

# ClassyCraft

Viktor Davidović RN 38/22

Viktor Milojević RN 52/22

## Uvod

Urednik UML dijagrama ClassyCraft je alat koji korisnicima omogućava kreiranje i menjanje UML dijagrama. ClassyCraft je program u obliku desktop aplikacije koji omogućava kreiranje novih entiteta, njihovo menjanje, kao i povezivanje entiteta različitim vrstama veza.

## Rečnik

**UML dijagram** - UML dijagrami su vizuelni alati koji pojednostavljaju složene sisteme koristeći oblike i linije za predstavljanje i objašnjenje različitih entiteta i njihovih veza.

**Entitet** - Opis jedne klase ili interfejsa unutar UML dijagrama. Prikazan grafički kao pravougaoni objekat.

**Veza (Odnos)** - Pokazuje različite vrste logičkih relacija unutar jednog dijagrama. Grafički je prikazana kao neka vrsta linije između entiteta.

**Interfejs** - Skup pravila koja definišu koje radnje objekat može da izvrši, pomažući različitim objektima da efikasno rade zajedno.

**Atribut** - U objektno-orijentisanom programiranju, atribut je kao karakteristika ili svojstvo koje opisuje objekat, kao što je njegova boja, veličina ili ime.

**Klasa** - Osnova objektno-orijentisanog programiranja. Opis nečega. Na primer, klasa "Osoba" može imati attribute ime, prezime, godište, JMBG...

**Metoda** - Niz operacija koji govori računaru kako da izvrši određeni zadatak.

## Konkurencija

U tabeli su prikazani konkurenti koji rade na kreiranju UML dijagrama slično kao i ClassyCraft i funkcionalnosti koje svaki od njih implementiraju:

Ime	Desktop klijent	Web verzija aplikacije	Mobilna verzija aplikacije	Koriscenje bez registracije	Besplatna verzija	Generisanje Java koda
ClassyCraft	✓	✗	✗	✓	✓	✓
Lucidchart	✗	✓	✓	✗	✓	✓
Visual Paradigm	✓	✓	✓	✗	✓	✓
Draw.io	✗	✓	✓	✓	✓	✗

StarUML	✓	✓	✗	✗	✗	✓
Umbrello UML Modeller	✓	✗	✗	✗	✗	✗
Gliffy	✗	✓	✗	✗	✗	✗
yEd	✓	✓	✗	✓	✓	✗

## Korisnici

ClassyCraft je svestran alat koji pojednostavljuje kreiranje i dokumentovanje UML dijagrama, pružajući korisničko iskustvo i početnicima i iskusnim korisnicima. On pojednostavljuje vizuelni prikaz dizajna sistema kroz UML dijagrame, poboljšavajući planiranje projekta, komunikaciju i dokumentaciju. Pored toga, nudi funkcionalnost za generisanje Java koda na osnovu UML dijagrama kao i generisanje UML dijagrama na osnovu Java koda.

Pre upotrebe ovog programa, poželjno je osnovno razumevanje UML koncepata i principa razvoja softvera.

Aplikacija je prvenstveno dizajnirana da zadovolji potrebe programera softvera, sistemskih arhitekata i obrazovnih institucija.

## Zahtevi

U nastavku se nalazi lista funkcija koju ClassyCraft treba da ispuni. Zbog jasnijeg opisa se uvode tri nivoa prioriteta:

- Visok (Program ne može da funkcioniše bez zadate funkcionalnosti)
- Srednji (Poželjno je da program ima zadatu funkcionalnost)
- Nizak (Program može da u potpunosti funkcioniše bez zadate funkcionalnosti)

## 1. Osnovne operacije nad elementima UML dijagrama

### 1.1 Kreiranje i brisanje UML elemenata

Korisnik treba da bude omogućen da lako doda UML elemente po svojoj želji, kao i da ih obriše sa dijagrama. Korisnik može kreirati klasu koja sadrži attribute i metode, ili interfejs koji sadrži samo metode. Prioritet je visok, rizik postoji u slučaju brisanja pogrešnih elemenata. Zainteresovana strana je korisnik.

### 1.2 Modifikovanje UML elemenata

Svaki UML element koji korisnik kreira, može se naknadno izmeniti. Prioritet je srednji, dok je rizik nizak. Zainteresovana strana je korisnik.

### 1.3 Raspoređivanje UML elemenata po radnoj površini

Korisnik može da UML elemente pomera po radnoj površini i tako ih raspoređuje na sebi najpregledniji način. Prioritet je srednji, rizik nizak, a zainteresovana strana je korisnik.

### 1.4 Vizuelizacija UML elemenata

UML elementi se grafički prikazuju na radnoj površini u trenutku kada ih korisnik kreira. Prioritet je visok, rizik nizak, a zainteresovana strana je korisnik.

## 2. Osnovne operacije nad UML vezama

### 2.1 Kreiranje i brisanje UML veza

Korisnik može da kreira i briše veze između entiteta. Pre povezivanja elemenata vezom, korisnik bira između 4 tipa veze, a to su: asocijacija, kompozicija, agregacija, dependency. Prioritet je visok, a rizik postoji u slučaju odabira pogrešnih veza. Zainteresovana strana je korisnik.

### 2.2 Modifikovanje UML veza

Korisnik je u mogućnosti da već kreiranim vezama promeni tip ili entitet koje povezuju. Prioritet je nizak. Rizik je nizak, a zainteresovana strana je korisnik.

### **3. Operacije nad celokupnim UML dijagramima**

#### **3.1 Kreiranje novog projekta**

Korisnik treba da bude u mogućnosti da kreira novi projekat koji može da sadrži više paketa (i podpaketa) u kojima se može nalaziti više UML dijagrama. Prioritet je visok, rizik je nizak. Zainteresovana strana: korisnik.

#### **3.2 Kreiranje novog paketa**

Korisnik treba da bude u mogućnosti da kreira novi paket, ili paket unutar paketa, koji može da sadrži više UML dijagrama. Prioritet je visok, rizik je nizak. Zainteresovana strana: korisnik.

#### **3.3 Kreiranje novog UML dijagrama**

Korisnik treba da bude u mogućnosti da u svakom trenutku napravi novi radnu površinu unutar prethodno kreiranog paketa na kojoj će moći da ispočetka kreira novi UML dijagram. Prioritet je visok, rizik je nizak. Zainteresovana strana: korisnik.

#### **3.4 Čuvanje i učitavanje UML dijagrama**

UML dijagram koji korisnik napravi treba da bude sačuvan negde na računaru u određenom obliku tako da je moguće kasnije mapu opet otvoriti. Prioritet je srednji, dok postoji rizik da se ceo rad korisnika izgubi zbog neispravnog čuvanja UML dijagrama. Zainteresovana strana: korisnik.

#### **3.5 Čuvanje UML dijagrama u obliku Java koda**

Korisnik treba da bude u mogućnosti da u svakom trenutku može da generiše Java kod na osnovu prethodno kreiranog UML dijagrama. Prioritet je srednji, dok postoji rizik da UML dijagram nije smisleno sklopljen. Zainteresovana strana: korisnik.

#### **3.6 Kreiranje UML dijagrama na osnovu Java projekta**

Korisnik treba da bude u mogućnosti da u svakom trenutku može da generiše UML dijagram na osnovu prethodno kreiranog Java projekta. Prioritet je nizak, dok postoji rizik da Java projekat nije smisleno napravljen. Zainteresovana strana: korisnik.

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

#### **4.1 Implementacija stabla struktura podataka**

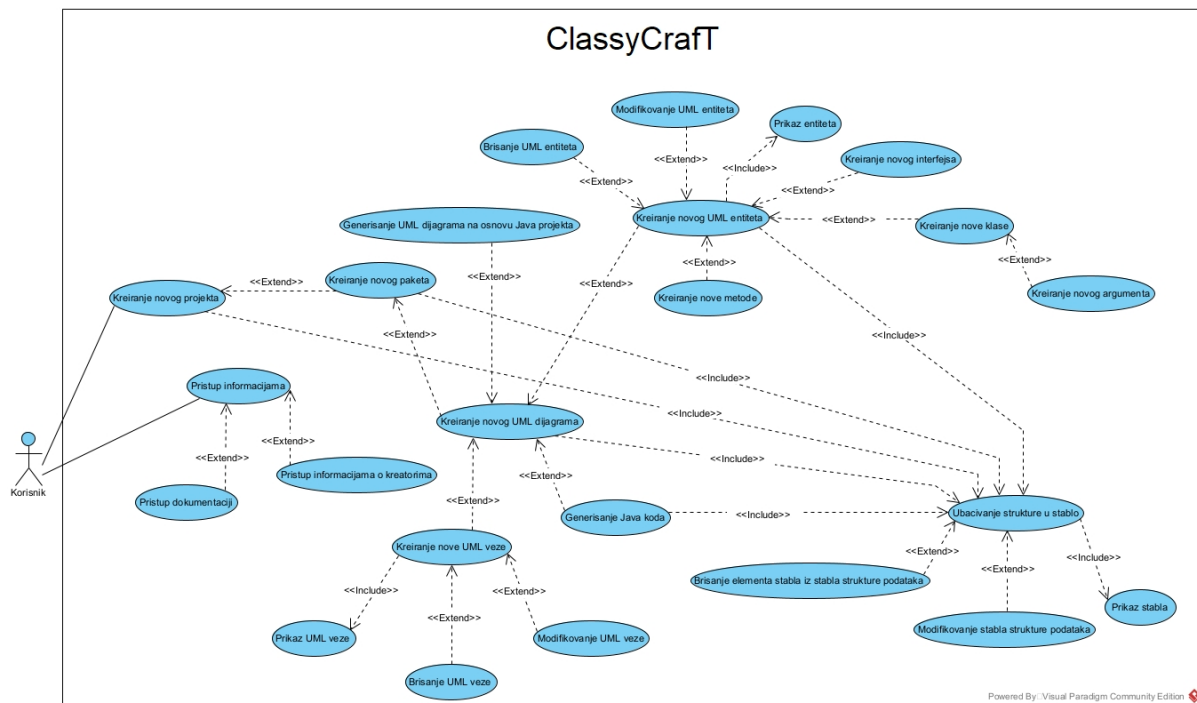
Aplikacija, u pozadini, sadrži stablo struktura podataka. U kojima se, redom po nivoima, nalaze: projekti, paketi i podpaketi, dijagrami. Oni su u stablu raspoređeni u odnosu na mesto u strukturi podataka koja je izabrana u vreme kreiranja nekog od ovih vrsta podataka.

#### **4.2 Prikaz stabla struktura podataka**

Stablo strukture podataka se u ClassyCraft aplikaciji nalazi sa leve strane i prikazano je kao lista projekata koji svaki ima u sebi padajuću listu paketa i podpaketa od kojih svaki ima svoju padajuću listu dijagrama. Klikom na strelicu pored imena projekta ili paketa otvaramo ili zatvaramo njegovu padajuću listu. Duplim klikom na neki od dijagrama, njegov sadržaj se prikazuje na radnoj površini sa desne strane.

## **Funkcionalni dijagram**

Ispod se nalazi slika UseCase dijagrama za aplikaciju ClassyCraft.



ClassyCraft UseCase dijagram

## UseCase

### Kreiranje novog projekta

Učesnik: Korisnik

Preduslov: Korisnik je otvorio ClassyCraft aplikaciju.

Opis: Korisnik uz pomoć menija bira ime projekta i lokaciju na disku na kojoj želi da se projekat kreira. Korisnik zatim potvrđuje akciju.

Izuzetak: Korisnik nije u mogućnosti da napravi neočekivanu grešku.

Koraci: Korisnik klikne dugme, unese ime projekta, unese autora projekta i unese mesto na disku gde će se projekat nalaziti.

Tačke proširenja: Moguće je kreiranje novog paketa, ubacuje se projekat u stablo strukture podataka.

Stanje: Kreira se novi projekat i prikazuje se u stablu strukture podataka.

### Kreiranje novog paketa ili podpaketa

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio projekat.

Opis: Korisnik uz pomoć menija bira projekat i u njemu pravi novi paket ili podpaket (paket unutar paketa), koji mogu da sadrže više UML dijagrama.

Izuzetak: Korisnik nije u mogućnosti da napravi neočekivanu grešku.

Koraci: Korisnik klikne na projekat ili paket, klikne na dugme, i unese ime paketa.

Tačke proširenja: Moguće je kreiranje novog UML dijagrama, ubacuje se paket u stablo strukture podataka.

Stanje: Kreira se novi paket i prikazuje se u stablu strukture podataka pod izabranim projektom.

### Kreiranje novog UML dijagrama

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio paket.

Opis: Korisnik uz pomoć menija briše sve elemente koji se trenutno nalaze na radnoj površini, a ukoliko radna površina ne postoji on napravi praznu i pri tom kreira novi UML dijagram.

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

Koraci: Korisnik klikne na paket, klikne na dugme i unese ime UML dijagrama.

Tačke proširenja: Moguće je generisanje Java koda, kreiranje novog UML entiteta i veze, dijagram se dodaje u stablo strukture podataka.

Stanje: Pojavljuje se prazna površina po kojoj korisnik može da kreira nove elemente i UML dijagram se ubacuje u stablo pod odabranim paketom

## **Kreiranje novog UML entiteta**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio UML dijagram.

Opis: Korisnik klikom na UML dijagram otvara dijalog za kreiranje novog UML entiteta i unosi njegovo ime, a potom bira njegovu poziciju klikom na radnu površinu.

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

Koraci: Korisnik klikne na UML dijagram, klikne na dugme, unese ime entiteta i onda klikne na radnu površinu.

Tačke proširenja: Moguće je brisanje UML entiteta, njegovo modifikovanje, kreiranje novih metoda, kreiranje klase ili interfejsa. Entitet se prikazuje na radnoj površini i u stablu strukture podataka.

Stanje: Novi UML entitet se dodaje na dijagram.

## **Kreiranje nove klase**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio UML dijagram.

Opis: Na radnoj površini se stvara nova klasa.

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

Koraci: Nakon kreiranja entiteta, korisnik klikne na dugme za klasu.

Tačke proširenja: Moguće je kreiranje novog argumenta.

Stanje: Radna površina dobija novi element koji je tipa klasa.

## **Kreiranje novog interfejsa**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio UML dijagram.

Opis: Na radnoj površini se stvara novi interfejs.

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

Koraci: Nakon kreiranja entiteta, korisnik klikne na dugme za interfejs.

Tačke proširenja: Nema daljih tački proširenja sem onih vezanih za kreiranje entiteta.

Stanje: Radna površina dobija novi element koji je tipa interfejs.

## **Kreiranje novog argumenta**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao klasu.

Opis: Na radnoj površini unutar klase se pojavljuje kreirani argument.

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

Koraci: Korisnik klikne na klasu, zatim na dugme za kreiranje novog argumenta, unosi njegovo ime, tip, vidljivost, da li je argument final/static.

Tačke proširenja: Nema daljih tački proširenja.

Stanje: Na radnoj površini se prikazuje kreiran argument u izabranoj klasi

## **Kreiranje nove metode**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio klasu ili interfejs

Opis: Na radnoj površini se unutar izabrane klase ili interfejsa prikazuje novonastala metoda. Ukoliko je metoda kreirana unutar klase koja već poseduje neke argumente onda se prikazuje ispod tih argumenata.

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

Koraci: Korisnik klikne na interfejs ili klasu, klikne na dugme za dodavanje metode. Unosi njeno ime, tip povratne vrednosti, vidljivost, da li je static.

Tačke proširenja: Nema daljih tački proširenja.

Stanje: Na radnoj površini se prikazuje kreirana metoda u izabranoj klasi ili metodi.

## **Kreiranje UML veze**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno napravio najmanje dva UML entiteta.

Opis: Na radnoj površini se između dve prethodno napravljene klase ili interfejsa pojavljuje novonastala veza. Veza može biti asocijacija, kompozicija, agregacija i veza zavisnosti (dependency), u zavisnosti od namene u UML dijagramu.

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

Koraci: Korisnik klikne na dugme, bira vrstu veze, zatim klikne na jedan pa na drugi entitet.

Tačke proširenja: Moguće je modifikovanje veze i njeno brisanje. Veza se prikazuje na radnoj površini.

Stanje: Na radnoj površini se prikazuje kreirana veza između željenih klasa ili interfejsa.

## **Modifikovanje stabla strukture podataka**

Učesnik: Korisnik

Preduslov: Korisnik je prethodno kreirao projekat, pakete ili dijagrame kojima želi da promeni mesto u stablu strukture podataka. Ukoliko se pomera paket, on je prazan i ne sadrži dijagrame ili podpakete.

Opis: Korisnik prevlačenjem pomera paket, podpaket ili dijagram u neki drugi paket (podpaket)

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

Koraci: Korisnik mišem prevlači element iz stabla na drugo mesto u stablu.

Tačke proširenja: Nema daljih tački proširenja.

Stanje: Stablo strukture podataka sa leve strane prikazuje izmenjeno stanje.

## **Modifikovanje UML entiteta**

Učesnik: Korisnik

Preduslov: Korisnik je prethodno kreirao UML entitet.

Opis: Korisnik može da promeni ime entiteta, kao i atribut ili metode koji se u njemu nalaze.

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

Koraci: Korisnik klikne na entitet, zatim može da promeni ime entiteta ili obriše neki atribut/metodu.

Tačke proširenja: Nema daljih tački proširenja.

Stanje: Izabranom UML entitetu se menjaju osobine i promene se prikazuju na radnoj površini.

## **Modifikovanje UML veze**

Učesnik: Korisnik

Preduslov: Korisnik je prethodno kreirao UML vezu.

Opis: Korisnik može da promeni vrstu veze kao i koje entitete ona spaja.

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

Koraci: Korisnik klikne na vezu, zatim bira vrstu veze u koju želi da je promeni.

Tačke proširenja: Nema daljih tački proširenja.

Stanje: Izabranoj UML vezi se menjaju osobine i promene se prikazuju na radnoj površini.

## **Brisanje elementa stabla iz stabla strukture podataka**

Učesnik: Korisnik

Preduslov: Korisnik je kreirao neki od elemenata stabla strukture podataka.

Opis: Korisnik odabirom i pritiskom na dugme briše odabrani element iz stabla strukture podataka. Ukoliko element ima podelemente prikazuje se upozorenje pre brisanja odabranog elementa.

Izuzetak: Korisnik može da obriše pogrešan element iz stabla strukture podataka.

Koraci: Korisnik klikne na neki element iz stabla, zatim klikne dugme za brisanje.

Tačke proširenja: Iz stabla se briše dati element.

Stanje: Izabrani element se briše iz stabla strukture podataka i taj element se više ne pojavljuje u grafičkom prikazu stabla.

## **Brisanje UML entiteta**

Učesnik: Korisnik

Preduslov: Korisnik je prethodno kreirao projekat i bar jedan dijagram u tom projektu.

Opis: Korisnik odabirom i pritiskom na dugme briše jedan UML entitet sa dijagrama.

Izuzetak: Korisnik može da obriše pogrešan UML entitet sa dijagrama.

Koraci: Korisnik klikne na entitet, zatim klikne na dugme za brisanje.

Tačke proširenja: Sa radne površine i iz stabla se briše entitet.

Stanje: Odabrani UML entitet je obrisao sa dijagrama i sve veze koje su bile povezane za taj element se takođe brišu.

## **Brisanje UML veze**

Učesnik: Korisnik

Preduslov: Korisnik treba da je već napravio UML vezu između dva UML entiteta.

Opis: Sa radne površine se uklanja odabrana veza.

Izuzetak: Korisnik može da obriše neku vezu koju nije nameravao, pri čemu će morati ponovo da je kreira.

Koraci: Korisnik klikne na vezu, zatim klikne na dugme za brisanje.

Tačke proširenja: Potrebno je brisanje uml veze sa radne površine.

Stanje: Radna površina gubi posmatranu vezu sa nje.

## **Vizuelizacija stabla strukture podataka**

Učesnik: Korisnik

Preduslov: Korisnik je kreirao bar jedan projekat u odabranom direktorijumu.

Opis: Stablo strukture podataka se prikazuje sa leve strane prozora aplikacije.

Izuzetak: Korisnik ne utiče na ovu funkcionalnost.

Koraci: Korisnik otvara aplikaciju.

Tačke proširenja: Nema daljih tačaka proširenja.

Stanje: Stablo strukture podataka je grafički prikazano sa leve strane.

## **Vizuelizacija UML entiteta**

Učesnik: Korisnik

Preduslov: Korisnik je kreirao UML entitet na dijagramu.

Opis: UML entitet se grafički predstavlja na radnoj površini.

Izuzetak: Korisnik ne utiče na ovu funkcionalnost.

Koraci: Korisnik kreira UML entitet.

Tačke proširenja: Nema tačaka proširenja.

Stanje: UML entitet je, nakon kreiranja, grafički prikazan na radnoj površini.

## **Vizuelizacija UML veze**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao UML vezu između dva UML elemenata.

Opis: Grafički se na radnoj površini prikazuje novonastala veza.

Izuzetak: Korisnik ne utiče na ovu funkcionalnost.

Koraci: Korisnik kreira UML vezu.

Tačke proširenja: Nema tačaka proširenja.

Stanje: Na radnoj površini se vizuelno prikazuje nova UML veza.

## **Generisanje Java koda**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao UML dijagram.

Opis: U novonastalom tekstualnom fajlu unutar paketa se pojavljuje Java kod baziran na kreiranom UML dijagramu.

Izuzetak: Korisnik nije smisljeno napravio UML dijagram.

Koraci: Korisnik klikne na dijagram, zatim klikne na dugme za generisanje java koda.

Tačke proširenja: Potrebno je da se novi fajl prikaže u stablu.

Stanje: U stablu strukture podataka se pojavljuje tekstualni fajl sa Java kodom u istom paketu gde se nalazi i izabrani UML dijagram.

## **Generisanje UML dijagrama na osnovu Java projekta**

Učesnik: Korisnik

Preduslov: Korisnik treba da je prethodno kreirao projekat i paket.

Opis: U odabranom paketu unutar projekta se pojavljuje UML dijagram baziran na kreiranom Java projektu.

Izuzetak: Java projekat nije napravljen kako treba.

Koraci: Korisnik klikne na paket i klikne na dugme za generisanje UML dijagrama na osnovu prethodno napravljenog Java projekta i izabere projekat sa diska.

Tačke proširenja: Potrebno je da se novonastali UML dijagram prikaže u izabranom paketu u stablu.

Stanje: U stablu strukture podataka se unutar izabranog paketa pojavljuje UML dijagram baziran na odabranom Java projektu.



## **Pristup informacijama**

Učesnik: Korisnik

Preduslov: Korisnik je pokrenuo ClassyCraft aplikaciju.

Opis: Korisnik koristeći meni pristupa informacijama vezanim za aplikaciju.

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

Koraci: Korisnik klikne dugme iz glavnog menija.

Tačke proširenja: Moguć je pristup dokumentaciji i pristup informacijama o kreatorima.

Stanje: Prikazuje se novi prozor sa informacijama vezanim za aplikaciju.

## **Pristup dokumentaciji**

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 bilo kakvu neočekivanu grešku pri ovoj funkcionalnosti.

Koraci: Korisnik klikne dugme iz glavnog menija.

Tačke proširenja: Nema tačaka proširenja.

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

## **Pristup informacijama o kreatorima**

Učesnik: Korisnik

Preduslov: Korisnik je pokrenuo ClassyCraft aplikaciju.

Opis: Korisnik koristeći meni pristupa informacijama vezanim za kreatore aplikacije.

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

Koraci: Korisnik klikne dugme iz glavnog menija.

Tačke proširenja: Nema tačaka proširenja.

Stanje: Prikazuje se novi prozor sa informacijama vezanim za kreatore.