

PYTHON TABANLI RANDEVU YÖNETİM SİSTEMİ

Hazırlayan

Bekir Sıddık Asil
701231018

Danışman

Doç. Dr. Ahmet Cüneyt Tantı

Giriş

Günümüzde her sektörde randevu yönetim sistemlerine ihtiyaç duyulmaktadır. Bu sistemler, işletmelerin müşterilerine daha iyi hizmet sunmalarını ve verimliliklerini artırmalarını sağlamaktadır. Özellikle süreçlerini manuel yöneten işletmelerde randevu yönetimi, büyük bir sorun olarak karşımıza çıkmaktadır. Bu işletmelerde müşteri bilgisi dahi kayıt edilmediği için bir müşteri veritabanı bulunmamaktadır. Bu da pazarlama faaliyetleri için büyük bir eksikliklerdir.

Bunun yanı sıra, bir çok küçük işletmenin randevu sistemi telefon üzerinden kişiler ile yönetilerek randevu alınmasına dayalıdır. Bu durumda müşterilerin tercih edebileceği farklı zaman dilimlerine ilişkin bilgileri bulunamadığı gibi randevularda karışıklık da olabilmektedir.

Bu konulara çözüm getirebilmek amacıyla, plug-in yapıya sahip bir SaaS ürünü dizayn etmek istedim. Bu proje sayesinde hayal ettiğim bu ürünün ilk adımlarını atma fırsatı buldum.

Projenin Amacı

Bu projenin amacı, küçük işletmeler için kolayca uygulanabilecek ve müşteri memnuniyetini artıracak bir randevu yönetim sistemi tasarlamaktır. Aynı zamanda eşimin sahibi olduğu moda evinin de benzer problemleri bulunmaktadır. Gerçek hayatta da kullanılmasını hedeflediğim bu proje ile ilk adımlarını attığım bu ürünü daha da ileriye taşıyarak ilerleyen süreçte hayata geçirmeyi amaçlamaktayım.

Sistem, aşağıdaki özellikleri içerecektir:

- Müşteri bilgilerinin kaydedilmesini sağlayacaktır.
- Tercih edilen güne ait farklı zaman dilimlerine ilişkin bilgileri sunacaktır.
- Randevu alma sürecini kolaylaştıracaktır.
- Randevularda karışıklık yaşanmasını önleyecektir.

Projenin Kapsamı

Bu projenin kapsamı, küçük işletmeler için kolayca uygulanabilecek ve müşteri memnuniyetini artıracak bir randevu yönetim sisteminin geliştirilmesini içermektedir. Sistem, aşağıdaki özellikleri içerecektir:

- **Müşteri kaydı ve müşteri bilgilerinin tutulması:** Sistem, müşteri bilgilerini kaydetmeyi ve saklamayı sağlayacaktır. Bu bilgiler, müşteri adı, soyadı, e-posta adresi, telefon numarası, adres vb. bilgileri içerecektir.
- **Randevu kaydı ve randevu bilgilerinin tutulması:** Sistem, randevuları kaydetmeyi ve saklamayı sağlayacaktır. Bu bilgiler, randevu tarihi, saati, yeri, müşteri adı, soyadı, randevu tipi vb. bilgileri içerecektir.

- **Randevu denenecek ürün tiplerinin listelenmesi ve randevu için müşteriye seçim yaptırılması:** Sistem, randevu için müşteriye farklı ürün tipleri sunacak ve müşteriden bir seçim yapmasını isteyecektir. Her randevu için maksimum bir ürün tipi seçilebilir.
- **Satıcı bilgisi ve satıcı-randevu eşleştirmesinin sistem tarafından yapılması:** Sistem, randevu için uygun bir satıcıyı belirleyecek ve randevuyu bu satıcıya atayacaktır.

İşlevsel Bağımlılıklar

Proje için tutulacak data aşağıdaki alanları içerecektir:

customer_id | customer_name | customer_surname | phone | email | appointment_id |
appointment_date | time_interval | product_type_id | product_type | product_type | salesperson_id
| salesperson_name | salesperson_surname

Buna göre işlevsel bağımlılıklar aşağıdaki gibi olacaktır:

- customer_id – {customer_name, customer_surname, phone, email}
- phone – {customer_id, customer_name, customer_surname, email}
- email – {customer_id, customer_name, customer_surname, email}
- appointment_id – {customer_id, appointment_date, time_interval, product_type_selection, salesperson_id}
- customer_id, appointment_date, time_interval – {salesperson_id}
- appointment_date, time_interval, salesperson_id – {customer_id}
- product_type_selection – {product_type_id, product_type}
- salesperson_id – {salesperson_name, salesperson_surname}

Normalizasyon Adımları

1. Normal Form:

Bütün veri alanları atomik formda tutulduğu için veri tabanı 1.Normal Form'a uygundur.

2. Normal Form:

Kısmi bağımlılıkları ortadan kaldırmak için adımlar atılmalıdır.

Bunun için appointment_date, time_interval, product_type_id, product_type, salesperson_id, salesperson_name, salesperson_surname alanları appointment_id'ye tamamen bağımlıdır.

Aynı şekilde customer_name, customer_surname, phone, email alanları da direk customer_id'ye bağımlıdır.

Bu nedenle, bu alanlar ayrı tablolara taşınmıştır. Yeni yapı aşağıdaki gibi olmuştur:

customers {customer_id, customer_name, customer_surname, phone, email}

appointments {appointment_id, appointment_date, time_interval, product_type_selection, salesperson_id}

product_type {product_type_id, product_type}

salesperson {salesperson_id, salesperson_name, salesperson_surname}

3. Normal Form:

3NF'e geçiş için geçişli işlevsel bağımlılıkları ortadan kaldırmak gerekir. phone ve email alanları customer_id'ye tamamen bağımlıdır. Bu nedenle bu alanları ayrı bir tabloya taşımak daha doğru olacaktır. Ancak projenin şu anki kapsamını düşündüğümüzde bu boyutta bir proje için bu yapının yeterli olabileceği düşünülmüştür. Ancak 3. Normal Form'a geçilecek olsaydı tablo yapısı aşağıdaki gibi olurdu:

customers {customer_id, customer_name, customer_surname}

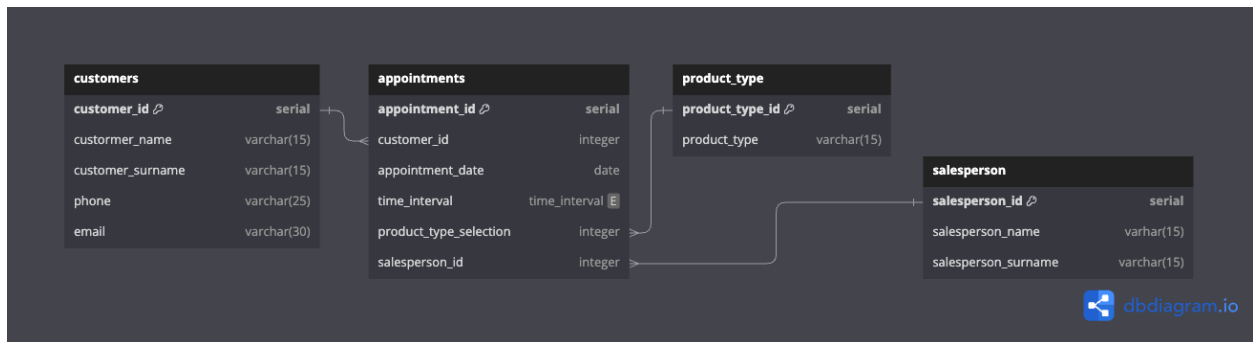
customers_info { customer_id, phone, email}

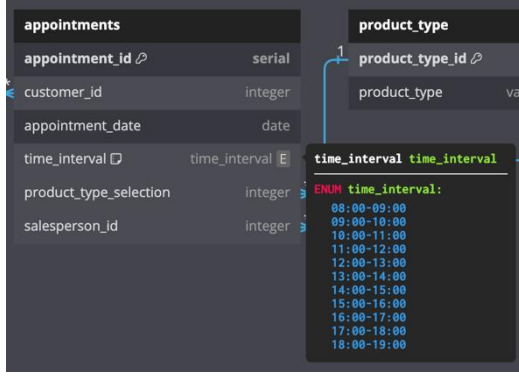
appointments {appointment_id, appointment_date, time_interval, product_type_selection, salesperson_id}

product_type {product_type_id, product_type}

salesperson {salesperson_id, salesperson_name, salesperson_surname}

Varlık İlişkisi Diyagramı (ERD)





time_interval alanındaki verilerin randevu süresinin bir saat olmasından ve çalışma saatlerinin belirli olmasından dolayı önden tanımlı olması gerekmekteydi bu nedenle bu alan enum olarak tanımlanmıştır.

Uygulama ile İlgili Detaylar

Uygulama python üzerinde geliştirilmiş olup bir gelinlik mağazasının randevu ve müşteri yönetim adımlarını ele almak üzere tasarlanmıştır. Daha da geliştirilerek öncelikle gerçek hayatta sahibi olduğumuz mağazanın randevu ve müşteri yönetimi geliştirilmek üzere önyüz bacağına eklenmesi planlanmaktadır. Zaman kısıtından dolayı önyüz geliştirmeleri eklenmemiştir.

Sırasıyla:

- Randevu kaydı olmayan müşterilerden öncelikle kişisel bilgiler alınır.
- Müşteri kaydı oluştuktan sonra randevu bilgileri alınır. Bu adımda girilen tarihteki rastgele atanan bir satıcının dolu olduğu saat aralıkları kullanıcıya listelenir ve uygun bir zaman aralığı için tercih yapması beklenir.
- Alınan bilgiler doğrultusunda randevu oluşturulur ve sonradan görüntülenmek istenirse telefon numarası bilgisi ile görüntüleme yapılabilir.

Proje ile İlgili Detaylar

Docker üzerinden ayağa kaldırılan bir postgresql DB sunucusu üzerinde postgresql Veri Tabanı oluşturulmuştur.

Python üzerinden psycopg2 ile Veri Tabanı'na bağlantı sağlanmış ve verinin görüntülenmesi, yeni verinin ilgili tablolara insert edilmesi sağlanmıştır.

Varlık İlişkisi Diyagramı (ERD) dbdiagram.io üzerinden hazırlanmıştır.

Kaynak Kodlar

1. bridal_shop.py → https://github.com/bekoasil/bridal_shop
2. bridal_shop.sql → https://github.com/bekoasil/bridal_shop

3. Proje Sunum Videosu: <https://youtu.be/DbD1SFybkQQ?si=SuNlqrYpWA21i-ye>