

Yemek Tarifi

1st Eyüp Ensar TOĞUŞLU

Kocaeli University

Computer Engineer

Kocaeli, Turkey

eyupensar89544@gmail.com

2nd Mahir Enes AKPINAR

Kocaeli University

Computer Engineer

Kocaeli, Turkey

mahirennesakpinar@gmail.com

Abstract—Bu çalışma, yemek tarifleri uygulaması geliştirmek için JavaFX teknolojilerini kullanarak bir masaüstü uygulama oluşturmayı amaçlamaktadır. Bu projede kullanıcıların kolayca tarif ekleyebileceği, güncelleyebileceği ve tarif detaylarına ulaşabileceği bir arayüz geliştirilmiştir. JavaFX kullanılarak kullanıcı dostu bir deneyim sağlanması hedeflenmiştir. Proje kapsamında, dinamik arama ve filtreleme özellikleri sunan, veritabanı yönetimi ve kullanıcı arayüzü tasarımı konusunda deneyim kazandırmayı amaçlayan bir uygulama geliştirilmiştir. Kullanıcıların ellerindeki mevcut malzemelerle hangi yemekleri yapabileceklerini görebilmesi ve tarifleri organize edebilmesi sağlanmıştır. Ayrıca, projenin algoritması ve temel kod yapısı ayrıntılı bir şekilde açıklanmış ve deneysel sonuçlar üzerinde değerlendirmeler yapılmıştır. Bu çalışma, veritabanı yönetimi ve algoritma geliştirme konularındaki becerileri geliştirmek ve kullanıcı dostu yazılım geliştirme hakkında deneyim kazanmak amacıyla gerçekleştirilmiştir. Uygulamanın geliştirilmesi sırasında JavaFX kullanılması, kullanıcı dostu bir grafiksel arayüz sağlanmasına olanak tanımış ve modern bir yazılım çözümü ortaya çıkarılmıştır.

Index Terms—component, formatting, style, styling, insert

I. GİRİŞ

Günümüzde dijital ortamda yemek tariflerine erişim artmakta ve yemek tarifleri uygulamaları kullanıcılar tarafından yoğun talep görmektedir. Bu çalışma, yemek tariflerini dijital bir platformda organize etmek ve kullanıcıların özelleştirilmiş tariflere kolayca ulaşmasını sağlamak amacıyla geliştirilmiştir. Proje, kullanıcıların yemek tariflerini kaydedebileceği, mevcut malzemelerle hangi tariflerin yapılabileceğini görebileceği ve tarifler arasında arama ve filtreleme yapabileceği bir masaüstü uygulama sunmayı amaçlamaktadır.

Projede, kullanıcı dostu bir grafiksel arayüz sağlanarak yemek tariflerinin kolay yönetimi hedeflenmiştir. JavaFX kullanılarak modern bir arayüz oluşturulmuş ve kullanıcının tarifler üzerinde hızlı ve etkin işlemler yapabilmesi sağlanmıştır. JavaFX, dinamik arayüz bileşenleri ve kolay entegre edilebilir yapısıyla, kullanıcıların tarif ekleme, tarif güncelleme, tarif silme, tariflerde arama yapma, belirli kriterlere göre tarifleri filtreleme ve malzeme yönetimi gibi işlemleri rahatça yapabilmesini sağlamıştır. JavaFX, özellikle Java Swing'e kıyasla daha zengin grafik bileşenleri ve daha kullanıcı dostu bir yapı sunarak, geliştiricilerin interaktif ve modern arayüzler geliştirmesine olanak tanımaktadır.

Projede amaçlanan özellikler şunlardır:

- **Dinamik Arama ve Filtreleme:** Kullanıcıların tarifler arasında dinamik bir şekilde arama yapabilmesi ve belirli kriterlere göre tarifleri filtreleyebilmesi sağlanmıştır. Bu sayede kullanıcılar, aradıkları tariflere hızlı bir şekilde ulaşabilirler.
- **Veritabanı Yönetimi:** Tariflerin ve malzemelerin saklanması için bir veritabanı kullanılmıştır. Veritabanı yönetimi, tariflerin güvenli bir şekilde saklanması, güncellenmesi ve gerektiğinde silinmesi gibi işlemleri içerir. Bu sayede, tariflerin kalıcı olarak saklanması ve yönetimi kolay hale getirilmiştir.
- **Kullanıcı Dostu Arayüz Tasarımı:** Uygulamanın kullanıcı dostu bir arayüzle tasarlanması, kullanıcıların tarifleri kolayca ekleyebilmesini, düzenleyebilmesini ve arayabilmesini sağlamıştır. Elektron teknolojisi kullanılarak modern ve estetik bir kullanıcı deneyimi sunulmuştur.
- **Malzeme Yönetimi ve Tarif Önerileri:** Kullanıcılar, mevcut malzemelerini belirterek bir malzemelerle hangi yemekleri yapabileceklerini görebilirler. Bu özellik, kullanıcıların ellerindeki malzemeleri değerlendirmesini kolaylaştırmakta ve israfı önlemeye yardımcı olmaktadır.

Bu proje, aynı zamanda yazılım geliştirme sürecinde veritabanı yönetimi, algoritma geliştirme ve kullanıcı arayüzü tasarımı konularında deneyim kazandırmayı hedeflemiştir. Kullanıcıların ihtiyaçlarını karşılayan, esnek ve modern bir çözüm sunmak amacıyla geliştirilen bu uygulama, JavaFX ve Electron teknolojilerinin gücünden yararlanarak hem görsel açıdan tatmin edici hem de işlevsel olarak güçlü bir yazılım ortaya koymuştur. Bu raporda, uygulamanın geliştirilme süreci, kullanılan yöntemler, algoritmalar ve elde edilen deneysel sonuçlar ayrıntılı olarak incelenecektir.

A. Amaç

- Dinamik arama ve filtreleme özelliklerine sahip bir masaüstü uygulaması geliştirilmesi.
- Bir uygulama içerisinde istenilen özellikteki ürünlerin filtrelenebilmesi ve sıralanması özelliklerini sağlamak.
- Veritabanı yönetimi ve algoritma geliştirme konularındaki becerilerin geliştirilmesi.
- Kullanıcı arayüzü tasarımı ve kullanıcı dostu yazılım geliştirme hakkında deneyim kazandırılması.

II. YÖNTEM

Projenin geliştirilmesinde JavaFX, kullanıcı arayüzü oluşturmak ve uygulama içi bileşenleri entegre etmek için tercih edilmiştir. JavaFX, masaüstü uygulamaları geliştirme konusunda zengin bileşenler ve grafiksel araçlar sunarak, kullanıcılara etkileşimli bir deneyim sağlamıştır. JavaFX, özellikle Java Swing'e göre daha kolay ve modern bir arayüz geliştirme süreci sunar. Dinamik ve zengin grafik bileşenleri sayesinde, uygulamanın kullanıcı etkileşimi ve genel kullanımı daha verimli hale getirilmiştir.

Proje kapsamında veritabanı yönetimi için MySQL kullanılmıştır. MySQL, veri güvenliği ve performans açısından güçlü bir veritabanı yönetim sistemi sunmaktadır. Tariflerin ve malzemelerin saklanması, güncellenmesi ve silinmesi işlemleri MySQL veritabanı üzerinden gerçekleştirilmiştir. Veritabanı sorgularının performansını artırmak amacıyla indeksleme teknikleri ve optimize edilmiş SQL sorguları kullanılmıştır. Bu sayede, tariflerin ve malzemelerin verimli bir şekilde yönetimi sağlanmıştır.

A. Teknik Zorluklar

Proje geliştirilirken bazı teknik zorluklarla karşılaşmıştır. JavaFX kullanılarak modern ve kullanıcı dostu bir arayüz oluşturmak, aynı zamanda veritabanı yönetimi gibi işlevleri entegre etmek bazı entegrasyon sorunlarına yol açmıştır. JavaFX'in bileşenlerinin esnekliği sayesinde bu sorunlar aşılmış olsa da, kullanıcı etkileşimlerinin veritabanına doğru bir şekilde yansıtılması ve veri tutarlılığının sağlanması üzerinde dikkatli bir şekilde çalışılmıştır. MySQL veritabanı ile Java arasındaki bağlantının sağlanması ve veri çakışmalarının önlenmesi amacıyla uygun kitleme mekanizmaları ve veri senkronizasyon teknikleri uygulanmıştır.

B. Teknolojik Avantajlar

JavaFX'in sağladığı avantajlardan biri, zengin grafiksel bileşenleri ve kolay özelleştirilebilir arayüz yapısıdır. Bu sayede, kullanıcı arayüzü daha interaktif ve kullanımı kolay bir şekilde tasarlanabilmektedir. JavaFX, özellikle Java Swing'e kıyasla daha dinamik ve zengin bileşenler sunar; bu durum, kullanıcıların daha iyi bir deneyim yaşamasını sağlamaktadır. Örneğin, JavaFX'in sunduğu TableView, ListView gibi bileşenler, tariflerin ve malzemelerin rahatça listelenmesini sağlamıştır. Bu bileşenlerin kullanımı, kullanıcı deneyimini iyileştirmek açısından büyük önem taşımaktadır. Ayrıca CSS kullanılarak JavaFX bileşenlerinin görünümü kolayca özelleştirilebilmiş ve uygulamanın estetik açıdan tatmin edici olması sağlanmıştır. JavaFX'in sunduğu zengin grafik ve multimedya özellikleri, bu projede kullanıcı dostu ve çekici bir arayüz oluşturmak için büyük bir

avantaj sağlamıştır. Bu özellikler, geliştiricilerin kullanıcı etkileşimlerine odaklanarak daha kullanışlı ve modern uygulamalar geliştirmesini mümkün kılmıştır.

C. Kullanılan Yöntem ve Yaklaşımlar

Uygulamanın geliştirilmesinde, çevik yazılım geliştirme (Agile) metodolojisi kullanılmıştır. Bu yöntem, kullanıcı geri bildirimlerinin hızlı bir şekilde alınarak uygulamaya yansıtılmasını ve proje sürecinde ortaya çıkan değişikliklere hızlıca adapte olunmasını sağlamıştır. Kullanıcı gereksinimlerine uygun olarak belirli aralıklarla yapılan testler ve geri bildirim oturumları, uygulamanın daha kullanıcı dostu olmasını sağlamıştır.

Veritabanı yönetimi için MySQL kullanılmıştır. MySQL, güvenli ve hızlı veri işleme kabiliyeti ile tariflerin ve malzemelerin güvenli bir şekilde saklanmasını ve yönetilmesini sağlamıştır. Veritabanı tabloları, tariflerin ve malzemelerin ilişkisel olarak saklanmasını sağlayacak şekilde tasarlanmıştır. Bu yapı, verilerin tutarlı bir şekilde saklanmasına ve hızlı sorgulamalar yapılmasına olanak tanımıştır.

Kullanıcı arayüzü tasarımında ise JavaFX'in sunduğu zengin grafiksel bileşenlerden faydalanılmıştır. JavaFX bileşenleri, masaüstü uygulamalarında yaygın olarak kullanılan form ve tablo yapılarının oluşturulmasını sağlamış ve kullanıcılara estetik ve işlevsel açıdan tatmin edici bir deneyim sunmuştur. Kullanıcıların tarif ekleyebilmesi, malzeme yönetimi yapabilmesi ve tarif araması gibi işlemleri kolaylaştırmak amacıyla arayüzde gezinme menüleri ve etkileşimli düğmeler kullanılmıştır.

D. Proje Mimarisi

Proje, temel olarak birkaç ana bileşene ayrılmıştır:

1) **HelloController (Ana Sayfa):** HelloController (Ana Sayfa): HelloController sınıfı, uygulamanın ana kontrol birimidir ve kullanıcının arama, filtreleme, tarif ekleme ve malzeme ekleme gibi temel işlevleri gerçekleştirmesini sağlar. Bu sınıf, kullanıcıların tarifler üzerinde kolayca işlem yapabilmesi için gerekli tüm etkileşimli bileşenleri (ComboBox, TextField, Button, GridPane, vb.) kontrol eder. Kullanıcı tarifleri arayabilir, sıralayabilir ve tarif ekleyebilir.

`tarifEkle()` metodu, yeni bir tarif eklenmesini sağlamak amacıyla yeni bir pencere açar. Bu pencere, kullanıcıya tarif bilgilerini girebileceği bir arayüz sunar ve ardından veritabanına kaydedilir.

`tarifAra()` metodu, kullanıcının arama çubuğuna girdiği bilgilere göre tarifleri filtreleyerek GridPane üzerinde gösterir. Arama, tarif adına veya malzemelere göre yapılabilir. SQL sorguları dinamik olarak oluşturulur ve sonuçlar GridPane'e eklenir.

`malzemeEkle()`: metodu, kullanıcının tariflere malzeme eklemesini sağlar. Bu işlev, yeni bir pencere açarak kullanıcının eklemek istediği malzemeleri seçmesine olanak tanır.

`sirala()`: metodu, kullanıcının belirli kriterlere (örneğin malzeme sayısı veya hazırlama süresi) göre tarifleri sıralamasını sağlar ve bu tarifleri `GridPane` üzerinde listeler.

2) **YemeklerController(Tarif Görüntüleme ve Detaylar)**: `YemeklerController` sınıfı, tariflerin detaylarını göstermek ve kullanıcıların tariflere ilişkin bilgi almasını sağlamak amacıyla kullanılmaktadır. Bu sınıf, tariflerin maliyet hesaplaması, yeterli malzeme kontrolü, kategori ve hazırlama süresi gibi bilgilerin kullanıcıya sunulmasını sağlar. Ayrıca, kullanıcılar tariflerin görsellerine tıklayarak tarif talimatlarını içeren ayrı bir pencere açabilirler.

Tarif Bilgilerini Gösterme: `setData()` metoduyla tariflerin adı, kategorisi, hazırlanma süresi gibi bilgileri ve tarif görseli ekrana yansıtılır. Bu sayede kullanıcı, tarifin içeriğini detaylı bir şekilde görebilir.

Malzeme ve Maliyet Kontrolü: Tarifin içerdiği malzemelerin yeterli olup olmadığını kontrol eder ve eksik malzeme maliyetini hesaplar. Bu işlem, kullanıcının malzemeleri tamamlayıp tamamlamadığına göre tarifin uygunluğunu belirlemesine yardımcı olur. Yeterli malzeme olduğunda panel yeşil renkte, eksik olduğunda ise kırmızı renkte gösterilir. Eksik malzemelerin maliyeti de kullanıcının bilgilendirilmesi amacıyla ekranda gösterilir.

Talimatların Gösterimi: Tarif görseline tıklanarak `openTalimatPanel()` metodu aracılığıyla tarifin adımlarını gösteren ayrı bir pencere açılır. Bu pencere, kullanıcının tarifin nasıl yapılacağına dair ayrıntılı bilgilere ulaşmasını sağlar

HelloController(Anasayfa) ile Entegrasyon: `YemeklerController`, ana kontrolcü olan `HelloController` ile etkileşim içinde çalışır. Bu sayede tariflerin güncellenmesi gerektiğinde ana sayfadaki tarif listesi de güncellenir ve kullanıcıya en güncel bilgi sunulur.

3) **TarifEkleController (Yeni Tarif Ekleme)**: `TarifEkleController` sınıfı, kullanıcıların yeni tarifler eklemesini sağlamak amacıyla kullanılır. Bu sınıf, tarifin adı, kategorisi, hazırlanma süresi, talimatları ve kullanılan malzemeler gibi bilgilerin alınması ve bu bilgilerin veritabanına kaydedilmesi için gerekli işlevleri sunar.

Yeni Tarif Ekleme: `TarifEkleButon()` metodu, kullanıcının girdiği tarif bilgilerini (isim, kategori, hazırlama süresi, talimatlar ve resim) veritabanına ekler. Bu süreçte tarif adı gibi bazı alanlarda veri kontrolü yapılır; örneğin aynı isimde bir tarif varsa kullanıcıya bir hata mesajı gösterilir.

Malzeme Ekleme: Kullanıcı, tarife eklemek istediği malzemeleri `MalzemeEkleButonu()` metodu ile seçebilir ve tarifin malzemeler tablosuna ekleyebilir. Malzeme bilgileri ve miktarı kullanıcıdan alınır ve tarif ile ilişkilendirilir.

Yeni Malzeme Ekleme: `YeniMalzemeEkle()` metodu, kullanıcının veritabanında mevcut olmayan yeni bir malzeme ekleyebilmesini sağlar. Bu işlem için yeni bir

pencere açılarak malzeme bilgileri girilir ve veritabanına eklenir.

Kategori ve Malzeme Yönetimi: Kategori seçimi, malzeme listesi ve tarif malzemelerinin tablo üzerinde gösterimi sağlanır. Kullanıcı, tarif için gerekli malzemeleri seçip ekleyebilir ve tarif ile ilişkilendirebilir.

HelloController ile Entegrasyon: Yeni bir tarif eklendiğinde, ana sayfadaki tarif listesi de güncellenir. Bu sayede kullanıcıya anlık ve güncel bir bilgi sunulur.

4) **MalzemeEkleController (Yeni Malzeme Ekleme)**: `MalzemeEkleController` sınıfı, kullanıcıların tariflere yeni malzemeler eklemesini sağlamak amacıyla kullanılır. Bu sınıf, malzeme adı, miktar, birim ve birim fiyat gibi bilgilerin alınması ve bu bilgilerin veritabanına kaydedilmesi için gerekli işlevleri sunar.

Malzeme Ekleme: `MalzemeEkle()` metodu, kullanıcının belirttiği malzeme bilgilerini veritabanına ekler. Kullanıcıdan alınan malzeme adı, miktar, birim ve fiyat bilgilerinin boş olup olmadığı ve geçerli formatta olup olmadığı kontrol edilir. Eğer malzeme zaten veritabanında varsa kullanıcıya hata mesajı gösterilir.

Duplicate Kontrolü: `malzemeZatenVarMi()` metodu, girilen malzemenin veritabanında daha önce eklenip eklenmediğini kontrol eder. Eğer aynı malzeme mevcutsa kullanıcıya hata mesajı gösterilir.

Veritabanına Ekleme: `malzemeEkle()` metodu, girilen malzeme bilgilerini veritabanına ekler ve başarılı bir ekleme durumunda kullanıcıya bilgi mesajı gösterir. Aynı zamanda tarif ekleme paneli güncellenerek kullanıcıya en güncel bilgi sunulur.

TarifEkleController ile Entegrasyon: Yeni bir malzeme eklendiğinde, tarif ekleme panelindeki malzeme listesi de güncellenir. Bu sayede kullanıcıya anlık ve güncel bilgi sunulması sağlanır.

5) **TalimatController (Tarif Güncelleme ve Silme)**: `TalimatController` sınıfı, kullanıcıların tariflerin detaylarını görüntülemesi, güncellemesi ve tarifleri silmesi işlevlerini gerçekleştirmek amacıyla kullanılır. Bu sınıf, tarifin adını, kategorisini, hazırlanma süresini ve talimatları kullanıcıya sunar ve bu bilgilerin güncellenmesi veya silinmesi işlemlerini yönetir.

Tarif Bilgilerini Gösterme: `setTalimat()` metodu, tarifin adı, kategorisi, hazırlanma süresi ve talimatları gibi bilgileri kullanıcıya sunar. Ayrıca, tarifile ilgili görsel de kullanıcıya gösterilir.

Tarif Silme: `delete()` metodu, kullanıcının seçtiği tarifi veritabanından siler. Silme işlemi, önce tarifin `Tarif-Malzeme` tablosundaki ilişkili verilerini temizler, ardından `Tarifler` tablosundaki kayıt silinir. Silme işlemi tamamlandığında, ana ekrandaki tarif listesi de güncellenir.

Tarif Güncelleme: `update()` metodu, kullanıcının tarif üzerinde yaptığı değişiklikleri veritabanına kaydeder. Kullanıcı, tarifin adını, kategorisini, hazırlanma süresini veya talimatları değiştirdiğinde bu metod devreye girer ve tarifin bilgileri güncellenir.

HelloController ile Entegrasyon: Tarifin silinmesi veya güncellenmesi durumunda, ana sayfadaki tarif listesi `closeWindowAndRefresh()` metodu ile güncellenir. Bu sayede kullanıcının yaptığı değişiklikler anında ana ekrana yansır ve kullanıcıya güncel bilgi sunulur.

E. Veritabanı Şeması ve UML Diyagramları

Veritabanı tasarımında, veri tutarlılığını ve bütünlüğünü sağlamak kritik öneme sahiptir. Bu doğrultuda, TarifMalzeme tablosu, verilerin doğru ve uyumlu bir şekilde ilişkilendirilmesini sağlamak amacıyla önemli bir rol oynar.

TarifMalzeme tablosunda, TarifID ve MalzemeID sütunları, sırasıyla Tarifler ve Malzemeler tablolarındaki birincil anahtarlara yabancı anahtar olarak tanımlanmıştır. Bu yapı, aşağıdaki avantajları sağlar:

- **Veri Tutarlılığı:** Yabancı anahtar kısıtlamaları sayesinde, TarifMalzeme tablosuna yalnızca mevcut Tarifler ve Malzemeler tablolarındaki kayıtlarla eşleşen değerler eklenebilir. Bu, yanlış veya geçersiz verilerin eklenmesini önler ve veritabanındaki verilerin tutarlı kalmasını sağlar.
- **Referential Integrity (Bağlılık Bütünlüğü):** TarifID veya MalzemeID değerleri, başka bir tabloda bulunmuyorsa TarifMalzeme tablosuna eklenemez. Aynı şekilde, Tarifler veya Malzemeler tablosundaki bir kayıt silinmek istendiğinde, bu kayda bağlı TarifMalzeme kayıtları da dikkate alınır ve silinme işlemi sırasında bu ilişkiler göz önünde bulundurulur. Bu sayede ilişkiler bozulmadan yönetilir.
- **Veri Bütünlüğü Sağlama:** Bu yapılar, veritabanı tasarımının esnekliğini artırır ve veri bütünlüğünü korur. Örneğin, tariflerin içerdiği malzemeler veya bu malzemelerin miktarları değişse bile, veritabanı yapısı ve kısıtlamaları sayesinde tüm bilgiler birbirleriyle uyumlu kalır.

Bu tür veri tutarlılığı önlemleri, veritabanı tasarımınızın gücünü ve güvenilirliğini artırır, ayrıca sistemin doğru çalışmasını garanti eder.

Veritabanı şeması ve UML diyagramları, sistemin yapısını ve ilişkilerini açıkça gösterir. Aşağıda sistemin nasıl organize edildiğine dair detaylı bir açıklama ve iki farklı UML diyagramı sunulmuştur.

Şekil 1, Malzeme, Tarifler, ve TarifMalzeme sınıflarını ve bunların birbirleriyle olan ilişkilerini göstermektedir. Bu diyagram, sistemin nasıl yapılandırıldığını ve her bir sınıfın hangi özelliklere sahip olduğunu ortaya koyar:

- **Malzeme Sınıfı:** Malzemeleri temsil eder ve malzemeID, malzemeAdi, toplamMiktar, malzemeBirim, ve birimFiyat gibi özelliklere sahiptir.
- **Tarifler Sınıfı:** Tariflerin temel özelliklerini içerir ve tarifID, tarifAd, kategori,

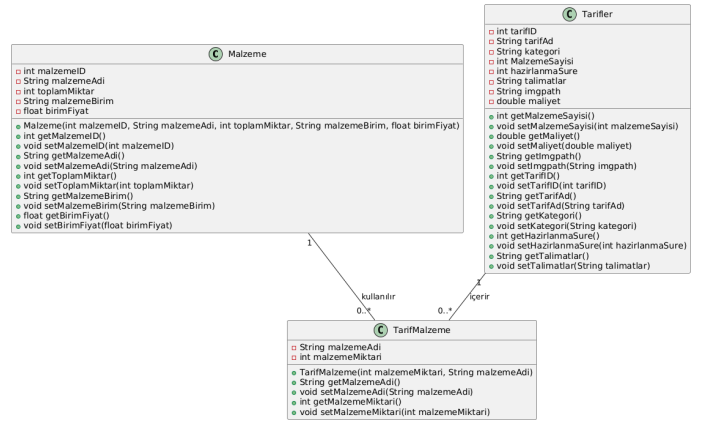


Fig. 1. Veritabanı UML Sınıf Diyagramı

hazirlanmaSure, talimatlar, ve impath gibi bilgileri tutar.

- **TarifMalzeme Ara Sınıfı:** Tarifler ve Malzeme arasındaki many-to-many ilişkiyi modellemek için kullanılan ara sınıftır. Bir tarifin hangi malzemeleri ve ne miktarda kullandığını belirler. Bu diyagram, Tarifler ve Malzeme arasında doğrudan bir ilişki olmadığını, ancak TarifMalzeme aracılığıyla dolaylı olarak bağlandığını gösterir.

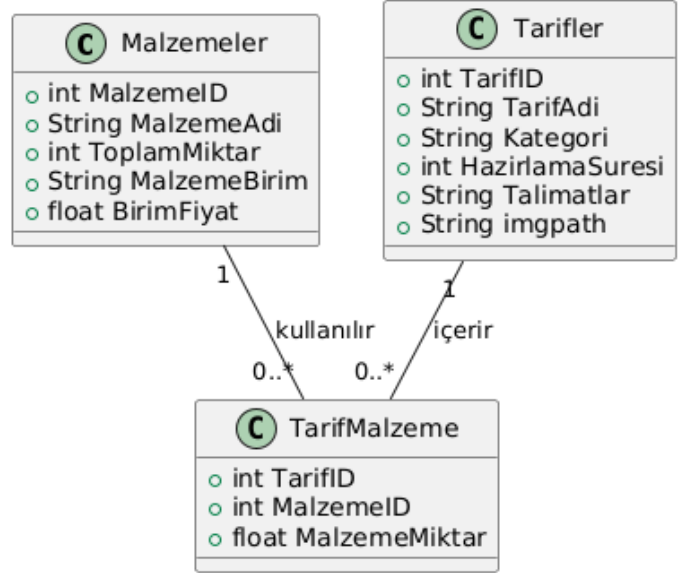


Fig. 2. Çoka Çok İlişki Veritabanı Şeması

Şekil 2, sistemdeki veritabanı tabloları arasındaki çoka çok ilişkiyi göstermektedir. Bu şema, UML sınıf diyagramından farklı olarak veritabanı perspektifinden ilişkilendirme yapısını gösterir:

- Tarifler ve Malzemeler tabloları birbirine doğrudan bağlanmaz. Bunun yerine, TarifMalzeme adlı bir ara tablo kullanılır.

- Ara tablo, tariflerin içerdiği malzemeleri ve miktarları tutar, böylece hem bir tarifin birçok malzeme içermesine hem de bir malzemenin birden fazla tarifte kullanılmasına olanak tanır.

III. DENEYSEL SONUÇLAR

Bu bölümde, uygulamanın geliştirilmesi ve test edilmesi sürecinde elde edilen deneysel sonuçlar sunulmaktadır. Proje testleri sırasında elde edilen bulgular ve uygulamanın işleyişine dair değerlendirmeler sunulmuştur. Bu bölümde, uygulamanın çeşitli ekran görüntülerine yer verilerek, kullanıcı arayüzünün işlevselliği ve kullanıcı deneyimi görsellerle desteklenmiştir.

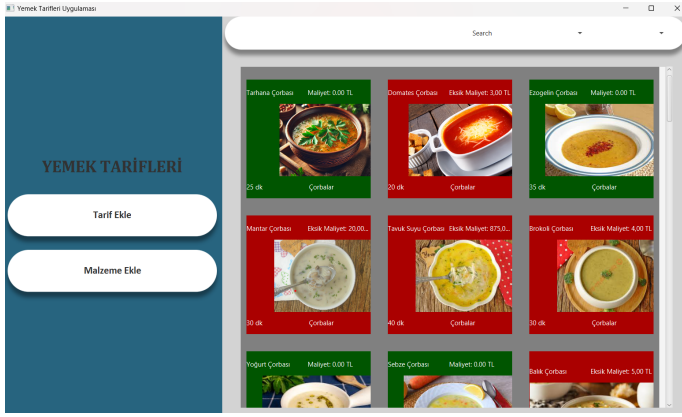


Fig. 3. Ana Sayfa

Uygulamanın ana sayfasında kullanıcıların mevcut tarifleri listeleyebildiği, tarifler arasında gezinebildiği ve tariflerin detaylarını görebildiği ekran görüntüsü sunulmuştur. Bu ekran, kullanıcıların arayüzle nasıl etkileşime geçtiğini ve tariflerin nasıl listelendiğini göstermektedir.

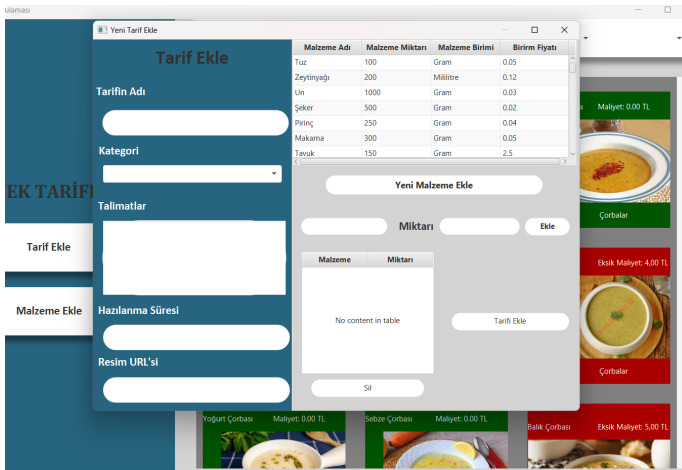


Fig. 4. Tarif Ekleme Ekranı

Kullanıcıların yeni tarif ekleyebildiği ekran görüntüsü sunulmuştur. Bu ekran, tarif adı, malzemeler, hazırlama

süresi gibi bilgilerin nasıl girildiğini göstermektedir. Kullanıcıların tarif oluşturma sürecinde karşılaştığı arayüz ve bu sürecin basitliği görsellerle desteklenmiştir.

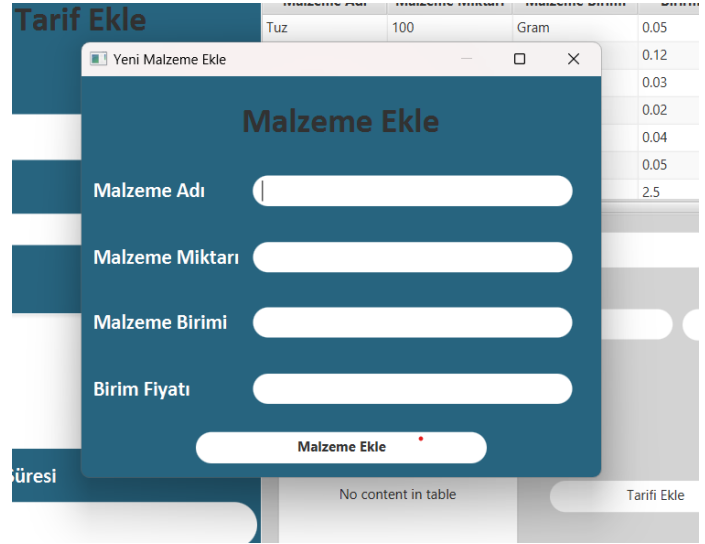


Fig. 5. Malzeme Ekleme Ekranı

Malzemelerin uygulamaya nasıl eklendiğini gösteren ekran görüntüsü sunulmuştur. Bu bölüm, kullanıcıların mevcut malzemeleri nasıl yönettiğini ve tarife uygun malzemeleri nasıl ekleyebildiğini gösterir.

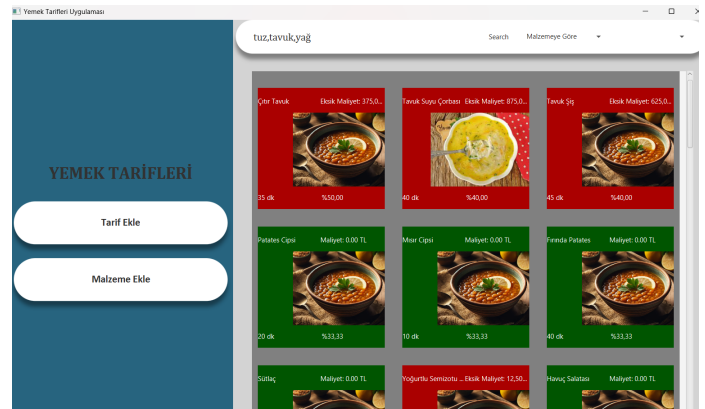


Fig. 6. Arama ve Filtreleme Sonuçları

Kullanıcıların belirli malzemelerle yapabilecekleri tarifleri bulmasına olanak tanıyan arama ve filtreleme işlevlerinin ekran görüntüleri sunulmuştur. Bu ekran, kullanıcıların mevcut malzemelerine göre uygun tarifleri nasıl bulduğunu ve arama sürecini nasıl yönettiğini gösterir.

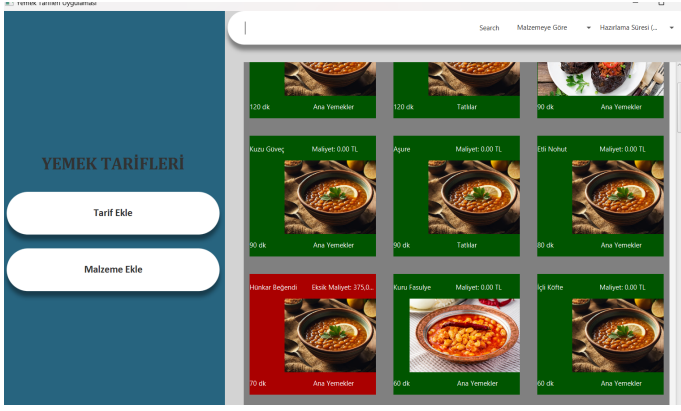


Fig. 7. Sıralama Sonuçları

Tariflerin farklı kriterlere göre (örneğin malzeme sayısı veya hazırlama süresi) sıralandığı ekran görüntüleri sunulmuştur. Bu bölümde, kullanıcıların tarifleri farklı şekillerde sıralayarak aradıkları tariflere daha kolay ulaşabildikleri gösterilmektedir.

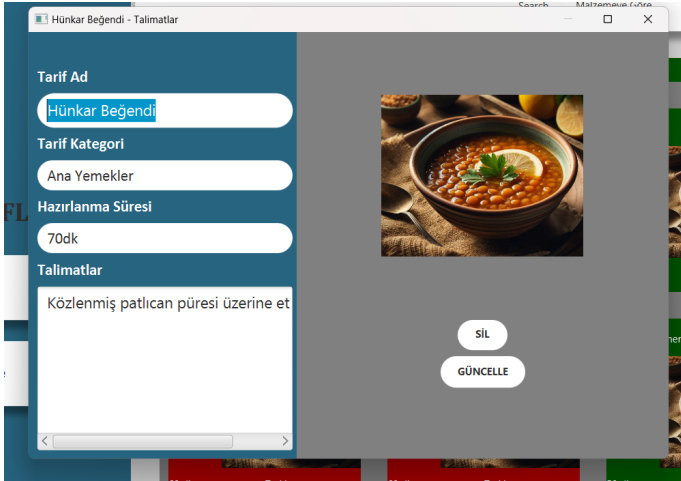


Fig. 8. Tarif Güncelleme ve Silme Ekranı

Talimatlar ekranında, kullanıcının tarif detaylarını güncelleyebildiği ve tarifi silebildiği bölümler gösterilmiştir. Kullanıcıların tariflerini düzenleme ve silme işlemlerini basit bir arayüzle nasıl gerçekleştirdiği ekran görüntüleriyle desteklenmiştir.

IV. SONUÇ

Proje sonucunda, kullanıcıların yemek tariflerini dijital bir platformda kolayca yönetebilmesi için işlevsel ve kullanıcı dostu bir masaüstü uygulama geliştirilmiştir. JavaFX teknolojisinin sunduğu zengin grafik bileşenleri ve veritabanı yönetimi sayesinde kullanıcıların tarif ekleme, malzeme yönetimi, arama ve filtreleme gibi işlemleri hızlı ve verimli bir şekilde yapabilmesi sağlanmıştır.

Uygulamanın sunduğu özellikler sayesinde kullanıcıların mevcut malzemelerle hangi tarifleri yapabileceklerini

görmeleri, tarifleri ekleyip düzenleyebilmeleri ve farklı kriterlere göre tarifleri arayıp sıralayabilmeleri mümkün hale gelmiştir. Kullanıcı dostu arayüz ve modern yazılım mimarisi, kullanıcıların bu işlemleri rahatlıkla yapmasını sağlamıştır.

Bu proje, yazılım geliştirme sürecinde veritabanı yönetimi, algoritma geliştirme ve kullanıcı arayüzü tasarımı gibi konularda önemli deneyimler kazanılmasına olanak tanımıştır. Ayrıca, kullanıcı gereksinimlerine uygun bir çözüm sunarak, yazılım geliştirme ve kullanıcı deneyimi konularında önemli kazanımlar elde edilmiştir. Geliştirilen uygulama, yemek tarifleri yönetimi konusunda kullanıcılara modern, kullanımı kolay ve işlevsel bir çözüm sunmaktadır.

REFERENCES

- [1] <https://www.youtube.com/watch?v=XIAzQ170kzM>
- [2] <https://github.com/k33ptoo/Drapo-Dashboard-JavaFX>
- [3] <https://www.youtube.com/watch?v=VOiFmZyGAds>
- [4] <https://www.youtube.com/watch?v=DIZtUGfEW0Q>