

Arhitekturni projekat

PlanIT Teams

Članovi tima:

Ana Stojanović 15905
Damjan Trifunović 15948
Milica Todorović 15937

Naziv tima:

Anonymous Mink

Sadržaj

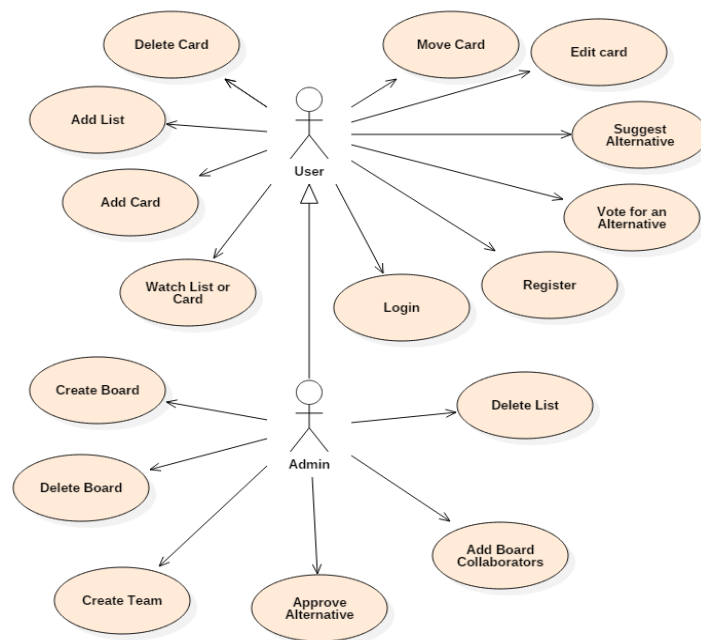
Domen problema.....	3
Funkcionalni zahtevi	3
Arhitekturni zahtevi	4
Nefunkcionalni zahtevi	4
Arhitekturni dizajn	4
<i>Arhitekturna skica</i>	<i>4</i>
<i>Strukturni pogled</i>	<i>5</i>
<i>Alokacioni pogled</i>	<i>5</i>
<i>Bihevioralni pogled</i>	<i>6</i>
Arhitekturni obrasci	6
<i>Layered</i>	<i>6</i>
<i>Publish-subscribe</i>	<i>7</i>
<i>MVVM (Model – View – View Model)</i>	<i>7</i>
<i>Repository</i>	<i>7</i>
Specifikacija biblioteka i programskih okvira	8

Domen problema

PlanIT Teams je desktop aplikacija namenjena timovima koji sarađuju na nekom projektu. Član tima kreira „kanban“ tablu i daje pristup ostalim članovima, tako da svi mogu da kreiraju kartice za određene zadatke, da ih premeštaju u različite liste na tabli i da predlažu drugačije prioritete zadataka (alternative datih listi).

Funkcionalni zahtevi

- Kreiranje korisničkog naloga i login
- Kreiranje timova (grupisanje korisnika u timove)
- Kreiranje kanban board-a
- Dodavanje timova ili drugih korisnika kao saradnike na *svoj* board
- Omogućavanje kolaboracije između članova board-a
- Dodavanje i izmena lista i kartica u okviru board-a
- Mogućnost praćenja (watch) odgovarajuće liste ili kartice (korisnik dobija in- app notifikaciju kada dodje do neke izmene na datoj listi ili kartici)
- Predlaganje alternative za postojeću listu i mogućnost glasanja od strane kolaboratora board-a za jednu alternativu. Klijenti koji prate datu listu bivaju obavešteni o aktivnom glasanju
- Sistem notifikacije zainteresovanih saradnika o promenama na posmatranim listama
- Skladištenje podataka



Use case dijagram sistema

Arhitekturni zahtevi

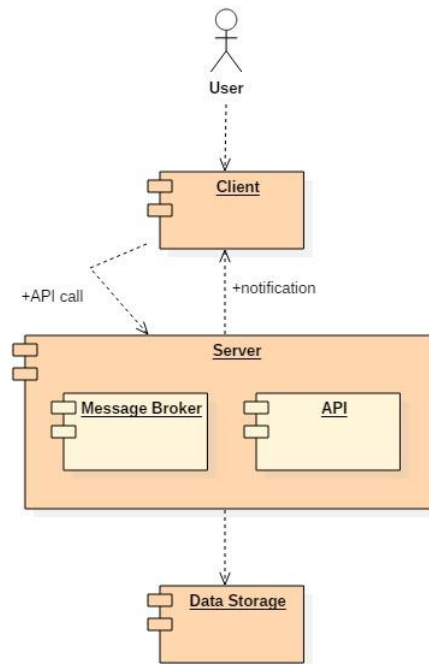
- Ostvarivanje sinhronne i asinhronne komunikacije između klijenta i servera
- Sistem će biti dostupan 7 dana nedeljno, 24 sata dnevno
- Potrebno je da šema baze podataka bude skrivena
- Potrebno je omogućiti trenutnu notifikaciju o događajima

Nefunkcionalni zahtevi

- Pouzdanost
- Performanse – potrebno je smanjiti vreme odziva i prilagoditi propusnu moć sistema
- Dostupnost – potrebno je da je aplikacija bude dostupna 24/7
- Modifikabilnost – potrebno je omogućiti relativno laku promenu sistema
- Skalabilnost – potrebno je da aplikacija može da podrži povećanje broja korisnika
- Lakoća korišćenja – potrebno je da aplikacija bude intuitivna i jednostavna za korišćenje

Arhitekturni dizajn

Arhitekturna skica

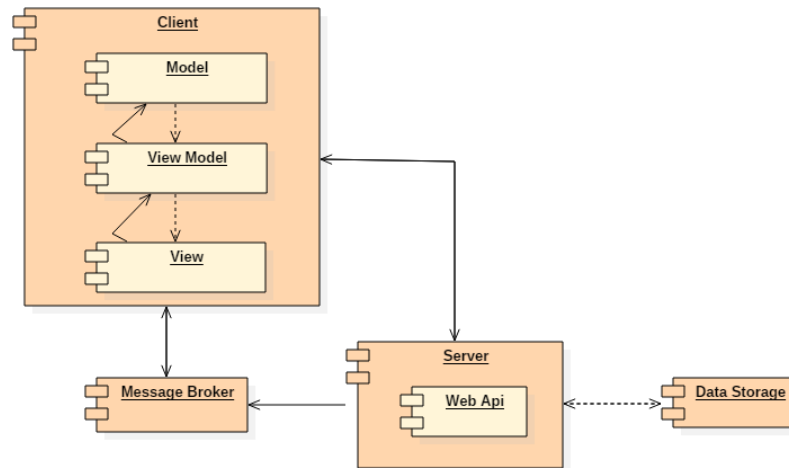


Arhitekturna skica

Strukturni pogled

Navedeni dijagram ilustruje strukturu sistema navodeći komponente sistema kao i njihovu međusobnu povezanost.

Struktura klijentske aplikacije zasnovana je na *MVVM* projektnom obrascu. Klijentska i serverska aplikacija ostvaruju sinhronu komunikaciju korišćenjem RESTful API-ja. Asinhrona komunikacija između klijentske i serverske aplikacije ostvarena je korišćenjem *message brokera*-a. Serverska aplikacija ostvaruje komunikaciju sa odgovarajućim *Data Storage*-om.

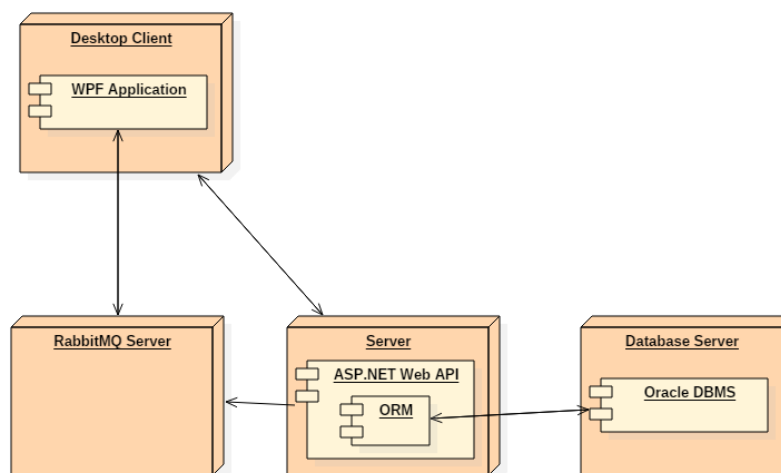


Dijagram strukture sistema

Alokacioni pogled

Navedeni dijagram ilustruje alokacioni pogled na sistem odnosno raspored prethodno opisanih komponenti na čvorovima (računarima).

Klijentska aplikacija je implementirana kao *WPF aplikacija* i predviđena je za računare sa Windows operativnim sistemom. Serverska aplikacija realizuje se kao *ASP.NET Web API*, dok se kao database server koristi *Oracle DBMS*. Za asinhronu komunikaciju između klijenta i servera, koristi se RabbitMQ message broker.

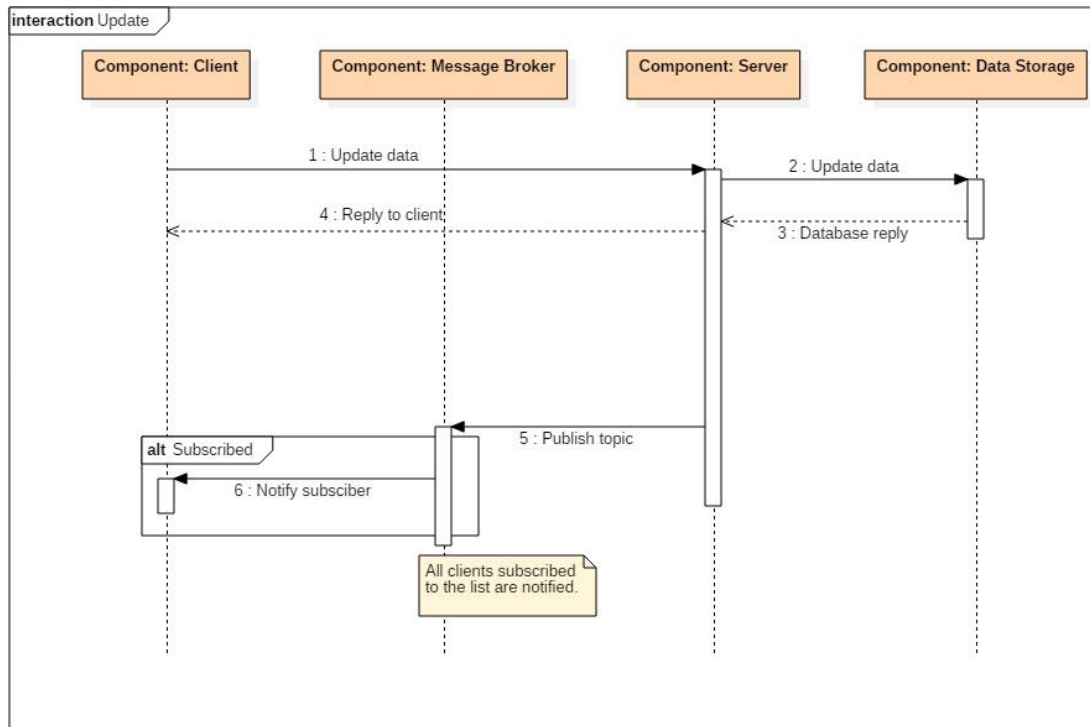


Dijagram raspoređivanja sistema

Bihevioralni pogled

Navedeni dijagram ilustruje bihevioralni pogled sistema na primeru operacije ažuriranja.

Klijent šalje zahtev za ažuriranje serveru, koji obrađuje zahtev i komunicira sa bazom po potrebi. Zatim, server ostvaruje komunikaciju sa message broker-om putem “publish” operacije. Svi klijenti koji posmatraju, subscribe-ovani su na, datu listu ili karticu (nad kojom je učinjena izmena) bivaju obavešteni da je došlo do izmena.



Dijagram sekvence sistema

Arhitekturni obrasci

Layered

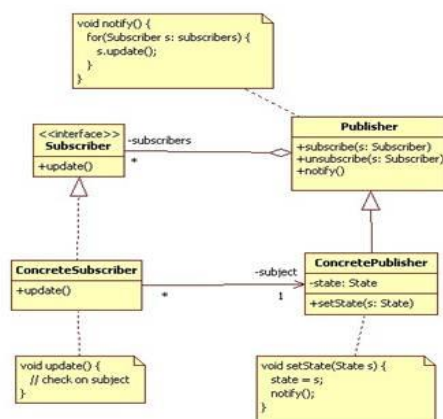
PlanIT Teams sistem će implementirati Layered (Slojeviti) arhitekturni obrazac.

Sistem će se sastojati iz tri sloja. Prvi sloj predstavlja klijentska aplikacija, drugi sloj se sastoji od message broker-a i serverske aplikacije, dok je treći sloj je baza podataka (data storage). Serverska aplikacija ostvaruje sinhronu komunikaciju sa klijentskom aplikacijom korišćenjem RESTful API-ja, dok message broker služi za asinhronu komunikaciju sa klijentom. Serverska aplikacija ostvaruje komunikaciju sa bazom koristeći ORM alat za mapiranje domenskih entiteta.

Publish-subscribe

PlanIT Teams sistem će implementirati Publish-subscribe arhitekturni obrazac (ovaj obrazac je sadržan u okviru *Message Broker* komponente).

Svaki board ima određen broj kolaboratora. Kada dodje do bilo koje promene u board-u, svim kolaboratorima je vidljivo da je došlo do promene. Kolaboratori mogu da se subscribe-uju na određene liste ili kartice u okviru board-a. U tom slučaju, subscribe-ovanim učesnicima stižu in-app notifikacije kojim se posebno naglašava da je došlo do promene na listama koje prate.

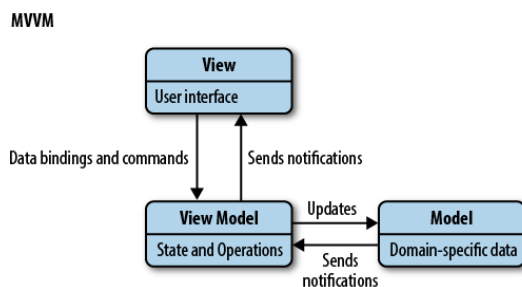


Dijagram publish-subscribe

MVVM (Model – View – View Model)

PlanIT Teams sistem će implementirati MVVM obrazac na strani klijentske aplikacije.

MVVM obrazac je modifikacija (varijanta) standardnog MVC obrazca. PlanIT Teams developeri su istraživanjem došli do zaključka da je MVVM najpogodniji obrazac za razvoj WPF aplikacije (klijentske aplikacije datog sistema).



Dijagram MVVM

Repository

PlanIT Teams sistem će sadržati centralizovanu bazu podataka.

Specifikacija biblioteka I programskih okvira

Windows Presentation Foundation (WPF) – Podsystem .NET frejmworka za izradu Windows klijentskih aplikacija

RabbitMQ .NET Client Library – Biblioteka koja omogućava komunikaciju između RabbitMQ server i .NET klijentske aplikacije

RabbitMQ Server – Message broker, implementira redove poruka i njemu se obraćaju korisnici redova

ASP.NET Web API aplikacija – Serverska aplikacija

Entity Framework – Objektno-relacioni mapper (ORM framework)

Oracle DBMS – Baza podataka