1. Odabrati jedan agilni metod koji će biti korišćen pri razvoju projektnog zadatka. Opisati izabrani model, objasniti zašto ste se odlučili za njega I primeniti model na Vašem projektu.

Za razvoj ovoj projektnog zadatka sam odabrao Scrum metod. Scrum je agilan okvir za upravljanje projektima, posebno popularan u razvoju softvera.Rad se organizuje u iteracijama poznatim kao "sprintovi" koji obično traju 2-4 nedelje.Ključne uloge u Scrum timu uključuju Scrum Mastera, Product Ownera i razvojni tim. Odabrao sam ovaj metod zbog njegove fleksibilnosti I iterativnog pristupa.

2. Koristeći Murov obrazac kreirati viziju proizvoda.

ZA radnike u agenciji za prodaju nekretnina KOJIMA je potrebna osnovna funkcionalnost unosa i pregleda podataka KOJA obezbedjuje detaljan unos svih podataka u sistem kao i pregled i sortiranje istih .ZA RAZLIKU OD drugih sistema koji su zastareli ,NAS PROIZVOD radnicima u agenciji pruza inovativne nacine vrsenja ovih usluga kao i jednostavno koriscenje Sistema.

STA: aplikacija koja omogucava laksi unos i pregled podataka o nekretninama

KO: ovaj proizvod je namenjen radnicima u agenciji za prodaju nekretnina

ZASTO: sistem donosi inovativan nacin izvrsavanja usluga kao sto su upis i pregled podataka

3. Napisati spisak funkcionalnih zahteva.

* Korisnik sistema ima mogucnost unosa svih potrebnih podataka o nekretninama
* Korisnik sistema ima mogucnost pregleda svih nekretnina u vidu kataloga
* Korisnik sistema ima mogucnost zakazivanja vremena za pregled nekretnina
* Korisnik sistema ima mogucnost filtriranja nekretnina po odredjenim kriterijumima(po tipu nekretnine,ceni,lokaciji…)
* Korisnik sistema ima moguconst kreiranja naloga i prijave na isti
* Korisnik ima mogucnost sortiranja podataka na nacin na koji zeli

4. Napisati nefunkcionalne zahteve.

Performanse: Sistem treba da ima vreme odziva manje od 2 sekunde za sve korisnicke akcije, kao I da podrzi istovremeni pristup od najmanje 100 korisnika vez gubitka performansi.

Dostupnost: Sistem treba da bude dostupan 99% vremena ukljucujuci odrzavanje.

Sigurnost: Sistem treba da ima primenjene sigurnosne mehanizme za autentifikaciju I autorizaciju korisnika, kao I enkripciju osetljivih podataka.

Skalabilnost: Sistem treba da bude skalabilan kako bi podrzao rastuci broj nekretnina I korisnika. Takodje je potrebne omoguciti dodavanje hardverskih resursa za poboljsanje performansi Sistema.

Interoperabilnost: Potrebno je omoguciti integrisanje sa drugim sistemima koji su vazni kao sto su platni sistemi.

Pouzdanost: Sistem treba da ima manje od 1% gresaka u svim funkcionalnostima, kao I da ima brz oporavak od gresaka bez gubljenja podataka.

Korisnicko iskustvo: Sistem mora imati responzivan dizajn, kao I biti prilagodjen korisnicima razlicitih nivoa tehnicke strucnosti.

5. Napisati od pet do sedam korisničkih priča za svaku grupu korisnika sistema.

Korisnicke price za radnike u agenciji:

* Kao radnik u agenciji, želim da mogu uneti detalje nove nekretnine, uključujući tip, adresu, kvadraturu, broj soba, cenu, opis i slike, kako bih ažurirao katalog nekretnina.
* Kao radnik u agenciji, želim pregledati sve rezervacije za pregled nekretnina, kako bih efikasno planirao i potvrdio rezervacije.
* Kao radnik u agenciji, želim imati mogućnost ažuriranja svog profila, uključujući kontakt informacije, kako bih osigurao tačnost podataka.
* Kao radnik u agenciji, želim imati pregled aktivnosti u sistemu, uključujući unose i izmene nekretnina, kako bih bio informisan o poslovnim događajima.
* Kao radnik u agenciji, želim imati mogućnost slanja poruka klijentima putem sistema, kako bih brzo i efikasno komunicirao o pitanjima i rezervacijama.

Korisnicke price za klijente:

* Kao potencijalni kupac, želim pregledati sve dostupne nekretnine na platformi, koristeći filtere kako bih pronašao nekretninu koja odgovara mojim potrebama.
* Kao kupac, želim rezervisati vreme za pregled nekretnine koja me interesuje, kako bih je bolje istražio pre nego što donesem odluku o kupovini.
* Kao kupac, želim imati mogućnost praćenja određenih nekretnina, kako bih primao obaveštenja o promenama u statusu ili ceni.
* Kao kupac, želim imati pristup pregledu finansijskih informacija o nekretninama, uključujući cene, poreze i druge relevantne podatke.
* Kao kupac, želim imati mogućnost slanja direktnih upita ili poruka radnicima u agenciji kako bih dobio dodatne informacije o nekretninama koje me interesuju.

6. Kreirati od tri do pet Persona.

Aleksandra – iskusna agentkinja za nekretnine:

Aleksandra, 40 godina, je stručnjakinja za nekretnine sa desetogodišnjim iskustvom u industriji. Kroz sistem prodaje nekretnina, Aleksandra efikasno upravlja svojim portfoliom, unosom novih nekretnina i praćenjem njihovog statusa. Sistem joj omogućava da personalizuje ponude prema klijentskim preferencijama, koristi analitiku za optimizaciju marketinških strategija, i pruža visok nivo usluge klijentima prilikom pregovora i zaključivanja transakcija.

Marko – mladi investitor u nekretnine:

Marko,28 godina, kao mladi investitor u nekretnine, koristi sistem kako bi pronašao optimalne investicione prilike. Kroz analizu trendova na tržištu, praćenje finansijskih podataka i upravljanje portfoliom, Marko donosi informisane odluke o kupovini. Sistem mu takođe pomaže u koordinaciji sa agentima, arhitektama i izvođačima radova tokom procesa investiranja, omogućavajući mu da vodi uspešne projekte.

Jovana – prvi put kupac:

Jovana, 26 godina, koja se prvi put upušta u svet nekretnina, koristi sistem kako bi olakšala proces kupovine. Kroz sistem, Jovana istražuje dostupne nekretnine, postavlja pitanja agentima i koristi funkcionalnosti sistema za rezervaciju pregleda. Sistem joj omogućava praćenje budžeta, primanje obaveštenja o novim nekretninama u skladu sa njenim preferencijama, i pruža joj pravovremene informacije o procedurama i dokumentaciji tokom celog procesa kupovine.

7. Za svaku personu kreirati od jedan do tri scenarija.

Scenario za Aleksandru:

Aleksandra, iskusna agentkinja za nekretnine, započinje svoj radni dan pristupanjem sistemu za prodaju nekretnina. Pregledava aktuelne ponude na tržištu, a zatim koristi sistem za unos novih nekretnina koje će dodati u svoj portfelj. Tokom dana, dobija upit od potencijalnog kupca za jednom od nekretnina koje su na njenom spisku. Koristi funkcionalnosti sistema kako bi brzo odgovorila na upit, pružila dodatne informacije i zakazala pregled. Na kraju dana, analizira analitičke izveštaje sistema kako bi identifikovala trendove i prilagodila marketinške strategije.

Scenario za Marka:

Marko, mladi investitor u nekretnine, ulazi u sistem kako bi analizirao trenutne investicione prilike. Koristi analitičke alate sistema kako bi pratio trendove na tržištu i identifikovao nekretnine koje predstavljaju dobru investiciju. Tokom dana, sarađuje sa agentima za nekretnine putem sistema, postavlja pitanja o specifičnostima određenih objekata i izrađuje plan investicija. Sistem mu pomaže u efikasnom upravljanju portfoliom, omogućavajući mu da prati performanse svake investicije i donosi informisane odluke.

Scenario za Jovanu:

Jovana, koja se prvi put upušta u kupovinu nekretnina, koristi sistem kako bi istražila dostupne objekte. Pregledava ponude, koristi funkcionalnosti sistema za filtriranje prema svojim preferencijama i pravi listu nekretnina koje bi želela videti. Kroz sistem, postavlja pitanja agentima o pravnim aspektima i dodatnim troškovima, koristi alate za praćenje statusa ponuda i primanje obaveštenja o promenama u cenama ili dostupnosti nekretnina. Na kraju dana, koristi sistem kako bi sačuvala listu favorita i planirala naredne korake u procesu kupovine.

8. Izabrati od jedan do tri stila softverske arhitekture, opisati i dijagramom predstaviti izabrane modele rešenja softverske arhitekture za Vaš sistem. Opisati prednosti i mane tih softverskih arhitektura.

Monolitna:

Monolitna arhitektura je klasičan pristup gde se cela aplikacija razvija, testira i implementira kao jedna jedinstvena celina. U kontekstu sistema za prodaju nekretnina, ova arhitektura bi uključivala centralizovani sistem koji obuhvata sve aspekte aplikacije, uključujući unos i pregled nekretnina, upravljanje rezervacijama, komunikaciju sa klijentima, analitiku, i druge funkcionalnosti.

Prednosti:

Jednostavnost Razvoja: Razvoj i održavanje su pojednostavljeni jer je cela aplikacija jedna jedinstvena celina.

Lakše Održavanje: Promene se vrše centralizovano, olakšavajući održavanje sistema.

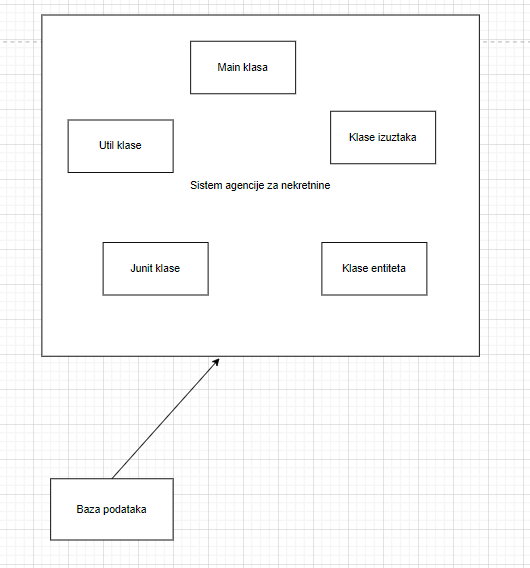
Brza Implementacija: Brza implementacija celog sistema.

Mane:

Ograničena Skalabilnost: Teže se skalira, posebno sa povećanjem veličine i kompleksnosti sistema.

Otežano Ažuriranje: Ako postoji potreba za promenom jednog dela, cela aplikacija se mora ažurirati.

Teško Razdvajanje Odgovornosti: Povećanje kompleksnosti može dovesti do teškog razdvajanja odgovornosti između različitih delova sistema.



Mikroservisna:

Mikroservisna arhitektura podrazumeva organizaciju sistema kao skupa nezavisnih i autonomnih servisa, svaki odgovoran za određenu poslovnu funkcionalnost. U kontekstu sistema za prodaju nekretnina, ova arhitektura bi uključivala različite mikroservise za unos nekretnina, upravljanje rezervacijama, analitiku, komunikaciju s klijentima, i druge funkcionalnosti. Svaki od ovih servisa bio bi zaseban entitet sa svojom bazom podataka.

Prednosti:

Skalabilnost: Svaki mikroservis može se skalirati nezavisno, omogućavajući prilagođavanje resursa prema potrebama.

Nezavisnost: Svaki servis može biti razvijan, implementiran i održavan odvojeno, pružajući veću fleksibilnost i brže isporuke.

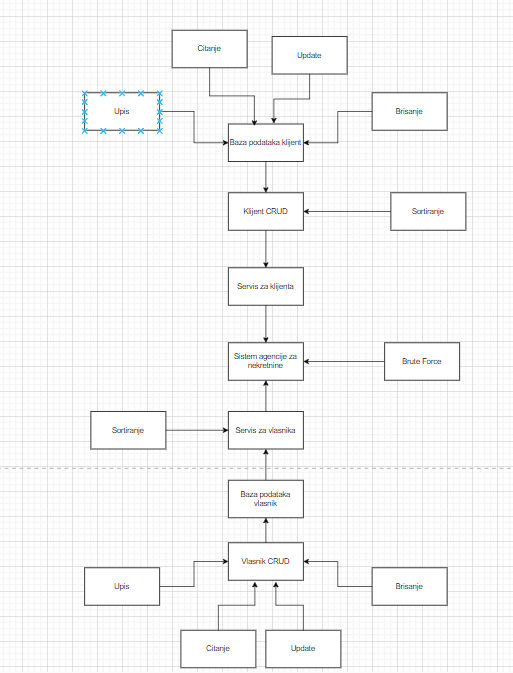
Fleksibilnost Razvoja: Različiti timovi mogu istovremeno raditi na različitim servisima, što ubrzava razvojni ciklus.

Mane:

Kompleksnost Mreže: Zahteva pažljivo upravljanje komunikacijom između mikroservisa, što može postati kompleksno s povećanjem broja servisa.

Operativni Izazovi: Održavanje i upravljanje velikim brojem servisa zahteva napredne alate i prakse.

Povećana Kompleksnost Razvoja: Razvoj, implementacija i testiranje zahtevaju dodatne alate i prakse kako bi se održala koherentnost sistema.



11. Izdvojiti komponente sistema koje mogu biti ponovo upotrebljene.

Modul za autentifikaciju korisnika moze biti izdvojen kao zasebna komponenta koja se moze integrisati u ostale delove sistema.

Modul za komunikaciju sa drugim servisima kao sto su servisi za placanje I procenu cene nekretnina.

Servisi za slanje poruka takodje mogu biti iskorisceni u razlicitim delovima Sistema.

12. Izdvojiti delove sistema koji mogu biti primer računarstva u oblaku i navesti koji bi to tip bio (IaaS, SaaS, PaaS). Takođe, izdvojiti one koji mogu biti zamenjeni već postojećim rešenjima, kao i one delove sistema za koje računarstvo u oblaku ne bi bilo najbolje rešenje (opsati zašto).

Servisi za autentikaciju i autorizaciju često postoje kao veća, specijalizovana rešenja koja se mogu koristiti kao usluge iz oblaka. Ovo je Saas tip.

Različite platforme za slanje poruka (npr. servisi za obaveštenja) često postoje u oblaku kao spoljni servisi koji se mogu integrirsati u sistem. Takodje SaaS tip.

SaaS za osnovne servise za plaćanje, dok IaaS može biti potreban ako se koriste specifični moduli za plaćanje koje je potrebno prilagoditi.

Cloud baze podataka omogućavaju efikasno upravljanje podacima bez potrebe za fizičkim održavanjem infrastrukture. IaaS pruža infrastrukturu za hostovanje baze podataka, dok PaaS dodatno pruža platformu koja olakšava upravljanje i održavanje baze podataka.

13. Izdvojiti komponente sistema koji mogu biti implementirani kao mikroservisi sa odvojenim bazama podataka.

Upravljanje Nekretninama:

Mikroservis koji se bavi unosom, izmenom i brisanjem podataka o nekretninama moze imati odvojenu bazu podataka.

Korisnički Profili:

Mikroservis koji čuva informacije o korisničkim profilima, uključujući podatke o preferencijama, istoriju pretraga i rezervacija moze imati odvojenu bazu.

Plaćanje i Finansije:

Mikroservis odgovoran za sve transakcije i finansijske operacije. Svaki servis ima svoju bazu podataka sa podacima o plaćanjima i finansijama.

Servis za Slanje Poruka:

Mikroservis koji omogućava slanje obaveštenja i poruka korisnicima. Svaki mikroservis ima odvojenu bazu podataka za praćenje istorije poruka.

15. Navedite i opišite atribute kvaliteta Vašeg sistema.

Performanse: Sistem efikasno odgovara na korisničke zahteve, pružajući brze odgovore i minimalno vreme odziva.

Dostupnost: Sistem je dostupan i funkcionalan uvek kada korisnici žele da ga koriste.

Sigurnost: Sistem je zasticen od neovlašćenog pristupa, privatni podaci su ocuvani I sprecena je zloupotreba istih.

Skalabilnost: Sistem ima mogucnost da raste i prilagodi se povećanju broja korisnika, podataka ili zahteva.

17. Kreirati od pet do deset scenarija testiranja (tabela je data u prilogu).

1. Testiranje unosa datuma:
   1. Proverava da li sistem unosi datum nastanka ugovora u pravilnom formatu

* Pravi se entitet ugovor
* Proverava se da li je datum nastanka validan
* Verifikacija da system pravilno obavestava o neispravnom formatu
  1. Proverava da li sistem unosi datum isteka ugovora u pravilnom formatu
* Pravi se entitet ugovor
* Proverava se da li je datum nastanka validan
* Verifikacija da sistem pravilno obavestava o neispravnom formatu

1. Testiranje dodavanja klijenata:
   1. Provera da li klijent nakon unosa postoji u bazi podataka
      * Otvara se konekcija sa bazom
      * Pravi se entitet klijent
      * Izvrsava se query sa dodavanjem klijenta u bazu
      * Proverava se da li klijent postoji
      * Verifikacija da sistem pravilno obavestava o nepostojecem ili neispravno unetim podacima
2. Testiranje dodavanja vlasnika nekretnine:
   1. Provera da li vlasnik nekretnine nakon unosa postoji u bazi podataka
      * Otvara se konekcija sa bazom
      * Prave se dva entiteta vlasnik
      * Izvrsava se query sa dodavanjem vlasnika u bazu
      * Proverava se da li vlasnici postoje u bazi
      * Verifikacija da sistem pravilno obavestava o nepostojecem ili neispravno unetim podacima
3. Testiranje unosa iznosa I naknade agenciji u ugovor:
   1. Proverava da li sistem unosi iznos I naknadu u pravilnom format
      * Pravi se novi ugovor
      * Proverava se da li su iznos I ugovor validni
      * Verifikacija da sistem pravilno obavestava o neispravnom formatu
4. Testiranje unosa broja telefona:
   1. Proverava da li sistem unosi broj teledona vlasnika I kljienta u pravilnom format
      * Pravi se novi vlasnik ili klijent
      * Proverava se da li su brojevi teelfona u validnom formatu
      * Verifikacija da sistem pravilno obavestava o neispravnom format

20. Pratiti razvoj projektnog zadatka verzioniranjem na Git-u.

https://github.com/ljuba016/agencija\_za\_nekretnine.git