

ClassyCraft

Uvod

ClassyCraft je korisna i inovativna aplikacija koja pojednostavljuje proces dizajniranja UML (Unified Modeling Language) dijagrama. Bez obzira da li ste softverski inženjer, sistemski arhitekta ili student koji uči o dizajnu softvera, ClassyCraft pruža intuitivan i efikasan alat za kreiranje, uređivanje i vizualizaciju UML dijagrama sa lakoćom. Sa različitim prilagodljivim elementima i jednostavnim interfejsom, ClassyCraft omogućava korisnicima da svoje koncepte dizajna softvera pretvore u stvarnost, čineći ga nezamenjivim alatom za sve koji su uključeni u razvoj softvera i modeliranje sistema.

Rečnik

Dijagram - Vizualna reprezentacija sistema ili njegovih komponenti.

UML dijagram - Vizualna reprezentacija sistema ili njegovih komponenti pomoću Unified Modeling Language (UML).

Klasa - Model, šablon za kreiranje objekata u objektno orijentisanom programiranju.

Interfejs - Model koji opisuje neko ponašanje, ima skup operacija. Svrha interfejsa jeste da neka klasa imlementira te operacije.

Objekat - Konkretna instanca klase, odnosno njena promenljiva (Objekat je isto što i instanca).

Atribut - Polje, odnosno promenljive neke klase koje su direktno deklarisanе u klasi.

Metoda - Funkcija kojom možemo da promenimo stanje objekta.

Nasleđivanje - Mehanizam za stvaranje nove klase na osnovu postojeće, odnosno mehanizam u kojem jedna klasa stiče svojstva druge klase.

Veza - Povezanost ili asocijacija između klasa u dijagramu.

Veza asocijacije - Veza koja govori o tome da se objekat jedne klase može pojaviti kao atribut u drugoj klasi.

Veza agregacije - Veza u kojoj objekti neke klase nisu usko vezani za drugu klasu. Primer: klasa škola i klasa student, ako obrišemo školu studenti će i dalje postojati.

Veza kompozicije - Veza u kojoj su objekti neke klase usko vezani za drugu klasu. Primer: klasa kuća i klasa soba, ako obrišemo kuću sobe ne mogu da postoje, jer one ne mogu da postoje ako ne postoji kuća.

Veza zavisnosti - Govori o tome da metoda neke klase može da koristi objekat neke druge klase (na primer kao argument, povratnu vrednost ili lokalnu promenljivu).

Korisnici

Interesna grupa nije ograničena, ClassyCraft je pristupačan svim korisnicima, bez obzira na profesionalnu oblast.

Aplikacija omogućava korisnicima da lako stvaraju prilagođene UML dijagrame, pružajući slobodu u odabiru i prilagođavanju elemenata dijagrama. Za korišćenje aplikacije potrebno je samo osnovno razumevanje UML koncepata, a prethodno iskustvo sa grafičkim programima je korisno, ali nije obavezno.

ClassyCraft je namenjen za računare i podržava rad sa minimalno jednim korisnikom.

Konkurencija

U tabeli su prikazane konkurentske aplikacije koje se bave kreiranjem UML dijagrama, poput ClassyCraT-a, i funkcionalnosti koje svaka od njih implementira:

Ime	Web aplikacija	Desktop aplikacija	Mobilna aplikacija	Intuitivnost	Pristupačnost	Kolaborativnost	Generisanje koda	Integracija
ClassyCraT	✗	✓	✗	✓	✓	✓	✓	✗
Lucidcraft	✓	✗	✓	✓	✓	✓	✗	✓
Star UML	✗	✓	✗	✗	✗	✓	✓	✓
Diagrams.net	✓	✗	✗	✓	✓	✓	✗	✗
PlantUML	✗	✗	✗	✗	✗	✓	✗	✓

Problem

ClassyCraT rešava kompleksnost i izazove povezane sa dizajniranjem i vizualizacijom UML dijagrama. Tradicionalni procesi ručnog crtanja dijagrama često su vremenski zahtevni i podložni greškama. Uz to, razumevanje i primena UML koncepata može biti prepreka za mnoge korisnike. Naš alat omogućava brzo kreiranje i uređivanje UML dijagrama, eliminisajući potrebu za komplikovanim softverima i pružajući intuitivan interfejs za korisnike bez obzira na njihovo iskustvo. Takođe, olakšava saradnju između članova tima koji se bave dizajnom softvera, čineći ga neprocenjivim alatom za sve korisnike uključene u razvoj softvera i modelovanje sistema.

Domen zahteva

U nastavku su navedene funkcije koje ClassyCraft treba da ostvari. Radi bolje strukture, razmotrimo tri nivoa prioriteta:

- Nizak (program može u potpunosti da radi bez zadate funkcionalnosti)
- Srednji (poželjno je da program ispunjava zadatu funkcionalnost, ali nije obavezno)
- Visok (program ne može da funkcioniše bez zadate funkcionalnosti)

Takođe zahteve ćemo podeliti u dve glavne kategorije, odnosno funkcionalne i nefunkcionalne zahteve. Svaka od ovih kategorija biće podeljena na tri podkategorije: osnovni, opcioni i napredni zahtevi.

Funkcionalni zahtevi

1. Osnovni zahtevi

1.1 Kreiranje projekta

Korisnik treba da bude u mogućnosti da kreira novi projekat. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik

1.2 Kreiranje paketa

Korisnik treba da bude u mogućnosti da u okviru svog projekta kreira novi paket. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.3 Kreiranje UML klasnog dijagrama

Korisnik treba da bude u mogućnosti da u okviru svog paketa kreira novi UML klasni dijagram. Prioritet je visok dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.4 Otvaranje toolbar-a sa elementima

Korisnik prilikom kreiranja UML klasnog dijagrama treba da bude u mogućnosti da pristupi svim ponuđenim elementima iz toolbar-a sa elementima tako što će mu se prikazati na ekranu. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.5 Prikaz toolbar-a sa elementima

Toolbar sa elementima treba grafički biti prikazan korisniku, kako bi mogao uspešno da bira elemente za kreiranje klasnog dijagrama. Prioritet je visok dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.6 Kreiranje elemenata

Korisnik treba da bude u mogućnosti da izabere i kreira neke od ponuđenih elemenata (klase, interfejsi i veze). Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.7 Kreiranje klasa

Korisnik treba da bude u mogućnosti da kreira klasu na radnoj površini kako bi kreirao UML dijagram. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.8 Kreiranje interfejsa

Korisnik treba da bude u mogućnosti da kreira interfejs na radnoj površini kako bi kreirao UML dijagram. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.9 Kreiranje veza

Korisnik treba da bude u mogućnosti da uspostavi neke od veza između klasa i interfejsa. Te veze su: veza generalizacije, realizacije, kompozicije, agregacije, asocijacije i veza zavisnosti. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.10 Prikaz elemenata

Korisnik prilikom rada na klasnom dijagramu treba da bude u mogućnosti da vidi sve kreirane elemente raspoređene na radnoj površini. Prioritet je visok, rizik je visok. Zainteresovana strana je korisnik.

1.11 Duplikacija elemenata

Korisnik treba da bude u mogućnosti da kopira postojeće elemente radi bržeg modelovanja. Prioritet je visok, dok je rizik srednji. Zainteresovana strana je korisnik.

1.12 Brisanje elemenata

Korisnik treba da bude u mogućnosti da obriše postojeće elemente sa radne površine. Prioritet je visok. Dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.13 Modifikovanje elemenata

Već kreirani elementi trebaju da budu u stanju da im se sadržaj kao i dizajn mogu promeniti. Naravno vezama samo dizajn. Prioritet je visok, dok je rizik srednji. Zainteresovana strana je korisnik.

1.14 Ponovno uspostavljanje veze (prepovezivanje)

Korisniku treba da se omogući da izvrši jednostavno prepovezivanje klasa i interfejsa. Prioritet je visok, dok je rizik srednji. Zainteresovana strana je korisnik.

1.15 Raspoređivanje po radnoj površini

Potrebno je da korisnik bude u mogućnosti da raspoređuje klase i interfejse po površini bez gubljenja veza. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.16 Dodavanje članova klasama i interfejsima

Korisnik treba da bude u mogućnosti da doda neke od članova klasama ili interfejsima. Za klasu su dostupni: atributi, metode i konstruktori. Dok su za interfejs dostupne samo metode. Prioritet je visok, dok je rizik srednji. Zainteresovana strana je korisnik.

1.17 Modifikovanje članova klasa i interfejsa

Korisnik treba da bude u mogućnosti da modifikuje članove klasama i interfejsima. Ovo modifikovanje se odnosi na promenu njihovog naziva. Prioritet je visok, dok je rizik srednji. Zainteresovana strana je korisnik.

1.18 Prikaz članova klasa i interfejsa

Već kreirani članovi trebaju biti grafički prikazani u okviru klase ili interfejsa na radnoj površini. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.19 Brisanje članova klasa i interfejsa

Korisnik treba da bude u mogućnosti da obriše kreirane članove unutar klasa i interfejsa. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.20 Implementacija stabla struktura podataka

Stablo strukture podataka treba da ponudi korisniku čist i organizovan pogled na hijerarhiju fajlova, prateći strukturu Project -> Paket -> Diagram. Prioritet je visok, dok je rizik srednji. Zainteresovana strana je korisnik.

1.21 Prikaz stabla struktura podataka

Korisnik prilikom rada na klasnom dijagramu treba da bude u mogućnosti da vidi sve elemente svog klasnog dijagrama raspoređene u stablo sa leve strane. Prioritet je visok, rizik je nizak. Zainteresovana strana je korisnik.

2. Opcioni zahtevi

2.1 Pristup informacijama

Korisnik treba da bude u mogućnosti da pristupi informacijama o samoj aplikaciji. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana su kreatori.

2.2 Otvaranje informacija o kreatorima

Korisnik treba da bude u mogućnosti da vidi osnovne informacije o kreatorima aplikacije. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana su kreatori.

2.3 Otvaranje dokumentacije

Korisnik treba da bude u mogućnosti da otvori dokumentaciju aplikacije. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana su kreatori.

2.1 Generisanje koda na osnovu dijagrama

Korisnik treba da bude u mogućnosti da na osnovu napravljenog UML dijagrama generiše kod. Prioritet je srednji, dok je rizik nizak. Zainteresovana strana je korisnik.

2.2 Čuvanje i učitavanje UML dijagrama

UML dijagram koji korisnik napravi treba da bude sačuvan negde na računaru u određenom obliku tako da je kasnije moguće taj UML dijagram opet otvoriti. Prioritet je srednji, dok je rizik nizak. Zainteresovana strana je korisnik.

2.3 Automatsko generisanje getera i setera za napravljeni atribut

Korisnik treba da bude u mogućnosti da automatski generiše geter i seter za napravljeni atribut. Prioritet je srednji, dok je rizik takođe srednji. Zainteresovana strana je korisnik.

2.4 Brisanje višestrukih elemenata odjednom

Korisnik treba da bude u mogućnosti da briše više elemenata dijagrama od jednom kako bi se povećala njegova produktivnost. Prioritet je srednji, dok je rizik nizak. Zainteresovana strana je korisnik.

2.5 Menjanje boje elemenata dijagrama

Osim pravljenja UML klasnog dijagrama korisniku treba biti pružena i mogućnost stilizovanja istog. Odnosno korisnik treba da bude u mogućnosti da menja boju elementima dijagrama. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

2.6 Mogućnost dodavanja komentara i opisa

Korisnik treba da bude u mogućnosti da doda komentare i opise na postojeće elemente dijagrama kako bi se poboljšala dokumentacija i razumevanje. Prioritet je srednji, dok je rizik nizak. Zainteresovana strana je korisnik.

2.7 Grupisanje elemenata

Korisnik treba da bude u mogućnosti da grupiše određene elemente klasnog dijagrama jer grupisanje pruža bolju organizaciju elemenata na dijagramu. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

3. Napredni zahtevi

3.1 Pravljenje drugih tipova dijagrama

Osim klasnog dijagrama korisniku treba biti pružena mogućnost kreiranja i drugih tipova UML dijagrama: use case dijagram, profilni dijagram, sekvencni dijagram... Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

3.2 Generisanje UML dijagrama na osnovu priloženog koda

Korisnik treba da bude u mogućnosti da na osnovu priloženog koda generiše UML dijagram. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

3.3 Real-time kolaboracija

Korisnicima treba biti pružena mogućnost da više korisnika zajedno uređuju dijagram u realnom vremenu. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

3.4 Analiza i validacija koda

Korisnik treba da bude u mogućnosti da za priloženi kod dobije povratnu informaciju da li on ispunjava određene standarde i pravila programiranja.

Nefunkcionalni zahtevi

1. Osnovni zahtevi

1.1 Brzina odziva

Aplikacija mora da bude u stanju da obezbedi brz odgovor na korisničke zahteve, sa vremenom odziva manjim od 200ms. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.2 Pouzdanost i stabilnost

Aplikacija treba da bude stabilna i pouzdana, sa minimalnim brojem rušenja ili grešaka. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

1.3 Prikaz napravljenih elemenata

Novonapravljeni pojmovi UML dijagrama trebaju da budu predstavljeni na grafičkoj površini kako bi korisnik imao pregled šta je napravio. Prioritet je visok, dok je rizik takođe visok. Zainteresovana strana je korisnik.

2. Opcioni zahtevi

2.1 Validacija UML dijagrama

Korisniku treba prikazati error poruku ako je pokušao da uradi nešto što po pravilu i razumevanju objektno orijentisanog programiranja nije dozvoljeno, na primer vezivanje dva interfejsa sa vezom generalizacije. Prioritet je srednji, dok je rizik nizak. Zainteresovana strana je korisnik.

2.2 Generisanje izveštaja i statistika

Aplikacija treba da bude u mogućnosti da generiše izveštaje o klasnom dijagramu, uključujući statistike o broju elemenata, veza i drugih karakteristika. Prioritet je nizak, dok je rizik nizak. Zainteresovana strana je korisnik.

3. Napredni zahtevi

3.1 Implementacija tehnologije za mašinsko učenje i veštačku inteligenciju

Aplikacija treba da implementira alate za mašinsko učenje i veštačku inteligenciju kako bi svojim korisnicima pružala funkcionalnosti poput automatskog generisanja sugestija za optimizaciju dizajna. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

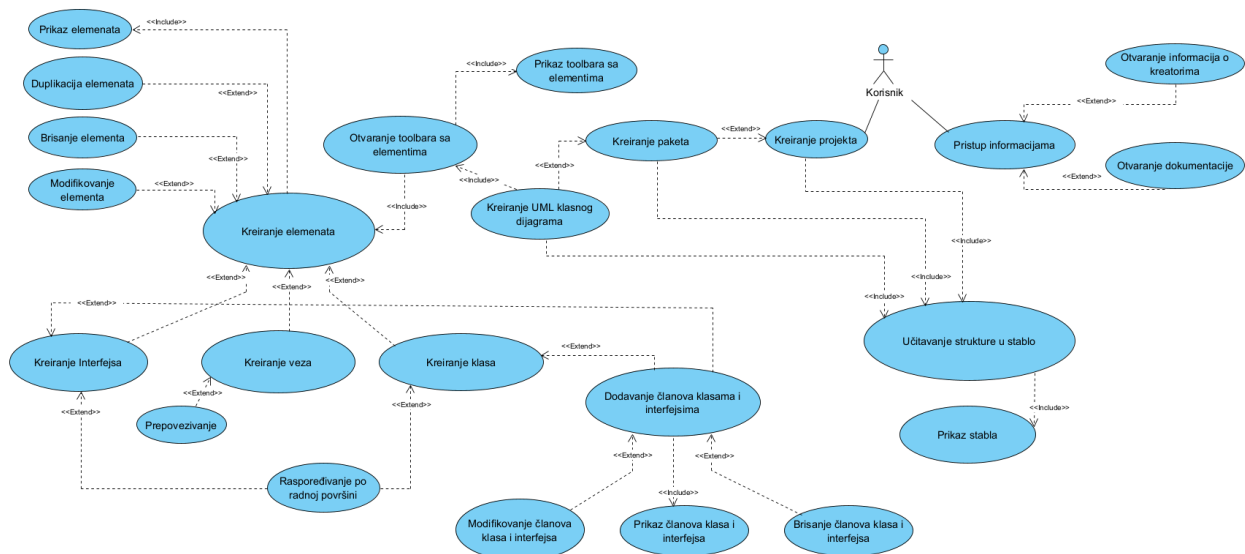
3.2 Istorija izmena i povratak na prethodnu verziju dijagrama

Korisnik treba da bude u mogućnosti da prati istoriju izmena na klasnom dijagramu kao i da ima mogućnost da se vrati na prethodne verzije dijagrama. Prioritet je nizak, dok je rizik takođe nizak. Zainteresovana strana je korisnik.

Funkcionalni dijagrami

Use Case diagram

Ispod se nalazi slika use case dijagrama softvera ClassyCraft



Kreiranje projekta

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio program ClassyCraft

Opis: Korisnik uz pomoć menija pravi novi projekat

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti

Stanje: U stablo se ubacuje novi projekat i prikazuje se njegovo ime sa leve strane aplikacije

Extension points: Kreiranje paketa

Kreiranje paketa

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio projekat

Opis: Korisnik može da u okviru svog projekta kreira novi paket, u okviru koga će biti u mogućnosti da kreira UML klasne dijagrame.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti

Stanje: U projekat se ubacuje novi paket i prikazuje se njegovo ime sa leve strane aplikacije podvučeno ispod projekta u kom se nalazi.

Extension points: Kreiranje UML klasnog dijagrama

Kreiranje UML klasnog dijagrama

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao paket

Opis: Korisnik može da u okviru svog paketa kreira novi UML klasni dijagram.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti

Stanje: U paket se ubacuje novi UML klasni dijagram odnosno novi fajl i prikazuje se njegovo ime sa leve strane aplikacije podvučeno ispod paketa u kom se nalazi.

Otvaranje toolbara sa elementima

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao UML klasni dijagram

Opis: Korisniku se otvara toolbar sa svim elementima (klase, veze...) ali je zaključan dok se ne kreira UML klasni dijagram.

Izuzetak: Korisnik može da pokuša da koristi toolbar a da prethodno nije napravio UML klasni dijagram, pri čemu će morati da ga kreira kako bi bio u mogućnosti ga koristiti.

Stanje: Aplikacija otvara toolbar sa svim elementima koje će korisnik moći da koristi.

Prikaz toolbara sa elementima

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao UML klasni dijagram.

Opis: Toolbar sa elementima se vizuelno prikazuje na određenom delu aplikacije.

Izuzetak: Korisnik ne utiče na ovu funkcionalnost.

Stanje: Na delu ekrana namenjen za prikaz našeg toolbara sa elementima, on se vizuelno prikazuje.

Kreiranje elemenata

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio toolbar sa elementima

Opis: Korisniku se pruža mogućnost da kreira neki od ponuđenih elemenata: klase, interfejsi i veze.

Izuzetak: Korisnik može da pokuša da kreira neki od elemenata a da mu se prethodno nije otvorio toolbar sa elementima.

Stanje: Aplikacija omogućuje korisniku da kreira neki od elemenata.

Extension points: Duplikacija elemenata, Brisanje elemenata, Modifikovanje elemenata, Kreiranje klase, Kreiranje veza, Kreiranje interfejsa.

Kreiranje interfejsa

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio odnosno otključao mogućnost korišćenja toolbara sa elementima.

Opis: Na radnoj površini se kreira interfejs.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Radna površina dobija novi element odnosno interfejs.

Extension points: Raspoređivanje po radnoj površini, Dodavanje članova klasama i interfejsima.

Kreiranje veza

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio odnosno otključao mogućnost korišćenja toolbara sa elementima.

Opis: Na radnoj površini se kreira neka od izabranih veza.

Izuzetak: Korisnik može da pokuša da uspostavi pogrešnu vezu između elemenata, tada mu se na ekranu pojavljuje upozorenje da takva veza nije moguća i biće primoran da ponovo uspostavi novu vezu.

Stanje: Radna površina dobija novi element odnosno vezu.

Extension points: Prepovezivanje

Prepovezivanje

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao neku od veza.

Opis: Korisnik može postojeću vezu da prepoveže odnosno da prespoji elemente.

Izuzetak: Korisnik može da pokuša da prepoveže elemente ali da ta veza nije dozvoljena između elemenata koje on želi da prepoveže, tada mu se na ekranu pojavljuje upozorenje da takva veza nije moguća i u tom slučaju ostaje veza koja je postojala pre prepovezivanja.

Stanje: Na radnoj površini se postojeća veza menja, odnosno drugi elementi će biti povezani.

Kreiranje klasa

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio odnosno otključao mogućnost korišćenja biblioteke sa elementima.

Opis: Na radnoj površini se kreira klasa

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Radna površina dobija novi element odnosno klasu.

Extension points: Raspoređivanje po radnoj površini, Dodavanje članova klasama i interfejsima.

Raspoređivanje po radnoj površini

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao klasu ili interfejs.

Opis: Korisnik može da izabere klasu ili interfejs i da ih rasporedi po površini po svojoj želji, pri čemu veza između njih ostaje i ne prekida se.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Klasa ili interfejs dobija novu poziciju na radnoj površini.

Modifikovanje elemenata

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao klasu ili interfejs.

Opis: Korisnik može da modifikuje klasu ili interfejs, tako što će im menjati sadržaj ili dizajn.

Izuzetak: Korisnik može da modifikuje klasu ili interfejs koju nije želeo da modifikuje. Tada će morati ručno da ispravi sve greške odnosno ručno da ukloni sve modifikacije.

Stanje: Klasa ili interfejs biva ažurirana odnosno promeniće se njen sadržaj ili dizajn.

Brisanje elemenata

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao neki od elemenata kako bi mogao ga obrisati.

Opis: Korisnik bira element i briše ga.

Izuzetak: Korisnik može da slučajno obrisati element koji nije želeo. Tada će morati da ga kreira ponovo kao i da ga modifikuje ukoliko je postojala nekakva modifikacija.

Stanje: Izabrani element biva obrisani sa radne površine.

Prikaz elemenata

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao neki od elemenata.

Opis: Grafički se na radnoj površini prikazuje novonastali element.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Na radnoj površini se vizuelno prikazuje kreirani element.

Duplikacija elemenata

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao klasu ili interfejs kako bi mogao da ih duplira.

Opis: Korisnik može da izabere klasu, interfejs i da ih duplira

Izuzetak: Korisnik ne može da duplira element ako ga u prvom planu nije ni kreirao. Dakle ovu funkciju tada uopšte neće moći da koristi.

Stanje: Na randoj površini se pojavljuje duplikat postojećeg elementa kog smo izabrali.

Dodavanje članova klasama i interfejsima

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao klasu ili interfejs.

Opis: Korisnik može da doda nekog člana u izabranu klasu ili interfejs. Članovi su: atribut, metoda, konstruktor.

Izuzetak: Korisnik može da doda konstruktor ili atribut u interfejs što nije dozvoljeno, tada biva upozoren i njegova radnja se ne izvršava.

Stanje: U okviru klase ili interfejsa se pojavljuje novi član, isto tako ova promena treba da bude vidljiva na radnoj površini na kojoj se nalazi ta klasa ili interfejs.

Extension points: Modifikovanje članova klasa i interfejsa, Brisanje članova klasa i interfejsa.

Modifikovanje članova klasa i interfejsa

Učesnik: Korisnik

Preduslov: Korisnik mora da je prethodno kreirao člana klase ili interfejsa

Opis: Korisnik može da modofikuje postojećeg člana.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Postojeći član biva ažuriran i ta promena se odražava na radnoj površini.

Prikaz članova klase i interfejsa

Učesnik: Korisnik

Preduslov: Korisnik mora da je prethodno kreirao člana klase ili interfejsa

Opis: U okviru klase ili interfejsa novonastali član treba da bude prikazan.

Izuzetak: Korisnik ne utiče na ovu funkcionalnost.

Stanje: Nakon kreiranja novog člana klase ili interfejsa, on biva prikazan u okviru te klase ili interfejsa. Isto tako biće vidljiv na celoj radnoj površini.

Brisanje članova klase i interfejsa

Učesnik: Korisnik

Preduslov: Korisnik mora da je prethodno kreirao člana klase ili interfejsa

Opis: Izabrani član u okviru klase ili interfejsa biva obrisani.

Izuzetak: Korisnik može da obriše člana kojeg nije nameravao, tada će morati ručno da ga doda nazad.

Stanje: Izabrani član biva obrisani iz klase ili interfejsa i ta promena se isto odražava i na radnu površinu odnosno, obrisani član više neće postojati.

Učitavanje strukture u stablo

Učesnik: Korisnik

Preduslov: Korisnik mora da je prethodno UML klasni dijagram.

Opis: Novokreirani element se ubacuje u naše stablo podataka.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Nakon kreiranja bilo koje strukture, ono se ubacuje u stablo komponentata.

Prikaz stabla

Učesnik: Korisnik

Preduslov: Korisnik mora da je prethodno kreirao UML klasni dijagram.

Opis: Na delu ekrana namenjen za prikaz našeg stabla ono se vizuelno prikazuje.

Izuzetak: Korisnik ne utiče na ovu funkcionalnost.

Stanje: Tokom korišćenja programa ClassyCraft je moguće videti trenutno stanje naših elemenata poređanih u strukturu podataka oblika stabla.

Pristup informacijama

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio program ClassyCraft.

Opis: Korisnik uz pomoć menija dolazi do liste sa svim informacijama vezanim za ClassyCraft.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Aplikacija prikazuje listu sa svim informacijama vezanim za ClassyCraft.

Extension points: Otvaranje informacija o kreatorima, Otvaranje dokumentacije.

Otvaranje dokumentacije

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio meni sa svim informacijama vezanim za ClassyCraft

Opis: Korisnik otvara kompletnu dokumentaciju za ClassyCraft koja se prikazuje na aplikaciji.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti.

Stanje: Aplikacija prikazuje nov prozor sa kompletnom dokumentacijom za ClassyCraft.

Otvaranje informacija o kreatorima

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno otvorio meni sa svim informacijama vezanim za ClassyCraft

Opis: Korisnik otvara nov prozor sa informacijama o kreatorima koji se prikazuje na aplikaciji.

Izuzetak: Korisnik ne može da napravi nikakvu neočekivanu grešku pri ovoj funkcionalnosti

Stanje: Aplikacija prikazuje nov prozor sa informacijama o njenim kreatorima.