SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Bobetić Valentina, 39918/11-R

Bradvica Hrvoje, 40630/11-R

Đuras Ivan, 39945/11-R

Kovaček Matija, 40008/11-R

Pantelić Antonia,

Projekt iz kolegija *Programsko inženjerstvo*

eNarudžba

Varaždin, 2014.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Projekt iz kolegija *Programsko inženjerstvo*

eNarudžba

Nositelj kolegija:

Prof. dr. sc. Vjeran Strahonja

Mentor:

Dr. sc. Zlatko Stapić

Varaždin, travanj 2014.

**Sadržaj**

Table of Contents

**No table of contents entries found.**

Uvod

U ovom projektu odlučili smo napraviti aplikaciju koja će uvelike pomoći kod ispunjavanja i zaprimanja narudžbi. Cilj nam je izraditi aplikaciju koja će omogućiti brzo naručivanje hrane u restoranima, zaprimanje narudžbe od strane djelatnika restorana, te da naručitelj može u svakom trenutku znati u kojoj je fazi njegova narudžba i kada je jelo pripremljeno.

Sam projekt podijelili smo u tri faze. U prvoj fazi definiramo projektni tim, odnosno koji su sudionici projekta i koja je njihova uloga u samoj izradi projekta. Svakome sudioniku dodijeliti ćemo zadatke koje će morati izvršiti i definirati ćemo nakon toga korake same provedbe projekta. Nakon što definiramo projektni tim, ići ćemo na specifikaciju i analizu zahtjeva korisnika, definirati koje su to kritične točke projekta na koje moramo posvetiti dodatnu pažnju i odabrati ćemo tehnologiju koju ćemo koristiti kako bismo na što bolje ispunili korisničke zahtjeve. Izraditi ćemo plan projekta, koliko će on trajati, kakav nam je budžet projekta i izraditi prvi dio projektne dokumentacije. Analizirati ćemo koje su to sve aktivnosti i koja je domena koju moramo zadovoljiti, te ćemo nakon toga to iz modelirati pomoću UML dijagrama.

U drugoj fazi nas očekuje samo modeliranje podataka i izrada ERA modela na temelju kojega ćemo iz generirati skriptu koja se kasnije koristi kod same izrade aplikacije. Ta skripta se naziva SQL skripta. U trećoj fazi i jednoj od najkompleksnijih i najzahtjevnijih faza, posvetiti ćemo se realizaciji samog programskog rješenja i izradi aplikacije koja će služiti za lakšu narudžbu jela iz restorana i lakše zaprimanje narudžbe. Aplikaciju ćemo na kraju testirati i vidjeti da li ima sve potrebne funkcionalnosti koje smo joj predefinirali i da li zadovoljava sve korisničke zahtjeve, te ako ne zadovoljava, dodatno ćemo ispraviti nastale greške.

Nakon toga, aplikaciju ćemo isporučiti naručitelju aplikacije, a same korisnike ćemo upoznati sa postojanjem takve aplikacije i educirati ih o njenom korištenju i primjeni u praksi.

Definiranje korisničkih zahtjeva

Namjena definiranja i specificiranja zahtjeva je izradi što je bolji softver koji će biti u skladu s određenim normama i smjernica koje se koriste kod implementacije i izgradnje softvera. Softver koji ćemo izradit je prvenstveno namijenjen za naručivanje hrane u nekom restoranu ili „fast food-u“, putem desktop aplikacije koja će omogućavati da se naručitelj prijavi u sustav preko prethodno obavljene registracije i tada može vidjeti ponudu jela/pića i naručiti određeno jelo uz način odabira plaćanja i da li želi dostavu. Naručitelj može komentirati i ocjenjivati hranu i djelatnike. S druge strane, djelatnik nekog restorana ili „fast food-a“ zaprima narudžbu kojoj dodijeliti različite statuse ili može ocjenjivati naručitelja pa se time stvara određeni bonitet i vidi se kakav je pojedini naručitelj. Ukoliko se dogodi da se određeni korisnik (naručitelj) pokaže kao lažna osoba, daje krive informacije vezane za lokaciju dostave, ne plati narudžbu itd., tada djelatnik restorana ima opciju da, ukoliko se pokaže da je probleme stvarao više puta, na osnovi dosadašnjih ocjena on korisniku može privremeno/permanentno zabraniti pristup aplikaciji. Administrator baze podataka će moći mijenjati ponudu hrane, dodavati ili brisati nešto u ponudi, te isto tako dodavati, brisati i uređivati postojeće djelatnike restorana, ali i same korisnike (naručitelje).

Sažetak uočenih i prethodno navedenih korisničkih zahtjeva je slijedeći:

* Registracija i prijava od strane naručitelja hrane
* Prijava od strane djelatnika u restoranu
* Pregled ponuda hrane, ispunjavanje narudžbe i odabir načina plaćanja
* Ocjena hrane i usluge od strane naručitelja
* Upravljanje korisnicima od strane djelatnika
* Ocjenjivanje naručitelja od strane djelatnika
* Zaprimanje narudžbe i dodavanje statusa narudžbi
* Mijenjanje ponude hrane

Cilj ovog softvera je da se na brži i lakši način obavi naručivanje hrane i da djelatnici mogu lakše zaprimiti narudžbu, te im se time olakšava posao i smanjuje se upotreba telefona kao sredstva naručivanja.

Da pojasnimo malo tko je to naručitelj. Naručitelj je osoba koja se prijavila u sustav, a prethodno se registrirala i koja je ispunila narudžbu i poslala je nekom restoranu ili „fast-foodu“. Djelatnik je osoba koja zaprima narudžbu i koji narudžbi dodjeljuje određene statuse koje ćemo kasnije opisati, te prosljeđuje narudžbu usmenim putem do kuhara.

Sam softver radi zasebno, nije dio nekog većeg sustava za koji bismo trebali raditi dodatne prekvalifikacije, no ako bude potrebno moguće ga je doraditi da se implementira u neki drugi sustav.

**Perspektive proizvoda i njegove funkcije**

Softver koji ćemo izraditi je desktop aplikacija koja će moći instalirati svi korisnici Windows operacijskih sustava. Softver će se moći preuzeti sa stranice restorana za koji će softver biti implementiran i na stranicama restorana koji će se odlučiti za ovaj softver. Nisu potrebne dodatne opcije i neka podešavanja za aplikaciju, dovoljno ju je instalirati na računalo, registrirati se i naručiti hranu.

Sama aplikacija je vrlo jednostavna za postavljanje na osobno računalo, isto tako i njezino korištenje. Nikakve posebne tehničke zahtjeve neće imati, sama uporaba aplikacije biti će što jednostavnije izvedena za korisnika, kao i za djelatnika restorana, tako da potreba za naknadnim educiranjem djelatnika neće biti potrebna. Isto tako ako se korisniku dogodi neka neočekivana greška, na samoj web stranici nalazit će se poseban kutak gdje će se moć obratiti djelatnicima restorana preko e-mail-a ili nazvati mobilnim uređajem, za daljnje upute od strane djelatnika.

Softver će se moći prilagoditi prema zahtjevima pojedinog korisnika ili prema određenom sustavu za koji će biti namijenjen. Naručitelj će poslati narudžbu, a djelatniku će se ona pojaviti kao „pop-up“ prozor, odnosno skočni prozor i prikazati će mu da ima novu narudžbu. Kod same registracije, naručitelj će obavezno morati unijeti slijedeće podatke: ime i prezime, korisničko ime i lozinku, točnu adresu, broj OIB-a, kontakt broj dok će kod prijave koristiti samo korisničko ime i lozinku. Lozinka će morati imati najmanje 8 znakova i kombinaciju slova i brojeva. Ako korisnik unese krivu lozinku ili korisničko ime, prikazati će mu se odgovarajuća poruka.

Ovisno o prijavi, da li je to naručitelj ili djelatnik, softver ima dva sučelja. Ako se radi o naručitelju, to je verifikacija prilikom same prijave u sustav, onda se otvara sučelje za naručitelja i obratno.

portovi

Aplikacije će se spajati na istu bazu podataka, koja će biti online baza podatka u kojoj će se nalaziti spremljeni upiti za pretragu ponude, odnosno jela.

**Karakteristike korisnika aplikacije**

Korisnici aplikacije su svi ljudi koji su zainteresirani za narudžbu u nekom trenutku. Softver nije specificirani za određeni segment ljudi, niti će se ikada bazirati. Ne zahtjeva veliku razinu informatičkog znanja ili znanja rada na računalu, no opet potrebno je znati neke od osnova na koji način instalirati i pokrenuti aplikaciju. Sama aplikacija je zamišljena da navodi korisnika, odnosno naručitelja na sve potrebne korake koje je potrebno izvršiti kako bi narudžba uspjela. Moguće su eventualne poteškoće kod registracije korisnika, pogotovo ljudi koji se nikada nisu s time susreli, no zbog toga ćemo implementirati i dodatnu pomoć onima koji negdje zapnu.

Aplikacija je više-manje korisnički orijentirana i korisnici se mogu lako njome služiti. Već unaprijed će biti predefinirane jedinice narudžbe, npr. komadi i slično, pa će unutar aplikacije jedino biti potrebno kliknuti na željeni odabir hrane, a isto tako će moći odabrati i način plaćanja, da li je to kartica, što onda olakšava rad prilikom dostave hrane, pa si dostavljač može unaprijed pripremiti uređaj koji mu je potreban ili će ipak izvršiti plaćanje gotovinom. Ako se dogodi da naručitelj se trenutno ne nalazi na adresi gdje inače stanuje, imati će opciju unosa trenutne adrese pa će tu biti dostavljena hrana. Nakon što je narudžba ispunjena, naručitelj će dobiti potvrdu narudžbe i moći će pratiti statuse narudžbe koje stavlja djelatnik. Statusi koje će biti implementirani u aplikaciju su sljedeći:

* NARUDŽBA JE ZAPRIMLJENA – Djelatnik u restoranu zaprima narudžbu koja mi se prikaže na ekranu i dodjeljuje joj status da je zaprimljena kako bi naručitelj mogao vidjeti da je poslao pravovaljanu narudžbu.
* NARUDŽBA JE NA REDU – Nakon što su sve narudžbe gotove, tada na red dolazi trenutna narudžba, te joj onda djelatnik dodjeljuje novi status kako bi naručitelj znao u kojoj je fazi narudžba.
* NARUDŽBA JE GOTOVA – Nakon što je narudžba ispunjena i jelo ispečeno, tada se daje naručitelju na znanje da može ili podići narudžbu ili da ona čeka na dostavu, ovisno što je naručitelj odabrao prilikom ispunjavanja narudžbe.
* DOSTAVA – Ako je naručitelj prilikom ispunjavanja narudžbe odabrao opciju da želi dostavu, tada djelatnik u restoranu stavlja ovaj status narudžbe onda kada je narudžba predana dostavljaču da je dostavi.
* DOSTAVLJENA – Dodjeljuje je djelatnik u restoranu, nakon što je narudžba dostavljena naručitelju.
* PODIGNUTO – Ako je naručitelj odabrao opciju da će sam podići narudžbu, tada nakon što je naručitelj došao i platio narudžbu, njoj se stavlja status od strane djelatnika restorana da je ona podignuta.

**Funkcije aplikacije i organizacija baze podataka**

Sustav će imati provjeru ispravnosti podataka prilikom prijave u sustav, na način da će provjeravati ispravnost unesenih podataka (korisničko ime i lozinka), da li su točni ili ne. Ukoliko nisu točni, pojaviti će se poruka. Ako aplikacija prestane s radom iz bilo kojeg razloga, vrijeme oporavka aplikacije će biti najbrže moguće i omogućiti će se korisnicima da rade dalje nesmetano. Ažuriranje aplikacije će se vršiti redovito i o tome obavještavati korisnike.

Baza podatak će biti online, to znači da je moguće pristupiti bazi podataka bilo kada. U bazi podataka će se nalazi ponuda svih vrsta jela pojedinog restorana, no za svaki restoran je moguće dopuniti bazu podataka. Uz svaku od ponuda hrane (jelovnika), biti će ispisani i sastojci koji se nalaze npr. sastojci pizze, veličina, cijena i mogućnost dodataka. Svaki od naručitelja priliko narudžbe, vidi koje su ponude trenutno aktualne i može onda to naručiti. Bazu podataka mijenja administrator na zahtjev samog korisnika aplikacije, odnosno vlasnika restorana.

Cijela aplikacija se vrti oko narudžbe hrane koja je izvršni, odnosno centralni događaj i kojemu se pridodaje velika pažnja.

Korisničke zahtjeve kao takve, smo mi sami prepoznali iz nekakvih vlastitih iskustava, pa kod definiranja istih nije bilo problema. Vrlo je bitno dobro definirati zahtjeve, da ne bi došlo do propusta prilikom njihova definiranja i prilikom njihove implementacije, te da se neke važne činjenice ne bi izostavile.

Plan projekta

Projektni plan izrađen je u alatu za upravljanje projektima naziva Microsoft Project. MS Project nam je omogućio jednostavnije planiranje projekta. U alatu smo definirali naš projekt tako da smo unijeli ime projekta, definirali vrijeme trajanja projekta (početni i završni datum), radno vrijeme i sl. Nakon početnih postavki unijeli smo sve planirane zadatke i njihovo trajanje, te potrebne resurse za izradu projekta. Zatim smo generirali grafički prikaz vremenskog plana (gantogram) iz kojeg je vidljiv slijed planiranih zadataka, te raspoloživost/opterećenost resursa. Na kraju smo generirali proračun troškova izrade projekta.

Za realizaciju softvera koristit ćemo fazni pristup razvoja softvera. Fazni pristup razvoja softvera karakteriziraju faze koje se izvode slijedno i to samo jednom u jednom razvojnom pothvatu ako je faza dobila pozitivnu ocjenu od strane korisnika. Kod faznog pristupa korisniku se isporučuju dijelovi, faze sustava koju čine određene funkcije definirane u projektu. Ovaj pristup omogućava korisniku da ima na raspolaganju faze softvera koje sadrže određene funkcionalnosti dok su ostale faze još u razvoju. Kompletna završna verzija softvera dobiva se spajanjem svih faza. Prednost ovog pristupa je ta što korisnik ima na raspolaganju dijelove softvera, prije konačne verzije softvera, odnosno završetka projekta, što omogućava rani uvid u određene funkcionalnosti softvera te na temelju toga korisnik može definirati nove zahtjeve. Što se tiče modela izabrali smo model objekata odnosno objektni pristup koji obuhvaća modele podatka i modele procesa. Modeli predstavljaju objekte te metode i poruke koje objekti razmjenjuju međusobno i s okolinom, odnosno prikazuju ponašanje sustava u radu.

1. **Projektni tim**

Projektni tim BrainStorm čini pet članova:

* Bobetić Valentina – Voditelj projekta
* Bradvica Hrvoje – Arhitekt
* Đuras Ivan – programer / tester
* Kovaček Matija – Programer / UI dizajner
* Pantelić Antonia – Analitičar

Svaki član ima određenu osnovnu ulogu u projektu. Uz osnovnu ulogu svaki član je upoznat s ostalim ulogama te djeluje i u tim područjima. Tako svaki član prolazi kroz većinu zadataka i u potpunosti je upoznat sa svim zadacima u projektu. U nastavku slijedi tablica s popisom članova, njihovim osnovnim ulogama i opisom aktivnosti same uloge.

|  |  |  |
| --- | --- | --- |
| **Član** | **Uloga** | **Aktivnosti uloge** |
| Bobetić Valentina | Voditelj projekta | Planira i vodi projekt, izrađuje projektnu dokumentaciju u MS Projectu, organizator, prati i koordinira rad tima, uspoređuje stvarno s planiranim, održava sastanke. |
| Pantelić Antonia | Analitičar | Analizira korisničke, poslovne zahtjeve za potrebom softverskog rješenja, primjenjuje UML za analizu zahtjeva, određuje metodologiju, tehnologiju rada u suradnji s razvojnim programerima, sudjeluje u razvoju i vođenju projekta. |
| Bradvica Hrvoje | Arhitekt | Definira arhitekturu informacijskog sustava, primjenjuje UML za dizajn sustava, kreira dijagrame slučajeva korištenja, dijagrame aktivnosti, podataka. Izrađuje dijagrame klasa te ERA model. |
| Kovaček Matija | Programer / UI dizajner | Izrađuje dizajn aplikacije/korisničkog sučelja, piše programski kod, sudjeluje u izradi korisničke dokumentacije. |
| Đuras Ivan | Programer / tester | Piše programski kod, testira aplikaciju, sudjeluje u izradi korisničke dokumentacije. |

Tablica 1. Uloga svakog člana projektnog tima

Kako bi mogli odrediti trajanje i troškove izrade projekta, prvo moramo definirati radno vrijeme članova tima i satnicu. Zbog studentski obaveza svi članovi tima rade od ponedjeljka do subote u radnom vremenu 13-19h. Cijena satnice za svakog člana tima iznosi 45kn/h.

U nastavku slijedi tablica s radnim vremenom, te satnicom svih članova tima.

|  |  |  |
| --- | --- | --- |
| **Radni dan** | **Radno vrijeme** | **Cijena rada po satu za sve članove** |
| Ponedjeljak | 13-19h | 45kn/h |
| Utorak | 13-19h |
| Srijeda | 13-19h |
| Četvrtak | 13-19h |
| Petak | 13-19h |
| Subota | 13-19h |

Tablica 2. Radno vrijeme tima

Za izradu projekta koristit ćemo vlastita, osobna računala. Osobna računala kao materijalni resurs ne ćemo uključivati u troškove projekta, u smislu troškova nabave, već kao trošenje/korištenje opreme koje je uračunato u cijenu rada svakog člana tima.

|  |  |
| --- | --- |
| **Materijalni resurs** | **Količina** |
| Stolno računalo | 2 |
| Laptop | 3 |

Tablica 3. Materijalni resursi korišteni kod izrade projekta

Za izradu projekta koristit ćemo sljedeću tehnologiju:

* MS Project – za izradu plana projekta i svih stavki koje obuhvaća plan projekta
* Visual Paradigm for UML – za izradu potrebnih UML dijagrama
* MS SQL Server 2008 R2 – za izradu baze podatka i ERA dijagrama
* Visual studio 2012 – za razvoj aplikacije pomoću programskog jezika C#
* Git – sustav za verzioniranje

1. Terminski plan projekta

-slike gantograma

1. Proračun i budžet projekta

-slika proračuna projekta iz MS Projecta / nisam siguran dal tu idu slike od gantograma ili se to odnosi pod podaci o provedbi projekata kaj ide iza UML dijagrama...

Ispod je ponuda naručitelju ..

1. Ponuda naručitelju

4. UML modeli

1. Dijagrami slučajeva korištenja (Use Case Diagram)

Dijagramom slučaja korištenja opisuje se funkcionalnost naše aplikacije eNarudžba te njezin odnos s okolinom odnosno korisnicima s gledišta vanjskog promatrača.

Dijagram se sastoji od učesnika, slučajeva korištenja i veza koje predstavljaju komunikaciju između učesnika i slučajeva korištenja te odnosa među samim slučajevima korištenja. U nastavku slijedi opis osnovnih koncepata .

U dijagramu slučajeva korištenja prikazani su sljedeći slučajevi korištenja:

* Prijava u aplikaciju
* Registracija korisnika
* Narudžba hrane
* Ponuda jelovnika
* Pregled (povijest) narudžbi
* Praćenje stanja narudžbe
* Ocjenjivanje hrane
* Ponovno naručivanje
* Zaprimljene narudžbe
* Promjena statusa narudžbe
* Upravljanje korisnikom

Slučaj korištenja *„Prijava u aplikaciju“* je prvi slučaj korištenja s kojim korisnik stupa u interakciju. Ishod te interakcije utječe na daljnje korištenje same aplikacije. *„Prijava u aplikaciju“* je slučaj korištenja gdje se korisnik prijavljuje u aplikaciju. U slučaju da korisnik nema korisničke podatke za prijavu može se registrirati preko proširenog slučaja korištenja *„Registracija korisnika“*.

Slučaj korištenja *„Registracija korisnika“* služi za registraciju korisnika kako bi se mogao kasnije prijavljivati u samu aplikaciju i koristiti njezine funkcionalnosti.

*„Ponuda jelovnika“* je slučaj korištenja gdje korisnik (naručitelj) može vidjeti ponudu jelovnika s opisom hrane, ocjenama hrane. Ako korisnik na temelju viđene ponude želi naručiti hranu, to mu je omogućeno jer je osnovni slučaj korištenja proširen proširenim slučajem korištenja *„Narudžba hrane“* .

*„Narudžba hrane“* je slučaj korištenja gdje korisnik (naručitelj) naručuje odabranu hranu u biranoj količini. Prilikom narudžbe korisnik navodi način plaćanja (kako bi se dostavljač mogao pripremiti), zatim odabire da li želi dostavu, te u slučaju da želi dostavu na drugu lokaciju (ako trenutno nije kod kuće), unosi dodatnu adresu dostave.

*„Pregled (povijest) narudžbi“* je slučaj korištenja gdje korisnik (naručitelj) može vidjeti svoju zadnju narudžbu te sve prošle narudžbe. Slučaj korištenja *„Pregled (povijest) narudžbi“* proširen je s ponašanjem proširenog slučaja korištenja *„Praćenje stanja narudžbe“* gdje korisnik po želji može pratiti što se događa s njegovom narudžbom. Zatim osnovni slučaj korištenja proširen je slučajem korištenja *„Ocjenjivanje hrane“* gdje korisnik može ocijeniti naručenu hranu. Zadnje proširenje osnovnog slučaja korištenja je prošireni slučaj korištenja *„Ponovno naručivanje“* gdje korisnik može ponovno naručiti određeno jelo koje je već ranije naručio (ako se to jelo još nudi).

*„Praćenje stanja narudžbe*“ je slučaj korištenja gdje korisnik (naručitelj) prati što se događa njegovom narudžbom. Kroz praćenje stanja narudžbe korisnik u svakom trenutku može znati što se događa s njegovom narudžbom (da li je na redu, dali se priprema ili dostavlja i sl.).

*„Ocjenjivanje hrane“* je slučaj korištenja gdje korisnik (naručitelj) ocjenjivanjem hrane iskazuje koliko je zadovoljan s određenim jelom, dali mu je bilo dobro i slično.

*„Zaprimljene narudžbe“* je slučaj korištenja s kojim korisnik (djelatnik u restoranu) stupa u interakciju. U tom slučaju korištenja korisnik vidi sve pristigle narudžbe. Za narudžbe koje su aktivne, spremne za daljnju obradu korisnik pomoću ponašanja proširenog slučaja korištenja

*„Promjena statusa narudžbe“* dodjeljuje narudžbi određene statuse koji označavaju u kojoj fazi se nalazi narudžba.

*„Upravljanje korisnikom“* je slučaj korištenja gdje korisnik (djelatnik u restoranu) ocjenjuje, komentira svoje naručioce hrane. Slučaj korištenja prvenstveno se koristi kada restoran im loša iskustva poput nepodignute narudžbe i slično s određenim naručiocima. Ako restoran ima više puta loše iskustvo s određenim naručiocem može mu jednostavno uskratiti svoju uslugu.

U dijagramu slučajeva korištenja prikazani su sljedeći učesnici:

* Korisnik - je generalizacija učesnika jer svi učesnici imaju zajedničko interakciju s slučajem korištenja 'Prijava u aplikaciju' te slučajem korištenja *„Registracija korisnika“*.
* Naručitelj - je primarni učesnik koji pokreće slučajeve korištenja. U interakciji sa sustavom pokreće sljedeće slučajeve korištenja
* Ponuda jela
* Pregled (povijest) narudžbi

te ostale slučajeve korištenja koji su uključeni u osnovni slučaj korištenja ili su proširenje prema osnovnom slučaju korištenja prethodno dva navedena slučaja korištenja.

* + - * Narudžba hrane
      * Praćenje stanja narudžbe
      * Ocjenjivanje hrane
      * Ponovno naručivanje
* Djelatnik u restoranu - je primarni učesnik koji u interakciji sa sustavom pokreće sljedeće slučajeve korištenja
  + - Upravljanje korisnikom
    - Zaprimljene narudžbe, koji je proširen proširenim slučajem korištenja „Promjena statusa narudžbe“

**Scenariji slučaja korištenja**

Postoje dva glavna scenarija za slučaj korištenja, odnosno jedan za svaku ulogu učesnika. Glavni scenarij za učesnika naručitelja je scenarij od pregleda ponude jelovnika, narudžbe hrane, praćenje narudžbe hrane pa sve do ocjene hrane.

Alternativni scenariji za učesnika naručitelja su scenariji:

* pregleda ponude jelovnika – gdje korisnik samo želi pregledati ponudu restorana
* pregled ponude jelovnika i narudžba hrane- gdje korisnik samo želi pregledati ponudu i naručiti
* pregled (povijest) narudžbi – gdje korisnik samo želi vidjeti što je sve u povijesti naručivao
* pregled (povijest) narudžbi i praćenje stanja narudžbe ili ocjenjivanje hrane ili ponovno naručivanje

Glavni scenarij za učesnika djelatnik u restoranu je scenarij pregleda zaprimljenih narudžbi i promjene statusa narudžbe.

SLIKA!!!!

1. Dijagrami slijeda (Sequence Diagram)

Izradom dijagrama slučajeva korištenja, dobili smo prikaz što bi naš sustav trebao raditi i dobili smo pregled prikaza sustava. Za svaki od scenarija slučajeva korištenja, napravili smo malo detaljniji prikaz onoga što se događa unutar pojedinog scenarija slučaja korištenja. Za opis pojedinih scenarija slučajeva korištenja upotrijebljen je glavni predložak kako bismo promatraču sustava na što jednostavniji način opisali što se događa u pojedinom scenariju slučaja korištenja. Na kratak, ali veoma dosljedan način prikazani su opisi pojedinih slučajeva korištenja.

Za svaki scenarij slučaja korištenja osim opisa, navodi se i ime autora, datum kada je opis generiran i napisan kao i vrijeme, koji su početni uvjeti za taj scenarij, a koji su završni uvjeti, odnosno ono što se poslije tog scenarija događa. Uz prethodno navedene detalje, još je su definirane i aktivnosti koje radi korisnik, u ovom slučaju naručitelj i što mu na temelju zahtjeva koji on šalje, odgovara sustav.

U dijagramu slučajeva korištenja prikazano je 11 slučajeva korištenja. Dijagram slijeda je prikazan samo za ?? dijagrama. Veza <<extend>> je opcionalna veza koja se može, ali se i ne mora izvesti prilikom izvođenja same aplikacije. Veza <<extend>> se odnosi na slijedeće slučajeve korištenja: *Narudžba hrane, Praćenje stanja narudžbi, Ponovno naručivanje, Ocjenjivanje hrane, Registracija korisnika i Upravljanje korisnikom.*

U nastavku su dani detaljni opisi slučajeva korištenja sa pripadajućim dijagramom slijeda, te njihovom specifikacijom.

* + 1. Ponuda jelovnika

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 20.04.2014. 23:13:01 |
| **Brief Description** | Korisnik aplikacije može vidjeti ponudu jelovnika koju restoran nudi, ponuda jelovnika sadrži osim same ponude hrane, opis hrane, koji sastojci ju čine, te ocjene i komentari korisnika koji su naručili određenu hranu. Korisnik na temelju ponude može naručiti hranu. |
| **Preconditions** | Korisnik treba biti prijavljen |
| **Post-conditions** | Pregledana ponuda hrane sa ili bez zahtjeva za narudžbom |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za pregled ponude jelovnika |  | | 2 |  | Prikaz forme s ponudom jelovnika | | 3 | Pregledava ponudu |  | | 4 | Odabire željenu hranu i šalje zahtjev za narudžbom hrane |  | | 5 | Zatvara formu s ponudom jelovnika |  | |

  Slika dijagrama slijeda!!!

* + 1. Pregled (povijest) narudžbi

|  |  |
| --- | --- |
| Author | Matija Kovaček |
| **Date** | 21.04.2014. 08:50:42 |
| **Brief Description** | Korisnik svaki puta kada realizira narudžbu, može ju ponovno vidjeti u pregledu (povijest) narudžbi. Korisnik može vidjeti što je sve naručio u određenoj narudžbi, zatim izborne mogućnosti su da može pratiti stanje aktivne narudžbe, ocijeniti naručenu hranu te ponoviti istu narudžbu (ponovno naručiti). |
| **Preconditions** | Korisnik treba biti prijavljen. |
| **Post-conditions** | Pregledana narudžba |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za pregledom (povijesti) narudžbi |  | | 2 |  | Prikaz forme s poviješću narudžbi | | 3 | Odabir narudžbu za pregled |  | | 4 |  | Prikaz informacija o narudžbi | | 5 | Šalje zahtjev za praćenje stanje aktivne narudžbe |  | | 6 | Šalje zahtjev za ocjenjivanje hrane |  | | 7 | Šalje zahtjev za ponovno naručivanje |  | | 8 | Zatvara formu s poviješću narudžbi |  | |

  Slika dijagrama slijeda!!!!

* + 1. Prijava u aplikaciju

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 20.04.2014. 20:09:23 |
| **Brief Description** | Korisnik unosi korisničke podatke za prijavu u aplikaciju. Prijava omogućuje daljnji rad s aplikacijom. |
| **Preconditions** | Korisnik treba biti registriran |
| **Post-conditions** | Korisnik koji je autoriziran ili odbijen |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za prijavom |  | | 2 |  | Prikaz forme za prijavom | | 3 | Unosi korisničke podatke |  | | 4 |  | Prikaz glavne forme | | 5 |  | Prikaz poruke o pogrešnoj prijavi | |

  Slika dijagrama slijeda!

* + 1. Pregled zaprimljenih narudžbi

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 11:45:42 |
| **Brief Description** | Korisnik (djelatnik u restoranu) pregledava dobivene narudžbe kako bi ih dalje proslijedio na obradu. Dodatna mogućnost je da djelatnik mijenja statuse narudžbe kako napreduje sam proces obrade narudžbe. |
| **Preconditions** | Korisnik (djelatnik u restoranu) mora biti prijavljen u aplikaciju. Da bi vidio zaprimljene narudžbe naručitelj prije mora naručiti hranu. |
| **Post-conditions** | Pregledana zaprimljena narudžba |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za pregledom zaprimljenih narudžbi |  | | 2 |  | Prikaz forme s zaprimljenim narudžbama | | 3 | Pregledava zaprimljene narudžbe |  | | 4 | Šalje zahtjev za promjenom statusa narudžbe |  | | 5 | Zatvara formu s pregledom zaprimljenih narudžbi |  | |

  Slika dijagrama slijeda!

* + 1. Promjena statusa narudžbe

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 12:06:16 |
| **Brief Description** | Korisnik zaprimljenoj narudžbi mijenja status, kako bi naručitelj znao u kojoj fazi se nalazi njegova narudžba. |
| **Preconditions** | Korisnik (djelatnik u restoranu) mora biti prijavljen u aplikaciju. Da bi promijenio status narudžbe, prije mora pregledati određenu narudžbu. |
| **Post-conditions** | Promijenjen status narudžbe. |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za promjenom statusa narudžbe |  | | 2 |  | Prikaz forme za promjenom statusa narudžbe | | 3 | Unosi/aktivira nove statuse |  | | 4 | Zatvara formu za promjenu statusa narudžbe |  | |

  Dijagram slijeda!

* + 1. Narudžba hrane

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 07:29:58 |
| **Brief Description** | Prije sam narudžbe, korisnik mora na temelju ponude jelovnika odabrati željenu hranu i količinu te hrane. Korisnik nakon što je odabrao željenu hranu, kreće u sam proces narudžbe gdje navodi potrebne podatke za ostvarenje narudžbe (da li želi dostavu, ovisno o prethodnoj stavci navodi način plaćanja, alternativnu adresu) te potvrđuje narudžbu. Korisnik može i odustati od narudžbe. |
| **Preconditions** | Korisnik mora biti prijavljen u aplikaciju te mora odabrati željenu hranu za narudžbu. |
| **Post-conditions** | Uspješno realizirana narudžba ili odustajanje od narudžbe |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za narudžbom |  | | 2 |  | Prikaz forme narudžbe | | 3 | Unosi podatke i potvrđuje narudžbu |  | | 4 |  | Prikaz poruke o uspješnoj narudžbi | | 5 |  | Prikaz glavne forme | | 6 |  | Prikaz poruke o neuspješnoj narudžbi | | 7 | Zatvara formu narudžbe |  | |

Dijagram slijeda!

* + 1. Praćenje stanja narudžbe

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 09:52:00 |
| **Brief Description** | Korisniku je omogućeno da za aktivnu narudžbu može pratiti njezino stanje. Stanja koja može vidjeti su: narudžba je zaprimljena, narudžba je na redu, narudžba je gotova, narudžba se dostavlja, narudžba je dostavljena, narudžba je podignuta. |
| **Preconditions** | Korisnik mora biti prijavljen u aplikaciju te mora imat aktivnu narudžbu. |
| **Post-conditions** | Vidljive promjene stanja narudžbe |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za praćenjem stanja narudžbe |  | | 2 |  | Prikaz forme za praćenje stanja narudžbe | | 3 |  | Prikaz informacija o stanju narudžbe | | 4 | Pregledava statuse narudžbe |  | | 5 | Zatvara formu za praćenje stanja narudžbe |  | |

  Dijagram slijeda!

* + 1. Ponovno naručivanje

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 10:27:01 |
| **Brief Description** | Korisnik može ponoviti istu narudžbu bez da prolazi pregled ponude jelovnika i odabira hrane. |
| **Preconditions** | Korisnik mora biti prijavljen u aplikaciju te mora imati barem jednu narudžbu u pregledu (povijesti) narudžbi. |
| **Post-conditions** | Ponovno naručena narudžba. |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za ponovnim naručivanjem |  | | 2 |  | Prikaz forme za ponovno naručivanje | | 3 | Unosi podatke i potvrđuje narudžbu |  | | 4 |  | Prikaz obavijesti o uspješnom ponovnom naručivanju | | 5 |  | Prikaz glavne forme | | 6 |  | Prikaz obavijesti o neuspješnm ponovnom naručivanju | | 7 | Zatvara formu za ponovno naručivanje |  | |

Dijagram slijeda

* + 1. Ocjenjivanje hrane

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 11:25:20 |
| **Brief Description** | Korisnik može ocjenjivati i komentirati naručenu hranu. |
| **Preconditions** | Korisnik mora biti prijavljen u aplikaciju te mora imati barem jednu narudžbu u pregledu (povijesti) narudžbi. |
| **Post-conditions** | Ocjenjena ili neocijenjena, komentirana ili ne komentirana hrana |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za ocjenjivanjem hrane |  | | 2 |  | Prikaz forme za ocjenjivanje hrane | | 3 | Ocjenjuje i komentira hranu |  | | 4 |  | Obavijest o pohranjenoj ocjeni, komentaru | | 5 | Zatvara formu za ocjenjivanje |  | |

  Dijagram slijeda!

* + 1. Registracija korisnika

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 20.04.2014. 21:44:27 |
| **Brief Description** | Da bi korisnik se mogao prijaviti u aplikaciju i kasnije koristiti njene mogućnosti, korisnik se ponajprije mora registrirati. |
| **Preconditions** | - |
| **Post-conditions** | Uspješno ili neuspješno registrirani korisnik |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za registracijom |  | | 2 |  | Prikaz forme za registraciju | | 3 | Unosi korisničke podatke |  | | 4 |  | Prikaz poruke o uspješnoj registraciji | | 5 |  | Prikaz forme za prijavu | | 6 |  | Prikaz poruke o neuspješnoj registraciji | |

  Dijagram slijeda

* + 1. Upravljanje korisnikom

|  |  |
| --- | --- |
| **Author** | Matija Kovaček |
| **Date** | 21.04.2014. 12:22:44 |
| **Brief Description** | Korisnik (djelatnik u restoranu) ocjenjuje, komentira svoje naručioce hrane u slučaju kada restoran im loša iskustva poput  nepodignute narudžbe i slično s određenim naručiocima. Ako restoran ima više puta loše iskustvo s određenim naručiocem može  mu  jednostavno uskratiti svoju uslugu. |
| **Preconditions** | Korisnik (djelatnik u restoranu) mora biti prijavljen u aplikaciju, te može samo to primijeniti na korisniku koji je naručio hranu. |
| **Post-conditions** | Zabilježba komentara/ocjene o naručitelju. |
| **Flow of Events** | |  |  |  | | --- | --- | --- | |  | **Actor Input** | **System Response** | | 1 | Šalje zahtjev za prikazom forme o upravljanju korisnikom |  | | 2 |  | Prikaz forme o upravljanju korisnikom | | 3 | Unosi komentare/ocjene naručitelju |  | | 4 |  | Prikaz obavijesti o pohranjenom komentaru, ocjeni. | | 5 | Zatvara formu za upravljanje korisnikom |  | |

**Dijagram slijeda!**

1. Dijagrami aktivnosti (Activity Diagram)

Za svaki od prethodno navedenih dijagrama slijeda napravili smo odgovarajući dijagram aktivnosti koji je još na nižoj razini od dijagrama slijeda. Dijagram aktivnosti prikazuje dvije dimenzije: vertikalna koja označava vremenski period poruka koje se šalju i primaju između instanci, dok horizontalna prikazuje koje sve instance objekta sudjeluju u interakciji, odnosno koji su to učesnici interakcije.

Prilikom same izrade dijagrama aktivnosti, definirali smo koji su to sve entiteti koji sudjeluju u interakciji i u pojedinom scenariju slučajeva korištenja. Imamo entitete djelatnik u restoranu, naručitelj, sustav, te aplikaciju koja kontrolira rad svih entiteta. Dijagram slijeda sastoji se od akcija koje su elementarne aktivnosti i ne mogu se rastaviti na manje dijelove. Na svakom je dijagramu također označeno područje terminiranja gdje naručitelj svaki puta može zaustaviti aplikaciju i njeno izvođenje.