

**ANKARA**  
**ÜNİVERSİTESİ**  
**MÜHENDİSLİK**  
**FAKÜLTESİ**  
**BİLGİSAYAR**  
**MÜHENDİSLİĞİ**



**BLM4537- IOS İLE MOBİL UYGULAMA**

**GELİŞTİRME I PROJESİ**

**MELİKE ŞAHİN**

**20290290**

**OCAK 2023**

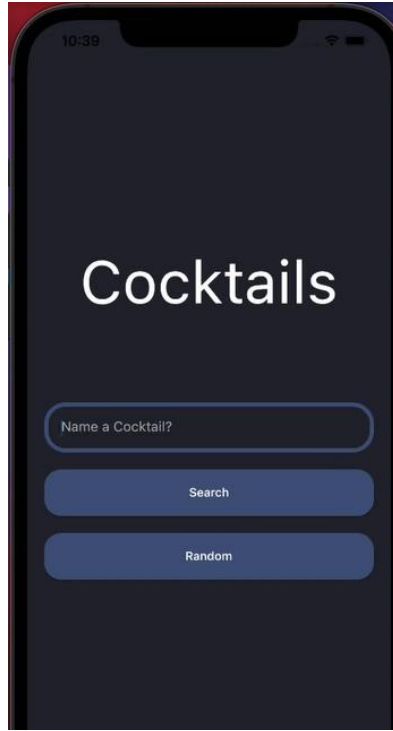
## 1. UYGULAMANIN TASARIMI

Cocktails uygulamasını IOS ile Mobil Uygulama Dersi için geliřtirdim. Projemde 2 panel bulunmaktadır. Bu paneller ařağıdakilerden oluřmaktadır:

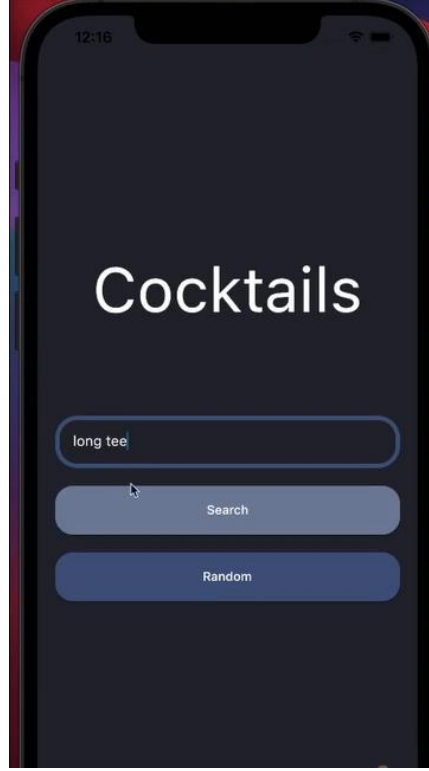
- 1- Kokteyl arama ekranı
- 2- Kokteyl tariflerinin bulunduğı ekran

### 1.1 Kokteyl Arama Ekranı

Kokteyl arama ekranında kokteyl ismi girilerek arama yapılabilir ve kokteylin tarifine, görseline ve malzemelerine ulaşılabilir. Aynı zamanda kokteyl arama ekranından random seçeneğı ile herhangi bir kokteyle ulaşılabilir.



řekil 1.1 Arama Ekranı



Şekil 1.1 Arama Ekranı

## 1.2 Kokteyl Tariflerinin Bulunduğu Ekran

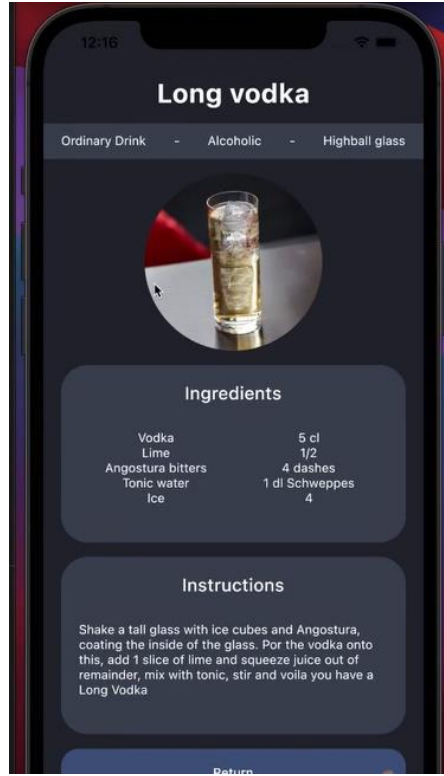
Kokteyl tariflerinin bulunduğu ekranda arama yapılan ya da random seçeneği sonucunda bulunan kokteyle ait görsel, malzemeler ve kokteylin tarifi karşımıza çıkmaktadır.



Şekil 1.2.1 Tarif Ekranı



Şekil 1.2.2 Tarif Ekranı



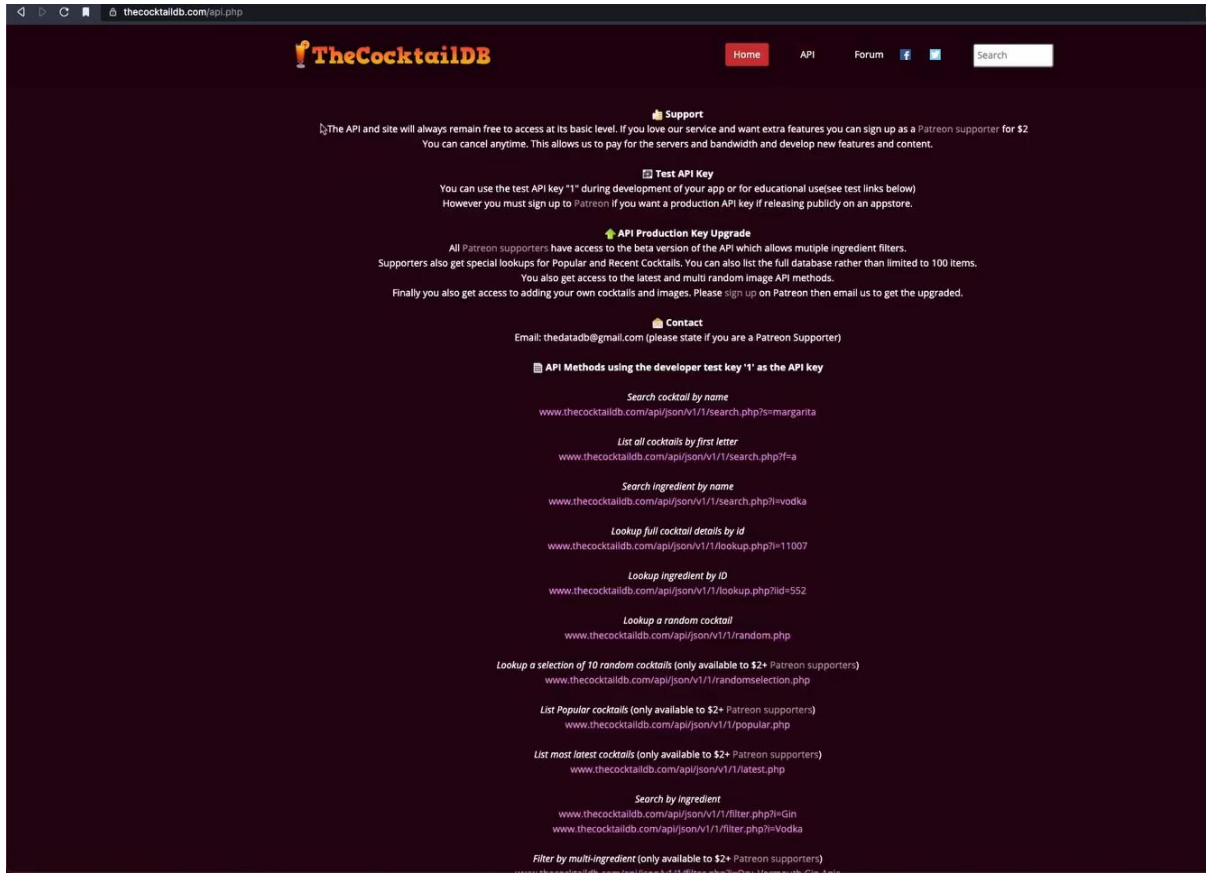
Şekil 1.2.3 Tarif Ekranı

## 2. API

Uygulamamızdaki kokteyl türlerine, malzemelerine ve tariflerine ulaşabilmek için çevrimiçi bir web sitesi olan TheCocktailDB üzerinden bir veritabanı ile uygulamamı bağladım.



Şekil 2.1 API Adresi



Şekil 2.1 API Adresi

### 3. BACKEND

Proje kapsamlı olduğu için kodların belli bir kısmı paylaşılacaktır. Kodlar hakkında daha detaylı bilgi için <https://github.com/melike-sahin/Cocktails> adresinden Github hesabımı ziyaret edebilirsiniz.

```
class Ingredient {  
    Ingredient({@required this.name, @required this.measure});  
  
    String name;  
    String measure;  
}
```

Yukarıda kokteyl tariflerindeki kullanılacak malzemelerin ismi ve ölçü miktarı bilgilerini taşıyan class tanımlanmıştır.

```
class CocktailManager {  
    String name;  
    String category;  
    String alcoholic;  
    String glassType;  
    String pictureUrl;  
    String instructions;  
    List<Ingredient> ingredients;  
  
    CocktailManager(  
        {this.name,  
        this.category,  
        this.alcoholic,  
        this.glassType,  
        this.pictureUrl,  
        this.instructions,  
        this.ingredients});  
}
```

Yukarıda uygulamada görüntülenecek olan kokteyl tarifi ile ilgili bilgileri tutan class tanımlanmıştır.

```
class InstructionWidget extends StatelessWidget {
  const InstructionWidget({@required this.instructions});
  final String instructions;
  @override
  Widget build(BuildContext context) {
    return Container(
      child: Column(
        children: [
          Text(
            "Instructions",
            style: kTableTextStyle,
          ), // Text
          SizedBox(height: 30),
          Expanded(
            child: SingleChildScrollView(
              scrollDirection: Axis.vertical,
              child: Text(instructions),
            ), // SingleChildScrollView
          ), // Expanded
        ],
      ), // Column
    ); // Container
  }
}
```

Yukarıda kokteyl tarifini görüntülemek için gereken widget tanımı yapılmıştır.



```

const kMainUrl = "https://www.thecocktaildb.com/api/json/v1/1/search.php?s=";
const kRandomUrl = "https://www.thecocktaildb.com/api/json/v1/1/random.php";

const kBackgroundColor = Color(0xFF1F2129);
const kComponentColor = Color(0xFF3C4D74);
const kGroupBackgroundColor = Color(0xFF383D4D);

const kHeaderTextStyle = TextStyle(
  fontSize: 30,
  fontWeight: FontWeight.w700,
);

const kTableTextStyle = TextStyle(
  fontSize: 20,
  fontWeight: FontWeight.w500,
);

const kBorderSide = BorderSide(
  color: kComponentColor,
  width: 5,
);

const kBoxDecorationStyle = BoxDecoration(
  color: kGroupBackgroundColor,
  borderRadius: BorderRadius.all(
    Radius.circular(30),
  ), // BorderRadius.all
); // BoxDecoration

const kButtonMinSize = Size(100, 55);

```

Yukarıda uygulamaya uygulanacak olan stil kuralları ve API URI sabitleri tanımlanmıştır.

```

Run | Debug | Profile
void main() {
  //portrait mode
  WidgetsFlutterBinding.ensureInitialized();
  SystemChrome.setPreferredOrientations([DeviceOrientation.portraitUp])
    .then((value) => runApp(Cocktails()));
}

class Cocktails extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    //theme -> dark mode
    return MaterialApp(
      home: SearchWindow(),
      debugShowCheckedModeBanner: false,
      theme: ThemeData.dark().copyWith(
        scaffoldBackgroundColor: kBackgroundColor,
      ),
    ); // MaterialApp
  }
}

```

Yukarıda main sürücü fonksiyonunda tüm uygulama servisleri başlatılmıştır.