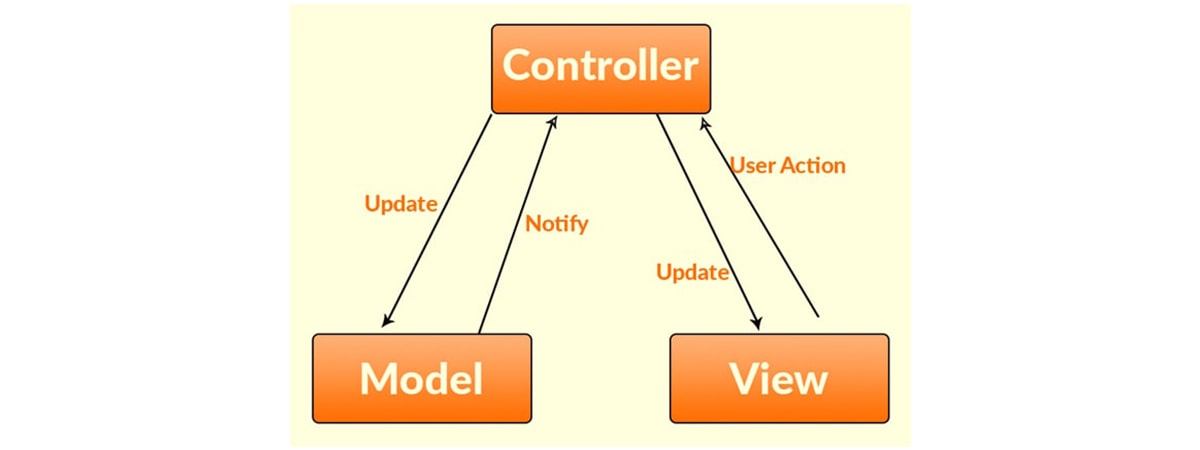
Glavna komponenta Microsoft SQL Servera je SQL Server Database Engine, koji upravlja skladištenjem, obradom i sigurnošću podataka. Database Engine obrađuje SQL naredbe, upravlja datotekama baze podataka, tabelama, indeksima, transakcijama i drugim aspektima baze podataka. Korisnici, uključujući SQL Server DBA-e i programere, koriste T-SQL naredbe za administraciju baze podataka, manipulaciju podacima, sigurnosnu zaštitu i druge zadatke.

# Smjernice za arhitektonsko projektovanje

Pri razvoju aplikacije, koristi se MVC arhitektura kako bi se postigla dobro strukturirana osnova. Ovakav model omogućava laku nadogradnju i proširenje te je jednostavan za razumijevanje sa strane programera. Modelovanje se obavlja pomoću softverskog alata Power Designer, koji omogućava kreiranje osnovnih paketa MVC modela i detaljno specificiranje potrebnih klasa, interfejsa i modela, kao i uspostavljanje veza među njima.

MVC označava Model View Controller i služi kao osnova za dizajn softvera. Softverska arhitektura pruža smjernice za konstrukciju ovih elemenata, dajući model kako bi trebali biti organizovani. Aplikacija se dijeli na tri osnovne komponente: model, pogled (View) i kontroler (Controller). Ova podjela omogućava jasno odvajanje logike obrade podataka od programiranja korisničkog interfejsa, olakšavajući razvoj aplikacija u objektno orijentisanim programskim jezicima.

MVC aplikacija komunicira s korisnikom putem pogleda, gdje korisnik šalje zahtjeve kontroleru, a kontroler zatim od modela traži izvršenje radnji i vraćanje rezultata. Rezultate zatim kontroler šalje u pogled, gdje ih korisnik može vidjeti. Ovaj proces je prikazan na Slici 1.



Slika 1. MVC okvir

## Model

Prvi nivo obuhvata lociranje svih podataka vezanih za sistem i njegov rad, uključujući obradu sadržaja sistema i moguća ažuriranja. Poslovna logika se odnosi na način na koji sistem manipuliše, transformiše i mijenja informacije radi njihove vizualizacije. Model omogućava korisnicima pristup informacijama bez potrebe za pretraživanjem različitih datoteka; dovoljno je samo unijeti potrebne parametre, a model će kroz prikaz pokazati željene informacije. Za izmjene u sistemu koristi se "kontroler", koji šalje informacije modelu. Umjesto direktnog korištenja SQL-a, informacije se često manipulišu kroz druge kontrolere i fokusiraju se na druge dijelove podataka, koji se sastoje od klasa i objekata.

## Pogled

Prikaz je stepenica na kojoj se informacije koje model šalje predstavljaju korisniku i prikazuju sadržaj u optičkom interfejsu. Odražava se programski kod, na kojem će se raditi i prikazati u korisničkom interfejsima.

## Kontroler

Kontroler je odgovoran za obradu zahtjeva korisnika koji su inicirani kroz model. Zahtjevi mogu obuhvatati pristup informacijama, uređivanje, kreiranje ili pretraživanje različitih vrsta podataka. On omogućava prilagođavanje načina prikaza informacija i načina na koji će se informacije obrađivati i predstavljati u "prikazu", što može uključivati promjene u načinu na koji model prezentuje informacije. Kontroler djeluje kao posrednik između modela i pogleda, prihvatajući zahtjeve i omogućavajući im da budu prikazani na različite načine. Konačni cilj kontrolera je osigurati da sve informacije budu pravilno obrađene od početka do kraja, ispunjavajući potrebe aplikacije ili programa koji se razvija.

# Smjernice za mehanizme

Mehanizmi za projekat “Law Bridge” su jednostavni, zahtijevajući osnovno poznavanje rada sa računarima i bazama podataka. Za korisnike koji nisu upoznati sa aplikacijama, dostupno je korisničko uputstvo i online pomoć, koje pruža detaljna objašnjenja o različitim dijelovima aplikacije, kao i često postavljena pitanja (FAQ).