Menza na pragu

Arhitekturni projekat

Verzija 1.0

Pregled izmena

|  |  |  |  |
| --- | --- | --- | --- |
| **Datum** | **Verzija** | **Opis** | **Autor** |
| 11.04.2021. | 1.0 | Inicijalna verzija | Miroslav, Damir i Stefan |
| 08.05.2021. | 1.1 | Prerađena verzija | Miroslav, Damir i Stefan |
|  |  |  |  |
|  |  |  |  |

Sadržaj

1. Cilj dokumenta 5

2. Opseg dokumenta 5

3. Reference 5

4. Predstavljanje arhitekture 5

5. Ciljevi i ograničenja arhitekture 5

6. Pogled na slučajeve korišćenja 5

6.1 Dijagrami slučajeva korišćenja 6

6.2 Kratak opis slučajeva korišćenja 6

6.2.1 Odabir jezika 6

6.2.2 Naručivanje 6

6.2.3 Uplata novca 6

6.2.4 Komentarisanje 6

6.2.5 Praćenje narudžbine 6

6.2.6 Potvrda dostave 6

6.2.7 Prijem narudžbine 7

6.2.8 Prosleđivanje narudžbine 7

6.2.9 Održavanje menija 7

6.2.10 Prijavljivanje 7

6.2.11 Registracija 7

6.2.12 Održavanje aplikacije 7

6.2.13 Konfigurisanje aplikacije 7

7. Pogled na logičku arhitekturu sistema 7

7.1 Pregled arhitekture – organizacija paketa i podsistema u slojeve 8

7.1.1 Korisnički interfejs 8

7.1.2 Aplikaciona logika 8

7.1.3 Pristup podacima 8

7.1.4 Razor stranice 8

7.1.5 ASP .Net 8

7.1.6 MySQL 8

8. Pogled na procese 9

8.1 Procesi 9

8.1.1 Web čitač 9

8.1.2 Web server 9

8.1.3 MySQL Server 9

8.1.4 Razor stranice 9

8.1.5 ASP .Net 9

9. Pogled na raspoređivanje sistema 10

9.1 Klijent 10

9.2 Web server 10

9.3 DBMS server 10

10. Pogled na implementaciju sistema 10

10.1 Model domena 10

10.2 Šema baze podataka 11

10.3 Komponente sistema 11

10.3.1 Komponente korisničkog interfejsa 12

10.3.2 Komponente aplikacione logike 12

10.3.3 Pristup podacima 12

11. Performanse 12

12. Kvalitet 13

Arhitekturni projekat

# Cilj dokumenta

Cilj ovog dokumenta je detaljni opis arhitekture aplikacije Menza na pragu.

# Opseg dokumenta

Dokument se odnosi na aplikaciju Menza na pragu koja će biti razvijen od strane tima Quickpix. Namena aplikacije je da omogući korisnicima jednostavno naručivanje i dostavu hrane iz menze kada nisu u mogućnosti da idu u linijski restoran.

# Reference

Spisak korišćene literature:

1. D01\_Predlog\_Projekta.
2. D02\_Vizija\_Sistema.
3. D03\_Raspored\_Aktivnosti.
4. D03\_Plan\_Realizacije.
5. D04\_Spec\_Zahteva.

# Predstavljanje arhitekture

Arhitektura sistema u dokumentu je prikazana kao serija pogleda na sistem: pogled na slučajeve korišćenja, pogled na logičku arhitekturu sistema, pogled na procese, pogled na razmeštaj komponenti sistema i pogled na implementaciju. Ovi pogledi su predstavljeni odgovarajućim UML dijagramima.

# Ciljevi i ograničenja arhitekture

Ključni zahtevi i sistemska ograničenja koja imaju značajan uticaj na izbor arhitekture i projektovanje sistema su:

1. Menza na pragu će biti implementirana kao Web aplikacija zasnovana na Razor stranicama i MySQL DMBS-u.
2. Klijentski deo aplikacije Menza na pragu će biti optimizovan za sledeće Web čitače: Microsoft Edge, Chrome, Opera kao i Firefox (Mozilla).
3. Svi zahtevi u pogledu performansi dati u moraju biti uzeti u obzir pri izboru arhitekture i razvoju sistema.

# Pogled na slučajeve korišćenja

U ovom odeljku je dat pogled na slučajeve korišćenja definisane u specifikaciji zahteva [5].

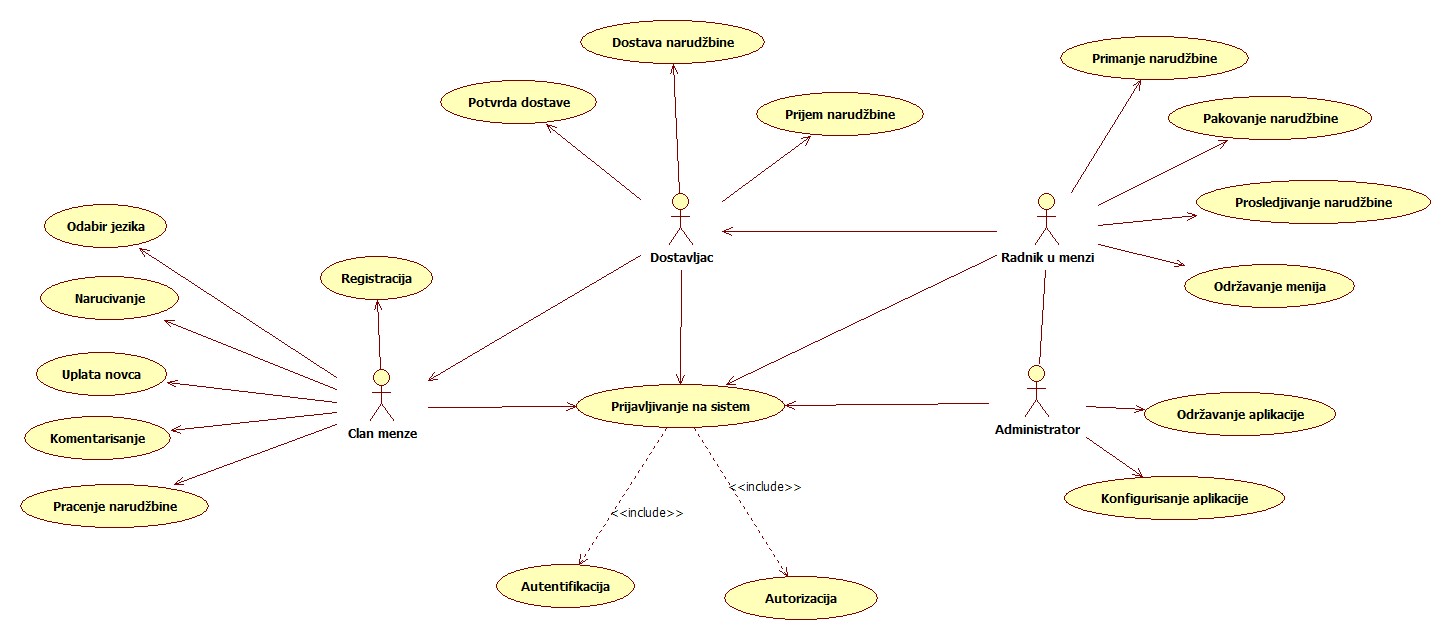
Slučajevi korišćenja aplikacije Menze na pragu su:

* Izbor jezika
* *Pregled informacija*
  + Pregled aktuelnog menija
  + *Pregled podataka o članovima*
    - Pregled podataka o određenom članu menze
    - Pregled podataka o istoriji naručivanja
* Prijavljivanje
* Ažuriranje podataka o članu
* Ažuriranje menija
* *Kreiranje i brisanje članova*
  + Kreiranje novog člana
  + Brisanje postojećeg člana
* Dodavanje radnika
* Komentarisanje
* Praćenje narudžbine
* Uplata novca

Ove slučajevi korišćenja mogu da iniciraju članovi menze, radnici u menzi, dostavljač ili administrator.

## Dijagrami slučajeva korišćenja

Osnovni UML dijagram koji prikazuje korisnike i slučajeve korišćenja aplikacije Menza na pragu prikazan je na sledećoj slici:



## Kratak opis slučajeva korišćenja

### Odabir jezika

Kratak opis: Izbor između srpskog i engleskog jezika za prikaz sadržaka aplikacije.

Akteri koji iniciraju slučaj korišćenja: član menze, radnik u menzi, dostavljač ili administrator

### Naručivanje

Kratak opis: Naručivanje željene hrane.

Akteri koji iniciraju slučaj korišćenja: član menze

### Uplata novca

Kratak opis: Uplata novca na kartici koji se koristi za plaćanje obroka.

Akteri koji iniciraju slučaj korišćenja: član menze

### Komentarisanje

Kratak opis: Ostavljanje utiska o kvalitetu dostave.

Akteri koji iniciraju slučaj korišćenja: član menze

### Praćenje narudžbine

Kratak opis: Prikaz lokacije dostavljača.

Akteri koji iniciraju slučaj korišćenja: član menze

### Potvrda dostave

Kratak opis: Dostavljač preko odgovarajuće stranice pritiskom na dugme potvrđuje uspešnost dostave.

Akteri koji iniciraju slučaj korišćenja: dostavljač

### Prijem narudžbine

Kratak opis: Primanje narudžbine od korisnika aplikacije (radnik u menzi) ili od radnika u menzi (dostavljač).

Akteri koji iniciraju slučaj korišćenja: dostavljač, radnik u menzi.

### Prosleđivanje narudžbine

Kratak opis: Radnik u menzi prosleđuje narudžbinu dostavljaču.

Akteri koji iniciraju slučaj korišćenja: radnik u menzi

### Održavanje menija

Kratak opis: Ažuriranje aktuelnog menija na dnevnom nivou.

Akteri koji iniciraju slučaj korišćenja: radnik u menzi

### Prijavljivanje

Kratak opis: Prijavljivanje korisnika u aplikaciju u cilju pristupa specifičnim funkcijama koje zahtevaju autorizaciju.

Akteri koji iniciraju slučaj korišćenja: član menze, radnik u menzi, dostavljač ili administrator.

### Registracija

Kratak opis: Članovi menze kreiraju nalog kako bi mogli da krenu sa korišćenjem aplikacije.

Akteri koji iniciraju slučaj korišćenja: član menze.

### Održavanje aplikacije

Kratak opis: Funkcionisanje aplikacije u svakom trenutku i brzo reagovanje na prijavljene greške i komentare.

Akteri koji iniciraju slučaj korišćenja: administrator.

### Konfigurisanje aplikacije

Kratak opis: Registrovanje zaposlenih, izmena podataka u bazi (izmena menija), dodavanje novih funkcionalnosti ukoliko je to potrebno.

Akteri koji iniciraju slučaj korišćenja: administrator.

# Pogled na logičku arhitekturu sistema

U ovom odeljku je dat pregled logičke arhitekture sistema. Ovaj pogled sadrži opis najznačajnijih klasa, njihove organizacije u pakete i podsisteme, i organizacija podsistema u slojeve. U cilju opisivanja dinamičkih aspekata arhitekture, ovaj odeljak može da uključi opise realizacije najznačajnijih slučajeva korišćenja. Da bi se ilustrovala veza između arhitekturno značajnih klasa, podsistema, paketa ili slojeva moguće je uključiti i odgovarajuće dijagrame klasa.

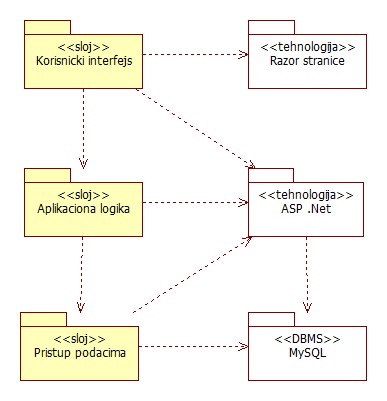
Logički pogled na aplikaciju Menza na pragu obuhvata 3 glavna paketa: Korisnički interfejs, Aplikaciona logika, Pristup podacima.

Paket *Korisnički interfejs* sadrži Razor stranice, C# skripte i multimedijalni sadržaj koji realizuju grafički dizajn i forme preko kojih korisnici sistema komuniciraju sa sistemom.

Paket *Aplikaciona logika* predstavlja srednji sloj sistema koji sadrži C# skripte zadužene za realizaciju funkcionalnosti specifičnih za domen sistema koji se razvija.

Paket *Pristup podacima* sadrži C# skripte koje predstavljaju interfejs za pristup, dodavanje i ažuriranje podataka koji se čuvaju u bazi podataka.

## Pregled arhitekture – organizacija paketa i podsistema u slojeve



### Korisnički interfejs

sloj

Ovaj sloj realizuje korisnički interfejs aplikacije. U njemu su sadržane svi multimedijalni sadržaji i C# skripte koje generišu Razor stranice preko kojih korisnici komuniciraju sa sistemom.

Sloj korisničkog interfejsa zavisi od sloja aplikacione logike, kao i paketa HTML i ASP .Net.

### Aplikaciona logika

sloj

Sloj aplikacione logike je srednji sloj u troslojnoj arhitekturi aplikacije Menza na pragu. Sadrži C# skripte koje realizuju funkcionalnost karakterističnu za domen primene aplikacije i uspostavljaju vezu između korisničkog interfejsa i sloja za pristup podacima.

Ovaj sloj zavisi od sloja za pristup podacima i ASP .Net paketa.

### Pristup podacima

sloj

Sloj za pristup podacima se nalazi na dnu troslojne arhitekture i sadrži C# skripte zadužene za pribavljanje, dodavanje i ažuriranje podataka koji se čuvaju u MySQL bazi podataka.

Ovaj sloj ne zavisi od drugih slojeva, ali je zavisan od ASP .Net paketa i MySQL baze podataka.

### Razor stranice

tehnologija

Tehnologija Razor stranica definiše gradivne elemente stranica koje se prikazuju u Web čitaču i koje omogućavaju prikaz formatiranih informacija i realizaciju formi za unos i ažuriranje podataka.

### ASP .Net

tehnologija

Tehnologija ASP .Net obezbeđuje mehanizam upravljanja prosleđenih podataka sa klijenta kao i prosleđivanje podataka ka klijentu. Ima za cilj upravljanej bazom podataka.

### MySQL

DBMS

MySQL predstavlja sistem za upravljanje bazama podataka koji će se koristiti za realizaciju aplikacije Menza na pragu.

# Pogled na procese

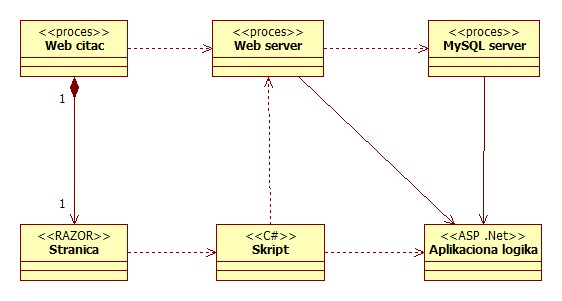
U ovom odeljku je sadržan pogled na procesnu arhitekturu sistema. Ovaj opis treba da sadrži specifikaciju različitih zadataka (procesa i niti) uključenih u rad sistema. Takođe je potrebno dati dijagrame koji pokazuju njihovu interakciju i konfiguraciju. Dodela objekata i klasa na određene zadatke takođe spada u opis procesne arhitekture.

Web aplikacije zasnovane na Razor stranicama imaju relativno jednostavan procesni model koji je u potpunosti pod kontrolom Web servera.

Ilustracije radi u nastavku je dat opis procesa uključenih u izvršenje aplikacije Menza na pragu kao Web aplikacije.

## Procesi

Na sledećem UML dijagramu klasa prikazani su procesi koji učestvuju u izvršenju aplikacije Menza na pragu. Dijagram je opšteg tipa i može se primeniti na bilo koju Web aplikaciju zasnovanu na Razor stranicama i MySQL bazi podataka.



### Web čitač

Web čitač je proces koji izvršava funkcionalnost aplikacije za prikaz Razor stranica dobijenih od nekog Web servera. U najopštijem slučaju Web čitač u jednom trenutku može da prikazuje samo jednu Razor stranicu.

Web čitač zavisi od Web servera koji generiše i vraća odgovarajuće podatke koji se posredstvom C# i Razor tehnologije pretvaraju u HTML sadržaj.

### Web server

Web server je proces koji izvršava funkcionalnost opsluživanja zahteva prispelih sa više Web čitača. Ukoliko je zahtevana Razor stranica, Web server inicira izvršenje C# procesa koji obrađuje odgovarajući skript i generiše sadržaj koji se vraća čitaču. Web server može paralelno da inicira veći broj C# procesa.

### MySQL Server

MySQL Server je proces koji izvršava funkcionalnost MySQL sistema za upravljanje bazama podataka. Ovaj proces može konkurentno da prihvati određen broj upita, izvrši ih nad bazom podataka i vrati rezultate procesu koji je upite postavio.

### Razor stranice

Razor stranice su ASP .Net sintaksa za realizaciju dinamičkih web stranica korišćenjem C#-a.

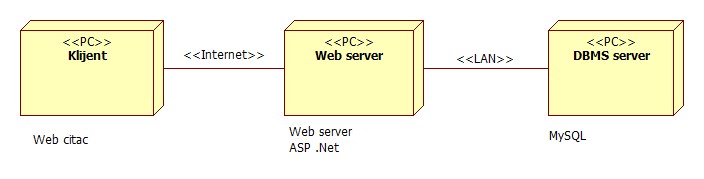
### ASP .Net

ASP .Net proces obavlja posao aplikacionge logike, proces može da zahteva usluge MySQL servera-a. Komunikacija između ASP .Net procesa i MySQL-a servera se obavlja preko prosleđivanja upita i vraćanja rezultata.

# Pogled na raspoređivanje sistema

Pogled na raspoređivanje sistema prikazuje različite fizičke čvorove za najopštiju konfiguraciju sistema. Fizičkim čvorovima koji predstavljaju procesore vrši se dodeljivanje identifikovanih procesa.

Na sledećoj slici dat je UML dijagram raspoređivanja aplikacije Menza na pragu.



## Klijent

Pristup aplikaciji Menza na pragu se obavlja preko klijentskih računara na kojima se izvršava Web čitač. Za povezivanje između klijenta i Web servera koristi se Internet infrastruktura tako da nema ograničenja u pogledu lokacije klijenta.

## Web server

Računar na kome se izvršava Web server opslužuje više klijenata koji pristupaju preko Interneta. Pored osnovnog procesa koji realizuje funkcionalnost Web servera, na ovom računaru mogu da se izvršavaju i ASP .Net procesi koji vrše obradu zadatih C# skripti. U najopštioj konfiguraciji DBMS se izvršava na posebnoj mašini koja je sa Web serverom u lokalnoj mreži (LAN).

## DBMS server

DBMS server je računar na kome se izvršava MySQL proces koji realizuje funkcionalnost sistema za upravljanje bazama podataka. Zbog sigurnosti podataka koji se na ovom računaru čuvaju pristup bazi je ograničen samo na računare iz lokalne mreže (LAN).

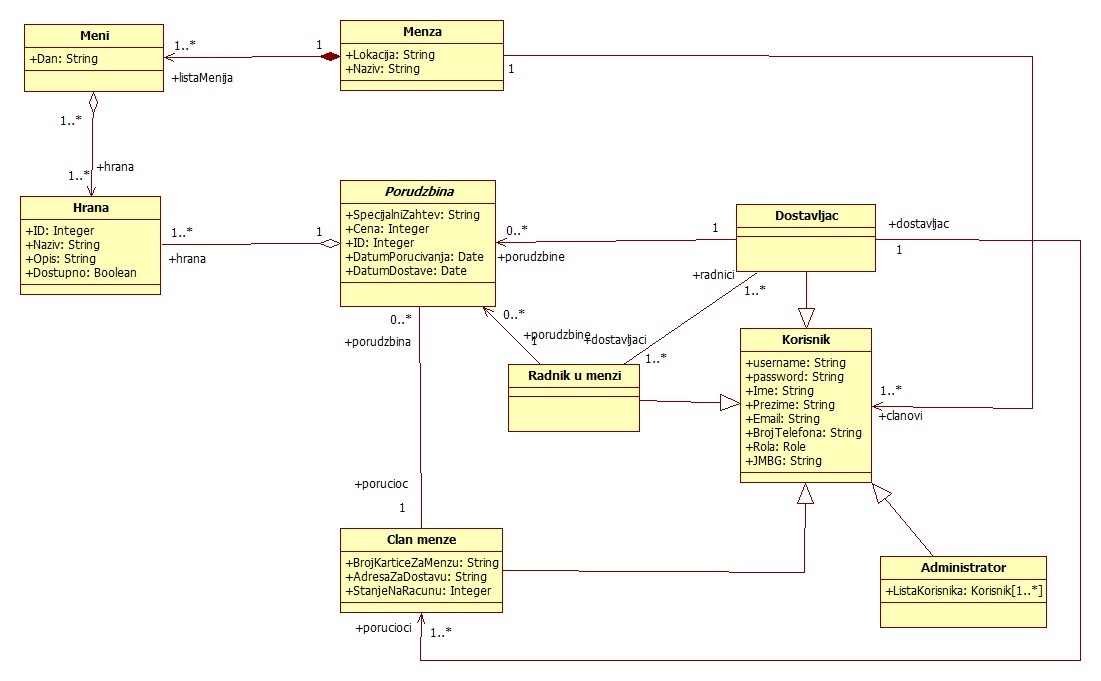
# Pogled na implementaciju sistema

Pogled na implementaciju prikazuje različite aspekte bitne za implementaciju sistema. U slučaju aplikacije Menza na pragu ovaj odeljak sadrži model domena, šemu baze podataka i prikaz komponenti sistema razvrstanih u ranije identifikovane pakete.

## Model domena

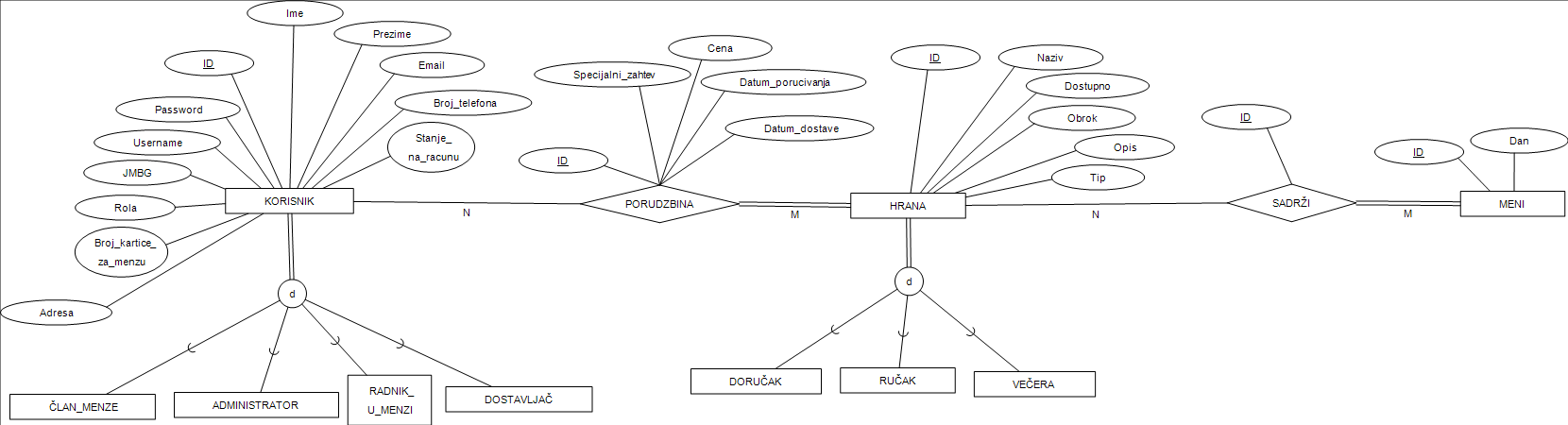
Model domena za koji se aplikacija Menza na pragu projektuje je ilustrovan UML dijagramom klasa. U njemu su prikazane domenske klase, neki od njihovih atributa, kao i veze koje se mogu identifikovati između njih.

Model domena predstavlja osnovu za projektovanje baze podataka, ali i identifikaciju nekih od komponenti (C# skripti) koje će biti implementirane.



## Šema baze podataka

Detaljna šema baze podataka je prikazana na sledećem dijagramu.



## Komponente sistema

*NAPOMENA: Menza na pragu projekat nije kompletno implementiran, tj. implementiran je na nivou arhitekturnog prototipa. Zbog ovoga sledeću sekciju trebate shvatiti kao ilustraciju kako bi dokument trebao da izgleda. Opis komponenti sistema za vaše projekte mora biti kompletan! Procena složenosti vašeg rešenja će zavisiti od opisa sistema koji ovde date.*

Komponente aplikacije Menza na pragu su C# skripte čiji će pregled biti dat po arhitekturnim slojevima. Za ilustraciju će biti korišćeni UML dijagrami komponenti, ali i dijagrami klasa. U slučajevima gde je C# skript prikazan kao klasa atributi predstavljaju ulazne podatke koji se uzimaju iz GET ili POST dela HTTP poruke, dok metodi predstavljaju funkcije definisane u okviru skripta.

### Komponente korisničkog interfejsa

Dizajn korisničkog interfejsa je obuhvaćen dvema komponentama:



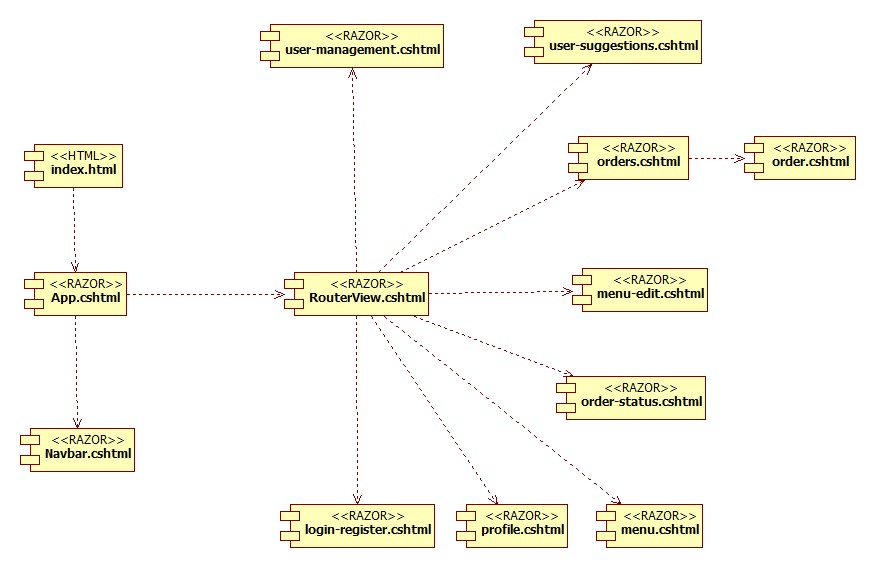
Komponenta **index.cshtml** koristi Razor stranice čiji sadržaj može da varira od parametra koji joj se proslede u zahtevu.

Komponenta **index.html** predstavlja osnovni HTML sadržaj koji je generisana preko C# skripti i Razor stranica.

Komponenta **main.css** predstavlja opis stilova za pojedine HTML elemente koji se javljaju na različitim stranicama.

### Komponente aplikacione logike

Komponente koje realizuju domen problema se uključuju isključivo preko **index.cshtml** komponente korisničkog interfejsa. Na taj način zadržavaju sva podešavanja stila definisana u ovom skriptu. Na sledećem dijagramu su prikazane komponente ovog sloja i njihove međusobne zavisnosti:



### Pristup podacima

Za pristup bazi podataka koristi se **MySQL Workbench**.

# Performanse

Izabrana arhitektura softvera podržava zahteve u pogledu broja korisnika koji mogu simultano pristupati sistemu i vremena odziva za pristup bazi podataka specificirane u zahtevima u pogledu performansi:

**Broj korisnika koji simultano pristupaju**

Sistem će da podrži do 1000 simultanih pristupa korisnika aplikacije.

**Vreme odziva za pristup bazi podataka**

Vreme potrebno za pristupanje bazi podataka u cilju izvršenje nekog upita ne sme da bude veće od 5 sekundi.

Zahtevane performanse su zadovoljene izborom tehnologija na kojima će sistem biti razvijen i definisane hardverske platforme.

# Kvalitet

Izabrana arhitektura softvera podržava zahteve u pogledu dostupnosti i srednjeg vremena između otkaza specificirane u zahtevima u pogledu pouzdanosti:

1. Aplikacija menza na pragu će biti dostupan 24 časa dnevno, 7 dana u nedelji. Vreme kada aplikacija nije dostupan ne sme da pređe 15%.
2. Srednje vreme između dva sukcesivna otkaza ne sme da padne ispod 72 sata.