

Lezzet Diyarı Mobil Uygulaması:Android Studio ve Firebase Kullanarak

Tuğçe Gül

Kocaeli Üniversitesi

Bilişim Sistemleri Mühendisliği: Mobil Uygulama Geliştirme Dersi

İzmit/Kocaeli

221307036@kocaeli.edu.tr

Bu çalışma, mobil platformda çalışan bir yemek tarifleri paylaşım uygulamasının geliştirilmesini amaçlamaktadır. **Lezzet Diyarı** adı verilen bu uygulama, kullanıcıların kolayca üye olmalarını, oturum açmalarını ve yemek tarifleri ekleyerek diğer kullanıcılarla paylaşımlarını sağlayan bir kullanıcı arayüzü sunmaktadır. Kullanıcılar ayrıca tarif detaylarını görüntüleyebilir, tariflere yorum yapabilir ve favori tariflerini listeleyebilir. Projede veri yönetimi için **Firestore Realtime Database** kullanılmış olup, kullanıcı deneyimi ve veri güvenliği öncelikli olarak ele alınmıştır. Bu çalışma, dijital tarif paylaşımının hızla artan popülaritesine katkı sağlayarak, yemek tarifleri paylaşımı için etkili bir mobil uygulama örneği sunmaktadır.

TANITIM

Yemek tarifleri paylaşımı, dijital platformlarda giderek daha popüler hale gelmektedir. Bu çalışma, mobil cihazlar üzerinden kullanıcıların yemek tarifleri paylaşımlarını ve tariflere erişimlerini kolaylaştırmak amacıyla geliştirilen bir uygulamanın incelenmesini ve geliştirilmesini ele almaktadır. Günümüzde yemek tarifleri paylaşım platformlarına olan talep artmakta ve kullanıcıların bu alanda keyifli, kolay ve güvenli bir deneyim yaşamaları önem kazanmaktadır. Bu projenin temel amacı, kullanıcıların mobil cihazları üzerinden yemek tariflerini kolayca ekleyebilmelerini, görüntüleyebilmelerini ve diğer tariflerle etkileşimde bulunabilmelerini sağlayan bir uygulama geliştirmektir. **Lezzet Diyarı** adlı uygulama, kullanıcı dostu bir arayüz ile tarif ekleme, detaylarını görüntüleme, tariflere yorum yapma ve favorilere ekleme gibi fonksiyonları sunarak kullanıcıların ihtiyaçlarını karşılamayı hedeflemektedir. Bu proje, dijital tarif paylaşım alanında kullanıcı dostu ve etkili bir çözüm sunmaktadır..

I.KULLANIM KOLAYLIĞI

Lezzet Diyarı uygulaması, kullanıcıların platformu kolayca kullanabilmeleri için özenle tasarlanmıştır. Uygulamanın temel özellikleri, kullanıcıların yemek tariflerini görüntüleme, ekleme ve tariflere etkileşimde bulunma deneyimlerini artırmayı hedeflemektedir.

A. Ara Yüz Tasarımı

Kullanıcıların rahatça gezinmesini ve istedikleri tariflere ulaşmasını sağlamak için sezgisel bir arayüz tasarlanmıştır. Kullanıcılar, tarif kategorilerine ve aradıkları yemek tarifine işlevsel bir arama özelliği ve basit navigasyon menüsü sayesinde kolaylıkla erişebilmektedir. Minimalist ve kullanıcı dostu tasarım, uygulamanın kullanımını daha keyifli hale getirmektedir..

B. Tarif Sunumu

Yemek tarifleri, görsel olarak çekici bir şekilde sunulmuş ve detaylı açıklamalarla desteklenmiştir. Kullanıcıların

tarifin malzemelerini, hazırlanış adımlarını ve servis önerilerini incelemeleri kolaylaştırılmıştır. Ayrıca, tariflere ait görseller, tarifin nasıl görüneceği hakkında kullanıcılara fikir vermektedir.

C. Yorum ve Favori İşlemleri

Kullanıcıların tariflere yorum yapabilmesi ve beğendikleri tarifleri favorilerine ekleyebilmesi için sade ve hızlı bir işlem akışı tasarlanmıştır. Favori tarifler, kullanıcıların profil sayfasından kolayca erişilebilecek şekilde düzenlenmiştir. Bu özellikler, kullanıcıların tariflerle etkileşim kurmasını ve kişisel bir tarif koleksiyonu oluşturmasını kolaylaştırmaktadır..

II.PROJENİN HAZIRLANMASI SÜRECİ

A. Veri Düzenleme ve Toplama

- **Tarif Kataloğu Oluşturma:** Lezzet Diyarı uygulamasında sunulacak yemek tariflerinin kategorize edilmesi ve kataloğun oluşturulması. Bu aşamada tariflerin isimleri, açıklamaları, malzeme listeleri, hazırlanış süreleri, porsiyon sayıları ve varsa görselleri gibi detaylar toplanmıştır.Ürün Sunumu ve Detaylandırma: Her ürün, görsel açıdan çekici bir şekilde sunulmalıdır. Ürün açıklamaları, özellikleri ve ölçüleri detaylı bir şekilde verilmelidir. Ürün fotoğrafları, ürünün gerçekçi ve net bir şekilde gösterilmesi için yüksek kalitede olmalıdır.
- **Kategori Yapısı:** Tariflerin "Çorbalar", "Salatalar", "Tatlılar" gibi farklı kategorilere ayrılması. Bu yapı, kullanıcıların tariflere kolayca ulaşmasını sağlamaktadır.Mobil Uyum ve Kullanılabilirlik: Uygulama, mobil cihazlarda da sorunsuz bir şekilde çalışmalı ve kullanıcılar için optimize edilmiş bir deneyim sunmalıdır.
- **Veri Yönetimi:** Kullanıcılar tarafından eklenen tariflerin veritabanında düzgün bir şekilde kaydedilmesi ve kategoriler altında listelenmesi için Firestore Realtime Database kullanılmıştır.
- **Analiz ve İyileştirme:** Test kullanıcılarından alınan geri bildirimlerle uygulamanın arayüzü ve işlevselliği sürekli olarak iyileştirilmiştir. Kullanıcı deneyimini artırmak için tarife erişim, arama ve favori yönetimi gibi süreçler optimize edilmiştir.

B.İçerik Oluşturma

- **Arayüz Tasarımı:** Kullanıcıların kolayca tariflere ulaşabilmesi ve tarif ekleyebilmesi için kullanıcı dostu bir arayüz tasarlanmıştır. Arayüz minimalist ve sezgisel olacak şekilde geliştirilmiş; kullanıcıların aradığı tariflere

zahmetsizce ulaşmasını sağlayan bir navigasyon menüsü oluşturulmuştur.

- **Tarif Sunumu ve Detaylandırma:** Her tarif, görsel açıdan çekici bir şekilde sunulmuş ve detaylı açıklamalarla desteklenmiştir. Tarif açıklamalarında malzemeler, hazırlanış adımları ve servis önerileri gibi bilgiler kullanıcıya sunulmuştur. Tarif görselleri, kullanıcıların yemek hakkında net bir fikir edinmelerini sağlamak amacıyla yüksek kalitede hazırlanmıştır.
- **Yorum ve Favori Yönetimi:** Kullanıcıların tariflere yorum yapabilmesi ve favori tariflerini bir listede toplayabilmesi için basit ve etkili bir işlem akışı tasarlanmıştır. Bu özellik, kullanıcıların tariflerle daha fazla etkileşimde bulunmasını sağlamıştır.

Lezzet Diyarı uygulamasında, kullanıcıların tarifleri kolayca görüntüleyebilmeleri, tarif ekleyebilmeleri ve favori tariflerini yönetebilmeleri için Navigation Component altyapısı ve

C. Navigation Component ve Fragment Kullanımı

Uygulama içerisinde Navigation Component ve Fragment yapıları tercih edilerek modern, esnek ve kullanıcı dostu bir navigasyon deneyimi sağlanmıştır. Farklı ekranların bağımsız olarak yönetilebilmesi için her bir özellik, ayrı bir Fragment sınıfında yapılandırılmıştır. Örneğin,

RecipeFragment yemek tariflerini listelemek için kullanılırken, **RecipeDetailFragment** seçilen bir tarife ait detayları gösterir ve **FavoritesFragment** kullanıcının favori tariflerini sergiler. Bu yaklaşım, kodun daha modüler, okunabilir ve sürdürülebilir olmasına katkı sağlamıştır.

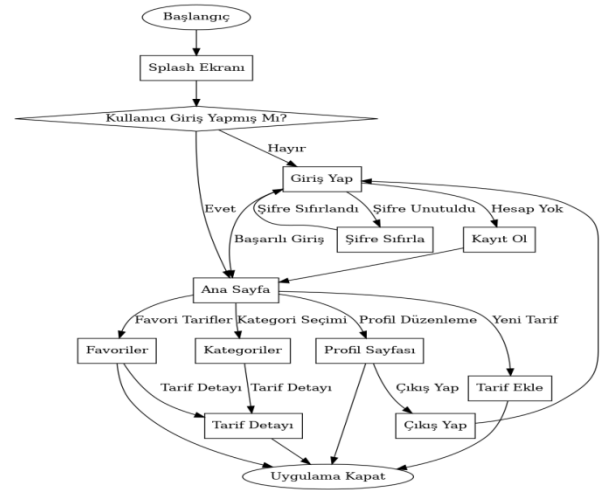
Fragment yapısının kullanılması, hem kullanıcı arayüzü bileşenlerinin yeniden kullanılabilirliğini artırmış hem de kodun farklı ekranlara kolayca uyarlanması sağlamıştır. Navigation Component ile entegre çalışan bu yapı, kullanıcıların ekranlar arasında kolay ve sorunsuz bir şekilde gezinmesine olanak tanımıştır. Navigation Component'in merkezi bir kontrol mekanizması sunan **nav_graph.xml** dosyası üzerinden çalışması, navigasyon yapısının hem düzenli hem de yönetilebilir olmasını sağlamıştır. Tür güvenliği sayesinde fragmentlar arasında veri aktarımı sırasında oluşabilecek hataların önüne geçilmiş ve geri yön tuşu desteği ile kullanıcılar önceki ekranlara kolayca dönebilmiştir.

Uygulamada ayrıca tariflerin kategorilere göre görüntülenmesini sağlayan bir **Navigation Drawer** tasarlanmıştır. Bu yapı, Firebase veritabanından kategoriye ait tarifleri çekip ilgili ekrana yükleyerek kullanıcıların tariflere hızlıca erişmesini sağlamıştır. Navigation Drawer ile fragmentlar arasında geçiş daha akıcı hale getirilmiş ve kategorilere göre düzenlenen tariflerin kullanıcı dostu bir arayüzde sunulması sağlanmıştır.

Navigation Component ve Fragment kullanımı, uygulamanın kod yapısının modüler ve test edilebilir olmasına büyük katkı sağlamıştır. Her fragment, yalnızca kendi sorumluluk alanındaki işlemleri yerine getirecek şekilde tasarlanmış, böylece uygulamanın farklı bölümleri arasında bağımsızlık sağlanmıştır. Bu yapı, yalnızca kodun okunabilirliğini değil, aynı zamanda sürdürülebilirliğini de artırmıştır. Sonuç olarak, bu yaklaşım hem kullanıcı

deneyimini geliştirmiş hem de uygulamanın daha düzenli ve hata olasılığı düşük bir altyapıya kavuşmasını sağlamıştır.

D.Akış Diyagramı ve Varlık-İlişki Diyagramı



D.1.) Ana Uygulamanın akış diyagramı

Uygulamanın akış diyagramı, kullanıcıların uygulama içindeki hareketlerini düzenli ve sezgisel bir şekilde planlamak amacıyla oluşturulmuştur. Akış, başlangıç ekranından itibaren kullanıcıyı sistemli bir şekilde yönlendirmektedir. Uygulama açıldığında kullanıcı, ilk olarak bir giriş kontrolüne tabi tutulur. Eğer kullanıcı oturum açmışsa, doğrudan ana sayfaya yönlendirilir. Oturum açmamış kullanıcılar ise giriş ekranına yönlendirilir. Giriş ekranında, kullanıcıların mevcut hesaplarıyla giriş yapmaları sağlanır. Ayrıca, şifrelerini unutan kullanıcılar için "Şifre Sıfırla" ve hesabı olmayanlar için "Kayıt Ol" seçenekleri sunulmuştur.

"Kayıt Ol" ekranı, yeni kullanıcıların sistemde hesap oluşturmalarını sağlar. Bu işlem tamamlandıktan sonra kullanıcı tekrar giriş ekranına yönlendirilir ve oturum açması beklenir. Şifre sıfırlama ekranı ise şifresini unutan kullanıcıların yeni bir şifre belirlemelerine olanak tanır. Şifre sıfırlama işlemi tamamlandığında kullanıcı, giriş ekranına döner.

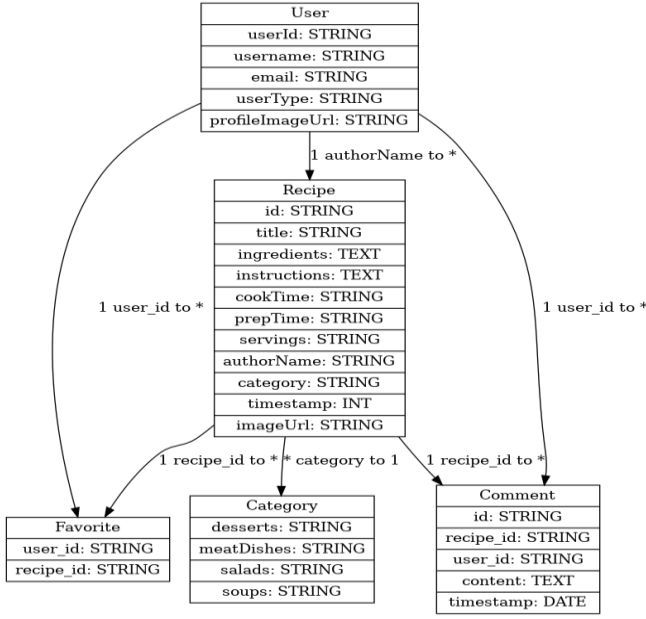
Başarılı bir girişin ardından kullanıcı ana sayfaya yönlendirilir. Ana sayfa, uygulamanın temel işlevlerine erişim sağlayan bir merkez ekranı görevi görür. Kullanıcı burada tarif ekleme, kategorilere göz atma ve favorilere erişme gibi işlemleri başlatabilir. "Tarif Ekle" özelliği ile kullanıcılar, yemek tariflerini sisteme ekleyebilir. Tarif ekleme işlemi sırasında tarifin adı, kategorisi, malzemeleri ve hazırlama talimatları gibi detaylar doldurulmaktadır.

Kullanıcı ana sayfadan bir tarif seçtiğinde "Tarif Detayı" ekranına yönlendirilir. Bu ekran, seçilen tarif ile ilgili tüm bilgileri içerir. Tarifin detaylarının yanı sıra kullanıcı, tarif hakkında yorum yapabilir veya tarifi favorilerine ekleyebilir. Uygulamada ayrıca tariflerin kategorilere göre incelenmesi de mümkündür. "Kategoriler" bölümü, tarifleri organize bir şekilde görüntüleme imkanı sunar. Kullanıcı, istediği kategoriye tıklayarak o kategoriye ait tariflere ulaşabilir ve detaylarına bakabilir.

Favori tariflerini görmek isteyen kullanıcılar "Favoriler" ekranına yönlendirilir. Bu ekranda kullanıcı daha önce favorilerine eklediği tarifleri görebilir ve bu tariflerin detaylarına erişebilir. Profil ekranı ise kullanıcıların kişisel bilgilerini görüntüleyip düzenlemelerini sağlar. Kullanıcı bu ekranda oturumunu kapatma işlemini de gerçekleştirebilir. Oturum kapatıldığında kullanıcı tekrar giriş ekranına yönlendirilir.

Son olarak, kullanıcı uygulamayı kapattığında tüm işlemler sonlandırılarak sistemden çıkış yapılır. Uygulamanın bu akış diyagramı, kullanıcı deneyimini optimize ederek her adımın belirli bir amaca hizmet etmesini sağlamaktadır. Ana sayfa, uygulamanın diğer tüm ekranlarına bağlanarak kullanıcının istediği işlemi kolayca gerçekleştirmesine olanak tanır. Bu yapı, hem işlevselliği hem de kullanıcı dostu bir arayüzü bir arada sunmaktadır.

D.2.) Varlık İlişki Diyagramı



Uygulamanın veritabanı yapısı, kullanıcılar, tarifler, kategoriler, yorumlar ve favoriler arasında güçlü bir ilişki modeli sunmaktadır. Bu yapı, uygulamanın tüm işlevlerini desteklemek için tasarlanmıştır ve ilişkiler, uygulama içerisinde gerçekleşen işlemleri tutarlı ve düzenli bir şekilde yönetmeyi sağlar.

User (Kullanıcı), uygulamanın temel varlıklarından biridir ve kullanıcılar benzersiz bir kimlik (userId) ile tanımlanır. Kullanıcılar; kullanıcı adı, e-posta adresi, profil resmi URL'si gibi bilgileri içerir. Kullanıcı türü (userType), normal kullanıcı veya farklı bir rol olup olmadığını belirler. Kullanıcılar, tarifler oluşturabilir, yorum yazabilir ve tarifleri favorilere ekleyebilir. Bir kullanıcı birçok tarif ekleyebilir ve yazdığı yorumlarla tariflere geri bildirim sağlayabilir. Aynı zamanda, favorilere tarif ekleyerek daha sonra erişim sağlayabilirler.

Recipe (Tarif), kullanıcılar tarafından oluşturulan içerikleri ifade eder. Tarifler, benzersiz bir kimlik (id) ile tanımlanır ve tarifin başlığı, malzemeleri, talimatları, pişirme süresi gibi detayları içerir. Tarifler bir kategoriye (category) ait olmalı ve görsel bir içerik için bir resim URL'sine (imageUrl) sahiptir. Her tarifin bir yazarı vardır ve bu yazarın kullanıcı adı, authorName alanıyla belirtilir. Tarifler, kullanıcılar tarafından favorilere eklenebilir ve farklı yorumlar alabilir.

Category (Kategori), tariflerin düzenli bir şekilde organize edilmesini sağlar. Tatlılar, et yemekleri, salatalar ve çorbalar gibi kategoriler, tariflerin daha kolay bulunmasına yardımcı olur. Her kategori birden fazla tarifi içerebilir, ancak her tarif yalnızca bir kategoriye ait olabilir.

Comment (Yorum), kullanıcıların tariflere geri bildirimlerini paylaşmalarına olanak tanır. Her yorum, benzersiz bir kimlik (id) ile belirlenir ve bir tarif (recipe_id) ilişkilendirilir. Ayrıca, yorumun hangi kullanıcı tarafından yazıldığını belirlemek için user_id kullanılır. Yorumlar, içeriği (content) ve yazılma tarihi/saatini (timestamp) içerir. Bir tarif birden fazla yoruma sahip olabilir, ancak her yorum yalnızca bir tarife ve bir kullanıcıya ait olabilir.

Favorite (Favoriler), kullanıcıların beğendiği tarifleri saklamalarını sağlar. Her favori kaydı, bir kullanıcı kimliği (user_id) ve bir tarif kimliğinden (recipe_id) oluşur. Bu yapı, bir kullanıcının birçok tarifi favorilere ekleyebilmesini sağlarken, her favori kaydı yalnızca bir kullanıcı ve bir tarif ile ilişkilidir.

Varlıklar arasındaki ilişkiler düzenli bir şekilde tasarlanmıştır. Bir kullanıcı birden fazla tarif ekleyebilir, ancak bir tarif yalnızca bir yazarı olabilir. Tarifler, bir kategoriye ait olurken bir kategori birçok tarifi içerebilir. Tarifler birden fazla yoruma sahip olabilir, ancak her yorum yalnızca bir tarife bağlıdır. Benzer şekilde, kullanıcılar birden fazla yorum yazabilir ve birden fazla tarifi favorilere ekleyebilir. Ancak her favori kaydı, yalnızca bir tarif ve bir kullanıcıya bağlıdır.

Bu yapı, uygulamanın işlevselliğini artırırken, veritabanı işlemlerinin tutarlı ve düzenli bir şekilde yapılmasını sağlar. Kullanıcıların tarif ekleme, yorum yapma ve favorilere ekleme gibi işlemleri, bu ilişkiler aracılığıyla desteklenmektedir.

IV.DENEYSEL SONUÇLAR

A.Projenin oluşturulması

Proje geliştirme sürecine başlamadan önce, uygulamanın temel tasarımını ve işlevlerini belirlemek istedim. İlk adım olarak, ekranın alt kısmında bir **navigasyon menüsü** oluşturarak uygulamanın ana yapısını ortaya koydum. Navigasyon menüsünü, **Tarifler, Favoriler, Tarif Ekle ve Giriş Sayfası** gibi temel sayfalarla yapılandırdım. Bu menüyü MainActivity.kt dosyasında tanımladıktan sonra, kullanıcı arayüzü tasarımlarını hazırlamaya başladım. Tasarımların ardından Firebase bağlantısını gerçekleştirdim ve uygulamanın işlevselliğini test etmek amacıyla ilk olarak LoginFragment üzerinde çalışmaya başladım. LoginFragment ı test ettikten sonra diğer fragmentların üzerinde çalıştım

A.GİRİŞ SAYFASI

Giriş ekranı, uygulamanın başlangıç noktasıdır ve kullanıcıların uygulamaya erişim sağlamak için kimlik doğrulama işlemlerini gerçekleştirdiği bölümdür. Kullanıcılar, bu ekran üzerinden mevcut hesaplarıyla giriş yapabilir, şifrelerini sıfırlayabilir veya yeni bir hesap oluşturabilir. Firebase Authentication altyapısı sayesinde güvenli bir giriş deneyimi sunan bu ekran, kullanıcı dostu bir arayüzle tasarlanmıştır. Giriş ekranı, uygulamanın diğer bölümlerine hızlı bir geçiş noktası sağlayarak kullanıcıların tariflere, favorilere ve diğer özelliklere erişimini mümkün kılar.

a.LoginFragment

LoginFragment, kullanıcıların uygulamaya giriş yapmasını sağlayan ana fragmenttir. Kullanıcılar, e-posta adreslerini ve şifrelerini girerek Firebase Authentication aracılığıyla kimlik doğrulama işlemlerini gerçekleştirebilir. Başarılı bir

giriş durumunda kullanıcı, RecipeFragment ekranına yönlendirilir. Eğer giriş sırasında hatalı bilgiler girilirse, kullanıcıya hata mesajı gösterilir ve gerekli düzeltmeleri yapması istenir. LoginFragment, kullanıcı dostu bir tasarımıyla hazırlanmış olup "Şifremi Unuttum" ve "Hesap Oluştur" gibi ek seçenekler sunar. Bu fragment, giriş işlemlerinin merkezini oluşturur ve navigasyon geçişleri ile diğer fragmentlere erişim sağlar.

b.ChangePasswordFragment

Kullanıcıların şifrelerini unutmaları durumunda yardımcı olan bir fragmenttir. LoginFragment üzerinden "Şifremi Unuttum" seçeneği seçildiğinde kullanıcı ChangePasswordFragment ekranına yönlendirilir. Bu ekran, kullanıcının e-posta adresine bir şifre sıfırlama bağlantısı göndermek için kullanılır. Firebase Authentication'ın sağladığı şifre sıfırlama mekanizması sayesinde, kullanıcılar güvenli bir şekilde şifrelerini yenileyebilir. Şifre sıfırlama işlemi tamamlandığında, kullanıcı yeniden LoginFragment ekranına dönerek giriş yapabilir.

c.SignUpFragment

Henüz hesabı olmayan kullanıcıların yeni bir hesap oluşturmasını sağlayan bir fragmenttir. LoginFragment ekranındaki "Hesap Oluştur" seçeneği seçildiğinde kullanıcı SignUpFragment ekranına yönlendirilir. Bu ekranda kullanıcıdan kullanıcı adı, e-posta ve şifre bilgileri alınır. Firebase Authentication ile entegre çalışan bu fragment, alınan bilgileri doğrular ve yeni bir kullanıcı kaydı oluşturur. Kullanıcı kaydı başarıyla tamamlandığında, kullanıcı tekrar LoginFragment ekranına yönlendirilir ve giriş yapması beklenir.

B.TARİF EKLEME SAYFASI

Tarif ekleme sayfası, kullanıcıların kendi tariflerini uygulamaya eklemesine olanak sağlayan bir bölümdür. Bu sayfa, kullanıcıdan tarif ile ilgili başlık, malzemeler, talimatlar, porsiyon bilgisi, hazırlık ve pişirme süresi gibi detayları alır. Ayrıca, kullanıcının tarif için görsel yüklemesi de sağlanır. Tarif ekleme işlemi tamamlandıktan sonra, tarif bilgileri Firebase Realtime Database'e kaydedilir ve diğer kullanıcıların erişimine açılır. Sayfa, kullanıcı dostu bir arayüzle tasarlanmış olup, tarif oluşturma sürecini basit ve etkili bir şekilde gerçekleştirmeyi amaçlar.

a.AddrecipeFragment

AddRecipeFragment, diğer fragmentlerle etkileşim halinde çalışır. Kullanıcı, giriş yapmadysa "Giriş Yap" butonu aracılığıyla LoginFragment ekranına yönlendirilir. Bu işlem, tarif ekleyebilmek için kullanıcıların önce giriş yapmasını zorunlu kılar. Eğer kullanıcı giriş yaptıysa, tarif ekleme ekranı aktif hale gelir. AddRecipeFragment, giriş kontrolü ve tarif ekleme süreçlerinde kritik bir rol oynar.

b.Kullanılan Adapter

Bu fragmentte ArrayAdapter sınıfı, Spinner bileşenlerini doldurmak için kullanılmıştır. Kategoriler, porsiyon sayıları ve süre bilgileri gibi seçenekler adaptör yardımıyla kullanıcıya sunulur. Firebase Realtime Database'den çekilen kategori bilgileri dinamik olarak Spinner bileşenine eklenir ve kullanıcı tarifi kategorisini buradan seçebilir. Servis ve süre seçenekleri ise sabit bir veri listesi olarak adaptöre tanımlanmıştır.

c.Kullanılan DataClass

Tarif ekleme işlemleri sırasında tarif bilgilerini düzenli bir şekilde temsil etmek ve Firebase Realtime Database'e kaydetmek için **Recipe** sınıfı kullanılmıştır. Bu sınıf,

tariflerin başlık, kategori, porsiyon bilgisi, hazırlık süresi, pişirme süresi, malzeme listesi, talimatlar, görsel bağlantısı, tarifi yazarı ve eklenme zamanı gibi detaylarını saklamak için tasarlanmıştır. Recipe sınıfı, tariflerin benzersiz bir kimlik (id) ile depolanmasını sağlayarak farklı tariflerin ayırt edilmesini mümkün kılar. Ayrıca, tarif görsellerinin Firebase Storage'da saklanması durumunda görsel bağlantısı (imageUrl) da bu sınıf üzerinden tutulur. Kullanıcıların tarifi eklediği tarih ve saati belirten zaman damgası (timestamp) da kaydedilerek tarifi sıralı bir şekilde görüntülenmesi sağlanır. Bu veri modeli, tariflerin eksiksiz bir şekilde saklanmasına olanak tanır ve diğer ekranlarda tarif detaylarının kolayca görüntülenmesini mümkün kılar.

C.FAVORİLER SAYFASI

Favoriler sayfası, kullanıcıların daha önce beğendiği veya kaydettiği tariflere hızlıca erişim sağlayan bir ekrandır. Bu sayfa, kullanıcıların tariflere olan ilgisini kaybetmeden favori listesine eklediği içerikleri görüntülemesine olanak tanır. Firebase Realtime Database ile entegre çalışarak her kullanıcının yalnızca kendi favori tariflerini görmesini sağlar. Favorilere eklenen tarifler, kullanıcının kimliğiyle ilişkilendirilmiş şekilde saklanır ve bu sayfa, giriş yapılmadığı sürece erişilemez durumdadır.

Favoriler sayfasının temel yapısını FavoritesFragment oluşturur. Bu fragment, favorilere eklenen tariflerin görselleştirilerek liste olarak gösterilmesini sağlar. Kullanıcı, listede bir tarif seçtiğinde, RecipeDetailFragment ekranına yönlendirilir ve tarifi tüm detaylarına erişebilir. Fragment, Firebase üzerinden favori tarifleri dinamik olarak çeker ve güncel bilgilerle kullanıcıya gösterir. Eğer bir tarif favorilerden çıkarılmak istenirse, Firebase Realtime Database'de ilgili düğümden silinir ve liste gerçek zamanlı olarak güncellenir.

Bu sayfada tariflerin listelenmesi için RecipeAdapter kullanılmıştır. Adaptör, tariflerin başlığı, görseli ve kısa bir açıklamasını içeren bilgileri RecyclerView üzerinde gösterir. Kullanıcı, listede bir tarif seçtiğinde tıklama olayı tetiklenir ve seçilen tarifi detayları başka bir ekranda açılır. Bu yapı, kullanıcı deneyimini geliştirerek favorilere eklenen tariflerin düzenli bir şekilde organize edilmesini sağlar.

Firebase Realtime Database ile tam entegre çalışan favoriler sayfası, her kullanıcının favorilerini ayrı bir şekilde yönetmesine olanak tanır. Tarif bilgileri, Recipe veri modeli kullanılarak işlenir. Bu model, tariflerin başlığı, kategorisi, malzemeleri, talimatları, hazırlık ve pişirme süreleri gibi bilgileri içerir. Ayrıca, tarifi görsel bağlantısı ve eklenme zamanı gibi detaylar da bu sınıf üzerinden takip edilir.

Sonuç olarak, favoriler sayfası, kullanıcıların beğendiği tariflere kolayca ulaşmasını sağlayan ve tarifi detaylı inceleyebilmesine imkan tanıyan bir ekran olarak tasarlanmıştır. Bu sayfa, kullanıcıların tarifleri kişisel bir liste halinde saklamasını ve düzenlemesini mümkün kılarak uygulama içi etkileşimi artırır.

D.ANASAYFA

Ana sayfa, uygulamanın başlangıç ekranını temsil eder ve kullanıcıların mevcut tariflere kolayca göz atmasını sağlar. Firebase Realtime Database ile entegre çalışan bu ekran, tarif bilgilerini dinamik bir şekilde alarak kullanıcıya bir liste halinde sunar. Liste, tariflerin görsellerini, başlıklarını ve kısa bilgilerini içerir. Kullanıcı, bir tarifi üzerine tıkladığında detay ekranına yönlendirilir ve tarifi daha ayrıntılı bilgilerini görüntüleyebilir. Gerçek zamanlı veritabanı desteği sayesinde, tariflerde yapılan değişiklikler anında ekrana yansır ve kullanıcı her zaman güncel bilgilerle karşılaşır.

Bu ekranın tasarımı, kullanıcı deneyimini geliştirmek üzere sade ve kullanışlı bir şekilde düzenlenmiştir. Tariflerin

düzenli bir şekilde listelenmesi için RecyclerView kullanılmış ve tariflere tıklanabilirlik özelliği eklenmiştir. Tarifler, Firebase Realtime Database'deki "recipes" düğümünden çekilen verilerle düzenlenir. Ana sayfa, kullanıcıların tarifler arasında kolayca gezinmesini sağlayarak, uygulamanın temel işlevlerinden birini yerine getirmektedir.

RecipeFragment (Ana Sayfa)

RecipeFragment, uygulamanın ana ekranını temsil eder ve kullanıcıların tariflere hızlıca erişmesini sağlar. Bu fragment, Firebase Realtime Database ile entegre çalışarak tarif bilgilerini dinamik bir şekilde alır ve bir liste (RecyclerView) halinde kullanıcıya sunar. Liste, tarifi görseli, başlığı ve diğer kısa bilgilerini içerir. Kullanıcı, bir tarifi üzerine tıkladığında RecipeDetailFragment ekranına yönlendirilir ve seçilen tarifi detayları görüntülenir. Firebase'in gerçek zamanlı güncelleme desteği sayesinde, tariflerde yapılan değişiklikler anında listeye yansır. Bu ekran, kullanıcıların tariflere kolayca göz atmasını sağlayan kullanıcı dostu bir ana sayfa işlevi görür.

a.RecipeDetailFragment

RecipeDetailFragment sınıfı, kullanıcıların bir tarifi detaylarını görüntüleyebileceği ve bu tarif hakkında yorum yapabileceği bir ekran olarak tasarlanmıştır. Bu fragment, tarif başlığı, yazar bilgileri, hazırlık süresi, pişirme süresi, malzemeler, talimatlar ve tarif görseli gibi bilgileri kullanıcıya sunar. Ek olarak, tarifi favorilere eklenmesi ve kullanıcı yorumlarının görüntülenmesi gibi etkileşimli özellikler barındırır. Yorum ve tarif detayları, düzenli bir yapı sağlamak için **TabLayout** bileşeni ile iki ayrı sekmede sunulmaktadır.

Fragment, tarife ait detay bilgilerini Firebase Realtime Database'den alır. Tarif bilgileri loadRecipeDetails() metodu kullanılarak yüklenir ve bu bilgiler ilgili arayüz bileşenlerine aktarılır. Tarif yazarına ait profil resmi ve kullanıcı adı gibi ek bilgiler de Firebase'den alınarak, tarifi yazarını tanıttıcı bir bölüm oluşturulmuştur. Tarif görseli Picasso kütüphanesi kullanılarak yüklenir ve bu, resimlerin hızlı ve verimli bir şekilde gösterilmesini sağlar.

Yorum sekmesi, kullanıcıların bir hakkında yorum yapabilmesini ve diğer kullanıcıların yorumlarını görüntüleyebilmesini sağlar. Yorumlar, Firebase'den loadComments() metodu ile yüklenir ve her bir yorum özel bir görünümle ekranda listelenir. Kullanıcı yorum yapmak istediğinde, giriş yapma durumu kontrol edilir ve kullanıcıya bir yorum ekleme alanı sunulur. Giriş yapmamış kullanıcılar, yorum yapmaya çalıştıklarında bir bilgilendirme mesajı ile uyarılır. Eklenen her yorum, tarifi Firebase üzerindeki "comments" düğümüne kaydedilir ve ayrıca yorumu yapan kullanıcının bilgileri de Firebase kullanıcı düğümünde saklanır.

Favori özelliği ise kullanıcının tarifleri favorilerine ekleyip çıkarmasını sağlar. toggleFavorite() metodu kullanılarak, tarif favorilere eklenir veya kaldırılır. Kullanıcının favori durumu Firebase üzerinden kontrol edilerek (checkIfFavorite()), favori ikonunun durumu güncellenir. Favorilere ekleme veya çıkarma işlemleri sırasında kullanıcıya bilgilendirme mesajları sunulur.

Tarif detayları ve yorumlar arasındaki geçiş, **TabLayout** bileşeni ile yönetilir. Kullanıcı tarif detaylarını görmek istediğinde, yorumlar gizlenir ve tarif bilgileri gösterilir. Yorum sekmesi seçildiğinde ise yorumlar görüntülenir ve yorum ekleme alanı aktif hale gelir. Bu düzenleme, kullanıcıların tarifle ilgili bilgileri ve etkileşimleri daha kolay bir şekilde yönetmesini sağlar.

Sonuç olarak, **RecipeDetailFragment**, kullanıcı deneyimini zenginleştiren detaylı bir tarif görüntüleme ve etkileşim

ekranıdır. Hem görsel hem de işlevsel açıdan kullanıcı dostu bir yapı sunar. Firebase Realtime Database ve Picasso gibi modern teknolojiler kullanılarak güçlü bir backend ve hızlı bir arayüz oluşturulmuştur.

b.Kullanılan Veri Modeli

Tarif bilgilerini temsil etmek için Recipe sınıfı kullanılmıştır. Recipe sınıfı, tarifi başlığı, yazarı, görsel bağlantısı, malzemeleri, talimatları, hazırlık ve pişirme süreleri gibi bilgileri içerir. Firebase ile entegre şekilde çalışarak tariflerin düzenli bir şekilde saklanmasını ve uygulama içerisinde kullanılmasını sağlar. Tarif detay ekranında ve tarif listesi ekranında, Recipe veri modeli kullanılarak veriler Firebase'den alınır ve ilgili alanlara yerleştirilir. Bu veri modeli, tariflerin yönetimini kolaylaştıran temel bir yapı sağlar.

c.Firebase Entegrasyonu

Tariflerin ve kullanıcı etkileşimlerinin yönetimi için Firebase Realtime Database ve Firebase Storage kullanılmıştır. RecipeFragment, Firebase'den tarif verilerini çeker ve listeye ekler. RecipeDetailFragment ise seçilen tarifi detaylarını ve yorumlarını Firebase'den alır. Kullanıcılar, tarifleri favorilere eklediklerinde veya yorum yaptıklarında, bu bilgiler Firebase'de gerçek zamanlı olarak saklanır ve güncellenir. Ayrıca, tarif görselleri Firebase Storage'da depolanır ve bağlantıları veritabanında saklanarak uygulamada dinamik bir şekilde yüklenir.

IV.HATALAR VE ÇÖZÜMLER

Uygulama geliştirme sürecinde karşılaşılan hatalar ve bu hataların çözüm yolları, projenin daha sağlam ve işlevsel bir yapıya kavuşmasını sağladı. Bu süreçte özellikle Android geliştirme ve Firebase entegrasyonunda çeşitli sorunlarla karşılaşıldı. Örneğin, "**Android Resource Linking Failed**" hatası, XML dosyalarında eksik veya hatalı tanımlamalar nedeniyle oluştu. Bu hata, drawable klasöründeki eksik görseller, hatalı tema tanımlamaları veya layout dosyalarında yanlış tanımlanan ID'lerden kaynaklandı. Bu sorunun çözümü için XML dosyaları dikkatlice incelendi ve eksik kaynaklar projeye dahil edilerek hatalı tanımlamalar düzeltildi.

Firebase bağlantısı sırasında da bazı sorunlar yaşandı. Özellikle Firebase Authentication ve Realtime Database yapılandırmalarında eksik veya yanlış konfigürasyonlar nedeniyle hatalar alındı. Bu durum, google-services.json dosyasının yanlış yerleştirilmesi veya Firebase'de gerekli izinlerin verilmemesi gibi nedenlerden kaynaklandı. Sorunları gidermek için dosya konumu doğru şekilde düzenlendi ve veritabanı okuma/yazma izinleri yeniden yapılandırıldı. Ayrıca, Firebase veritabanına yapılan sorguların doğru referanslarla çalıştığı doğrulandı.

Navigasyon bileşeni (Navigation Component) ile çalışırken bazı geçişlerde IllegalArgumentException hatasıyla karşılaşıldı. Bu durum, nav_graph.xml dosyasındaki eksik tanımlamalar veya tür uyumsuzluğu nedeniyle oluştu. Bu sorunun çözümünde tüm geçişler gözden geçirildi ve safeArgs kullanılarak tür güvenliği sağlandı. Ayrıca, Fragmentlar arasında doğru veri aktarımı yapılması için gereken düzenlemeler yapıldı.

RecyclerView kullanımı sırasında, Firebase'den alınan verilerin RecyclerView listesine yansıtılmaması problemiyle karşılaşıldı. Bu sorun, adapter'in doğru şekilde güncellenmemesi nedeniyle oluştu. Çözüm olarak,

Firestore'den veri çekme işlemleri sırasında `notifyDataSetChanged()` fonksiyonu eklenerek liste güncellemeleri sağlandı.

Tarif ekleme ekranında kullanıcıların seçtiği resimlerin `ImageView` bileşenine yüklenememesi bir diğer sorun olarak karşılaşıldı. Bu hata, resim URI'sinin yanlış tanımlanmasından kaynaklandı. Resim seçimi işlemi için kullanılan kod tekrar düzenlenerek URI değerleri doğru şekilde ayarlandı ve `ImageView` bileşenine yansıtılması sağlandı.

Tarif detay ekranında yorumların yüklenmesi sırasında Firestore'den gelen verilerin eksik olması veya hatalı referanslar nedeniyle yorumlar görüntülenemedi. Bu durumun çözümü için yorumların çekildiği veritabanı referansları kontrol edildi ve yorumların olmadığı durumlarda kullanıcıya bilgilendirme mesajı gösterildi. Ayrıca, yorum ekleme işlemi sırasında Firestore Authentication ile kullanıcı doğrulama işlemleri geliştirildi.

Bu hataların çözüm süreci, hem uygulamanın kararlılığını artırdı hem de projeye yönelik değerli öğrenme deneyimleri sundu. Tüm bu sorunlar giderildikten sonra uygulama, kullanıcı dostu bir yapıya kavuştu ve güvenilir bir şekilde çalışması sağlandı.

V.SONUÇ

Uygulamanın geliştirilme süreci, kullanıcı deneyimi, performans ve kullanılabilirlik açısından olumlu sonuçlar doğurmuştur. Geliştirilen uygulama, yemek tariflerini paylaşmayı ve keşfetmeyi kolaylaştıran kullanıcı dostu bir arayüz ve yüksek performanslı bir altyapı sunmaktadır. Kullanıcıların uygulama içindeki deneyimleri detaylı şekilde incelendiğinde şu sonuçlara ulaşılmıştır:

- **Yüksek Performans:** Uygulama, hızlı yükleme süreleri ve akıcı geçişler ile kullanıcıların beklentilerini karşılamaktadır. Ana sayfa, tarif detayları ve favoriler gibi bölümler arasında gezinme sırasında herhangi bir gecikme yaşanmamış, performanslı bir deneyim sunulmuştur. Firestore ile gerçekleştirilen veri alışverişi sayesinde tariflerin hızlı bir şekilde yüklenmesi sağlanmıştır.
- **Kullanıcı Dostu Arayüz:** Arayüz tasarımı, kullanıcıların tariflere kolayca erişebileceği ve işlemlerini rahatça gerçekleştirebileceği şekilde düzenlenmiştir. Kategorilere ayrılmış düzen, kullanıcıların aradıkları tariflere hızlıca ulaşmasını sağlamış; net ve anlaşılır bir navigasyon yapısı oluşturulmuştur. Özellikle tarif detay ekranındaki sekmeli yapı, tarif bilgilerine ve yorumlara erişimi kolaylaştırmıştır.
- **Duyarlılık ve İstikrar:** Uygulama, farklı cihazlarda ve internet bağlantı hızlarında istikrarlı bir şekilde çalışmaktadır. Düşük bağlantı hızlarında bile Firestore'den veri çekme işlemleri sırasında gecikmeler minimum düzeyde tutulmuştur. Uygulamanın bu duyarlılığı, kullanıcıların kesintisiz bir deneyim yaşamasını sağlamıştır.
- **Optimize Edilmiş Kullanıcı Deneyimi:** Tarif ekleme, favorilere tarif ekleme ve yorum yapma gibi işlemler, kullanıcıların beklentilerine uygun bir şekilde optimize edilmiştir. Kullanıcılar, istedikleri tarife kolayca ulaşabilir, tarifleri

favorilerine ekleyebilir ve tariflere yorum yaparak diğer kullanıcılarla etkileşime geçebilir. Bu özellikler, kullanıcıların uygulama içindeki deneyimini daha keyifli ve etkili hale getirmiştir.

Sonuç olarak, uygulama hem kullanıcıların ihtiyaçlarını karşılayan hem de kolay kullanılabilir bir yapıya sahiptir. Bu süreçte kullanılan modern teknolojiler ve düzenli kod yapısı, uygulamanın performansını artırmış ve kullanıcıların deneyimlerini iyileştirmiştir. Elde edilen bu sonuçlar, uygulamanın temel hedeflerine ulaştığını ve gelecekte yapılabilecek geliştirmeler için sağlam bir temel oluşturduğunu göstermektedir.

VI. REFERANSLAR

- <https://www.youtube.com/watch?v=m2G55JsCffg>
- <https://developer.android.com/studio/intro>
- <https://developer.android.com/build>
- <https://developer.android.com/tools>
- <https://www.udemy.com/course/android-mobil-uygulama-gelistirme-egitimi-kotlin/learn/lecture/20248534?start=0#overview>
- <https://www.udemy.com/course/android-bootcamp-program-kotlin/learn/lecture/39540138?start=0#overview>
-

GitHubLinki:

- <https://github.com/tgcx12/mobiluygulamam>

Google Drive Linki:

- https://drive.google.com/drive/u/1/folders/1-sKW8OrGnd6G-zx_vgUKoEOQYq5gNGpE
-

