

T.C.
ONDOKUZ MAYIS ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



Yazılım Mühendisliği Laboratuvarı
Kütüphane Otomasyonu Gereksinim Raporu
AHMET GÜNEŞ
19060387

İçindekiler

1	Giriş.....	3
1.1	Amaç.....	3
1.2	Kapsam	3
1.3	Tanımlar ve Kısaltmalar	3
1.3.1	Terim ya da Kısaltma	3
1.3.2	Tanım	3
1.3.	Dokümana Genel Bakış	3
2.	Genel Tanıtım	4
2.1.	Ürüne Bakış.....	4
2.2.	Mimari Şema.....	4
2.2.1.	Model (Veri Modeli)	4
2.2.2	View (Kullanıcı Arayüzü)	4
2.2.3.	Controller (Kontrolcü)	5
2.3.	Veritabanı Tasarımı	5
2.4.	Varsayımlar ve Bağımlılıklar	5
2.4.1.	Platrom Bağımlılıkları:.....	6
2.4.2.	Üçüncü Taraf Kütüphaneler:	6
2.4.3.	Donanım ve Yazılım Bağımlılıkları.....	6
2.4.4.	Ağ Bağlantısı Bağımlılıkları:	6
2.4.5.	Kullanıcı Davranışı ve Kullanılabilirlik Bağımlılıkları.....	6
3.	Fonksiyonel Gereksinimler	6
3.1.	Kullanıcı İşlevselliği:.....	6
3.1.1.	Üye İşlevselliği	6
3.1.2.	Yönetici İşlevselliği	7
3.2.	Sistem İşlevselliği	7
3.2.1.	Veri Yönetimi	7
3.2.2.	Mesajlaşma Sistemi	7
4.	Kullanılabilirlik Gereksinimleri	7
4.1.	Arayüz Tasarımı	8
5.	Kullanım – Durum Diyagramları	10
5.1.	Kitap Rezervasyonu.....	10
5.2.	Yönetici Kitap Ekleme.....	11
5.3.	İletişim	11
6.	Yardımcı Bilgiler	12
7.	Referanslar.....	12

Kütüphane Otomasyonu Yazılım Gereksinimleri Tanıtımı

1 Giriş

1.1 Amaç

Bu kütüphane otomasyon uygulaması, kullanıcıların kütüphane işlemlerini daha etkin ve verimli bir şekilde yönetmeyi amaçlar. Kullanıcılar, kütüphane kaynaklarını kolayca bulabilir, ödünç alabilir, iade edebilir ve kişisel bilgilerini güvenli bir şekilde yönetebilirler. Uygulama, kullanıcı dostu arayüzü ve kapsamlı işlevleriyle kullanıcıların ihtiyaç duydukları bilgilere hızlıca erişmelerini sağlar, böylece kütüphane süreçlerini optimize etmeyi hedefler. Ayrıca, yönetici paneli aracılığıyla kütüphane envanterine eklenen kitapları sisteme kolayca entegre etme ve kitap takibi gibi kolaylıklar sunar.

1.2 Kapsam

Bu uygulama, kütüphane kaynaklarına erişimi artırarak kullanıcıların bilgiye daha hızlı ve verimli bir şekilde ulaşmalarını sağlamayı amaçlamaktadır. Kullanıcılar, çeşitli arama ve filtreleme seçenekleriyle istedikleri kitapları kolayca bulabilir ve rezervasyon yapabilirler. Ayrıca, yönetici paneli kütüphane personelinin envanteri daha etkin bir şekilde yönetmesine ve mevcut kaynakları izlemesine olanak tanır. Bu şekilde, kullanıcılar daha kolay erişilebilir bir kütüphane deneyimi yaşarken yöneticiler de kaynakları verimli bir şekilde yönetebilirler.

1.3 Tanımlar ve Kısaltmalar

1.3.1 Terim ya da Kısaltma	1.3.2 Tanım
Envanter	Envanter, bir işletmenin varlıklarının listesidir.
DB	Database (Veri Tabanı)
ER	Entity Relationship (Varlık İlişki)
MVC	Model, View, Controller
RN	React Native

1.3. Dokümana Genel Bakış

Bu belge, mobil kütüphane uygulamasının gereksinimlerini, tasarımını ve mimarisini detaylı olarak tanımlamaktadır. Bu bölümde, projenin genel amacı, kapsamı ve referanslar hakkında bilgi verilmiştir. Mobil kütüphane uygulamasının genel özellikleri ve işlevselliği tanımlanmış, proje gereksinimleri ayrıntılı olarak incelenmiş ve fonksiyonel gereksinimler, performans gereksinimleri, kullanılabilirlik gereksinimleri ve diğer önemli gereksinimler listelenmiştir. Uygulamanın kullanıcı arayüzü, mimari şeması ve diğer tasarım unsurları açıklanmıştır. Bu doküman, projenin başarılı bir şekilde geliştirilmesine yardımcı olmak için oluşturulmuştur.

2. Genel Tanıtım

2.1. Ürüne Bakış

Kütüphane Otomasyonu Mobil Uygulaması, kütüphane yönetimi için bir çözüm sunar. Basit ve anlaşılabilir arayüzü hem kütüphane personelinin hem de kullanıcıların ihtiyaçlarını karşılamayı amaçlar. Bu ürün, kitap arama ve ödünç alma gibi temel işlevlerin yanı sıra, kullanıcıların kütüphane envanterine erişmesini ve etkileşimde bulunmasını sağlar. Kapsamlı kitap veritabanı sayesinde, kullanıcılar geniş bir kitap koleksiyonunda arama yapabilir ve istedikleri kitapları kolayca bulabilirler. Profil yönetimi özelliği ile kullanıcılar kütüphane kartlarını yönetebilir ve güncelleyebilirler. Ayrıca, kitap arama ve rezervasyon modülü sayesinde, kullanıcılar istedikleri kitapları bulabilir ve rezerve edebilirler. Entegre ödünç alma işlem süreci ile kitapları ödünç almak ve iade etmek daha kolay hale gelir.

2.2. Mimari Şema

Bu mimari şema, mobil kütüphane uygulamasının Model-View-Controller (MVC) tasarım desenini temel aldığını açıkça göstermektedir. MVC, uygulamanın yapısını organize etmek ve farklı sorumlulukları ayırmak için kullanılan bir yaklaşımdır. Uygulamanın her bir bileşeni, Model, View ve Controller olarak tanımlanmıştır. Aşağıda, MVC mimari şemasıyla uygulamanın ana bileşenlerini ve bunların işlevlerini gösteren bir açıklama sunulmaktadır.

2.2.1. Model (Veri Modeli): Model, uygulamanın veri tabanı işlemlerini ve iş mantığını içeren katmandır. Bu katmanda, veri tabanı işlemleri gerçekleştirilir ve kullanıcı talepleri işlenir. Kütüphane verileri bu katmanda yönetilir ve gerektiğinde güncellenir. Örneğin, kullanıcıların ödünç aldığı kitapların kaydedilmesi ve iade tarihlerinin kontrol edilmesi gibi işlemler bu katmanda gerçekleştirilir.

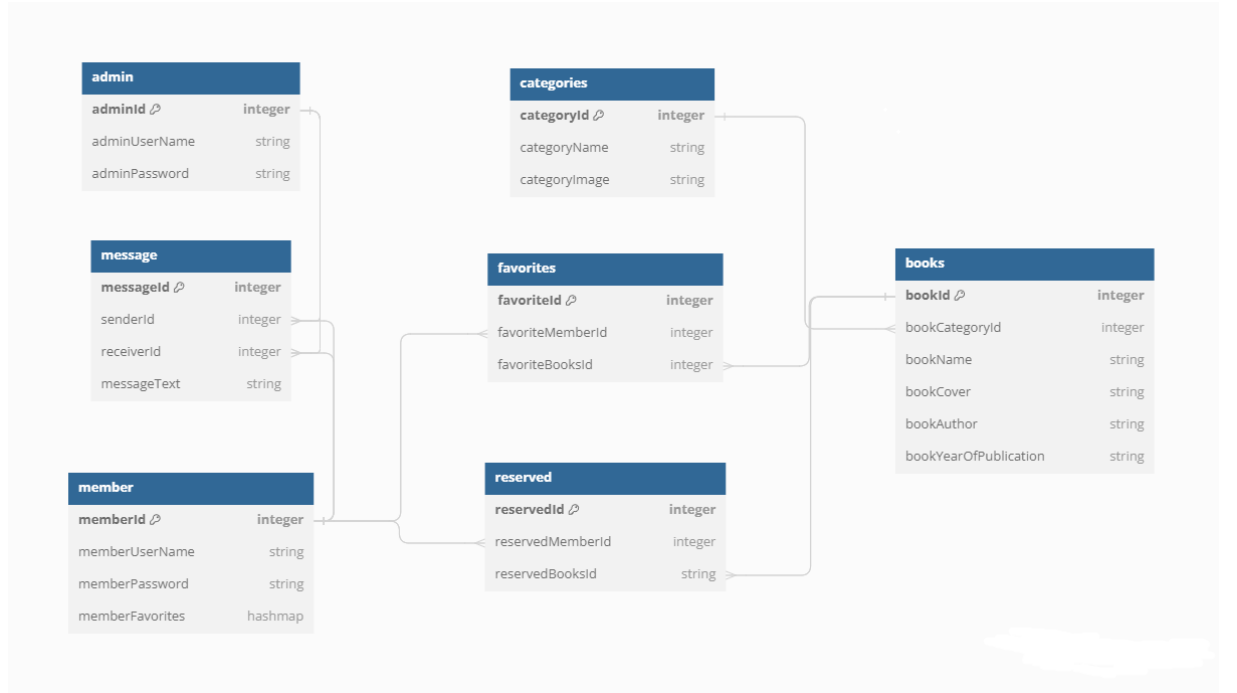
2.2.2 View (Kullanıcı Arayüzü): View, kullanıcıların uygulamayla etkileşimde bulunduğu kısmı temsil eder. Mobil uygulama, kullanıcıların kitapları araması, ödünç alması, iade etmesi ve hesaplarını yönetmesi için optimize edilmiştir. Bu katmanda, kullanıcı arayüzü tasarlanır ve kullanıcıya görüntülenir.

2.2.3. Controller (Kontrolör): Controller, Model ve View arasındaki etkileşimi yöneten ve uygulamanın iş mantığını kontrol eden katmandır. Kullanıcı talepleri Controller tarafından alınır, uygun Model işlemleri çağrılır ve sonuçlar View katmanına iletilir. Örneğin, kullanıcının bir kitabı ödünç almak istemesi durumunda, bu talep Controller tarafından alınır, Model tarafından işlenir ve sonuç View katmanında kullanıcıya gösterilir.

2.2.4. Veritabanı Katmanı: Veritabanı Katmanı, uygulamanın veri tabanı ile iletişim kurduğu katmandır. Bu katmanda, Firestore Database gibi bir veri tabanı hizmeti kullanılarak veri alışverişi gerçekleştirilir. Kullanıcıların kitapları araması, ödünç alması, iade etmesi gibi işlemler bu katman üzerinden gerçekleştirilir ve gerekli veriler burada saklanır.

2.3. Veritabanı Tasarımı

Kütüphane otomasyonu veritabanı sisteminin temel özellikleri aşağıda varlık-ilişki (entity – relationship model) modelinde (ER modeli) gösterildiği gibidir. [Diyagram çizimi için <https://dbdiagram.io> sitesi kullanılmıştır.]



Şekil 1 DB - ER Modeli

Bağlantılar şekilde gösterildiği gibidir.

2.4. Varsayımlar ve Bağımlılıklar

Her yazılım projesi, geliştirme süreci boyunca dikkate alınması gereken çeşitli varsayımlar ve bağımlılıklarla karşılaşır. Bu bağımlılıklar, projenin teknik altyapısı,

kullanılan araçlar ve hedeflenen platformlar gibi bir dizi faktörü içerir. Bu başlık altında, mobil kütüphane uygulaması projemizin geliştirilmesi ve işletilmesi sırasında önemli olan bu varsayımlar ve bağımlılıkları inceleyeceğiz. Bu, projenin başarılı bir şekilde tamamlanması ve istenen işlevselliğin elde edilmesi için gereken temel unsurları anlamamıza yardımcı olacaktır.

2.4.1. Platform Bağımlılıkları: React Native kullanılarak geliştirilmiştir, bu nedenle mobil uygulama hem iOS hem de Android platformlarında çalışabilir olmalıdır. Bu, geliştirme sürecinde her iki platforma da uyumlu kod yazılmasını gerektirir.

2.4.2. Üçüncü Taraf Kütüphaneler: Firebase Firestore veritabanını kullanmaktadır ve ayrıca navigasyon için React Navigation, state yönetimi için Redux Toolkit ve ikonlar için Icon kütüphanelerine bağımlıdır. Bu kütüphaneler, geliştirme sürecinde kullanılan önemli araçlardır ve uygun şekilde entegre edilmelidir.

2.4.3. Donanım ve Yazılım Bağımlılıkları: Uygulamanın, minimum işletim sistemi sürümü ve bellek gereksinimleri gibi donanım ve yazılım gereksinimlerini karşılaması gerekmektedir. Bu, farklı mobil cihazların uyumluluğunun test edilmesini gerektirebilir.

2.4.4. Ağ Bağlantısı Bağımlılıkları: Uygulama, Firebase Firestore üzerindeki veritabanı işlemleri için internet bağlantısına bağımlıdır. Bu nedenle, kullanıcıların sürekli olarak internet erişimine sahip olması gerekmektedir. İnternet bağlantısı olmadan uygulama çalışmaz.

2.4.5. Kullanıcı Davranışı ve Kullanılabilirlik Bağımlılıkları: Kullanıcılar, uygulama içinde kitap rezervasyonu yapabilir, kitaplar hakkında bilgi alabilir ve yöneticiye mesaj gönderebilirler. Bu işlevlerin kullanıcı dostu bir arayüzle sunulması ve kullanıcıların ihtiyaçlarını karşılaması önemlidir. Ayrıca, yöneticinin kullanıcıların kitap alışverişlerini ve iadelerini takip etmesi için kullanıcı dostu bir arayüz sağlanmalıdır.

3. Fonksiyonel Gereksinimler

3.1. Kullanıcı İşlevselliği: Kullanıcılar, uygulama içinde çeşitli işlevleri gerçekleştirebilirler. Bu işlevler, uygulamanın amacına ve kullanıcıların ihtiyaçlarına yönelik olarak belirlenmiştir. Kullanıcı işlevselliği, aşağıdaki alt başlıklar altında incelenmiştir.

3.1.1. Üye İşlevselliği

- Giriş Yapma: Üyeler, mevcut hesaplarına giriş yapabilirler.
- Kayıt Olma: Yeni kullanıcılar, uygulamada bir hesap oluşturabilirler.

- Kitap Araştırma: Üyeler, uygulama içindeki kitapları arayabilir, filtreleyebilir ve detaylarına erişebilirler.
- Kitap Rezervasyonu: Üyeler, istedikleri kitapları rezerve edebilirler.
- Mesaj Gönderme: Üyeler, yöneticiye mesaj gönderebilirler.
- Hesap Bilgilerini Güncelleme: Üyeler, hesaplarıyla ilgili bilgileri güncelleyebilirler.

3.1.2. Yönetici İşlevselliği

- Giriş Yapma: Yöneticiler, mevcut hesaplarına giriş yapabilirler.
- Tüm Kitapları Görüntüleme: Yöneticiler, uygulama içinde bulunan tüm kitapları görüntüleyebilirler.
- Kitap Ekleme ve Silme: Yöneticiler, yeni kitaplar ekleyebilir veya mevcut kitapları kaldırabilirler.
- Üyelerle İletişim: Yöneticiler, üyelere gelen mesajlara yanıt verebilirler.
- Kitap Ödünç Alma Takibi: Yöneticiler, hangi üyenin hangi kitabı ödünç aldığını ve iade tarihlerini takip edebilirler.
- Üye Hesaplarını Yönetme: Yöneticiler, üyelerin hesaplarını güncelleyebilir veya kaldırabilirler.

3.2. Sistem İşlevselliği

3.2.1. Veri Yönetimi: Veri yönetimi, uygulamanın Firebase Firestore veritabanı üzerinden veri alışverişi yapması ve kullanıcıların verilerini güvenli bir şekilde saklaması ve güncellemesi işlevlerini içerir. Bu işlevler şu şekilde detaylandırılabilir.

- Kullanıcı Bilgileri: Kullanıcıların hesap bilgileri (ad, soyad, e-posta, vb.) Firebase Authentication ile yönetilir.
- Kitap Veritabanı: Tüm kitapların bilgileri (adı, yazarı, yayın tarihi, kategorisi, vb.) Firebase Firestore veritabanında saklanır.
- Rezervasyonlar: Kullanıcıların yaptığı kitap rezervasyonları ve ödünç almaları Firebase Firestore üzerinde izlenir.

3.2.2. Mesajlaşma Sistemi

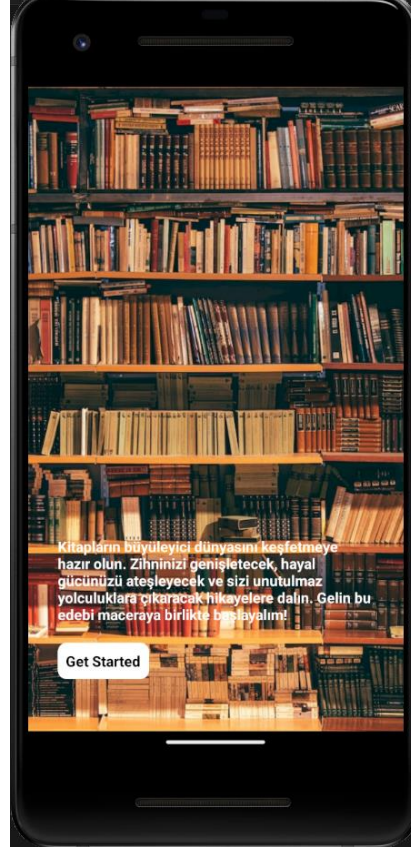
Mesajlaşma sistemi, kullanıcıların yöneticiye mesaj gönderebilmesini ve yöneticinin bu mesajlara yanıt verebilmesini sağlar. Bu işlevler şu şekilde detaylandırılabilir.

- Mesaj Gönderme: Kullanıcılar, uygulama içinde yöneticiye mesaj gönderebilirler. Mesajlar Firebase Firestore üzerinde saklanır.
- Mesaj Alma ve Yanıt Verme: Yöneticiler, kullanıcılardan gelen mesajları görüntüleyebilir ve yanıtlayabilirler. Bu işlem Firebase Firestore üzerinde gerçekleşir.

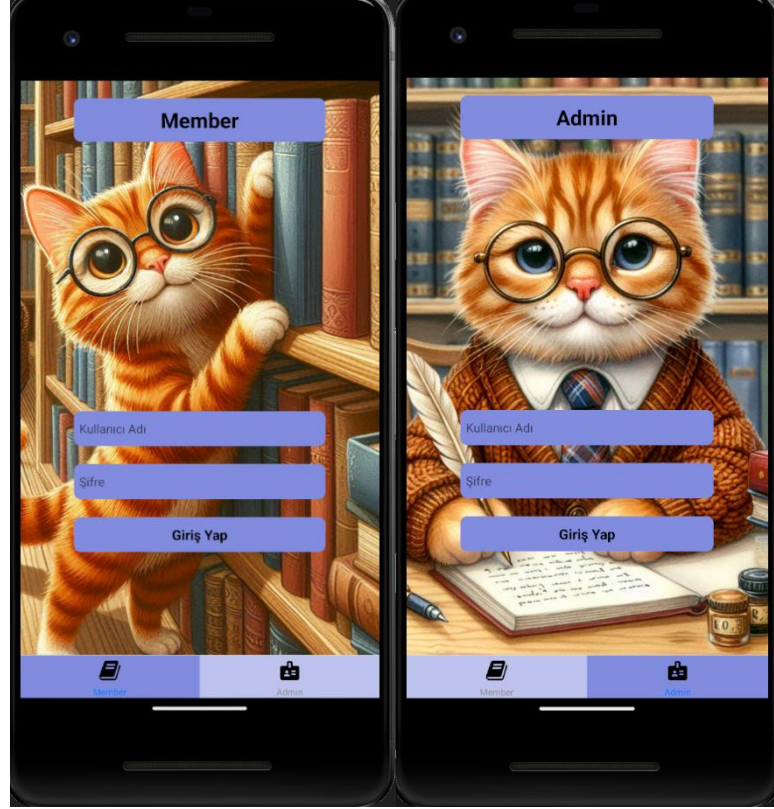
4. Kullanılabilirlik Gereksinimleri

4.1. Arayüz Tasarımı

Bu bölüm, kullanıcıların uygulama içinde gezinirken karşılaşacakları görsel ve işlevsel öğeleri içerir. Fotoğraflar, kullanıcılara sunulacak arayüzün nasıl görüneceğini gösterir. Şekil 2’de anasayfanın nasıl görüneceği yer almaktadır. Kullanıcıyı giriş ekranında kitap okumaya teşvik eden bir söz bulunmaktadır.

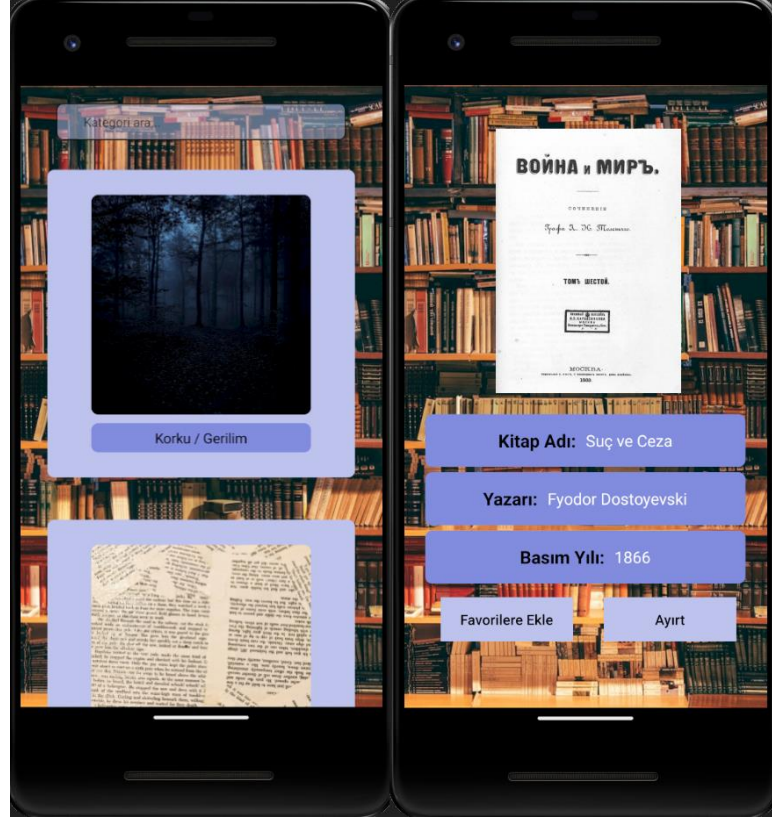


Şekil 2 Anasayfa Görünümü



Şekil 3 Giriş Ekranları

Her bir kullanıcı tanımı için, uygun arkaplanlarla hazırlanan giriş ekranları yukarıda görüldüğü gibidir. [Kedi fotoğrafları bing.com/images/create sitesi kullanılarak tasarlanmıştır.]



Şekil 4 Kategoriler ve Kitap Detayları Ekranları

Yukarıda ise kitapların kategoriler ekranı ve arama sekmesi ve kitap detayları sekmeleri taslak halde görülmektedir.

5. Kullanım – Durum Diyagramları

5.1. Kitap Rezervasyonu

Aktörler	Üye
Tanım	Üyenin kitap rezerve etmesi
Önkoşul	Üyenin önceki aldığı kitapları geri vermiş olması
Sonkoşul	Üye kitabı ödünç alır
Önceliği	1
Kullanım sıklığı	Çok sık
Ana akış	<ul style="list-style-type: none"> • Üye arayüzü açar • Giriş ekranından girişini yapar • İsteddiği kategoriden kitabı seçer • Kitabı butondan rezerve etmiş olur.
Alternatif akış	<ul style="list-style-type: none"> • Üye arayüzü açar • Giriş ekranından girişini yapar

	<ul style="list-style-type: none"> İstediği kategoriden kitap hali hazırda başkası tarafından alınmıştır. Alan kişinin iade tarihini görebilir. Sisteme o gün tekrar giriş yaptığında kitabı rezerve edebilir.
--	---

5.2. Yönetici Kitap Ekleme

Aktörler	Yönetici
Tanım	Kütüphaneye yeni kitap gelmesi
Önkoşul	Giriş yapan kullanıcının yönetici olması
Sonkoşul	Kitabın envantere eklenmesi
Önceliği	1
Kullanım sıklığı	Az
Ana akış	<ul style="list-style-type: none"> Yönetici arayüzü açar Giriş ekranından girişini yapar Kitap ekleme sekmesine gelir Gerekli kategoriye seçer ve kitap bilgilerini girer. Kitap kullanıma açılır
Alternatif akış	<ul style="list-style-type: none"> Yönetici arayüzü açar Giriş ekranından girişini yapar Kitap ekleme sekmesine gelir Kitapla ilgili kategori bölümü yoktur. Kitaba uygun bir kategori sekmesi oluşturur. Kitap bilgilerini girer. Kitap kullanıma açılır

5.3. İletişim

Aktörler	Yönetici, Üye
Tanım	Kullanıcıların (Yönetici veya Üye) kendi aralarında haberleşmesi
Önkoşul	Sadece üye – yönetici arası mesajlaşma olması
Sonkoşul	İletişimin gerçekleşmesi
Önceliği	1
Kullanım sıklığı	Çok
Ana akış	<ul style="list-style-type: none"> Üye girişini yapar

	<ul style="list-style-type: none"> • Bir sorusu varsa veya bir sorunla karşılaşırsa kendi panelinden ilgili sekmeye girer • Yöneticiye mesaj gönderir.
Alternatif akış	<ul style="list-style-type: none"> • Yönetici girişini yapar • Üyeye bir şey söylemek, bir konuda uyarmak için kendi panelinden ilgili sekmeye girer • Üyeye mesaj gönderir.

6. Yardımcı Bilgiler

Proje henüz yapım aşamasındadır.

Link:

<https://github.com/ahmetguness/LibraryApp>

7. Referanslar

1. Perforce. How to Write a Product Requirements Document (PRD). Perforce Blog. Erişim tarihi: 25 Nisan 2024, <https://www.perforce.com/blog/alm/how-write-product-requirements-document-prd>
2. wikiHow. How to Write a Requirements Document. Erişim tarihi: 26 Nisan 2024, <https://www.wikihow.com/Write-a-Requirements-Document>
3. dbdiagram.io. Online Database Diagram Tool. Erişim tarihi: 4 Mayıs 2024, <https://dbdiagram.io/>
4. Bing. Bing Images. Erişim tarihi: 4 Mayıs 2024, <https://www.bing.com/images/create?FORM=GENILP>
5. Chumpolm. (2018). IEEE Standard 830-1984: Recommended Practice for Software Requirements Specifications. Erişim tarihi: 28 Nisan 2024, <https://chumpolm.wordpress.com/wp-content/uploads/2018/09/ieee-std-830-1984.pdf>