Dijagram klasa (UML Class Diagram) prikazuje klase i veze između klasa unutar nekog sustava, drugim riječima dijagram klasa prikazuje statički prikaz sustava s obzirom na vremensku komponentu pomoću skupa objekata slične strukture, ponašanja u veza. Valja napomenuti da dijagram klasa ili razreda ne opisuje događaje, stanja, aktivnosti ili bilo kakvu vremensku promjenjivu karakteristiku sustava koji se modelira. Po definiciji: *Dijagram klasa* (engl. Class Diagram) predstavlja skup elemenata strukture kao što su klase, sučelja, a ponekad i same instance (objekti), njihovu unutrašnju strukturu te međusobnu povezanost određenim tipovima relacija.1 Tu se još može nadodati kako objekti predstavljaju entitet iz stvarnog svijeta ili apstrakciju nečega sa dobro definiranim granicama i smisao sa sustavom.2 Svaka klasa ima svoj naziv, atribut (ako je zadan i potreban) koji se ima svoj naziv i tip podataka i operacije koja djeluje nad klasom koje imaju ime, te ulazne i izlazne parametre. Klase unutar klas dijagrama se povezuju pomoću osnovnih veza generalizacije i asocijacije koji predstavljaju interakciju unutar sustava, te način veza i nasljeđivanja unutar klas dijagrama.

Unutar našeg sustava nalaze se sljedeće klase, *DBZalihaPive*, *InformacijePivi*, *EvidencijaZalihaPive*, *OcjenaKorisnika*, *BrojStola*, *Narudžba*, *PonoviRundu*, *Košarica*, *PovezivanjeBlagajne*, te četiri „Interfac-a“3  *Rulet*, *PrikazSvihArtikla*, *TraženjePoParametrima* i *PrikazJednogArtikla*. Klase unutar dijagrama su povezani vezama agregacije, asocijacije, kompozicije i korištenja:

*DBZalihaPive* predstavlja klasu koja se povezuje sa bazom podataka, te obrađuje i izvršava zadane uvjete (upite) koji manipuliraju sa bazom podataka.

*InformacijePivi* predstavlja klasu koja manipulira sa samim podacima koji će se unijeti proslijediti bazi podataka. Ova klasa ima zadatak dodavanja novih artikala u bazu, brisanja više nepostojećih artikala, provjeri dvojnog unosa artikla, te u slučaju postojanja njegovog brisanja i izmjeni nekih parametara već postojećeg artikla unutar baze.

1. UML dijagram klasa i ERA modeliranje, Programsko inženjerstvo, Fakultet organizacije i informatike, Varaždin; 2011/2012
2. UML-dijagrami: Zbirka primjera i riješenih zadataka, Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva 2013
3. Sučelje (engl. Interface) opisuje potpise i svojstva koje klase, koje ga implementiraju moraju sadržavati

*EvidencijaZalihaPive* predstavlja klasu koja ažurira bazu podataka svaki put kada se neka narudžba izda. Ova klasa je dosta važan za naš sustav iz razloga što u samim programskim sučeljima ne budu prikazani podaci onih artikli kojih nema na stanju (skladištu) . Razlog tome je ideja da kupac ima samo uvid u one artikla koji se sami nalaze u ugostiteljskom objektu.

*PrikazSvihArtikla* predstavlja stereotipiziranu klasu koja ima zadatak prikaza svih artikala koji se nalaze unutar baze podataka.

*TraženjePoParametrima* predstavlja stereotipiziranu klasu koja traži neki artikl ili grupu artikla zadanih po nekom od definiranih parametara.

*Rulet* predstavlja stereotipiziranu klasu koja na malo drugačiji način prikazuje artikle koji se nalaze unutar ugostiteljskog objekta. Uz pomoć ruleta korisnik igra „mini-igru“ koja nakraju pomoću slučajnog odabira korisniku izdaje artikl.

*PrikazJednogArtikla* predstavlja stereotipiziranu klasu koja prikazuje sve podatke jednog artikla nakon što se korisnik odlučio za neki artikl

*OcjenaKorisnika* predstavlja klasu koja će pamtiti sve artikle koji su bili prodani i po njima izraditi listu najtraženijih (najviše naručenih) artikla i najbolje ocjenjenih artikla od strane korisnika. Ova klasa će se povezivati sa narudžbom svaki put kada se narudžba ispuni i od nje dobivati sve podatke narudžbi, izračunati trenutno najviše tražene artikle i proslijediti ih na programsko sučelje.

*BrojStola* predstavlja klasu koja unutar sebe sadrži trenutnu poziciju (stol) vanjskog uređaja koji je spojen na sustav. Svaki put prije ego što se narudžba ispuni klasa *BrojStola* šalje ispis trenutnog uređaja koji se koristi te se zajedno sa narudžbom ispisuju kako bi konoba znao koji stol je naručio robu.

*Košarica*  predstavlja klasu koja pamti trenutnu još uvijek na potvrđenu narudžbu. Klasa sprema sve podatke važne za stvaranje narudžbe traženog artikla, te nudi opciju ponovnog vraćanja unatrag na neki prijašnji preglednik u slučaju da korisnik hoće napraviti još jednu narudžbu.

*Narudžba* predstavlja klasu koja povezuje sve podatke narudžbe i sprema ju za slanje narudžbe na samu izlaznu jedinicu (blagajnu) ugostiteljskog objekta. Prima podatke sadržane u košarici i broj stola na kojem se nalazi uređaj te šalje blagajni.

*PonoviRundu* predstavlja klasu koja pamti zadnju proslijeđenu narudžbu te nudi korisniku opciju ponavljanja zadnje narudžbe. Glavna ideja je pojednostavljenje korištenja ustava korisnicima koji „su si malo popili“.

*PovezivanjeBlagajne* predstavlja klasu koja se povezuje sa izlaznom jedinicom (blagajnom) i sa sustavom FINE kako bi dobila „Jednostavni identifikacijski račun (JIR)“ koji je zakonom obavezan za svaki izdani račun.

