
MBoard Documentation

Release 1.0

itucsdb1745-Yunus Güngör

Dec 21, 2017

CONTENTS

Team itucsd1745

Members

- Yunus Güngör

MBoard web tabanlı bir iletişim ve sosyal ağıdır. Kullanıcılar tüm kullanıcıların görebildiği mesajlar ekleyebilir ve eklenen mesajlara cevap yazabilir. Yazılan cevaplar kullanıcılar tarafından oylanabilir ve cevapların görüntülenme sırası oylara göre belirlenir.

Contents:

KULLANICI KILAVUZU

MBoard uygulaması kullanıcıları ana sayfa ile karşılar

Kullanıcılar cevapları ve mesajları herhangi bir hesap oluşturmadan görebilirler ama mesaj veya eklemek, cevaplara oy vermek isteyen bir kullanıcı hesap oluşturmalı veya giriş yapmalıdır. Sağ üstteki “login” tuşu kullanılarak giriş yapılabilir veya üye olunabilir.

Giriş yapmış olan kullanıcılar misafir kullanıcıların yapamadığı, cevap veya mesaj ekleme, cevapları oylama gibi işlemleri gerçekleştirebilirler. Misafir kullanıcıların bu işlemleri gerçekleştirebilmeleri için sisteme giriş yapmaları gerekmektedir. Bir kere giriş yapan kullanıcıyı, sistem bir sonraki ziyaretinde tanıyacak ve otomatik olarak giriş yapacaktır. Kullanıcı hesabı oluşturmak giriş yapmak kadar kolay, “login” tuşunu kullanarak yönlendirilen sayfada kullanılmak istenen kullanıcı adı ve şifrenin girilmesi hesap oluşturmak için yeterlidir.

Sistemde üç farklı kullanıcı bulunmaktadır:

Kullanıcılar

- Misafir Kullanıcı
- Normal Kullanıcı
- Admin Kullanıcı

1.1 Yunus Güngör tarafından tamamlanan parçalar

1.1.1 Ana Sayfa

Misafir Kullanıcı

Misafir kullanıcı giriş yapmak için sağ üst köşede bulunan “Login” butonunu kullanabilir. Kullanıcın giriş yapması ile, misafir kullanıcı normal kullanıcı haline gelir.

Herhangi mesaja ait cevaplar görüntülenmek istendiğinde “Answers” butonuna basılarak o mesaja ait mesajlar ve o mesajların oyları ekranda görüntülenebilir.

Cevapların sol tarafında cevabı yazan kullanıcının adı ve resmi bulunmaktadır.

Cevap yazmak için kullanıcıların giriş yaparak normal kullanıcı olması gerekmektedir. Giriş yapmak için login sayfası kullanılabilir.

Eğer misafir kullanıcı daha önce hesabını kaydetmediyse, başka bir kullanıcı tarafından kullanılmayan bir kullanıcı adı ve daha sonra hatırlayabileceği bir şifreyi login sayfasındaki forma girip, “Log in/Sign Up” tuşuna bastıktan sonra, kullanıcının şifreyi tekrar girmesi istenecektir.

Oluşturulan hesabın şifresi veya profil fotosu profil sayfasından değiştirilebilir ancak kullanıcı adının değiştirilmesine izin verilmez.

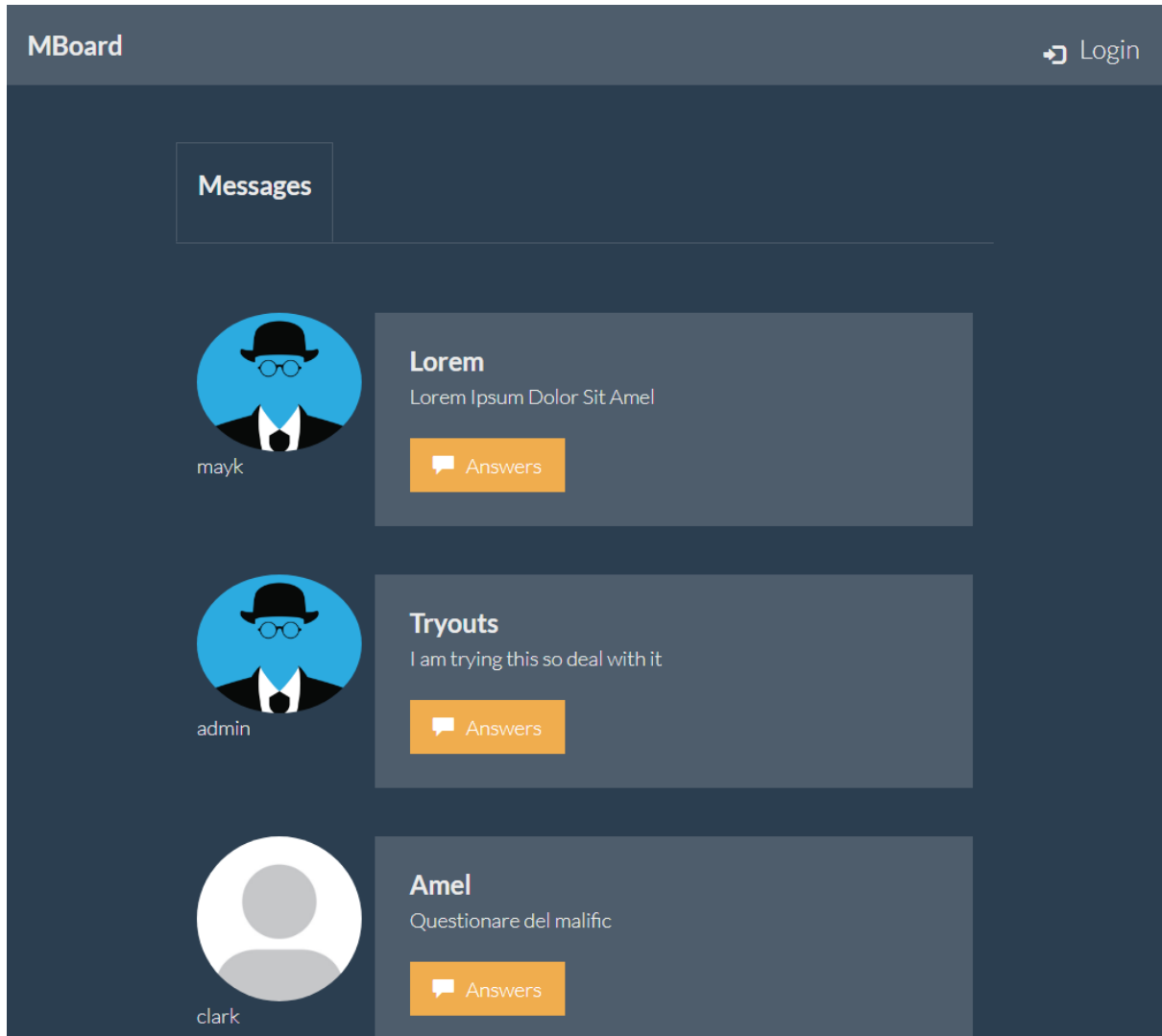


Fig. 1.1: Misafir kullanıcı ana sayfada mesajları ve cevapları görüntüleyebilir veya giriş yapabilir.

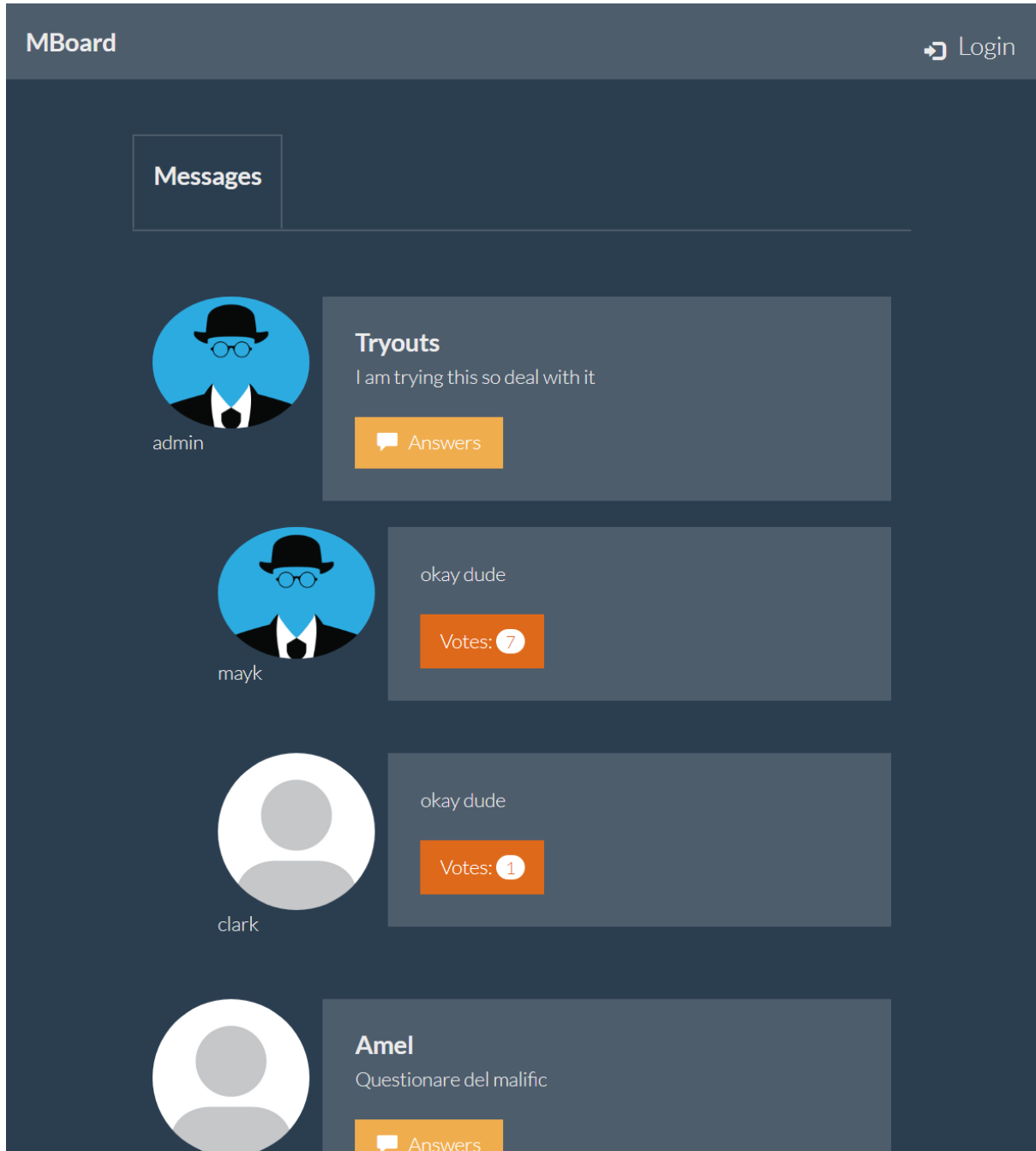


Fig. 1.2: "Answers" butonuna basıldıktan sonra cevaplar mesajın altında görüntülenebilir hale gelir.

MBoard [Login](#)

Log In

Username:

Password:

[Log in / Sign Up](#)

You can sign up with logging in if you dont have an account

To create a new account in the system just login with desired username and password, system will crate a new user if an account does not exists

© 2017, Yunus G ng r

Fig. 1.3: Daha  nce kaydolunan kullanıcı adı ve şifre login sayfasındaki forma girilerek ve “Log in/Sign Up” butonuna basılarak giriş yapılabilir.

MBoard

Login

Log In

Username: db

Password:

Sign Up

No user found please press sign up to add user or go back for trying again

You can sign up with logging in if you dont have an account

To create a new account in the system just login with desired username and password, system will crate a new user if an account does not exists

© 2017, Yunus Güngör

Fig. 1.4: Verilen ekstra mesajlar ile sistemin nasıl kullanılabileceği ile ilgili ipuçları veya geri bildirimler yapılmaktadır. Login sayfasında sistemde bulunmayan bir kullanıcı adı girildiğinde login sayfası otomatik olarak üye ekleme moduna geçer. Kullanıcının şifresini tekrar girmesi ile yeni kullanıcı sistemde oluşturulabilir. Bu durumda kullanıcı adı değiştirilemez. Eğer kullanıcı yanlış kullanıcı adı girdiyorsa tarayıcıdaki geri tuşunu kullanarak geri dönebilir ve tekrar deneyebilir.

Normal Kullanıcı

Kullanıcı giriş yaptıktan veya sistem otomatik olarak tanıdıktan sonra ana sayfa ekranı “Profile” butonu, “Add Message” sekmesi, her mesajda “Reply” butonu ve her cevapta “Up” butonu ile birlikte görüntülenir.

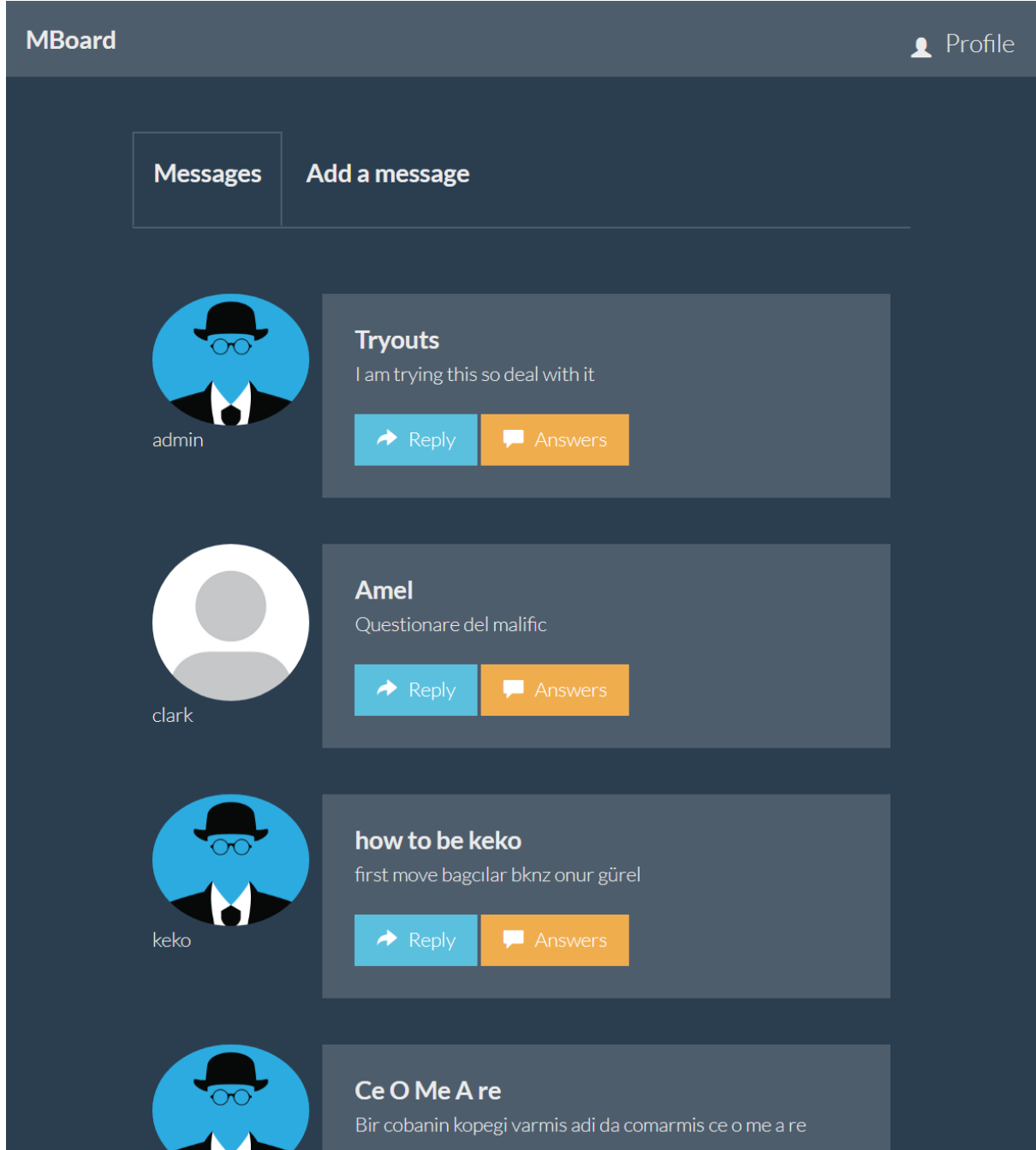
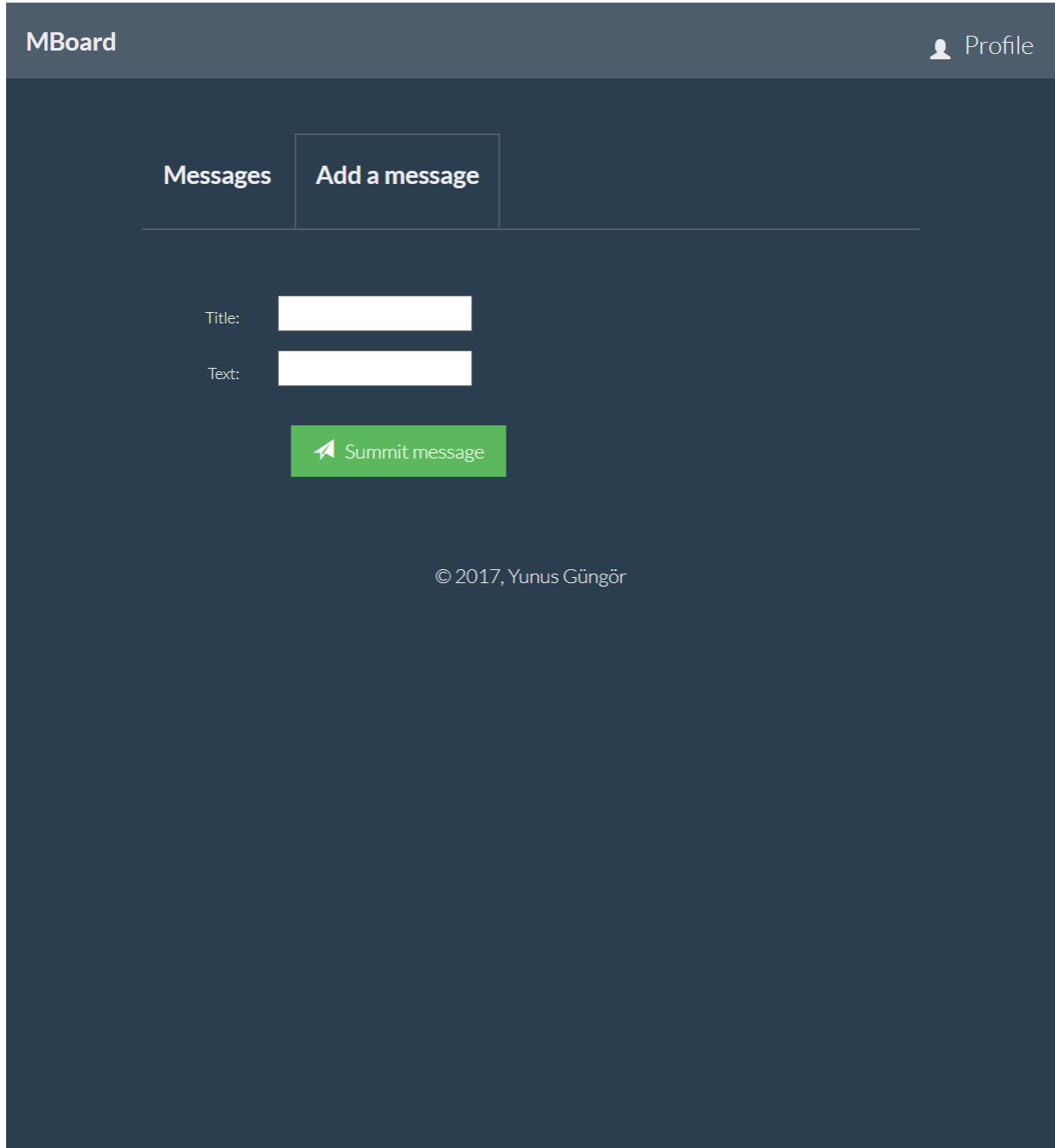


Fig. 1.5: Ana sayfa ekranından “Profile” butonu kullanılarak profil sayfasına erişilebilir. Kullanıcı ansayfada bulunan “Add Message” sekmesini kullanarak yeni bir mesaj ekleyebilir.

“Add Message” sekmesine tıklandığında mesaj eklemek için gerekli olan form ekranda görünür olacaktır. Bu formda başlık ve mesaj bilgileri doldurulduktan sonra “add message” butonuna basarak yeni mesaj sisteme eklenebilir.

Herhangi bir cevap eklemek için, cevap eklenmek istenen mesajın üstünde bulunan “Reply” tuşu kullanılabilir.



The screenshot shows a web interface for MBoard. At the top, there is a dark blue header bar with the text 'MBoard' on the left and a 'Profile' link with a user icon on the right. Below the header, the main content area has a dark blue background. It features two tabs: 'Messages' and 'Add a message'. The 'Add a message' tab is active. Below the tabs, there are two input fields: 'Title:' and 'Text:'. Below these fields is a green button with a white paper plane icon and the text 'Summit message'. At the bottom center of the main content area, there is a copyright notice: '© 2017, Yunus Güngör'.

Fig. 1.6: Mesaj eklemek için doldurulması gereken form.

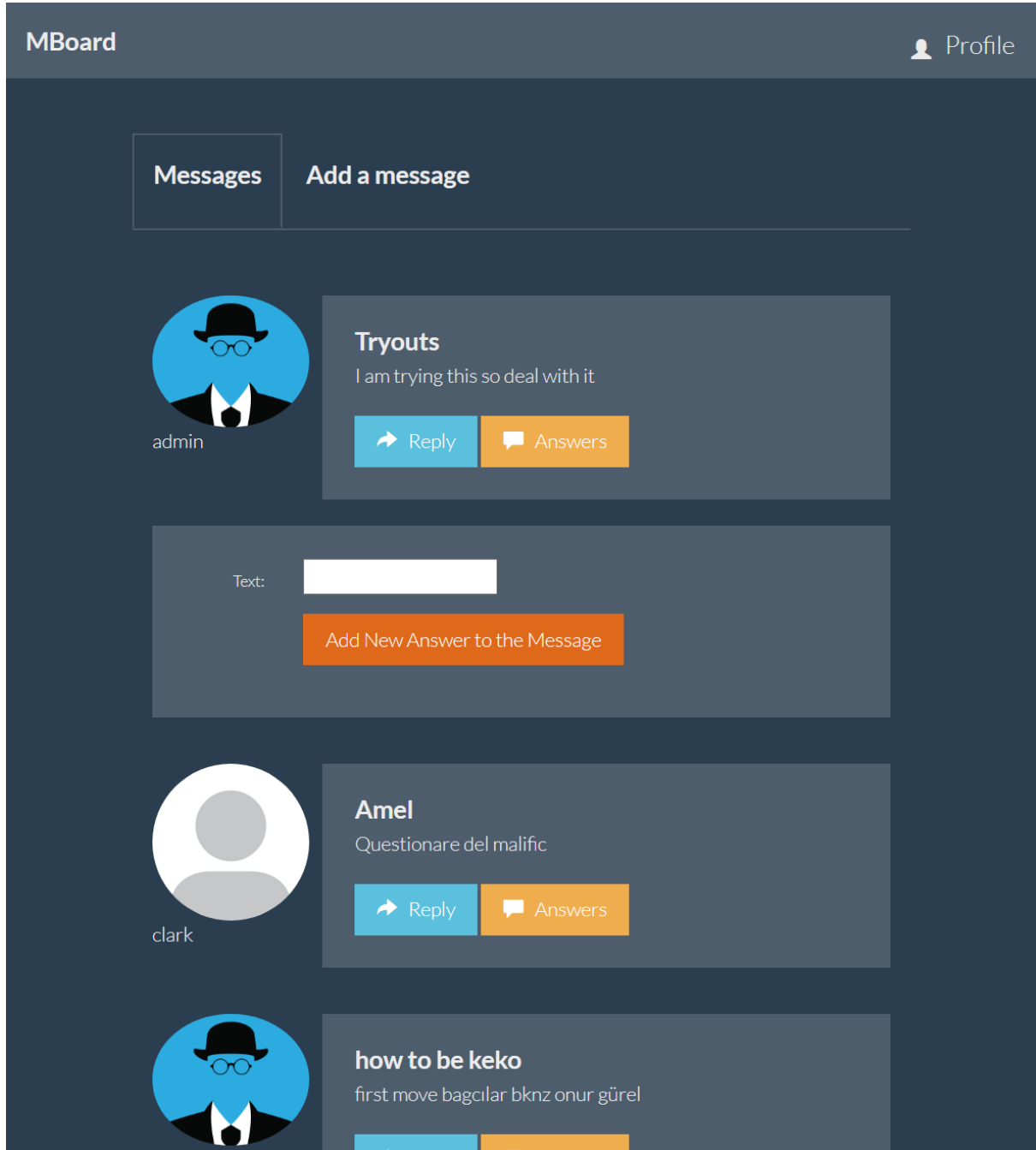


Fig. 1.7: “Reply” tuşuna basıldığında görüntülenen form doldurularak ve “Add New Answer to the Message” butonu kullanılarak, mesaja cevap eklenebilir. Tekrar “Reply” tuşuna basıldığında veya butona basıldığında form tekrar gizlenir.

Herhangi bir cevaba oy vermek için oy verilmek istenen cevabın üstünde bulunan “Up” butonu kullanılabilir. Oy verildikten sonra ana sayfa yenilenir ve cevaplar oy sayısına göre tekrar sıralanır. Kullanıcıların oy verme sınırı bulunmamaktadır. Herhangi bir cevabın daha yukarda görüntülenmesi verilen oyların sayısına bağlıdır. Daha çok oy alan cevap daha yukarda görüntülenir.

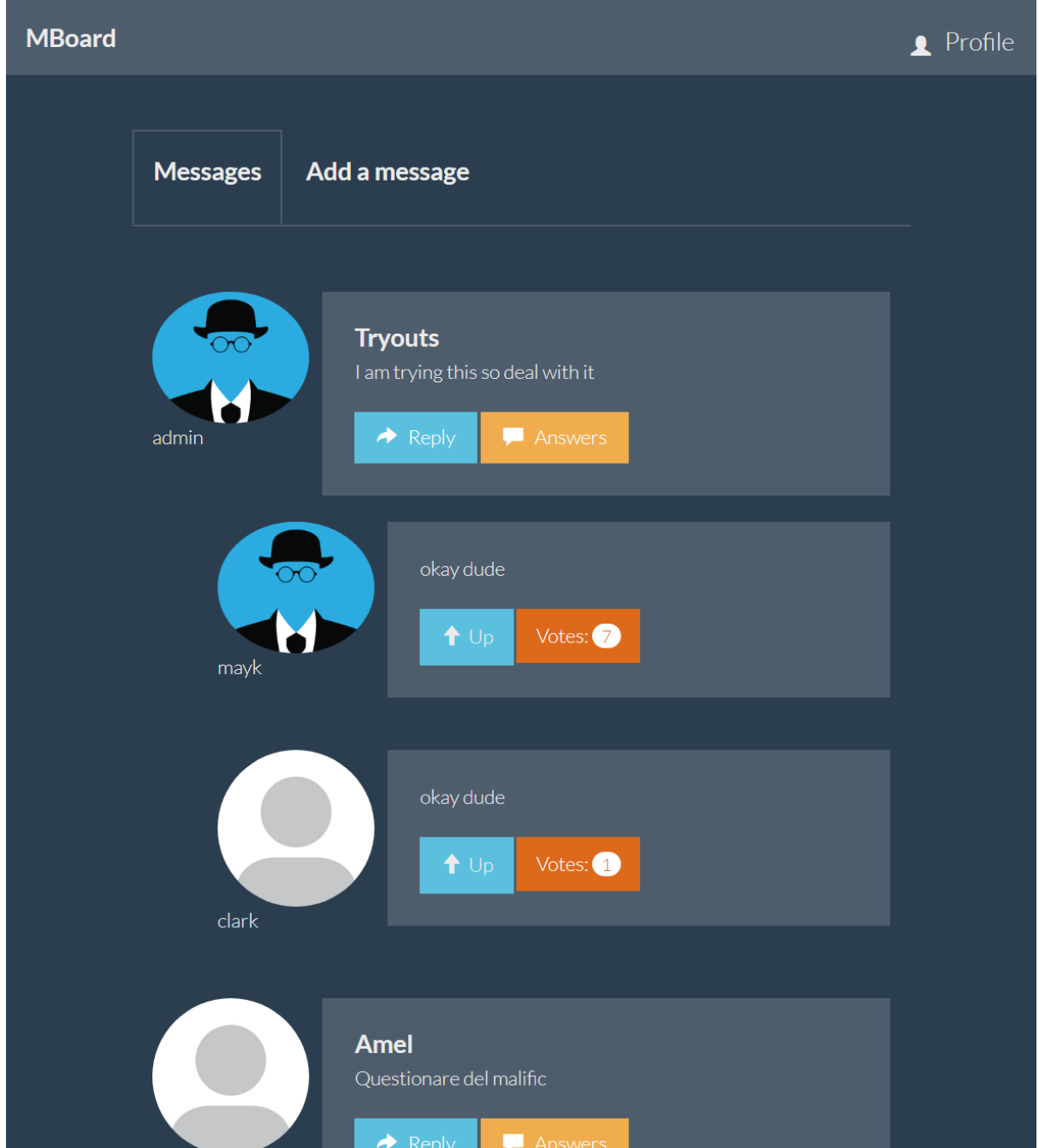


Fig. 1.8: “Up” butonun görüntülenebilmesi için kullanıcının giriş yapmış olması ve herhangi bir mesaja ait olan cevapları görüntülüyor olması gerekmektedir.

Admin Kullanıcı

Admin kullanıcılar normal kullanıcıların yapabildiği tüm işlemleri normal kullanıcı gibi yerine getirebilir. Bu işlemler: Mesaj ekleme, cevap ekleme, profil sayfasına erişebilme ve cevapları oylayabilmektir. Bu işlemlerin yanında ana sayfada mesaj ve cevapları düzenleyebilirler.

Admin kullanıcılar eklenen herhangi bir mesaj veya cevabı silebilir veya düzenleyebilir. Mesaj veya

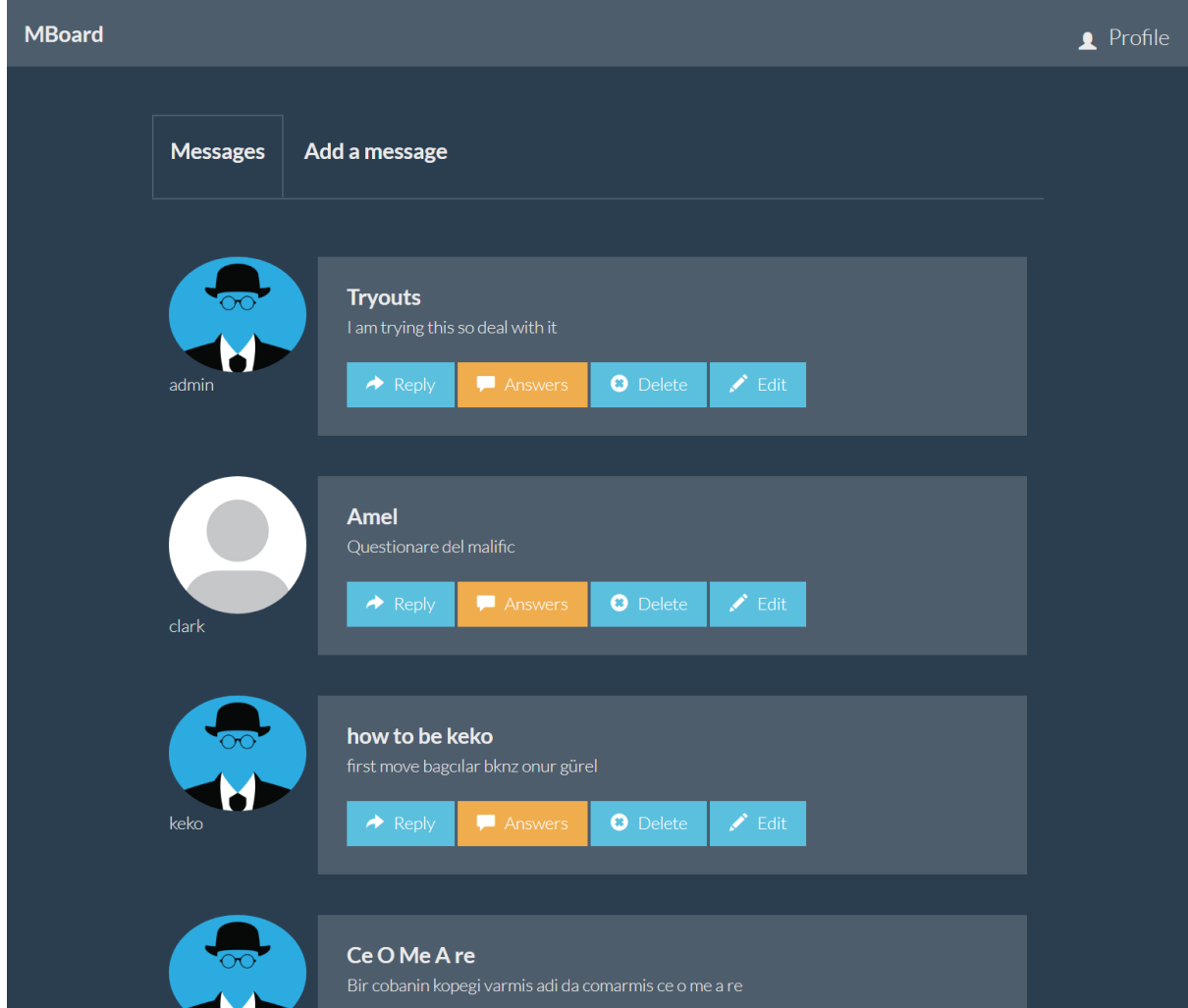


Fig. 1.9: Admin kullanıcı ana sayfada bulunan mesajların üzerindeki “Delete” ve “Edit” butonlarına erişim sağlayabilir.

cevapların üzerinde bulunan “Delete” butonuna basıldığında o mesaj veya cevap silinir. Silinen mesaja ait olan tüm cevaplar da mesajla birlikte silinir. “Edit” butonuna basıldığında ise yeni bir sayfada düzenleme formu daha önce bulunan mesaj veya cevap verisiyle birlikte görüntülenir. İstenilen düzenlemeler yapıldıktan sonra “Summit Message” butonu ile yapılan değişiklikler kaydedilir.

Fig. 1.10: Cevap Düzenleme ekranı

1.1.2 Profil Sayfası

Misafir Kullanıcı

Misafir kullanıcı profil sayfasına erişemez. Profil sayfasına sadece sisteme giriş yapmış olan normal kullanıcılar veya admin kullanıcılar ulaşabilir.

Normal Kullanıcı

Normal kullanıcı profil sayfasından hesabından çıkış yapabilir, şifresini değiştirebilir, profil resmini değiştirebilir, diğer kullanıcıları görüntüleyebilir, diğer kullanıcıları arkadaş olaral ekleyebilir veya en iyi arkadaşını değiştirebilir.

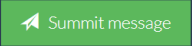
Sağ üstteki “Log Out” butonu ile kullanıcılar hesabından çıkış yapabilir. “Change Password” bölümündeki formu doldurarak, “Change Password” butonuna basarak kullanıcı şifresi değiştirilebilir. “Change Picture” bölümündeki formda yapılan değişiklikler, “Change Picture” butonuna

MBoard [Profile](#)

Edit Message

Title: Tryouts

Text: I am trying this so deal

 Summit message

© 2017, Yunus Güngör

Fig. 1.11: Mesaj Düzenleme ekranı

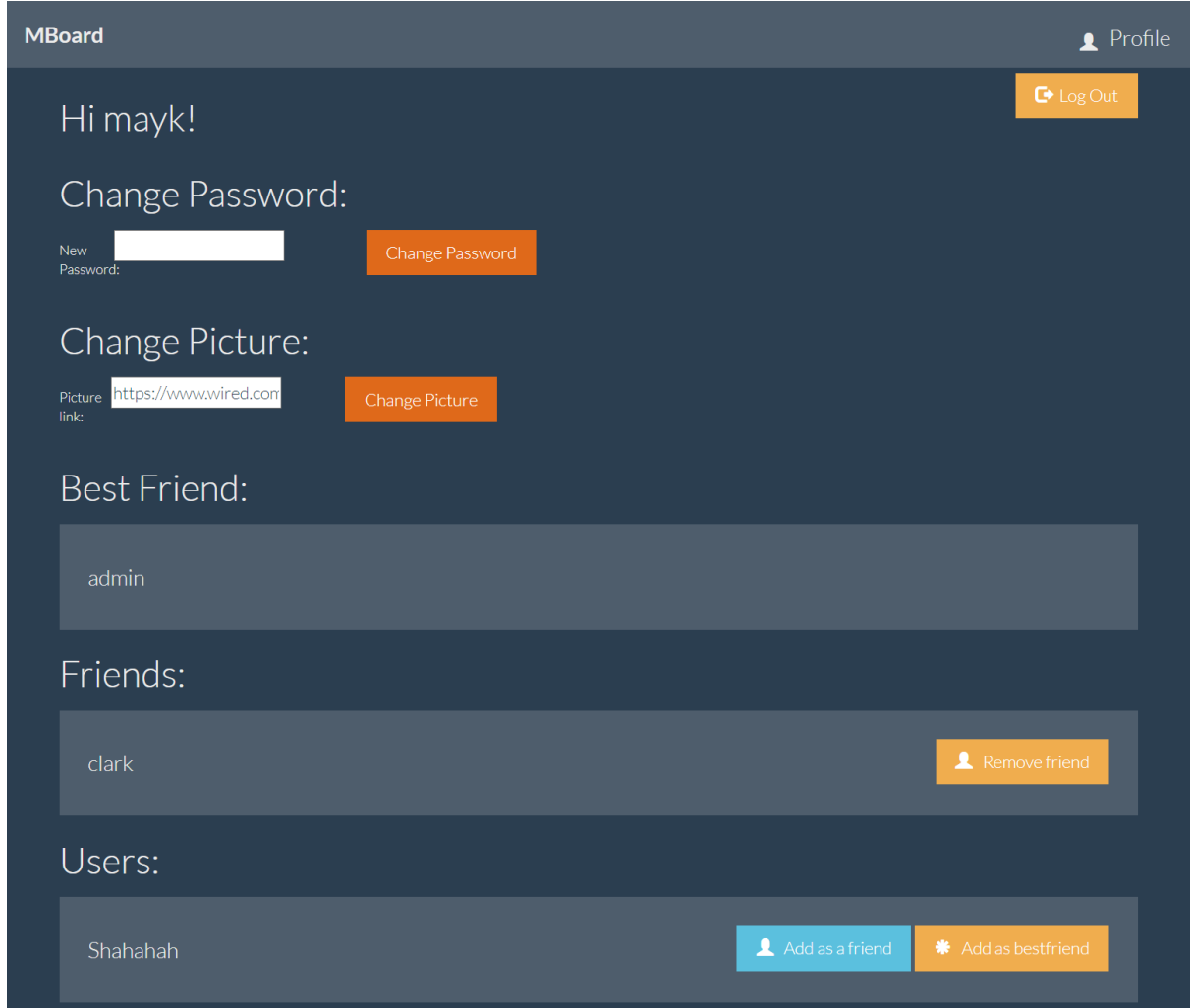


Fig. 1.12: Profil ekranında kullanıcıyı kullanıcı ismiyle karşılayan bir mesaj bulunmaktadır.

basarak kaydedilebilir. Bu bölüm (“User Picture”) kullanıcının profil fotosunu bulundurur. Sistemin dosya depolama yeteneği olmadığı için profil fotoğrafları başka bir site üzerinden link ile kullanılabilir. “Best Friend” bölümünde kullanıcının en iyi arkadaşı görüntülenebilir. Bu bölümün varsayılan değeri “admin” dir. Herhangi bir en iyi arkadaş bulunamazsa sistem “admin” i en iyi arkadaş olarak görüntülemektedir. “Friends” bölümünde eklenen arkadaşlar görüntülenebilir. Arkadaşların üzerinde “Remove Friend” butonu kullanılarak o kullanıcı arkadaşlıktan çıkarılabilir.

“Users” bölümünde sistemdeki tüm kullanıcılar görüntülenebilir. Her kullanıcının üzerinde bulunan “Add as a friend” butonu kullanılarak o kullanıcı arkadaş olarak eklenebilir. Eklenen arkadaşlar “Friends” bölümünde görüntülenebilir. “Add as bestfriend” butonu kullanılarak kullanıcı en iyi arkadaş olarak eklenebilir. En iyi arkadaş olarak eklenen kullanıcının en iyi arkadaşı eklemeyi yapan kullanıcı olarak düzenlenir. Yani en iyi arkadaş butonu iki kullanıcı için de en iyi arkadaşı değiştirir.

Admin Kullanıcı

Admin kullanıcı normal kullanıcının profil sayfasında yapabildikleri dışında kullanıcıları silebilir veya başka bir kullanıcıyı admin kullanıcı yapabilir.

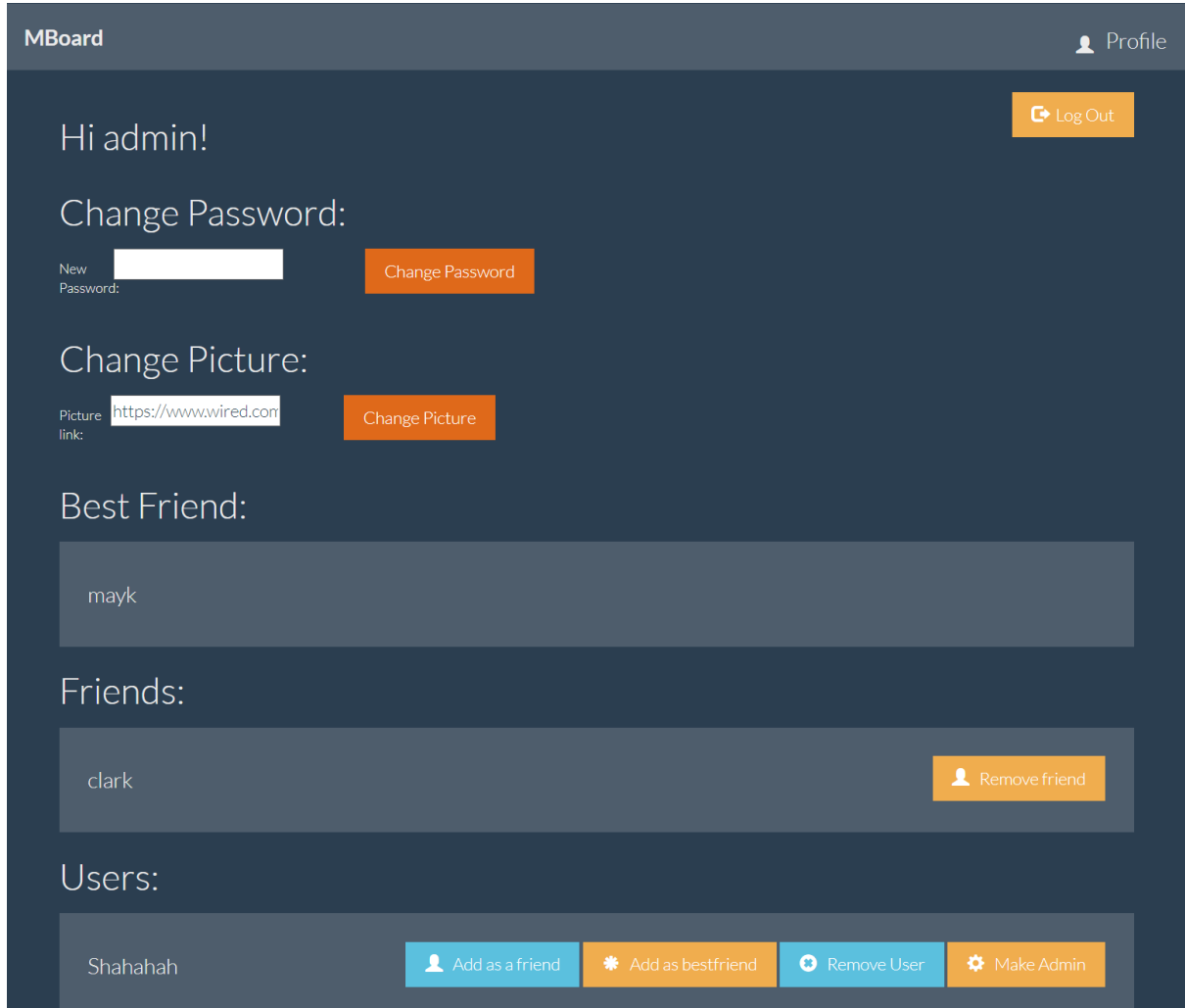


Fig. 1.13: Admin kullanıcı profil sayfasına giriş yaparsa, “Users” bölümünde kullanıcılara ait “Remove User” ve “Make Admin” butonlarını görebilir.

Kullanıcıların üzerindeki “Remove User” butonuna basarak o kullanıcı sistemden silinebilir. Sistemde bulunan o kullanıcının eklediği tüm mesajlar ve cevaplar o kullanıcıyla birlikte sistemden silinir. Kullanıcılar üzerindeki “Make Admin” butonuna basılarak normal bir kullanıcı admin kullanıcı seviyesine getirilebilir.

Ayrıca admin kullanıcı giriř yaptıktan sonra tarayıcı üzerinden site adresindeki “/reset” adresine girerse sistemde bulunan tüm veriler temizlenir ve yeniden kurulumu yapılır. Eğer sistemde hiçbir kullanıcı yoksa admin kullanıcı şartı aranmaz. Admin kullanıcı tarayıcı üzerinden site adresindeki “/dummy” adresine girerse sisteme daha önceden hazırlanmış verilerin giriři yapılır. Sistemin yeniden kurulumundan sonra sisteme tek bir kullanıcı eklenir ve bu kullanıcı admin kullanıcı olarak ayarlanır. Bu kullanıcının kullanıcı adı “admin” ve şifresi “admin” dir.

GELİŞTİRİCİ KILAVUZU

2.1 Veritabanı Tasarımı

Veritabanında 3 varlık ve 4 tablo bulunmaktadır. Tablolardan 1 tanesi varlıklar arası bağlantı kurmak için kullanılmıştır. Veritabanındaki varlıklar ve bağlantılar aşağıdaki diagramda görülebilir. Bu varlıkların yönetimi (ekle,sil,düzenle,ara) ile ilgili fonksiyonların tümü “app/models/database.py” dosyasının içinde “Database” sınıfında bulunmaktadır.

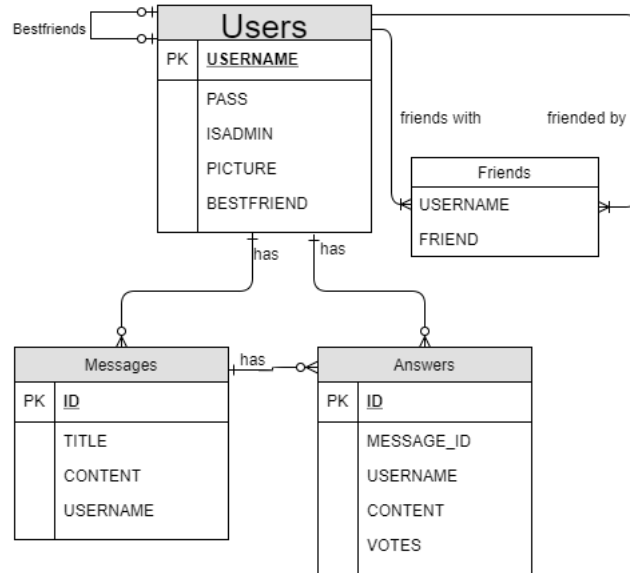


Fig. 2.1: Veritabanı yapısı

2.2 Kod

Bu uygulamada kodlar klasörlendirilmiş ve obje tabanlı bir yaklaşım izlenmiştir. Ana dizinde bulunan “app” klasörü uygulamanın kaynak kodlarını içerir. Diğer klasör ve dosyalar uygulamanın genel yapısının kurulması ve çalıştırılması ile ilgilidir. “app” klasörü içindeki “models” klasöründe uygulamada kullanılan objeler yer almaktadır. “handlers.py” web sayfalarının adreslerinin işlenmesi ve yönlendirilmesini kontrol etmektedir ve “forms.py” uygulamada kullanılan tüm girdi formlarını bulundurmaktadır. Bunların yanında “server.py” dosyasında uygulamayı başlatmak için gerekli kodlar ve konfigürasyonlar yer almaktadır. “templates” klasöründe ise html ve css kullanılarak yazılmış arayüz dosyaları bulunmaktadır.

Uygulama “server.py” dosyasının python ile çalıştırılması ile başlatılabilir.

Uygulamada bazı 3. parti eklentiler ve temalar kullanılmıştır:

Uygulama **python** kullanılarak yazılmıştır, python’ın yanında 3. parti anaçatılar ve eklentileri kullanılmıştır.

Kullanılan anaçatı yapıları

- Bootstrap
- Flask

Eklentiler

- psycopg2
- passlib
- flask_login
- Flask-WTF
- wtforms_components

Tema ve arayüz eklentileri

- Superhero bootstrap teması
- UI Comments arayüzü

Tema ve arayüz eklentileriyle ilgili daha fazla bilgi kaynak koddaki “Citations and used libraries.txt” dosyasında bulunabilir.

Uygulamadaki veritabanı tabloları “/reset” linkine girilmesiyle tetiklenen “resetEverything()” fonksiyonu ile oluşturulmuştur. Aynı zamanda uygulamadaki “/dummy” linkine girilmesi tetiklenen “insertDummy()” fonksiyonu ile sisteme belirli bir miktarda veri otomatik olarak eklenebilir. Bu iki fonksiyon da “adminStuff.py” dosyasındaki “adminCommands” sınıfında bulunmaktadır.

```
def resetEverything():
    if not current_user.is_anonymous and current_user.is_admin:
        with dbapi2.connect(current_app.config['dsn']) as connection:
            cursor = connection.cursor()

            query = "DROP TABLE IF EXISTS ANSWERS"
            cursor.execute(query)

            query = "DROP TABLE IF EXISTS MESSAGES"
            cursor.execute(query)

            query = "DROP TABLE IF EXISTS FRIENDS"
            cursor.execute(query)

            query = "DROP TABLE IF EXISTS USERS"
            cursor.execute(query)

            query = "CREATE TABLE USERS (USERNAME VARCHAR(50) PRIMARY KEY,
→PASS VARCHAR(120), ISADMIN BOOLEAN DEFAULT FALSE, PICTURE VARCHAR(500)
→DEFAULT 'https://www.wired.com/wp-content/uploads/2015/11/GettyImages-
→134367495.jpg', BESTFRIEND VARCHAR(50) REFERENCES USERS(USERNAME) ON
→DELETE SET DEFAULT DEFAULT 'admin') "
            cursor.execute(query)

            query = "CREATE TABLE MESSAGES (ID SERIAL PRIMARY KEY, TITLE
→VARCHAR(50), CONTENT VARCHAR(500), USERNAME VARCHAR(50) REFERENCES
→USERS(USERNAME) ON DELETE CASCADE) "
            cursor.execute(query)

            query = "CREATE TABLE ANSWERS (ID SERIAL PRIMARY KEY, MESSAGE_ID
→INTEGER REFERENCES MESSAGES(ID) ON DELETE CASCADE , USERNAME
→VARCHAR(50) REFERENCES USERS(USERNAME) ON DELETE CASCADE, CONTENT
→VARCHAR(500), VOTES INTEGER DEFAULT 0) "
            cursor.execute(query)

            query = "CREATE TABLE FRIENDS (USERNAME VARCHAR(50) REFERENCES
→USERS(USERNAME) ON UPDATE CASCADE, FRIEND VARCHAR(50) REFERENCES
→USERS(USERNAME) ON UPDATE CASCADE) "
```



```

        cursor.execute(query)

        hashedPass = pwd_context.encrypt('admin')
        query = "INSERT INTO USERS (USERNAME,PASS,ISADMIN) VALUES (%s,
→ %s, %s)"
        cursor.execute(query, ('admin',hashedPass,True))

        connection.commit()
        logout_user()

```

```

def insertDummy():
    if not current_user.is_anonymous and current_user.is_admin:
        with dbapi2.connect(current_app.config['dsn']) as connection:
            cursor = connection.cursor()

            hashedPass = pwd_context.encrypt('123456')
            query = "INSERT INTO USERS (USERNAME,PASS,ISADMIN) VALUES (%s,
→ %s, %s)"
            cursor.execute(query, ('mayk',hashedPass,False))

            hashedPass = pwd_context.encrypt('123456')
            query = "INSERT INTO USERS (USERNAME,PASS,ISADMIN, PICTURE)
→VALUES (%s, %s, %s, %s)"
            cursor.execute(query, ('clark',hashedPass,False,'http://
→ishowmy.support/img/user-icon-360x360.jpg'))

            query = "INSERT INTO MESSAGES (TITLE,CONTENT,USERNAME) VALUES
→(%s, %s, %s)"
            cursor.execute(query, ('Lorem', 'Lorem Ipsum Dolor Sit Amel',
→'mayk'))

            query = "INSERT INTO MESSAGES (TITLE,CONTENT,USERNAME) VALUES
→(%s, %s, %s)"
            cursor.execute(query, ('Tryouts', 'I am trying this so deal
→with it', 'admin'))

            query = "INSERT INTO ANSWERS (MESSAGE_ID,USERNAME,CONTENT,
→VOTES) VALUES (%s, %s, %s, %s)"
            cursor.execute(query, ('2', 'mayk', 'okay dude', '5'))

            query = "INSERT INTO ANSWERS (MESSAGE_ID,USERNAME,CONTENT,
→VOTES) VALUES (%s, %s, %s, %s)"
            cursor.execute(query, ('2', 'clark', 'okay dude', '1'))

            query = "INSERT INTO MESSAGES (TITLE,CONTENT,USERNAME) VALUES
→(%s, %s, %s)"
            cursor.execute(query, ('Amel', 'Questionare del malific', 'clark
→'))

            query = "INSERT INTO ANSWERS (MESSAGE_ID,USERNAME,CONTENT,
→VOTES) VALUES (%s, %s, %s, %s)"
            cursor.execute(query, ('3', 'mayk', 'lorem ipsum dolar', '2'))

            query = "INSERT INTO ANSWERS (MESSAGE_ID,USERNAME,CONTENT,
→VOTES) VALUES (%s, %s, %s, %s)"
            cursor.execute(query, ('3', 'admin', 'hello buddy', '3'))

            query = "INSERT INTO ANSWERS (MESSAGE_ID,USERNAME,CONTENT,
→VOTES) VALUES (%s, %s, %s, %s)"
            cursor.execute(query, ('3', 'mayk', '42', '5'))

            connection.commit()

```

```
logout_user()
```

Kodda bulunan `current_user`, `dbapi2`, `current_app.config['dsn']`, `pwd_context`, `logout_user` komutları sırasıyla `flask_login`, `psycopg2`, “`server.py`” konfigürasyonu, `passlib`, `flask_login` kütüphaneleri veya dosyalarından elde edilmektedir.

2.2.1 Yunus Güngör tarafından tamamlanan parçalar

Uygulamada bulunan tüm tablo yapıları ve kodlar Yunus Güngör tarafından tamamlanmıştır.

Uygulamada 3 farklı varlık bulunmaktadır:

Varlıklar

- Kullanıcılar (Users)
- Mesajlar (Messages)
- Cevaplar (answers)

Kullanıcılar (Users)

Kullanıcılar varlığında bulunan özellikler:

- USERNAME PRIMARY KEY Her kullanıcıya has kullanıcı adı
- PASS Kullanıcıların şifresi, hash fonksiyonundan geçtikten sonra depolanır
- ISADMIN Kullanıcının admin kullanıcı olup olmadığını belirtir
- PICTURE Kullanıcı resminin adresini tutar
- BESTFRIEND Kullanıcının en iyi arkadaşını tutar

Kullanıcılar (Users) varlığında kullanılan sorgular:

- **Kullanıcı ekleme**

```
query = "SELECT PASS FROM USERS WHERE USERNAME = %s"
cursor.execute(query, (username,))
if(cursor.fetchall()):
    return 'User already exists'
query = "INSERT INTO USERS (USERNAME, PASS) VALUES (%s, %s)"
```

- **Kullanıcı bilgilerini görüntüleme**

```
SELECT * FROM USERS WHERE USERNAME = %s
```

- **Kullanıcı şifresini getirme**

```
query = "SELECT PASS FROM USERS WHERE USERNAME = %s"
cursor.execute(query, (username,))
password = cursor.fetchall()
if password is not None and len(password)==1:
    return password[0][0]
```

- **Kullanıcı şifresini değiştirme**

```
if form.validate_on_submit():
    hashedPass = pwd_context.encrypt(form.data['password'])
    db.update_pass(current_user.username,hashedPass)
    flash('Updated Password')
```

```
query = "UPDATE USERS SET PASS=%s WHERE USERNAME=%s"
cursor.execute(query, (password, username))
```

- **Kullanıcı fotoğrafını değiştirme**

```
query = "UPDATE USERS SET PICTURE=%s WHERE USERNAME=%s"
cursor.execute(query, (picture, username))
connection.commit()
```

- **Kullanıcıyı silme**

```
if current_user.is_admin and username != 'admin':
    db = Database()
    db.del_user(username)
    return redirect(url_for('site.home_page'))
```

```
DELETE FROM USERS WHERE USERNAME=%s
```

- **Kullanıcıyı admin kullanıcı yapma**

```
if current_user.is_admin and username != 'admin':
    db = Database()
    db.adminize(username)
    return redirect(url_for('site.home_page'))
```

```
UPDATE USERS SET ISADMIN=TRUE WHERE USERNAME=%s
```

- **Tüm kullanıcıları görüntüleme**

```
SELECT USERNAME FROM USERS
```

- **Kullanıcıları en iyi arkadaş olarak düzenleme**

```
query = "UPDATE USERS SET BESTFRIEND=%s WHERE USERNAME=%s "
cursor.execute(query, (username1, username2,))
cursor.execute(query, (username2, username1,))
```

- **Kullanıcıların en iyi arkadaşını görüntüleme**

```
SELECT BESTFRIEND FROM USERS WHERE USERNAME=%s
```

- **Kullanıcıyı arkadaş olarak ekleme**

```
query = "SELECT * FROM FRIENDS WHERE USERNAME = %s AND FRIEND=%s"
cursor.execute(query, (username1, username2,))
f1 = cursor.fetchall()
cursor.execute(query, (username2, username1,))
f2 = cursor.fetchall()
if (len(f1) <= 0 or len(f2) <= 0):
    #no relationship
    query = "INSERT INTO FRIENDS VALUES(%s, %s)"
    cursor.execute(query, (username1, username2,))
```

- **Kullanıcıyı arkadaşlıktan çıkarma**

```
query = "SELECT * FROM FRIENDS WHERE USERNAME = %s AND FRIEND=%s"
cursor.execute(query, (username1, username2,))
f1 = cursor.fetchall()
cursor.execute(query, (username2, username1,))
f2 = cursor.fetchall()
if (len(f1) > 0 or len(f2) > 0):
```

```
#confirmed friendship
query = "DELETE FROM FRIENDS WHERE USERNAME = %s AND FRIEND=%s"
cursor.execute(query, (username1, username2,))
```

- Kullanıcının arkadaşlarını görüntüleme

```
SELECT FRIEND FROM FRIENDS WHERE USERNAME = %s
```

Mesajlar (Messages)

Mesajlar varlığında bulunan özellikler:

- ID PRIMARY KEY Her mesaja özel olan kimlik numarası
- TITLE Mesajın başlığı
- CONTENT Mesajın içeriği (gövdesi)
- USERNAME Mesaja sahip olan kullanıcı

Mesajlar (Messages) varlığında kullanılan sorgular:

- Mesaj ekleme

```
INSERT INTO MESSAGES (TITLE, CONTENT, USERNAME) VALUES (%s, %s, %s)
```

- Mesaj görüntüleme

```
SELECT * FROM MESSAGES WHERE ID = %s
```

- Mesajların tamamını görüntüleme

```
SELECT * FROM MESSAGES LEFT OUTER JOIN USERS ON MESSAGES.USERNAME = USERS.
↪ USERNAME
```

- Mesajı silme

```
if current_user.is_admin:
    db=Database()
    db.del_message(message_id)
    return redirect(url_for('site.home_page'))
```

```
DELETE FROM MESSAGES WHERE ID=%s
```

- Mesajı Düzenleme

```
if current_user.is_admin:
    form = AddMessageForm()
    db=Database()
    message = db.get_message(message_id)
    if form.validate_on_submit():
        message = Message(form.data['title'], form.data['text'])
        message.id = message_id
        db.edit_message(message)
        flash('message edited')
        return redirect(url_for('site.home_page'))
    return render_template('message.html', form=form, message=message)
return redirect(url_for('site.home_page'))
```

```
UPDATE MESSAGES SET TITLE=%s, CONTENT=%s WHERE ID=%s
```

Cevaplar (Answers)

Cevaplar varlığında bulunan özellikler:

- ID PRIMARY KEY Her mesaja özel olan kimlik numarası
- MESSAGE_ID Cevabın ait olduğu mesajın kimlik numarası
- USERNAME Cevaba sahip olan kullanıcı
- CONTENT Cevabın içeriği
- VOTES Cevabın aldığı toplam oy sayısı

Cevaplar (Answers) varlığında kullanılan sorgular:

- Cevap ekleme

```
INSERT INTO ANSWERS (MESSAGE_ID, USERNAME, CONTENT) VALUES (%s, %s, %s)
```

- Cevap görüntüleme

```
SELECT * FROM ANSWERS INNER JOIN USERS ON ANSWERS.USERNAME=USERS.USERNAME_  
↪WHERE ID = %s
```

- Bir mesaja ait tüm cevapları tamamını görüntüleme

```
query = "SELECT * FROM MESSAGES"  
cursor.execute(query)  
messages = cursor.fetchall()  
answers = {}  
for message in messages:  
    query = "SELECT * FROM ANSWERS LEFT OUTER JOIN USERS ON ANSWERS.  
↪USERNAME=USERS.USERNAME WHERE MESSAGE_ID=%s ORDER BY VOTES DESC"  
    cursor.execute(query, (message[0],))  
    answer = cursor.fetchall()  
    answers[message[0]]=answer  
return answers
```

- Bir cevabın oy sayısını arttırma

```
UPDATE ANSWERS SET VOTES = VOTES + 1 WHERE ID=%s
```

- Cevabı silme

```
if current_user.is_admin:  
    db=Database()  
    db.del_answer(message_id)  
    return redirect(url_for('site.home_page'))
```

```
DELETE FROM ANSWERS WHERE ID=%s
```

- Cevabı Düzenleme

```
if current_user.is_admin:  
    form = AddAnswerForm()  
    db=Database()  
    answer = db.get_message_answer(answer_id)  
    print(answer.text)  
    if form.validate_on_submit():  
        answer = MessageAnswer(form.data['text'])  
        answer.id = answer_id  
        db.edit_message_answer(answer)  
        flash('answer edited')  
        return redirect(url_for('site.home_page'))
```

```
    return render_template('answer.html', form=form, answer=answer)
    return redirect(url_for('site.home_page'))
```

```
UPDATE ANSWERS SET CONTENT=%s WHERE ID=%s
```