**ClassyCrafT**

**Uvod**

Urednik UML dijagrama, ClassyCrafT, omogućava korisniku da napravi i menja UML dijagram. ClassyCrafT je desktop aplikacija koja ima mogućnost da pravi različite entite, dozvoljava menjanje, kao i da vrši međusobno povezivanje različitim vrstama veza.

**Rečnik**

UML dijagram - UML dijagram je vizualni alat koji koristi grafičke simbole i oblike kako bi predstavio različite aspekte softverskog sistema ili procesa, pomažući u razumevanju i komunikaciji među ljudima koji rade na razvoju ili analizi softvera.

Klasa – Klasa je osnova Objektno orijentisanog programiranja. Predstavlja opis necega. Na primer klasa

“Telefon”, bi imala atribute ime, marka, serijski broj itd.

Veza – Veza je logicka konekcija između objekata koja je na UML-u predstavljena nekom vrstom linije.

Atribut – Karakteristika koja opisuje neku klasu.

Entitet – Entitet je opis klase unutar UML dijagrama.

**Konkurencija**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Ime | Desktop klijent | Web verzija aplikacije | Mobilna verzija | Besplatno | Uputstva koriscenja | Generisanje JAVA koda |
| ClassyCrafT | ✔ | ❌ | ❌ | ✔ | ✔ | ✔ |
| Lucidchart | ❌ | ✔ | ✔ | ❌ | ❌ | ✔ |
| Draw.io | ❌ | ✔ | ✔ | ✔ | ✔ | ❌ |
| StarUML | ✔ | ❌ | ❌ | ✔ | ✔ | ✔ |
| PlantUML | ❌ | ✔ | ❌ | ✔ | ✔ | ✔ |
| Microsoft Visio | ✔ | ✔ | ❌ | ❌ | ✔ | ❌ |

**Korisnici**

Za korišćenje ove aplikacije potrebno je osnovno znanje o konceptu UML dijagrama, kao I shvatanje čemu on služi.

Program je namenjen za jednog korisnika (jedan računar istovremeno).

U ovoj aplikacijii korisnik sam podešava izgled i strukturu njegovog UML dijagrama po želji.

ClassyCrafT je namenjen za studente prve godine, kao i za korisnike koji žele da olakšaju implenetaciju koda njihovog programa i da ga vizuelno predstave drugim ljudima koji bi ga tako lakše razumeli.

**Zahtevi**

U navedenom tekstu se nalaze funkcionalsnosti aplikacije, poredjane u tri prioriteta:

* Nizak (program može u potpunosti da radi bez zadate funkcionalnosti)
* Srednji (poželjno je da program ispunjava zadatu funkcionalnost)
* Visok (program ne može da funckioniše bez zadate funkcionalnosti)

**1. Osnovne operacije nad entitetima klasnog Dijagrama**

**1.1 Kreiranje i brisanje entiteta**

Korisnik je u mogućnosti da doda I da obriše entitet po želji kako bi kreirao svoj UML Dijagram. Prioritet je visok, rizik je samo pri brisanju.

**1.2 Modifikovanje entiteta**

Već kreiran entitet ima mogućnost izmene . Prioritet je srednji, rizik ne postoji.

**1.3 Rasporedjivanje entiteta po ravnoj površini**

Korisnik ima mogućnost da rasporedi entitete po površini po njegovoj želji bez gubljenja veza. Prioritet je srednji, rizik ne postoji.

**1.4 Vizuelizacija entiteta**

Korisnik pri pravljenju entiteta ima grafički prikazan njegov entitet na radnoj površini, kako bi imao preglednost šta je sve uradio.Prioritet visok, rizik ne postoji.

**2. Osnovne operacije nad vezama klasnog dijagrama**

**2.1 Kreiranje i brisanje veza**

Korisnik je u mogućnosti da međusobno poveže dva ili više entiteta različitim vrstama veza, kao I da obriše postojeću vezu imeđu njih. Prioritet je visok, rizik samo da ne obriše ili pogrešno poveže dva entiteta.

**2.2 Modifikovanje veza**

Korisnik je u mogućnosti da proneni vrstu već postojeće veze izmedju njegovih entiteta. Prioritet je srednji, rizik nizak.

**3. Osnovne operacije nad celokupnim klasnim dijagramom**

**3.1 Kreiranje novog projekta**

Korisnik ima mogućnost da napravi novi projekat te tako kreće od početka sa njim i potom smišlja njegovu implementaciju za UML, takodje može imati više UML dijagrama na njegovom projektu.Prioritet visok, rizik ne postoji.

**3.2 Kreiranje novog UML dijagrama**

Korisnik ima mogućnost da u svakom trenutku napravi čistu, novu, radnu površinu na već postojećem projektu i da krene da pravi novi UML dijagram. Prioritet je visok, rizik nizak.

**3.3 Čuvanje i učitavanje UML dijagrama**

Korisnik ima mogućnost da sačuva trenutni UML dijagram, kao I da učita već postojeći dihjagram (sačuvan). Prioritet srednji, rizik nizak.

**4. Organizacija struktura podataka korišćenih za pravljenje UML dijagrama**

**4.1 Implementacija stabla struktura podataka**

Sve komponente UML dijagrama treba da budu raspoređene u strukturu stabla kako bi se ograničilo dodavanje neželjenih elemenata na sam UML dijagram. Svaka struktura ima svog roditelja u koji može da bude dodata ili dete koje može da bude dodato u nju. Prioritet je visok a rizik je nizak.

**4.2 Prikaz stabla struktura podataka**

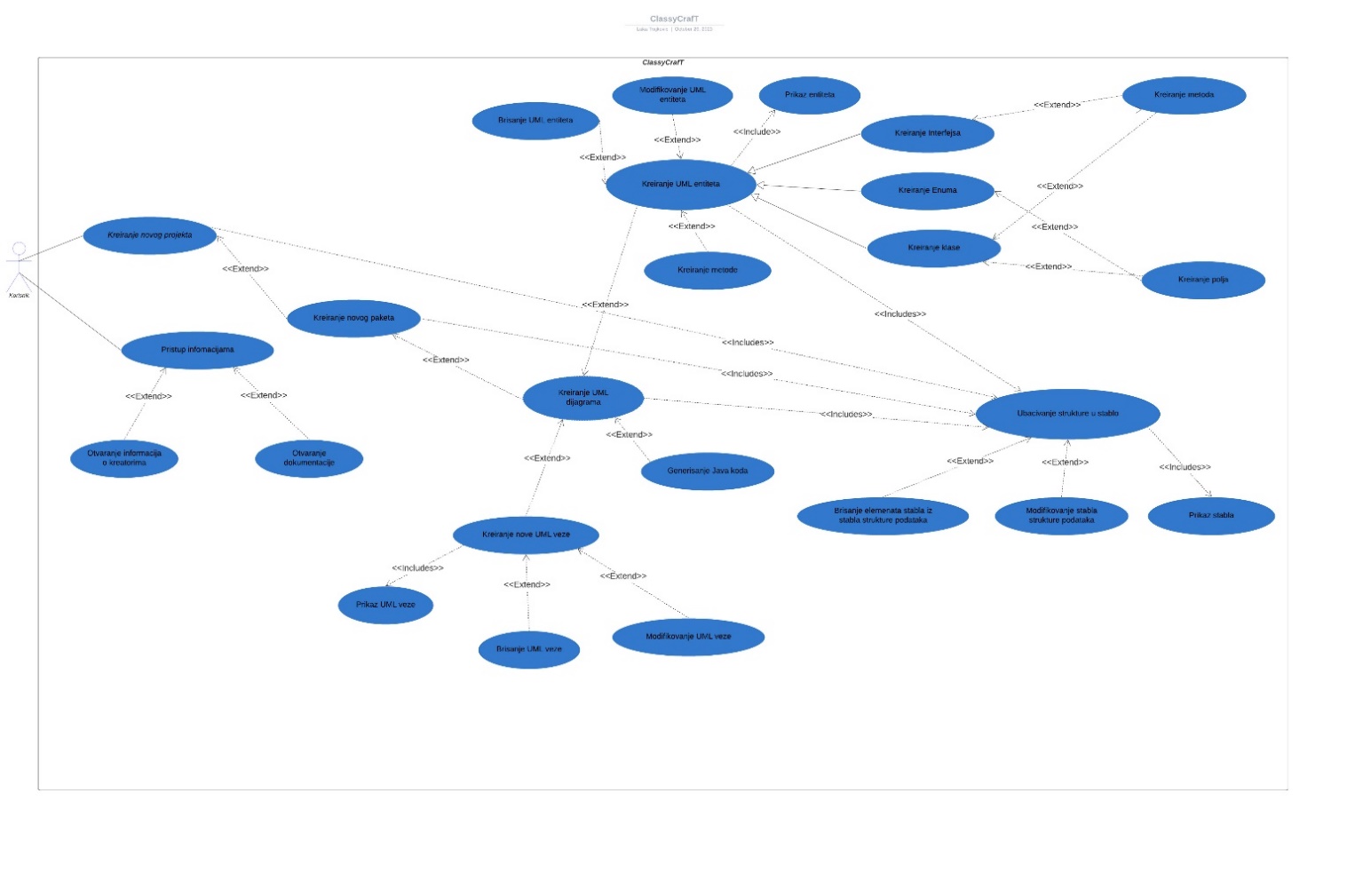
Korisnik prilikom rada na UML dijagramu treba da bude u mogućnosti da vidi sve elemente svog UML dijagrama raspoređene u stablo sa leve strane. Prioritet je srednji, rizik ne postoji.

**4.3 Prikaz biblioteke šablona**

Korisnik prilikom kreiranja UML dijagrama treba da bude u mogućnosti da pristupi svim ponuđnim šablonima iz biblioteke šablona tako što će mu se prikazati na ekranu. Prioritet je nizak, dok rizik ne postoji.

**Funkcionalni dijagram**

Ispod se nalazi slika UseCase dijagrama projekta ClassyCrafT.



**UseCase**

**Kreiranje novog projekta**

Akter: Korisnik.

Preduslov: Korisnik je otvorio ClassyCrafT aplikaciju.

Koraci: Korisnik je kliknuo File./ Korisnik je kliknuo New./ Korisnik je kliknuo Project.

Tačke proširenja: Potrebno je ispisati projekat u stablo.

**Kreiranje novog paketa**

Akter: Korisnik.

Preduslov: Korisnik je kreirao novi projekat.

Koraci: Korisnik je kliknuo na new Package.

Tačke proširenja: Potrebno je dodati paket u stablo.

**Kreiranje novog UML dijagrama**

Akter: Korisnik.

Preduslov: Korisnik je kreirao novi paket.

Koraci: Korisnik je kliknuo I dodao novu radnu površinu.

Tačke proširenja: Novi dijagram je dodat kao prazna radna površina.

**Kreiranje novog UML entiteta**

Akter: Korisnik.

Preduslov: Korisnik je kreirao radnu površinu.

Koraci: Korisnik je kliknuo da doda novi entitet.

Tačke proširenja: Entitet je prikazan na radnoj površini..

**Brisanje UML entiteta**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Korisnik je kliknuo da obriše označeni entitet.

Tačke proširenja: Entitet se gubi sa radne površine i briše iz stabla.

**Modifikovanje UML entiteta**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Korisnik je kliknuo na opciju za izmenu entiteta.

Tačke proširenja: Entitet menja njegov izlged i strukturu u zavisnosti šta je korisnik promenio.

**Kreiranje metode**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Korisnik dodaje metode u svoj enetitet po njegovoj želji unosom teksta preko tastature.

Tačke proširenja: U entitet se dodaje njegov novi metod.

**Prikaz entiteta**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Automatski se prikazuje korisnikov entitet na njegovoj radnoj površini.

Tačke proširenja: Na radnoj površini se prikazuje korisnikov entitet.

**Kreiranje Interfejsa**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Korisnik bira tip entiteta da bude Interfejs odabirom u maloj ekranskoj formi klikom na miš.

Tačke proširenja: Entitet dobija tip Interfejsa.

**Kreiranje Enuma**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Korisnik bira tip entiteta da bude Enum odabirom u maloj ekranskoj formi klikom na miš.

Tačke proširenja: Entitet dobija tip Enuma.

**Kreiranje Klase**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet.

Koraci: Korisnik bira tip entiteta da bude Klasa odabirom u maloj ekranskoj formi klikom na miš.

Tačke proširenja: Entitet dobija tip Klase.

**Kreiranje argumenata**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML entitet i odabrao njegov tip.

Koraci: Preko tastature dodaje u njegov entiet argumente po želji.

Tačke proširenja: Entitet dobija novi argument.

**Kreiranje UML veze**

Akter: Korisnik.

Preduslov: Korisnik je kreirao UML dijagram.

Koraci: Korisnik je kliknuo da doda vezu, potom izabrao vrstu veze.

Tačke proširenja: EntitetI su spojeni nekom od vrsta veza koju je korisnik odabrao.

**Brisanje UML veze**

Akter: Korisnik.

Preduslov: Korisnik je dodao vezu izmedju entiteta.

Koraci: Korisnik je kliknuo na vezu koju želi da obriše, potom je kliknuo na dugme za brisanje.

Tačke proširenja: Između dva entiteta je obrisana UML veza.

**Modifikovanje UML veze**

Akter: Korisnik.

Preduslov: Korisnik je dodao vezu izmedju entiteta.

Koraci: Korisnik je kliknuo na vezu koju želi da modifikiju, potom je izabrao vrstu modifikaciju kojom želi da promeni svoju vezu.

Tačke proširenja: Data veza je promenjena u odnosu na korisnikovu želju.

**Prikaz UML veze**

Akter: Korisnik.

Preduslov: Korisnik je kreirao vezu između entiteta.

Koraci: Potrebno je samo da korisnik ispuni uslov kreiranja veze, potom se ona automatski pojavluje na njegovoj radnoj površini.

Tačke proširenja: Na ekranu se automatski dodaje veza između odabranih entiteta koju je korisnik izabrao.

**Generisanje UML koda**

Akter: Korisnik.

Preduslov: Korisnik je napravio UML dijagram.

Koraci: Poželjno je da kada se korisnik u potpunosti odluču da je njegom UML dijagram gotov, klikne na opciju generisanja njegovog UML koda.

Tačke proširenja: Ipsisuje se kod na osnovu njegovog UML dijagrama.

**Pristup informacijama**

Akter: Korisnik.

Preduslov: Korisnik je pokrenuo ClassyCrafT aplikaciju.

Koraci: Korisnik je kliknuo pristupu informacija.

Tačke proširenja: Korisniku se prikazuje na ekranu informacije koje moze da odabere da vidi.

**Otvaranje iformacija o kreatorima**

Akter: Korisnik.

Preduslov: Korisnik je kliknuo da pristupi informacijama.

Koraci: Korisnik je kliknuo na informacije o kreatorima.

Tačke proširenja: Na ekranu se prikazuju inforamcije o kreatorima.

**Otvaranje dokumentacije**

Akter: Korisnik.

Preduslov: Korisnik je kliknuo da pristupi informacijama.

Koraci: Korisnik je kliknuo otvaranje dokumentacije.

Tačke proširenja: Na ekranu se prikazuju stari fajlovi koje je korisnik nekada u prošlosti pravio.

**Ubacivanje strukture u stablo**

Akter: Korisnik.

Preduslov: Korisnik je napravio projekat.

Koraci: Automatski se prikazuje stablo na ekranu, korisnik ne treba ništa da uradi.

Tačke proširenja: Na ekranu se prikazuje stablo.

**Brisanje elementa stable**

Akter: Korisnik.

Preduslov: Korisnik je napravio, UML dijagram ili UML enetitet ili UML vezu.

Koraci: Korisnik je kliknuo označio element koji želi da obriše, potom kliknuo na dugme za brisanje.

Tačke proširenja: Sa ekrana se briše izabrani element.

**Modifikovanje stable**

Akter: Korisnik.

Preduslov: Korisnik je napravio, UML dijagram ili UML enetitet ili UML vezu.

Koraci: Korisnik .

Tačke proširenja: Na ekranu se prikazuje novo modifikovano stablo.

**Prikaz stabla**

Akter: Korisnik.

Preduslov: Korisnik je napravio neki UML dijagram, kako bi se napravio stablo.

Koraci: Automatski se prikazuje stablo na ekranu.

Tačke proširenja: Na ekranu se prikazuju stablo.