

**MOBİL PROGRAMLAMA**

**PROJE ÖDEVİ**

|  |  |
| --- | --- |
| İsim-Soyisimler | Eray Berk DALKIRAN  Yiğit ÇETİNKAYA |
| Numaralar | 201817007  201817017 |
| Fakülte | Teknoloji Fakültesi |
| Bölüm | Bilgisayar Mühendisliği |
| Ders | BMT-342 |
| Öğretmenler | Doç. Dr. İbrahim Alper DOĞRU  Arş. Gör. Ömer Ayberk ŞENCAN |

**İÇİNDEKİLER**

[1. Özet 4](#_Toc136984333)

[2. Tasarım ve Mimari 4](#_Toc136984334)

[2.1. AppCompatActivity 4](#_Toc136984337)

[2.2. Intent 4](#_Toc136984338)

[2.3. AsyncTask 4](#_Toc136984339)

[2.4. Bundle 5](#_Toc136984340)

[2.5. View 5](#_Toc136984341)

[2.6. Adapter 5](#_Toc136984342)

[2.7. AdapterView 5](#_Toc136984343)

[2.8. ArrayAdapter 5](#_Toc136984344)

[2.9. Button: 5](#_Toc136984345)

[2.10. EditText 6](#_Toc136984346)

[2.11. Spinner 6](#_Toc136984347)

[2.12. TextView 6](#_Toc136984348)

[2.13. Toast 6](#_Toc136984349)

[3. Görev Dağılımı 8](#_Toc136984350)

[3.1. Frontend Geliştirici (ERAY BERK DALKIRAN) 8](#_Toc136984352)

[3.2. Backend Geliştirici (YİĞİT ÇETİNKAYA) 8](#_Toc136984353)

[4. Zaman Yönetimi 9](#_Toc136984354)

[4. Kullanılan Teknolojiler 10](#_Toc136984355)

[4.1. Android Studio 10](#_Toc136984356)

[4.2. Java 10](#_Toc136984357)

[4.3. XML 10](#_Toc136984358)

[4.4. Microsoft SQL Server (MSSQL) 10](#_Toc136984359)

[5. Özellikler ve İşlevler 11](#_Toc136984360)

[5.1. Kullanıcı Profilleri 11](#_Toc136984361)

[5.2. Çeşitli Bağış Seçenekleri 11](#_Toc136984362)

[5.3. Haritalar 11](#_Toc136984363)

[6. Son Kullanıcı Faydaları 11](#_Toc136984364)

[6.1. Kolay ve Hızlı Bağış İşlemi 11](#_Toc136984365)

[6.2. İhtiyaç Sahiplerinin Erişimi 12](#_Toc136984366)

[6.3. Verimli Kaynak Yönetimi 12](#_Toc136984367)

[6.4. Kullanıcı Dostu Arayüz 12](#_Toc136984368)

[6.5. Bağış Kültürü 12](#_Toc136984369)

[7. SQL Tablolarının Yapıları 13](#_Toc136984370)

[8. Uygulama İçi Görseller 14](#_Toc136984371)

# 1. Özet

"Charity Bridge" uygulaması, ihtiyaç sahipleriyle bağışçıları bir araya getirmeyi amaçlayan bir bağış uygulamasıdır. Uygulama, ihtiyaç sahiplerinin farklı bağış paketlerine erişebildiği ve bağış yapabilme imkanına sahip olduğu bir platform sunmaktadır. Proje, iki kişi tarafından geliştirilmiş olup, Android Studio ve Java mobil programlama dili kullanılarak gerçekleştirilmiştir. Figma üzerinde adım adım planlama yapılarak backend ve frontend görevleri ayrıştırılmış ve MSSQL veritabanı kullanılmıştır. Proje kısa bir süre içerisinde başarıyla tamamlanmıştır.

# 2. Tasarım ve Mimari

Charity Bridge uygulamasının tasarımı ve mimarisi, çeşitli Android bileşenleri ve kütüphaneleri kullanılarak oluşturulmuştur. Aşağıda proje geliştirme sürecinde yer alan bazı önemli uygulama bileşenleri ve kütüphanelerinden bazıları yer almaktadır:



## AppCompatActivity

Charity Bridge uygulamasının tüm aktiviteleri, AppCompatActivity sınıfından türetilmiştir. Bu sınıf, Android uygulamalarında kullanılan standart aktivite sınıfıdır ve uygulamanın etkileşimli özelliklerini destekler.

## Intent

Uygulama içinde farklı aktiviteler arasında geçiş yapmak, veri iletmek veya başka uygulamaları başlatmak için Intent sınıfı kullanılmıştır. Intent, aktiviteler arası iletişimi sağlayan bir yapıdır.

## AsyncTask

Uygulama içinde arka planda yapılan işlemleri yönetmek için AsyncTask sınıfı kullanılmıştır. Bu sınıf, uzun süren işlemleri UI (Kullanıcı Arayüzü) üzerinde yapılabilir hale getirmek için kullanılan bir mekanizmadır.

## Bundle

Aktiviteler arasında veri taşımak için Bundle sınıfı kullanılmıştır. Bundle, anahtar-değer çiftlerini saklamak için kullanılan bir veri yapısıdır.

## View

Kullanıcı arayüzündeki temel yapı elemanlarından biridir. Uygulama içinde farklı görünümleri oluşturmak ve kullanıcı etkileşimini sağlamak için kullanılmıştır.

## Adapter

Veri listelerini görüntülemek ve bu verilere erişim sağlamak için Adapter sınıfı kullanılmıştır. Örneğin, bağış paketlerini listeleyen bir ListView veya RecyclerView için bir Adapter kullanılmış olabilir.

## AdapterView

Adapter ile veri bağlama ve görüntüleme işlemlerini yöneten bir arayüzdür. Örneğin, ListView veya Spinner gibi görüntüleme bileşenleri AdapterView arayüzünü uygular.

## ArrayAdapter

ArrayAdapter, bir dizi veriyi AdapterView'e bağlamak için kullanılan bir sınıftır. Örneğin, ListView'e veri listesini bağlamak için ArrayAdapter kullanılmış olabilir.

## Button:

Kullanıcıların tıklama eylemlerini algılamak için Button bileşeni kullanılmıştır. Kullanıcı bağış yapma işlemi gibi etkileşimleri gerçekleştirmek için kullanılan bir bileşendir.

## EditText

Kullanıcının metin girişi yapmasını sağlayan bir bileşendir. Bağış yapılacak miktar gibi kullanıcı girişine ihtiyaç duyan alanlarda kullanılmıştır.

## Spinner

Kullanıcının bir listeden bir öğeyi seçmesini sağlayan bir bileşendir. Örneğin, bağış paketi seçimi için Spinner kullanılmış olabilir.

## TextView

Metin göstermek için kullanılan bir bileşendir. Kullanıcıya bağış yapılan miktarı veya diğer bilgileri göstermek için kullanılmıştır.

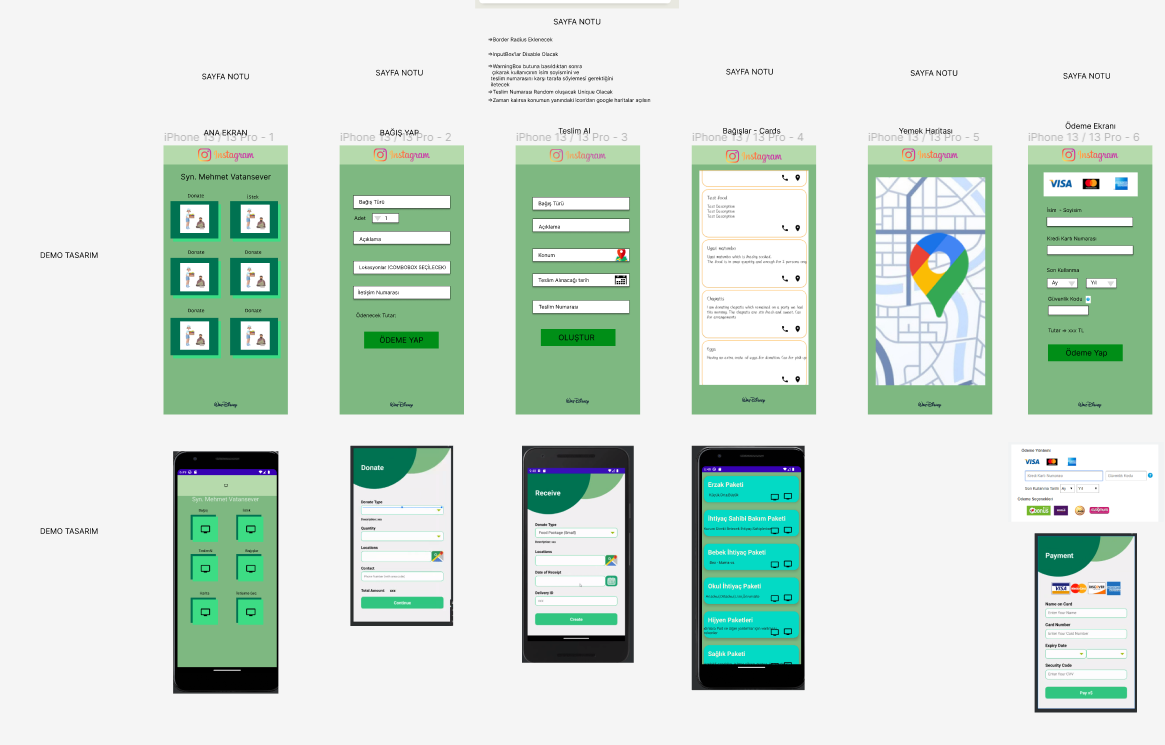
## Toast

Kullanıcıya geçici mesajlar göstermek için kullanılan bir mekanizmadır. Örneğin, bağış işlemi tamamlandığında kullanıcıya bir Toast mesajı gösterilebilir.

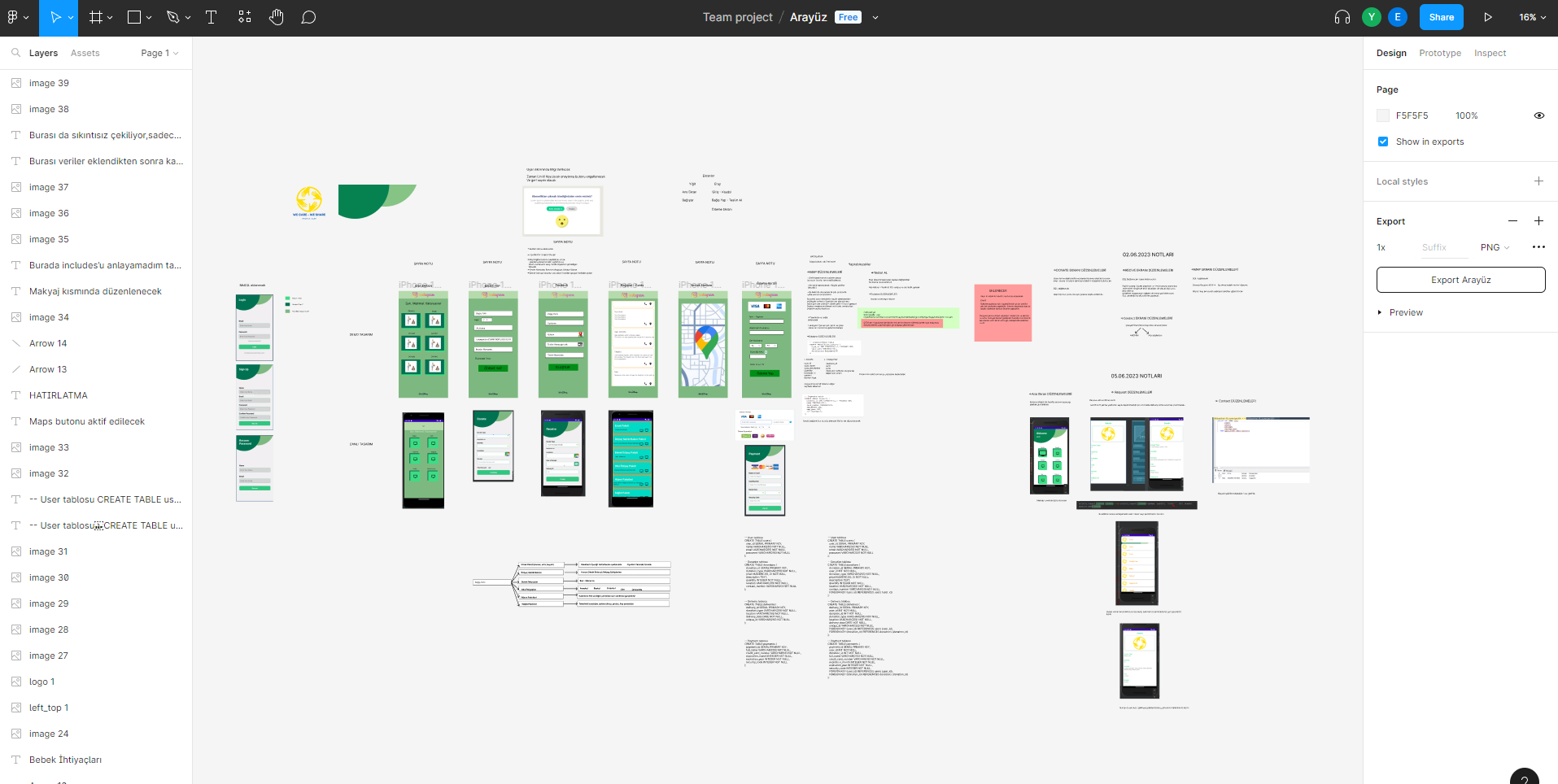
Bu listedeki bileşenler ve kütüphaneler, Charity Bridge uygulamasının kullanıcı arayüzünün oluşturulmasında ve kullanıcı etkileşimlerinin yönetilmesinde önemli bir rol oynamıştır. Bu sayede kullanıcılar uygulama üzerinden bağış yapabilir, bağışları görüntüleyebilir ve ihtiyaç sahipleri bu bağışlara erişebilir.

Aşağıda yer alan resimlerde Figma üzerinde yapılan genel çalışmaları ve geliştirme aşamasında yer alan notlarla birlikte anlık ifadeleri görebilmekteyiz.

**Görsel 1 ---**



**Görsel 2 ---**

****

# 3. Görev Dağılımı

Charity Bridge uygulamasının geliştirilmesi sürecinde, görevlerin etkin bir şekilde dağıtılması ve takibi için aşağıdaki görev dağılımı gerçekleştirilmiştir:



## Frontend Geliştirici (ERAY BERK DALKIRAN)

• Kullanıcı arayüzünün tasarımını gerçekleştirdi.

• XML dosyalarında kullanıcı etkileşimlerini sağlayan düğmeleri, giriş alanlarını ve diğer bileşenleri oluşturdu.

• Arayüzün görüntülenmesi için layout yapılarını düzenledi.

• Kullanıcı etkileşimlerini işleyen Java kodunu yazdı.

• Arayüzdeki verilerin düzgün şekilde gösterilmesi ve kullanıcıya geri bildirimler sağlandı.

• Yapılan sayfalar olarak; kullanıcının etkileşime girebileceği sayfaların tasarımlarıyla fiziksel olarak kullanabilir hale getirilmesini görev aldı ve backend kısmına bu aşamaların aktif bir şekilde kullanılmasını sağlaması için sayfaların iletilmesini sağladı.

## Backend Geliştirici (YİĞİT ÇETİNKAYA)

• Veritabanı işlemlerini yönetmek için MSSQL kullanarak backend kodunu yazdı.

• Bağış verilerini saklamak ve ihtiyaç sahiplerinin erişimini sağlamak için gerekli tabloları oluşturdu.

• Bağış yapıldığında veritabanına kaydedilen işlemleri gerçekleştirdi.

• Veritabanı bağlantısını sağladı ve güvenli bir şekilde veri alışverişi yapılmasını sağladı.

• Backend kodunda hata yönetimi ve güvenlik önlemlerini uyguladı.

• Yapılan sayfalar olarak; frontend geliştiricisinden aldığı sayfalardan; gerekli fonksiyonların oluşturulması ve DataBase tarafında yönetilmesini sağlayacak kodların entegrasyonunu sağlaması üzerine genel yapıyı oluşturdu ve uygulamayı teste sokarak olası risk durumlarını giderdi. Görev dağılımı sayesinde her bir geliştirici, kendi alanında sorumluluk üstlenerek projenin başarılı bir şekilde tamamlanmasını sağladı. Sürekli iletişim ve işbirliği içinde çalışarak birlikte hedeflere ulaşıldı.

Görev dağılımı sayesinde her bir geliştirici, kendi alanında sorumluluk üstlenerek projenin başarılı bir şekilde tamamlanmasını sağladı. Sürekli iletişim ve işbirliği içinde çalışarak birlikte hedeflere ulaşıldı.

# 4. Zaman Yönetimi

Charity Bridge projesinde, etkili zaman yönetimi için aşağıdaki stratejiler kullanıldı:

• Proje yönetimi ve iş takibi için GitHub kullanıldı. Projeye ait görevler, iş birimleri (branch) ve ilerleme, GitHub üzerindeki proje deposunda (repository) takip edildi. Bu sayede her bir geliştirici, kendi görevlerini takip edebilir ve projenin ilerlemesini kontrol edebilir.

• Başlangıçta bir zaman çizelgesi oluşturuldu ve her bir görevin tahmini süreleri belirlendi. Görevlerin tamamlanması ve ilerleme takibi için düzenli olarak toplantılar yapıldı.

• Geliştirme sürecinde her bir görev, önceliklere göre sıralandı. Öncelikli görevler öncelikle tamamlandı ve ardından diğer görevlere geçildi.

• İşbirliği ve iletişim, zaman yönetimi açısından kritik bir rol oynadı. Her bir geliştirici, birbirleriyle sürekli olarak iletişim halinde oldu, sorunları tartıştı ve gerektiğinde birlikte çözümler üretti.

• İlerleme ve zaman yönetimi düzenli olarak kontrol edildi. Gerektiğinde zaman çizelgesi güncellendi ve ayarlamalar yapıldı.

Bu zaman yönetimi stratejileri sayesinde, Charity Bridge projesi zamanında ve başarıyla tamamlandı. Düzenli iletişim ve işbirliği ile projenin hedefleri doğrultusunda ilerlenerek verimli bir geliştirme süreci sağlandı.

# Kullanılan Teknolojiler

## Android Studio

Uygulamanın geliştirilmesi sürecinde IDE olarak Android Studio kullanıldı. Bu platform, kod yazmayı, hata ayıklamayı ve uygulamanın genel performansını izlemeyi kolaylaştırır. Ayrıca, geniş araç ve özellik yelpazesiyle geliştiricilere çok yönlü bir geliştirme ortamı sağlar.

## Java

Uygulamanın temel kodlama dili olarak Java kullanıldı. Java'nın hızı, güvenilirliği ve geniş kütüphane desteği, mobil uygulama geliştirmede mükemmel bir seçimdir. Bu dil, ayrıca geliştiricilere yazılımın karmaşık bölümlerini etkili bir şekilde yönetme ve optimize etme imkanı sunar.

## XML

Kullanıcı arayüzlerini tasarlamak için XML işaretleme dili seçildi. XML, kullanıcı etkileşimlerini ve arayüz tasarımlarını yönetmek için özelleştirilebilir bir yapı sağlar. Ayrıca, XML'nin esnekliği, kullanıcı deneyimini iyileştirmek için ince ayarların yapılmasına olanak sağlar.

## Microsoft SQL Server (MSSQL)

Veritabanı yönetim sistemi olarak Microsoft SQL Server (MSSQL) kullanıldı. MSSQL, işlem gücü, güvenilirlik ve güvenlik özellikleriyle bilinen güçlü bir veritabanı yönetim sistemidir. Ayrıca, büyük veritabanları ve iş yükleri için mükemmel bir seçimdir ve karmaşık veri sorgulama ve analizini kolaylaştıran kapsamlı araçları ve özellikleri vardır. Bu proje için MSSQL, verilerin düzgün bir şekilde organize edilmesi, saklanması ve alınması için kullanıldı.

# Özellikler ve İşlevler

## Kullanıcı Profilleri

Kullanıcıların kendi profillerini oluşturabilmesini sağlar. Bu sayede, kullanıcılar bağış geçmişlerini takip edebilir, ayarlarını özelleştirebilir ve bağışlarını yönetebilirler.

## Çeşitli Bağış Seçenekleri

Kullanıcıların çeşitli ihtiyaçlara yönelik bağış yapabilmelerine olanak sağlar. Bu; erzak bağışları, yiyecek, giyecek, eğitim malzemeleri vb. bağış türlerini içermektedir.

## Haritalar

İhtiyaç sahipleri, uygulama içerisindeki haritalar üzerinden erzaklarını nereden alabileceklerini kolaylıkla görebilir ve akabinde eyleme geçebilir.

# Son Kullanıcı Faydaları

Charity Bridge uygulaması, kullanıcılarına bir dizi fayda sağlamayı amaçlamaktadır. Bu faydalar, bağış yapmak isteyen bireyler ve ihtiyaç sahibi bireyler arasında etkili bir köprü oluşturarak gerçekleştirilir. Aşağıda, uygulamanın sağladığı son kullanıcı faydalarını açıklayan bir rapor yer almaktadır:

## Kolay ve Hızlı Bağış İşlemi

Charity Bridge uygulaması, kullanıcılarına kolay ve hızlı bir bağış yapma imkanı sunar. Kullanıcılar, uygulama aracılığıyla birkaç adımda bağış yapabilirler. Bağış paketleri ve ödeme seçenekleri kullanıcılara sunulur, böylece istedikleri miktar ve bağış seçeneğiyle hızlıca bağış yapabilirler.

## İhtiyaç Sahiplerinin Erişimi

Uygulama, ihtiyaç sahiplerine de önemli bir fayda sağlar. İhtiyaç sahipleri, Charity Bridge uygulaması üzerinden bağış paketlerine erişebilir ve bu paketlerden kendileri için uygun olanı seçebilirler. Bu sayede ihtiyaç sahipleri, ihtiyaçlarını karşılamak için daha fazla kaynağa ulaşabilir ve destek alabilirler.

## Verimli Kaynak Yönetimi

Charity Bridge uygulaması, bağışçılar ve ihtiyaç sahipleri arasındaki bağlantıyı sağlayarak kaynakların daha verimli bir şekilde yönetilmesini sağlar. Bağışçılar, ihtiyaç sahiplerinin gerçek ihtiyaçlarına yönlendirilir ve bu sayede kaynaklar daha etkili bir şekilde kullanılır. Bu da toplumda daha fazla insana yardım edilmesini sağlar.

## Kullanıcı Dostu Arayüz

Uygulama, kullanıcı dostu bir arayüze sahiptir. Basit ve sezgisel bir tasarıma sahip olan uygulama, kullanıcıların kolaylıkla navigasyon yapabilmesini ve istedikleri bilgilere hızlı bir şekilde erişebilmesini sağlar. Böylece kullanıcılar, bağış yapma sürecinde herhangi bir zorluk yaşamadan uygulamayı kullanabilirler.

## Bağış Kültürü

Charity Bridge uygulaması, kullanıcılarına kolaylık, verimlilik ve etkinlik sağlayarak toplumda bağış kültürünün yayılmasına katkıda bulunur. Bağış yapmak isteyen bireylerin ve ihtiyaç sahibi bireylerin bir araya gelmesini sağlayarak toplumsal yardımlaşmanın artmasına katkıda bulunur. Bu sayede, uygulama son kullanıcılar için önemli bir değer ve fayda sunar.

# SQL Tablolarının Yapıları

1. SQL Tablolarının Yapıları:
2. Users Tablosu:
3. user\_id: int veri tipinde, boş olmayan bir kullanıcı kimliği.
4. name: 255 karaktere kadar olan varchar veri tipinde, boş olmayan bir kullanıcı adı.
5. email: 255 karaktere kadar olan varchar veri tipinde, boş olmayan bir e-posta adresi.
6. password: 255 karaktere kadar olan varchar veri tipinde, boş olmayan bir şifre.
7. phone\_number: 255 karaktere kadar olan varchar veri tipinde, boş olmayan bir telefon numarası.
8. Receive Tablosu:
9. receive\_id: int veri tipinde, boş olmayan bir alım kimliği.
10. donation\_id: int veri tipinde, bir bağış kimliği.
11. date\_of\_receipt: 255 karaktere kadar olan varchar veri tipinde, bir alımın tarihi.
12. delivery\_id: int veri tipinde, bir teslimat kimliği.
13. Payments Tablosu:
14. payment\_id: int veri tipinde, boş olmayan bir ödeme kimliği.
15. user\_id: int veri tipinde, bir kullanıcı kimliği.
16. card\_number: 16 karaktere kadar olan varchar veri tipinde, bir kart numarası.
17. exp\_month: int veri tipinde, bir son kullanma ayı.
18. exp\_year: int veri tipinde, bir son kullanma yılı.
19. cvv: 4 karaktere kadar olan varchar veri tipinde, bir CVV numarası.
20. Locations Tablosu:
21. location\_id: int veri tipinde, boş olmayan bir konum kimliği.
22. city: 255 karaktere kadar olan varchar veri tipinde, bir şehir.
23. branch: 255 karaktere kadar olan varchar veri tipinde, bir şube.
24. latitude: 255 karaktere kadar olan varchar veri tipinde, bir enlem değeri.
25. longitude: 255 karaktere kadar olan varchar veri tipinde, bir boylam değeri.
26. DonationTypes Tablosu:
27. type\_id: int veri tipinde, boş olmayan bir bağış türü kimliği.
28. type\_name: 255 karaktere kadar olan varchar veri tipinde, bir bağış türü adı.
29. description: 255 karaktere kadar olan varchar veri tipinde, bir bağış türü açıklaması.
30. price: int veri tipinde, bir bağış türü fiyatı.
31. Donations Tablosu:
32. donation\_id: int veri tipinde, boş olmayan bir bağış kimliği.
33. type\_id: int veri tipinde, bir bağış türü kimliği.
34. user\_id: int veri tipinde, bir kullanıcı kimliği.
35. location\_id: int veri tipinde, bir konum kimliği.
36. quantity: int veri tipinde, bir bağış miktarı.
37. price: int veri tipinde, bir bağış fiyatı.
38. Contacts Tablosu:
39. contact\_id: int veri tipinde, boş olmayan bir iletişim kimliği.
40. name: 255 karaktere kadar olan varchar veri tipinde, bir iletişim adı.
41. phone: 255 karaktere kadar olan varchar veri tipinde, bir iletişim telefon numarası.

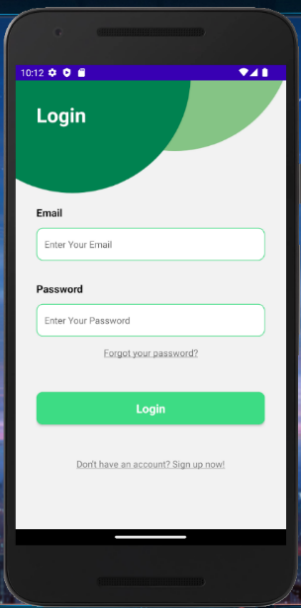
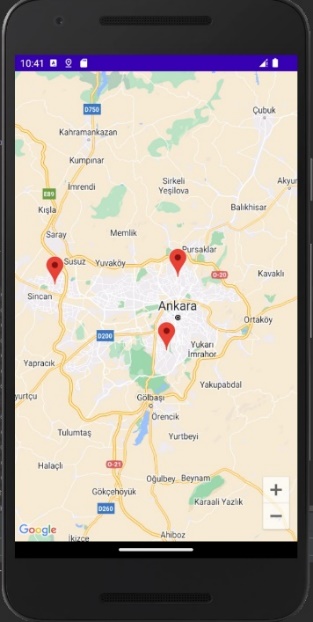
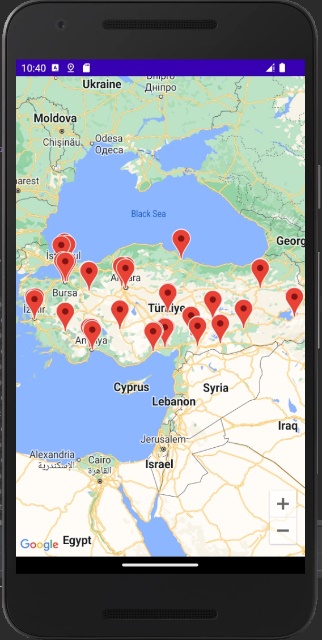
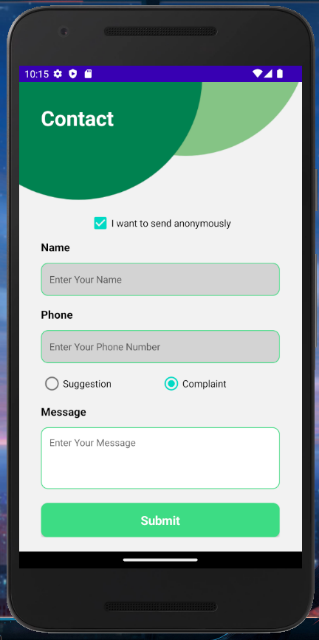
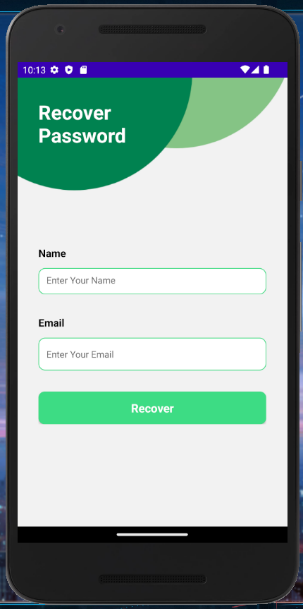
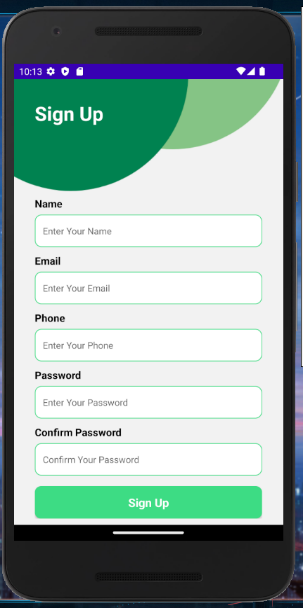
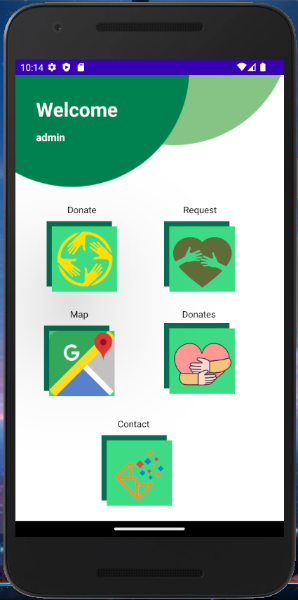
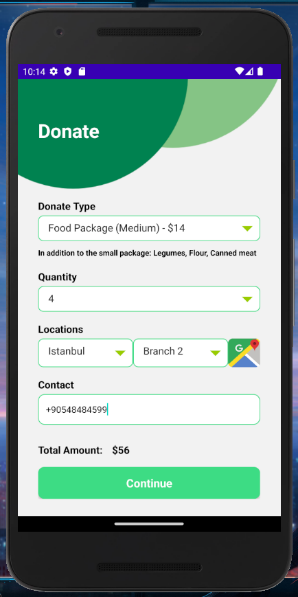
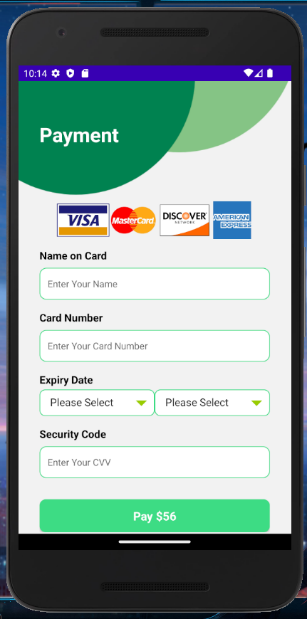
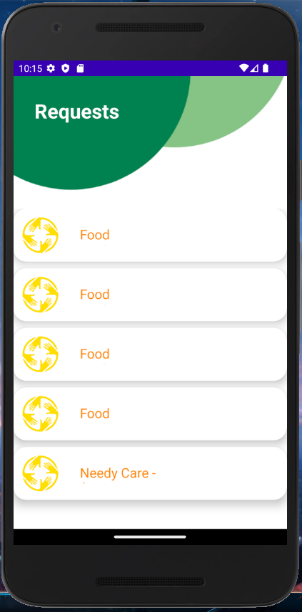
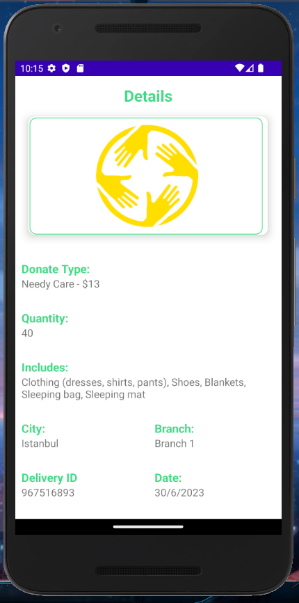
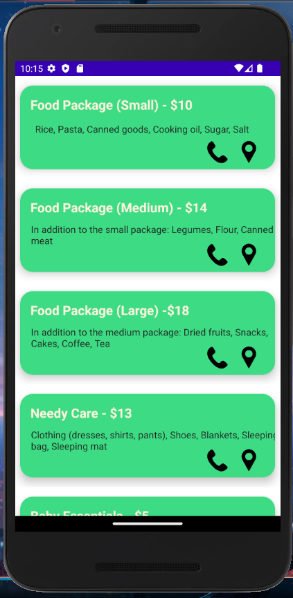
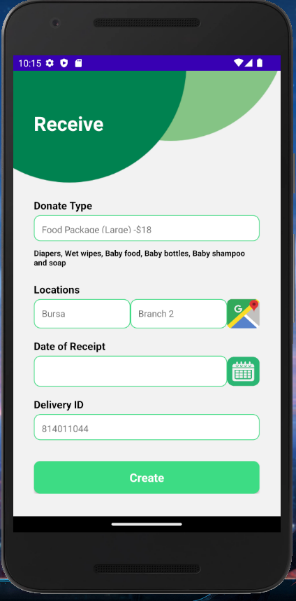
id: int veri tipinde, boş olmayan bir iletişim kimliği.

name: 100 karaktere kadar olan varchar veri tipinde, bir iletişim adı.

1. phone: 20 karaktere kadar olan varchar veri tipinde, bir iletişim telefon numarası.
2. message: 2147483647 karaktere kadar olan text veri tipinde, bir iletişim mesajı.

message\_type: 20 karaktere kadar olan varchar veri tipinde, bir iletişim mesajı türü.

# Uygulama İçi Görseller

# Uygulama Toplantıları

