**Projektna dokumentacija – Refrigerator**

**Studenti: Nikola Grbac, Josip Ćosić**

**Smjer: Poslovna informatika**

**Kolegij: Programsko inženjerstvo**

**Mentori: doc. dr. sc. Tihomir Orehovački, Nikola Tanković, dipl. ing.**

**1. Sažetak**

Dokumentacija ukratko sadrži opis cijele aplikacije kao i podjelu u dvije kategorije: **Sistemska dokumentacija** koja je namijenjena softverskim inženjerima ili članovima razvojnog tima, gdje se opisuju zahtjevi, građa i funkcioniranje aplikacije te time se omogućuje sam razvoj aplikacije i njeno kasnije održavanje. **Korisnička dokumentacija** jednostavno je namijenjena korisnicima te im omogućuje da koriste aplikaciju te im istovremeno opisuje funkcionalnosti aplikacije na njima razumljiv način. Dokumentacija također ukratko sadrži uvod, gdje se daje opis aplikacije koja se razrađuje kao i ciljano tržište, korisnike i glavne prednosti koje su moguće i ostvarive pri uvođenju aplikacije.

Motivacija sadrži unutar sebe detaljno razrađeno ciljano tržište gdje bi se aplikacija prvenstveno plasirala te implementirala. Opis postojećih ali i konkurentskih rješenja (druge aplikacije iste namjene i funkcionalnosti) te uz nju i priložena **SWOT** analiza u kojoj se opisuju snage (strenght), odnosno atributi kojima aplikacija se ističe, slabosti (weaknesses) ili mane unutar aplikacije i njenih funkcionalnosti, prilike (oportunities) u kojima aplikacija parira konkurentima ili čak ima bolja svojstva te prijetnje (threats) odnosno oni čimbenici zbog kojih aplikacija ima svojstva da ne uspije na tržištu ili ne može parirati konkurentskim aplikacijama istoga tipa ili iste namjene.

Razrada funkcionalnosti odnosi se na zasebno opisivanje i grupiranje funkcionalnosti po grupama korisnika (administrator, menadžer, korisnik...). Gdje se prilaže "Use Case" dijagram cijelog sustava (aplikacije). Razrada ključnih korisničkih interakcija s aplikacijom kroz "Use Case Sequence" dijagram. Prilaganje prototipova sučelja za aplikaciju.

Implementacija pomoću dokumentiranja klasnih dijagrama na kojima su vidljivi načini rješavanja ključnih funkcionalnosti unutar aplikacije gdje se vidi koje su GTK klase koriste u View-u, te na koji način su povezane s klasama modela (Presenter, Model, View). Priložene sve klase i njene sastavnice uz objašnjenje značaja tih sastavnica.

Korisničke upute pomoću kojih se pojašnjava način korištenja aplikacije uz pomoć screeshotova sučelja aplikacije.

**2. Uvod**

Aplikacija koja se razrađuje vodi se pod imenom **Refrigerator**. Glavno (ciljano tržište) aplikacije su ugostiteljski objekti (restorani, kafići, fast food-ovi...). Zbog velikog obujma posla koji se nalazi u toj industriji (turizam), mnogi ugostiteljski objekti (gore navedeni) tijekom ljetnih mjeseci imaju puno posla oko praćenja namirnica potrebnih za rad objekta i njenog praćenja od ulaska u objekt (u našem slučaju npr. frižider) pa do izlaska (potrošnje) odnosno potrošnja namirnice za određeno jelo. Glavni korisnici u našem slučaju bilo bi osoblje, što ovisi u kojem ugostiteljskom objektu se aplikacija implementira i koristi u velikom obujmu.

Korisnici (zaposlenici) u kafiću bili bi konobari koji bi pomoću aplikacije pratili trenutno stanje zaliha određenog pića (kava, alkoholna pića, topli napitci...) te primanjem narudžbe od strane dobavljača unosili dobivene namirnice u aplikaciju te time imali točnu količinu određene namirnice što bi u krajnjem slučaju olakšavalo narudžbe, primanje tih narudžbi te smanjivali bi se troškovi jer ne bi se dolazilo do situacije da se određena namirnica naruči više nego što je potrebno u tom trenutku što ima za rezultat bespotrebno trošenje financijskih sredstava. Također korisnici bi bili i kuhari u restoranima koji su u većini slučajeva zaduženi za narudžbu namirnica potrebnih za spravljanje raznih jela. Imamo isti slučaj uporabe gdje kuhar provjera stanje frižidera putem aplikacije te ima na uvid što mu nedostaje i u kojoj količini, što smanjuje birokraciju (upisivanje što nedostaje na komadić papira te eventualni gubitak njega).

Glavne prednosti koje bi se ostvarile uvođenjem aplikacije prvenstveno jesu lakše praćenje zaliha, trenutnog stanja frižidera što uvelike olakšava narudžbu prema dobavljaču te i primanje namirnica te unos dobivenih namirnica unutar aplikacije. Mogućnost dodavanja recepata što uvelike koristi restoranima, gdje je moguće unijeti određeni recept te potrebne namirnice za spravljanje tog jela. Također postoji mogućnost izmjene recepta, odnosno mijenjanje nekih namirnica unutar određenog jela kao i pregled predloženih recepata te praćenje roka isteka svake namirnice. Sve te funkcionalnosti na kraju doprinose fluidnijem radu objekta u kojem se aplikacija koristi, smanjenje troškova (nema bespotrebnog naručivanja viška namirnica), smanjenje birokracije, odnosno uporaba papira i drugog potrošnoga materijala.

**3. Motivacija**

Ciljano tržište aplikacije kao što je i prije spomenuto su ugostiteljski objekti (restorani, kafići, fast food-ovi...). Objekti koji imaju karakteristiku da primaju puno narudžbi, odnosno namirnica te je potrebno voditi računa o njihovom stanju (količini) čime se olakšava cijeli posao naručivanja zbog mogućnosti provjere točne količine određenih namirnica.

Što se tiče postojećih rješenja, koja možemo ujedno i zvati i konkurentska rješenja tu imamo pametne frižidere (smart fridges ili internet fridges) kao zasebnu cjelinu bijele tehnike. Na tržištu postoje već duže od jednoga desetljeća, gdje je LG 2000. godine plasirao prvi pametni frižider s tadašnjom cijenom od $20,000 no prodaja mu nije zaživjela jer su tada korisnici smatrali kao beskorisno i jako skupo rješenje. Također na ruku im nije išlo (pametnim frižiderima) tada su imali sigurnosne propuste (virusi, malware...). 2000. godine iste godine kad je LG plasirao svoj pametni frižider ruska kompanija Kaspersky Lab poznata po svom anti-virusnom programu Kaspersky upozorila je sve proizvođače da pametni frižideri i sva ostala bijela tehnika povezana izravno na Internet ima mogućnost biti ugrožena. No u novije doba to se sve popravilo te pametni frižideri postali glavni krik tehnologije u kućanskim aparatima. Tako da današnji pametni frižideri imaju raznovrsne mogućnosti kao što su sprječavanja miješanja mirisa, čuvanje svježine svih namirnica, omogućavanje raspodjele namirnica po vrsti, ali i prema potrebama starijih i mlađih članova obitelji, za lakše odlaganje i uzimanje namirnica. Novi frižideri renomiranih proizvođača (Samsung, Gorenje, LG...) imaju tzv. Twin Cooling Plus sistem što u biti znači da sustav koristi dva zasebna protoka zraka i preciznu kontrolu temperature. Također pametni frižideri imaju i mogućnost biti pokretani na različitim operativnim sustavima kao što su Android ili IOS.

Gledajući na sve navedene prednosti (karakteristike) pametnih frižidera naspram desktop aplikacije vidljivo je da konkurencija postoji te da je se teško s njom nositi na svim poljima, odnosno nositi se s onime što oni nude. Jedina prednost desktop aplikacije bila bi eventualno njena cijena. U sljedećoj SWOT analizi bit će navedene prednosti uvođenja aplikacije u rad.

**SWOT analiza**

**Strenghts (snage)**

1. Lakoća korištenja aplikacijom
2. Implementacija
3. Distribucija
4. Cijena(?)

**Weaknesses (slabosti)**

1. Nema podrške za croos platforme
2. Postojanje samo desktop aplikacije
3. Mali broj funkcionalnosti u odnosu na konkurentske proizvode
4. Slab rezultat (promoviranja) marketinga

**Oportunities (prilike)**

1. Mogućnost promoviranja preko društvenih mreža (Facebook, Instagram...)
2. Privlačenje novih klijenata (manji ugostiteljski objekti)

**Threats (prijetnje)**

1. Negativna reakcija korisnika (negativan feedback)
2. Inovativnija konkurentska rješenja
3. Malen broj implementiranja

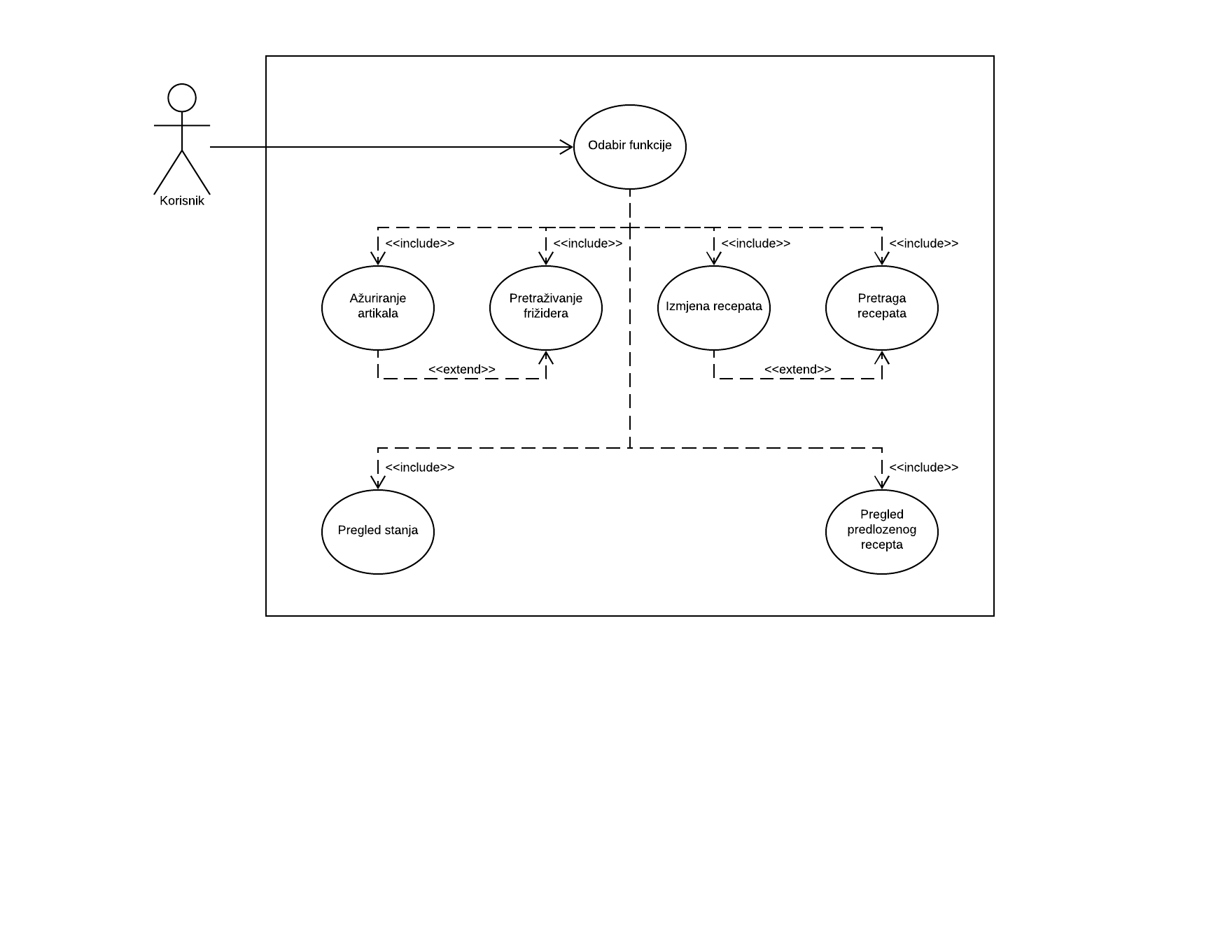
Glavne predispozicije, ono što je potrebno za imati funkcionalnu aplikaciju i njeno korištenje je desktop računalo ili laptop po mogućnosti s operativnim sustavom Windows (7, 8, 8.1 ili 10). Potrebna je baza podataka (Sqlite), zbog unosa podataka (namirnica). Dodaci kao što su baza sqlite, paketi system.sqlite i dapper. Dapper je objektno-relacijski paket koji pruža okvir za mapiranje modela domene usmjerenih na objekt u relacijsku bazu podataka te je prijeko potreban za pravilno funkcioniranje aplikacije.

Prednosti uvođenja aplikacije nisu vidljive samo na korisnicima (zaposlenicima) iako oni najvećim dijelom koriste aplikaciju u svom svakodnevnom radu na poslu. Prednosti, iako ih je moguće vidjeti u dugom roku vidljive su na ugostiteljskim objektima koji baziraju svoj rad pomoću aplikacije. Jedan od prednosti korištenja aplikacije a koja se ne tiče korisnika bila bi ušteda vremena i financijskih sredstava. Nema bespotrebnog zapisivanja čega i koliko određene namirnice fali. Dovoljno je pogledati stanje unutar aplikacije čime se dobiva izravan uvid u namirnice i njihov broj. Druga prednost je vidljiva na troškovima, koristeći aplikaciju objekti imaju mogućnost uštedjeti određenu novčanu svotu u dugom roku (npr. godinu dana) gdje se ne naručuje namirnica koje ima dovoljno za određeni dan, gdje ne postoji situacija da namirnicu propadaju gdje u ovom slučaju aplikacija ima funkcionalnost upisivanja dana kada je namirnica unesena te se može pratiti rok isteka.

Organizacije ili poduzeća koja komuniciraju izravno s objektom koji koristi aplikaciju nemaju ikakvih izravnih poboljšanja na svoje poslovanje. Objekt naručuje određenu količinu namirnica kod dobavljača, pomoću aplikacije tako da samo ima mogućnost provjere čega i koliko joj treba. Nema direktne povezanosti sa sustavom određenog dobavljača koji bi imao uvid u naše stanje zaliha i time sam pravio narudžbu i vršio dostavu nje.

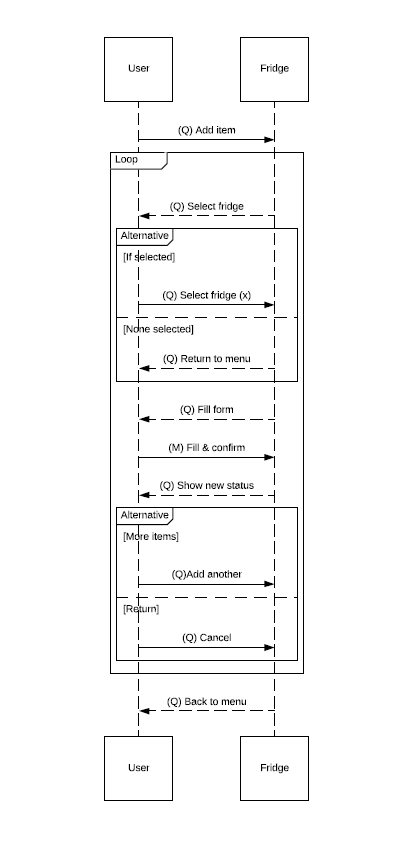
**4. Razrada funkcionalnosti**

Jedini korisnik u našem slučaju bio bi zaposlenik (osoblje) određenog ugostiteljskog objekta. Konobar, kuhar ili pomoćni kuhar. Dolje navedeni glavni "Use Case" cijeloga dijagrama opisuje funkcionalnost.



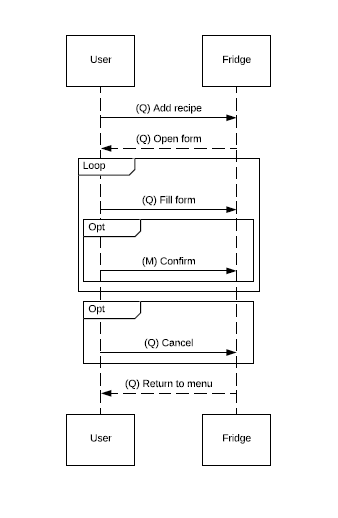
Korisnik (konobar, kuhar, pomoćni kuhar...) dolaskom na početno sučelje ima za odabrati neku od funkcija. Jedna od opcija je "Ažuriranje artikla" što znači jednostavno da se artiklu može mijenjati ime i količina te možemo dodati artikl i brisati ga. Druga mogućnost je dodavanje nove namirnice u frižider, gdje upisujemo ime i količinu. Sljedeće funkcije tiču se recepata gdje imamo isti princip. "Izmjena recepta" odnosi se na mijenjanje određenih namirnica, neke namirnice možemo brisati a neke dodavati. "Pretraga recepata" ima opciju pretrage baze u kojoj su recepti stvoreni. "Pregled stanja" odnosi se na provjeru trenutnoga stanja zaliha namirnica u frižideru. "Pregled predloženog recepta" ima mogućnost izbacivanja random recepta s time da ne postoji mogućnost izbacivanja istog rezultata dvaput.

"Use Case Sequence" dijagram za dodavanje artikla u frižider.

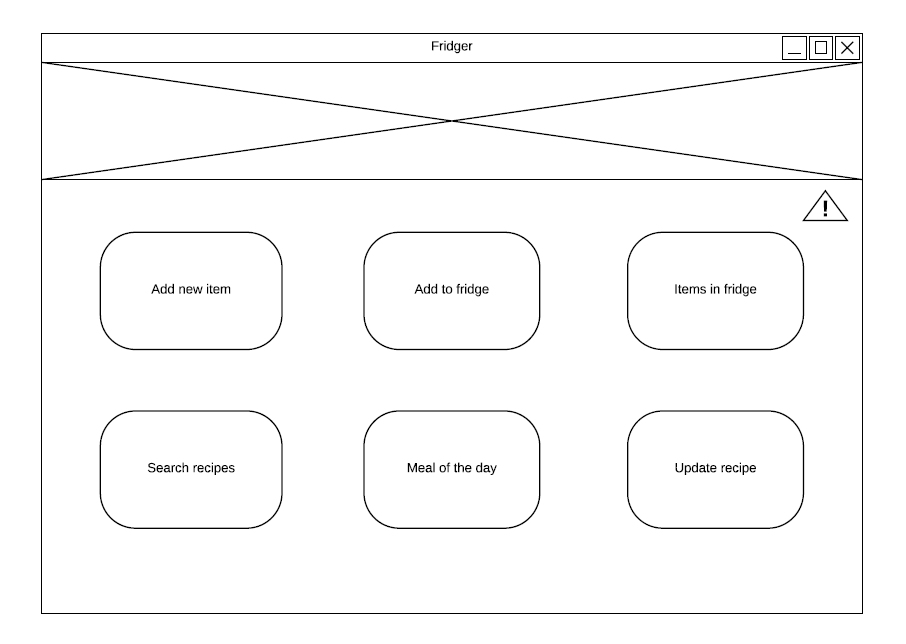


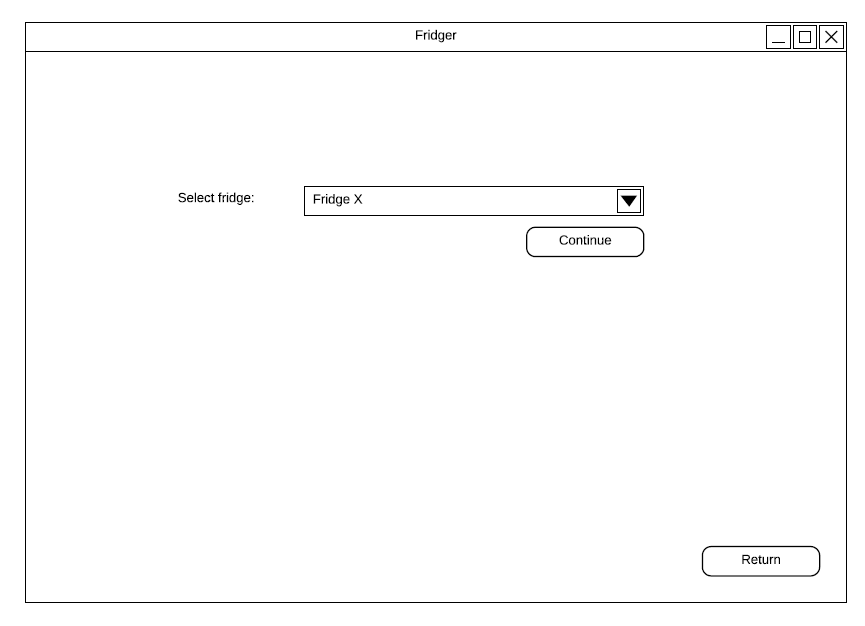
Gore navedeni "Use Case Sequence" dijagram opisuje funkcionalnost dodavanja artikla u frižider. Prvo imamo "Query event" gdje se zahtjeva prikaz informacija, odnosno zahtjeva se dodavanje artikla u frižider, imamo nasljeđivanje. U ovom slučaju imamo loop koji se izvršava dok se određeni frižider ne odabere, nasljeđivanje također imamo, uvjetno izvođenje, ako se frižider odabere nastavlja se dalje, ako ne loop se izvršava dok se odabere opcija frižidera. Ako se odabere frižider korisniku se vraća generalizacijom ispunjenje forme ("Query event"), korisnik ispunjava formu te ga vraća u frižider nasljeđivanjem, gdje imamo pojavu mutacije, odnosno promjena podataka u sustavu (stanje frižidera se mijenja). Korisniku se vraća rezultat, trenutno stanje, generalizacijom. Sljedeće opcije, uvjetnog izvođenja su, unošenje još artikala ili povratak na izbornik na stranu korisnika.

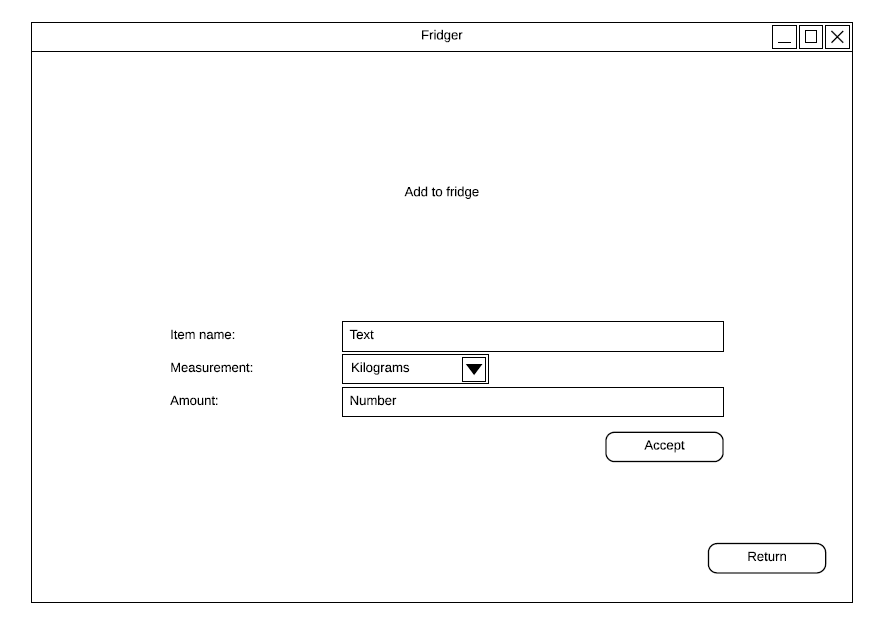
"Use Case Sequence" dijagram za dodavanje recepata u frižider.

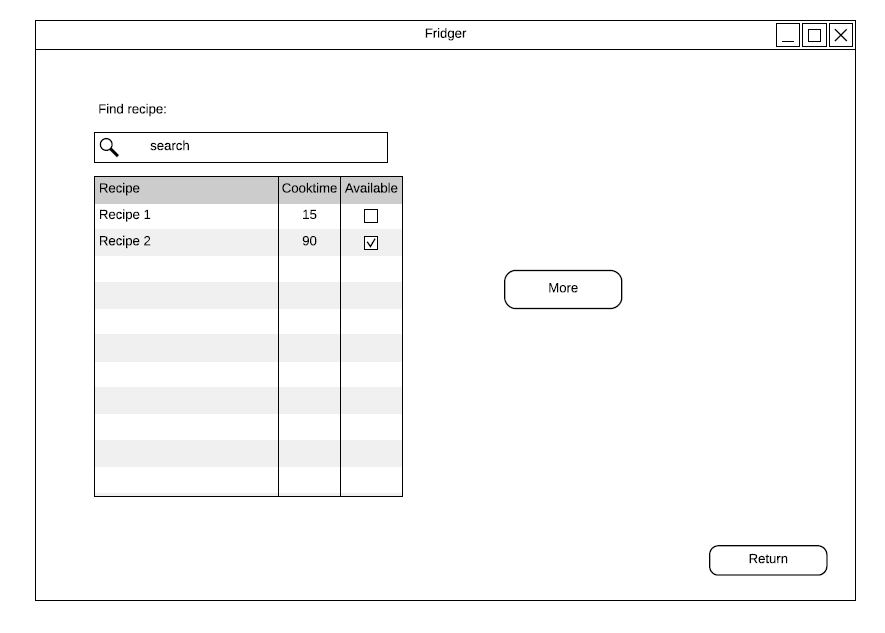


Gore navedeni "Use Case Sequence" dijagram opisuje funkcionalnost dodavanja recepta u frižider. Prvo imamo "Query event" gdje se zahtjeva prikaz informacija, odnosno zahtjeva se dodavanje recepta u frižider, imamo nasljeđivanje. Frižider vraća odgovor na upit generalizacijom na korisnika. U ovom slučaju imamo loop koji se izvršava dok se upit ne ispuni, nasljeđivanje imamo, uvjetno izvođenje, ako se upit ispuni nastavlja se dalje, ako ne loop se izvršava dok se ne upit ne ispuni. Imamo opciju opcionalnog izvođenja ("Query event") gdje se odbija opcija ispunjavanja upita te se generalizacijom vraća na korisnika.

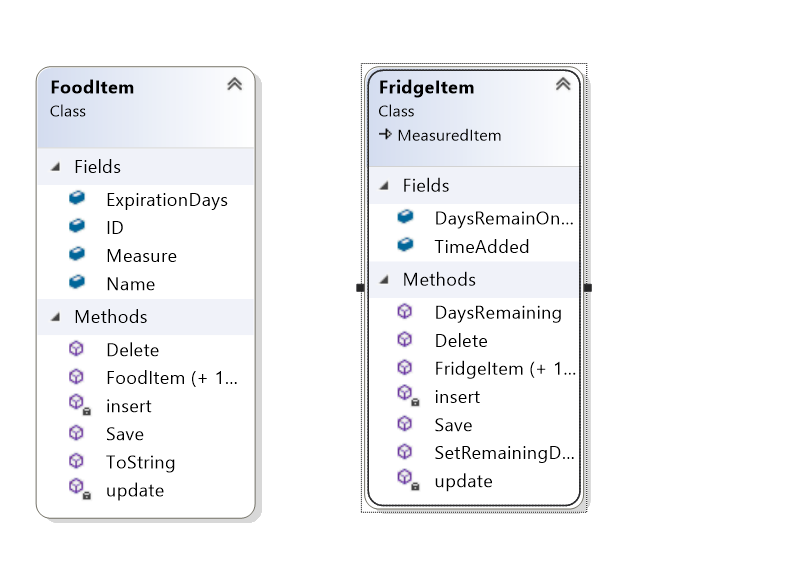






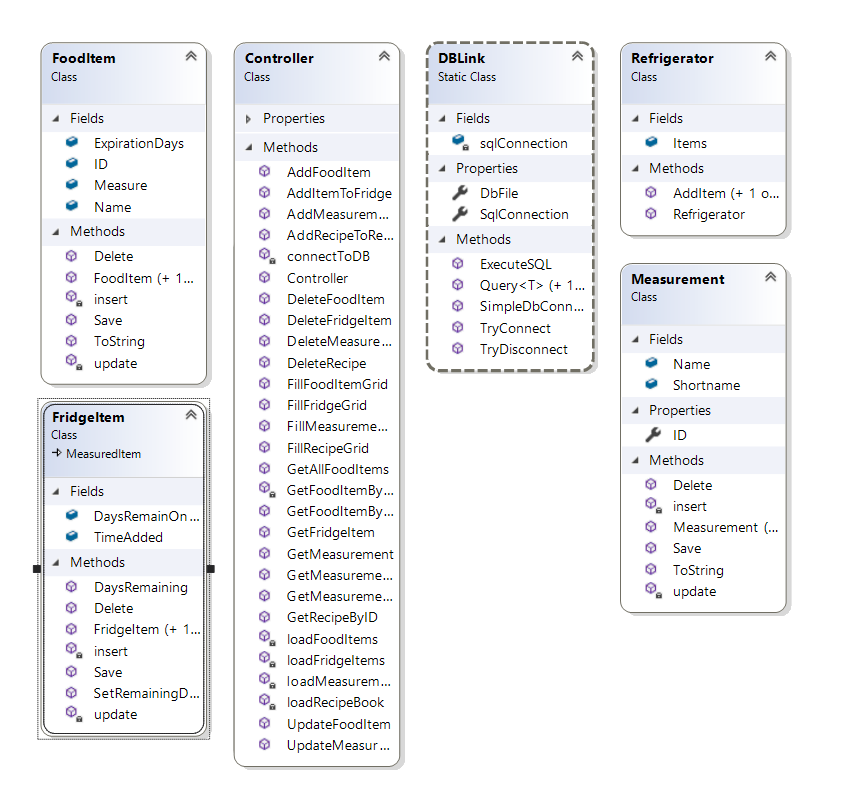


Sljedeća slika prikazuje klasni dijagram, za artikl u našem slučaju, Food Item i Fridge Item.



**5. Implementacija**

Jedna od ključnih funkcionalnosti je dodavanje namirnica u frižider, opisana klasnim dijagramom.



Korištene klase su FoodItem, Controller, Refrigerator, FridgeItem, Measurment te DBLink. Klasa FoodItem sastoji se od metoda Delete sa opcijom brisanja namirnice iz baze. FoodItem (+1) je kontruktor. Insert dodavanje u bazu. Save za spremanje u bazu te update za ažuriranje namirnice. Sastoji se od polja koji su vidljivi na sučelju, ID, Measure, Name i ExpiratonDays. Klasa FridgeItem sastoji se od metoda DaysRemaining, preostali dani namirnice prije njenog isteka roka.

Delete za brisanje iz baze, FridgeItem je konstruktor, insert je za unos, Save za spremanje, SetRemaining za postavljanje broja dana i update za ažuriranje. Controller klasa sastoji se od svojstava fridge, recipeBook, foodItemStorage i measurmentStorage. Glavne metode su AddFoodItem, dodavanje namirnica, AddItemToFridge, dodavanje namirnice u frižider, AddMeasurement, dodavanje mjera, AddRecipeToBook, dodavanje recepata u knjigu recepata, ConnectToDB spajanje na bazu podataka, DeleteFoodItem, brisanje namirnice iz baze, DeleteFridgeItem brisanje namirnice iz frižidera, UpdateFoodItem, ažuriranje namirnice, promjena imena i količine i UpdateMeasurment, ažuriranje mjera.

Klasa Measurement sastoji se od polja prikazanih na sučelju Name i ShortName (kratica). Metode su Delete, brisanje mjera iz baze, insert za unos mjera, Save za spremanje mjera, update za ažuriranje mjera, promjena imena i kratice. Klasa Refrigerator sastoji se od polja Items, dodavanje. Metode su AddItem i Refrigerator. Klasa DBLink sastoji se od polja sqlConnection, metoda ExecuteSQL za izvršavanje sql naredbi, Query za ispis naredbi, TryConncet proba spajanja i TryDisconnect za odspajanje.

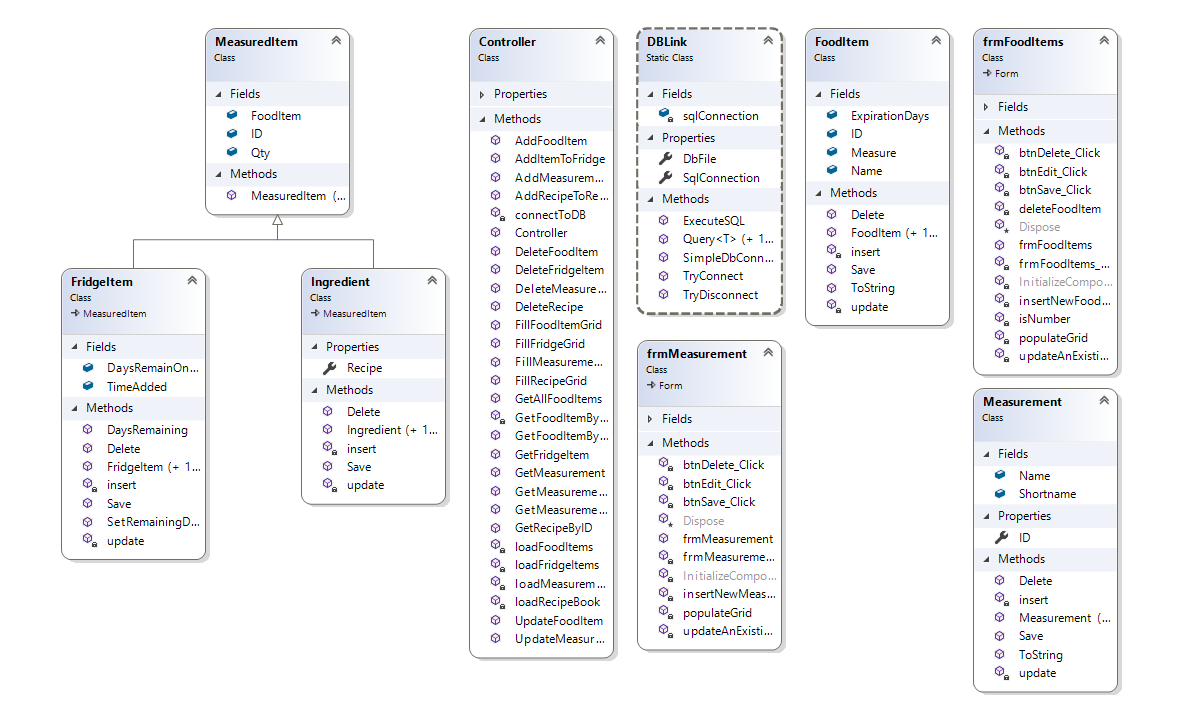
**Funkcionalnost dodavanja mjera**

Sastoji se od klasa Measurement, frmMeasurement, Controller i DBLink.



Klasa Measurement sastoji se od polja vidljivih na sučelju, Name i Shortname. Metode unutar klase su Delete za brisanje mjere, insert za unos mjere, Save za spreamanje mjera u bazu, update za ažuriranje. Controller klasa sastoji se od svojstava fridge, recipeBook, foodItemStorage i measurmentStorage. Glavne metode su AddFoodItem, dodavanje namirnica, AddItemToFridge, dodavanje namirnice u frižider, AddMeasurement, dodavanje mjera, AddRecipeToBook, dodavanje recepata u knjigu recepata, ConnectToDB spajanje na bazu podataka, DeleteFoodItem, brisanje namirnice iz baze, DeleteFridgeItem brisanje namirnice iz frižidera, UpdateFoodItem, ažuriranje namirnice, promjena imena i količine i UpdateMeasurment, ažuriranje mjera. Klasa frmMeasurement sastoji se od metoda btnDelete\_Click za spajanje sa dugmom za brisanje, btnSave\_Click za spremanje, btnEdit\_Click za uređivanje mjera. InsertNewMeasurement dodavanje nove mjere i updateAnExisting, ažuriranje postojeće mjere. Klasa DBLink sastoji se od polja sqlConnection, metoda ExecuteSQL za izvršavanje sql naredbi, Query za ispis naredbi, TryConncet proba spajanja i TryDisconnect za odspajanje.

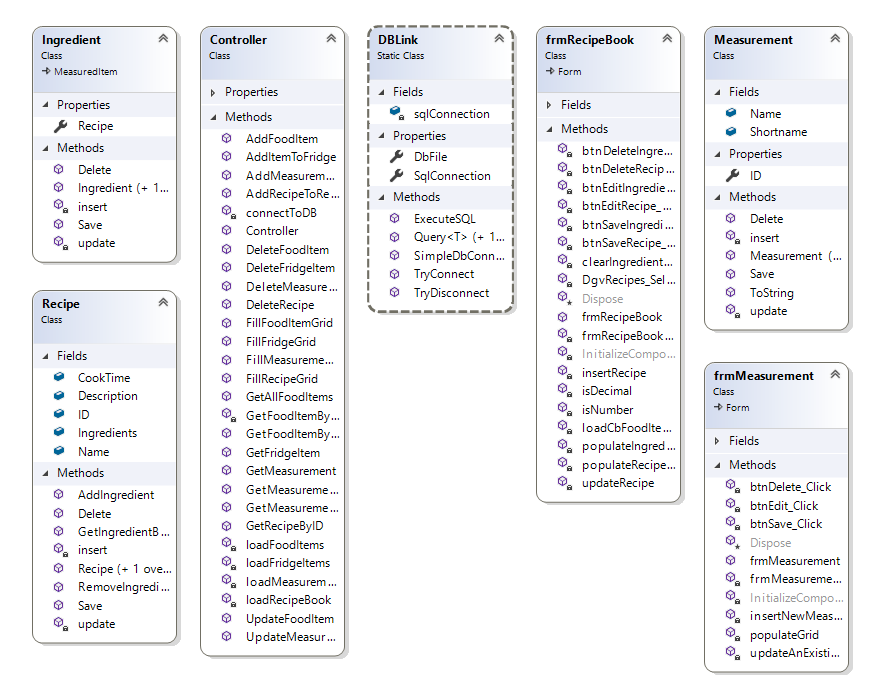
**Funkcionalnost dodavanja namirnica u frižider**



Sastoji se od klasa MeasuredItem, FridgeItem, Ingridient, Controller, DBLink, frmMeasurement, FoodItem, frmFoodItem te Measurement. Klasa frmMeasurement sastoji se od metoda btnDelete\_Click za spajanje sa dugmom za brisanje, btnSave\_Click za spremanje, btnEdit\_Click za uređivanje mjera. InsertNewMeasurement dodavanje nove mjere i updateAnExisting, ažuriranje postojeće mjere. Klasa DBLink sastoji se od polja sqlConnection, metoda ExecuteSQL za izvršavanje sql naredbi, Query za ispis naredbi, TryConncet proba spajanja i TryDisconnect za odspajanje. Klasa Measurement sastoji se od polja vidljivih na sučelju, Name i Shortname. Metode unutar klase su Delete za brisanje mjere, insert za unos mjere, Save za spreamanje mjera u bazu, update za ažuriranje.

Controller klasa sastoji se od svojstava fridge, recipeBook, foodItemStorage i measurmentStorage.Glavne metode su AddFoodItem, dodavanje namirnica, AddItemToFridge, dodavanje namirnice u frižider, AddMeasurement, dodavanje mjera, AddRecipeToBook, dodavanje recepata u knjigu recepata, ConnectToDB spajanje na bazu podataka, DeleteFoodItem, brisanje namirnice iz baze, DeleteFridgeItem brisanje namirnice iz frižidera,UpdateFoodItem, ažuriranje namirnice, promjena imena i količine i UpdateMeasurment, ažuriranje mjera. Klasa Ingredient sastoji se od toga da pripada klasi Recipe, metode su Delete za brisanje, insert za unos sastojka, Save za spremanje sastojka te update za ažuriranje.

**Funkcionalnost dodavanja recepata**

****

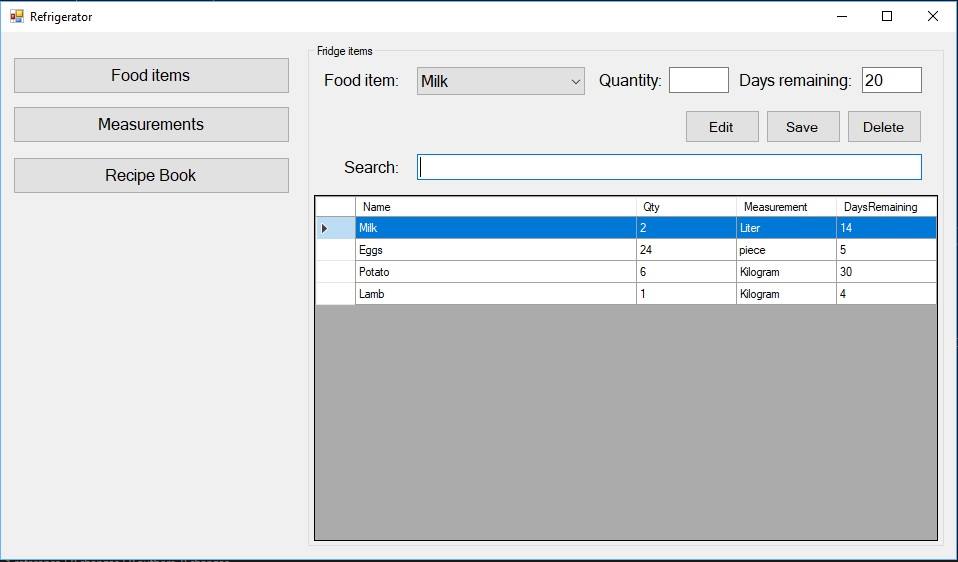
Sastoji se od klasa Ingredient, Recipe, Controller, DBLink, frmRecipeBook, Measurement i frmMeasurement. Controller klasa sastoji se od svojstava fridge, recipeBook, foodItemStorage i measurmentStorage.Glavne metode su AddFoodItem, dodavanje namirnica, AddItemToFridge, dodavanje namirnice u frižider, AddMeasurement, dodavanje mjera, AddRecipeToBook, dodavanje recepata u knjigu recepata, ConnectToDB spajanje na bazu podataka, DeleteFoodItem, brisanje namirnice iz baze, DeleteFridgeItem brisanje namirnice iz frižidera, UpdateFoodItem, ažuriranje namirnice, promjena imena i količine i UpdateMeasurment, ažuriranje mjera. Klasa Ingredient sastoji se od toga da pripada klasi Recipe, metode su Delete za brisanje, insert za unos sastojka, Save za spremanje sastojka te update za ažuriranje.

Klasa frmMeasurement sastoji se od metoda btnDelete\_Click za spajanje sa dugmom za brisanje, btnSave\_Click za spremanje, btnEdit\_Click za uređivanje mjera. InsertNewMeasurement dodavanje nove mjere i updateAnExisting, ažuriranje postojeće mjere. Klasa DBLink sastoji se od polja sqlConnection, metoda ExecuteSQL za izvršavanje sql naredbi, Query za ispis naredbi, TryConncet proba spajanja i TryDisconnect za odspajanje. Klasa Measurement sastoji se od polja vidljivih na sučelju, Name i Shortname. Metode unutar klase su Delete za brisanje mjere, insert za unos mjere, Save za spreamanje mjera u bazu, update za ažuriranje.

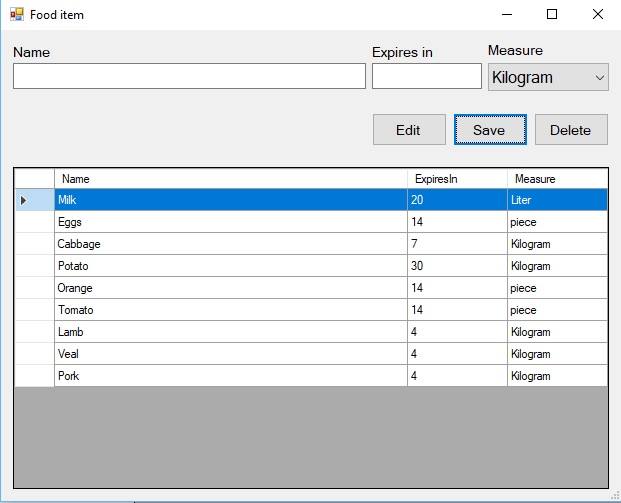
Klasa DBLink sastoji se od polja sqlConnection, metoda ExecuteSQL za izvršavanje sql naredbi, Query za ispis naredbi, TryConncet proba spajanja i TryDisconnect za odspajanje. Klasa Measurement sastoji se od polja vidljivih na sučelju, Name i Shortname. Metode unutar klase su Delete za brisanje mjere, insert za unos mjere, Save za spreamanje mjera u bazu, update za ažuriranje. Klasa Ingredient sastoji se od toga da pripada klasi Recipe, metode su Delete za brisanje, insert za unos sastojka, Save za spremanje sastojka te update za ažuriranje. Klasa Recipe sastoji se od polja vidljivih na sučelju CockTime, vrijeme potrebno za spravljanje jela, Description, kratki opis jela, ID, Ingredients što su sastojci i Name ime. Metode su AddIngredient dodavanje sastojaka u bazu, Delete, njihovo brisanje, RemoveIngredient, odnosno micanje sastojka, Save za spremanje sastojaka i update za ažuriranje sastojaka. Klasa frmRecipeBook sastoji se od metoda btnDeleteIngredient za spajanje sa dugmom za brisanje, btnSaveIngredient za spremanje, btnEditIngredient za uređivanje sastojaka.

**6. Korisničke upute**

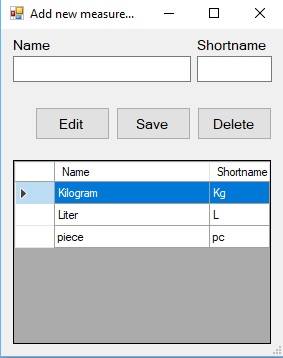
Sljedeće slike prikazuju stvarna sučelja aplikacije te njima se pojašnjava način korištenja aplikacije.



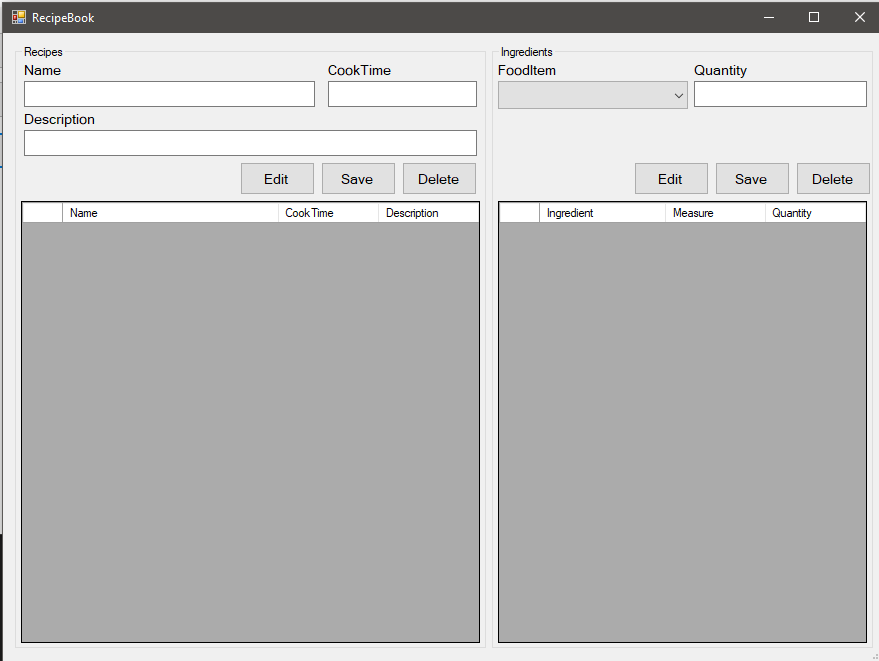
1. slika prikazuje glavni izbornik. Lijeva strana sadrži glavne funkcije. Prva je **Food items**, pritiskom na nju otvara nam se sučelje. Imamo opcije dodavanja namirnice, količine i postavljanja roka isteka. Dolje navedena funkcija **Edit** služi za izmjenu tih parametara, **Save** služi spremanje u bazu te **Delete** koji služi za brisanje iz baze. Nakon toga imamo **Measurements** gdje imamo opcije za dodavanje mjera, njihovih imena, kratica. Te imamo naposljetku **Recipe Book**, knjigu recepta gdje je moguće pregledavati bazu recepata, dodavati recepte, njihovu izmjenu i brisanje.



2. slika prikazuje sučelje za dodavanje namirnica. Vrlo jednostavan i intuitivan meni. Gornja lijeva strana prikazuje **Name** gdje unosimo ime namirnice, nakon toga imamo **Expires in**, opcija za stavljanje broja dana kad namirnica istječe, odnosno njen rok trajanja te s krajnje desne strane nalazi se **Measure** gdje imamo opcije u kilogramima (kg) i litrama (lt). Također imamo i tipku **Edit** prvu na donjoj polovici ekrana, kojom je moguće mijenjati ime, rok isteka i mjeru. Pritiskom na donju tipku **Save**, namirnica se sprema te ju je moguće vidjeti u donjem prostoru predviđenom za provjeru stanja zaliha.



3. slika prikazuje sučelje za dodavanje nove mjere. Uz unaprijed definirane mjere za kilograme (kg) i litre (lt). Korisnik ima mogu mogućnost dodavanja svojih mjera. Npr. moguće je dodati mjeru umjesto kg da stavi se grami i slično tome, umjesto litara da se stavi mililitri. Proces dodavanje mjere isti je kao u slučaju dodavanja namirnice. Gornji lijevi kut imamo **Name**, unosi se puno ime željene mjere, sljedeća opcija je **Shortname**, kratica mjere. Dolje navedena dugmad **Save**, služi za spremanje mjere u bazu. **Edit** za izmjenu imena ili kratice te **Delete** za brisanje željene mjere. Stvorene mjere spremaju se u donji prostor te su vidljive korisniku.



4.slika prikazuje funkcionalnost **Recipe Book**, odnosno knjigu recepata. Imamo mogućnost unosa recepata po svojo želji. U gornji dio unosimo ime recepta pod **Name**, **Cook Time** jednostavno označava vrijeme potrebno za spravljanje jela te njegov opis **Description**. Dolje nam se nalaze opcije **Edit** za izmjenu recepata, **Save** za spremanje te **Delete** za brisanje recepta iz baze. Desna strana odnosi se potrebne namirnice za spremanje određenog jela. **Food item** ima opciju unosa namirnica, **Quantity** za njihovu količinu. Također imamo opciju **Edit** za ažuriranje namirnica, **Save** za spremanje tokom unosa namirnica i **Delete** za brisanje namirnica.