

Yemek Tarif Rehberi Uygulaması

1. Cihangir Emre ER
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
220202036@kocaeli.edu.tr

2. İsmail Özer
Bilgisayar Mühendisliği
Kocaeli Üniversitesi
Kocaeli, Türkiye
220202093@kocaeli.edu.tr

□

I. ÖZET

Bu çalışma, kullanıcıların yemek tariflerini ekleyebileceği, mevcut malzemeleri kullanarak yapılabilecek tarifleri filtreleyebileceği bir masaüstü uygulamasını içermektedir. Uygulama, dinamik arama, malzemeye göre tarif önerisi ve çeşitli filtreleme özellikleri sunmaktadır. Bu rapor, uygulamanın algoritmik yapısını, deneysel sonuçlarını ve geliştirilen özellikleri kapsamaktadır.

II. GİRİŞ

Bu projenin amacı, bir yemek tarif rehberi uygulaması oluşturarak, kullanıcıların tarifleri kolaylıkla ekleyip düzenlemesine ve mevcut malzemelerine göre yapılabilecek yemekleri keşfetmesine olanak tanımadır. Proje, kullanıcı dostu bir arayüz ile yemek tariflerinin verimli bir şekilde yönetilmesi ve kullanıcıya önerilerde bulunulması üzerine odaklanmıştır.

Bu çalışmanın amacı, yemek tariflerinin organize edilmesi ve kullanıcıların mevcut malzemelere göre yemek planlaması yapabilmesi için pratik bir çözüm sunmaktır. Yemek tarifi uygulaması, günlük yaşamda yemek hazırlama sürecini daha düzenli ve kolay hale getirmekte, aynı zamanda kullanıcıların zaman ve maliyet açısından tasarruf sağlamalarına yardımcı olmaktadır. Dijital ortamda tarif saklama, arama, filtreleme ve maliyet hesaplama gibi özellikleri bir araya getirerek, kullanıcıların mutfaktaki deneyimlerini geliştirmek hedeflenmiştir.

III. YÖNTEM

Uygulama, C# programlama dili kullanılarak geliştirilmiştir. Kullanıcı arayüzü basit ve anlaşılır olacak şekilde tasarlanmış olup, veritabanı yönetimi için SQL Server kullanılmıştır. Kullanıcılar tarifleri veritabanına kaydedebilir, tariflerin detaylarına erişebilir, tarifleri güncelleyebilir ve silebilirler. Ayrıca, malzeme sayısına, hazırlama süresine ve maliyet aralığına göre tarifler filtrelenir. Projede kullanılan algoritmalar, tarif ve malzeme yönetimi için optimize edilmiştir.

Projenin geliştirilmesi sırasında, uygulamanın işlevselliğini artırmak ve kullanıcı dostu bir deneyim sağlamak amacıyla birkaç temel aşama izlenmiştir. Bu aşamalar arasında veritabanı tasarımı, kullanıcı arayüzü geliştirilmesi, arama ve filtreleme işlevleri ile tarif öneri sistemi yer almaktadır.

Uygulama, kullanıcıların tarifleri dijital ortamda saklayabilmesi, mevcut malzemelere göre öneri alabilmesi ve tarif maliyetlerini hesaplayabilmesi amacıyla C# ve Entity Framework kullanılarak geliştirilmiştir.

1. Veritabanı Tasarımı

Uygulamanın temel veritabanı yapısı üç ana tabloyu içermektedir: Tarifler, Malzemeler ve Tarif-Malzeme İlişkisi. Bu yapı, tariflerin ve malzemelerin ayrı ayrı yönetilmesine ve tariflerin malzemelerle ilişkilendirilmesine olanak tanır.

- Tarifler Tablosu:** Bu tabloda her tarifin benzersiz bir kimliği (TarifID), adı (TarifAdı), kategorisi (Kategori), hazırlanma süresi (HazırlamaSuresi) ve talimatları (Talimatlar) tutulmaktadır. Tarifler yalnızca bir kategori ile ilişkilendirilmiştir (örneğin, "Ana Yemek", "Tatlı").
- Malzemeler Tablosu:** Malzemeler tablosu, her malzemenin benzersiz kimliğini (MalzemeID), adını (MalzemeAdı), toplam miktarını (ToplamMiktar), birimini (MalzemeBirim, örneğin "kg", "lt") ve birim fiyatını (BirimFiyat) içermektedir. Bu sayede tariflerin maliyetini hesaplamak mümkün hale gelmiştir.
- Tarif-Malzeme İlişkisi Tablosu:** Tarifler ve malzemeler arasındaki many-to-many ilişkiyi yönetmek amacıyla, TarifID ve MalzemeID'yi içeren bu tablo oluşturulmuştur. Her tarifteki malzemelerin miktarlarını tutmak için MalzemeMiktar alanı eklenmiştir. Bu yapı, aynı malzemenin birden fazla tarifte kullanılabilmesini ve her tarifte farklı miktarlarda olmasını sağlar.

Veritabanı, 3NF (Üçüncü Normal Form) kurallarına göre tasarlanmış ve normalizasyon uygulanarak veri tekrarı ve tutarsızlıkların önüne geçilmiştir [1].

2. Kullanıcı Arayüzü Geliştirme

Uygulamanın kullanıcı arayüzü, basit ve sezgisel olacak şekilde geliştirilmiştir. Ana ekran, tüm tariflerin listelendiği bir DataGridView bileşeni içermektedir. Bu liste, tarif adı, hazırlama süresi ve maliyet bilgilerini içerir ve kullanıcının herhangi bir tarifin detaylarını görüntülemesine olanak tanır.

Arayüzde ayrıca tarif ekleme, güncelleme ve silme işlemlerini gerçekleştiren butonlar bulunmaktadır. Kullanıcı, bir tarif eklerken tarif adını, kategorisini, hazırlama süresini ve malzemelerini girebilir. Malzeme ekleme işlemi sırasında, kullanıcı mevcut malzemeleri seçebilir veya yeni bir malzeme tanımlayabilir. Uygulamada, tariflerin yapılabilirliği durumuna göre renklendirme (yeşil: yapılabilir, kırmızı: yapılamaz) kullanılarak kullanıcıya görsel bir geri bildirim sunulmaktadır.

3. Arama ve Filtreleme İşlevleri

Arama ve filtreleme işlevleri, kullanıcının tarifleri daha hızlı ve etkili bir şekilde bulmasını sağlamak amacıyla geliştirilmiştir. Arama işlevi, tarif adı veya kullanılan malzemeler üzerinden yapılabilir. Filtreleme seçenekleri arasında tarifin kategorisi, hazırlama süresi ve maliyeti bulunmaktadır. Bu özellikler, kullanıcıların tarifleri özel kriterlere göre listelemesini mümkün kılar.

- Tarif Adına Göre Arama: Kullanıcı, tarif adını girerek doğrudan arama yapabilir.
- Malzemeye Göre Arama: Kullanıcı, ellerinde mevcut olan malzemeleri girdikten sonra, sistem bu malzemelerle yapılabilecek tarifleri önermektedir. Eşleşme yüzdesi hesaplanarak en uygun tarifler kullanıcıya sunulmaktadır.

PSEUDO KOD

Function GetAllRecipes():

Initialize empty list 'recipes'

Define SQL query to select recipe ID, name, and preparation time

Connect to the database using 'connectionString'
Execute SQL query

For each row in the result:

Get 'TarifID', 'TarifAdi', and 'HazirlamaSuresi'

Calculate the cost of the recipe using 'CalCost' function

Add the recipe to 'recipes' list

Return 'recipes'

End Function.

Function GetAllIngredients():

Initialize empty list 'ingredients'

Define SQL query to select ingredient names

Connect to the database using 'connectionString'

Execute SQL query

For each row in the result:

Add the ingredient name to 'ingredients' list

Return 'ingredients'

End Function

```
public partial class MainForm : Form
{
    private System.Windows.Forms.DataGridView dgvRecipes;

    1 reference
    public MainForm()
    {
        InitializeComponent();
        pictureBox1.SendToBack();
    }

    // Form yüklendiğinde kullanıcının karşısına çıkanlar
    1 reference
    private void MainForm_Load(object sender, EventArgs e)
    {
        AddMisCostColumn();
        dgvRecipes.ColumnHeaderMouseClick += dgvRecipes_ColumnHeaderMouseClick;
        cbCostRange.SelectedIndexChanged += cbCostRange_SelectedIndexChanged;
        LoadRecipes();
        LoadIngredients();
        LoadKategoriToCombobox();
        LoadCostRangesToCombobox();
    }
}
```

```
// Tarifleri görüntüler ve checkbox durumuna göre renklendirir
3 references
private void DisplayRecipes(List<dynamic> recipes)
{
    dgvRecipes.DataSource = null;

    // Eğer "MatchPercentage" sütununa göre sıralamak isteniyorsa
    if (recipes.Count > 0 && recipes.First().GetType().GetProperty("MatchPercentage") != null)
    {
        // MatchPercentage değerine göre büyükten küçüğe sıralıyoruz
        recipes = recipes.OrderByDescending(r => r.MatchPercentage).ToList();
    }

    // Veri kaynağını güncelle
    dgvRecipes.DataSource = recipes;

    if (recipes.Count > 0)
    {
        dgvRecipes.Columns["TarifID"].Visible = false;
        dgvRecipes.Columns["Maliyet"].HeaderText = "Maliyet";

        if (dgvRecipes.Columns.Contains("MatchPercentage"))
        {
            dgvRecipes.Columns["MatchPercentage"].HeaderText = "Eşleşme Yüzdesi";
        }

        if (dgvRecipes.Columns.Contains("Yapilabilir"))
        {
            dgvRecipes.Columns["Yapilabilir"].Visible = false;

            foreach (DataGridViewRow row in dgvRecipes.Rows)
            {
                bool yapilabilirMi = Convert.ToBoolean(row.Cells["Yapilabilir"].Value);
                decimal gerekenMaliyet = 0;

                // Eğer tarif yapılamazsa eksik malzeme maliyetini hesapla ve renklendir
                if (!yapilabilirMi)
                {
                    int tarifId = Convert.ToInt32(row.Cells["TarifID"].Value);
                    gerekenMaliyet = AppHelper.CalMisssinging(tarifId); // Eksik malzeme maliyetini hesapla
                    row.Cells["GerekenMaliyet"].Value = gerekenMaliyet;
                    row.DefaultCellStyle.BackColor = Color.Red; // Yapılamayan tarif kırmızıya boyanır
                }
                else
                {
                    row.Cells["GerekenMaliyet"].Value = DBNull.Value; // Yapılabilen tarifler için eksik maliyet yok
                    row.DefaultCellStyle.BackColor = Color.LightGreen; // Yapılabilen tarif yeşile boyanır
                }

                // Checkbox false ise tüm renkleri varsayılan beyaz olarak ayarla
                if (!cbshowSomeRecipes.Checked)
                {
                    row.DefaultCellStyle.BackColor = Color.White;
                }
            }
        }
    }
}
```

```
foreach (DataGridViewRow row in dgvRecipes.Rows)
{
    bool yapilabilirMi = Convert.ToBoolean(row.Cells["Yapilabilir"].Value);
    decimal gerekenMaliyet = 0;

    // Eğer tarif yapılamazsa eksik malzeme maliyetini hesapla ve renklendir
    if (!yapilabilirMi)
    {
        int tarifId = Convert.ToInt32(row.Cells["TarifID"].Value);
        gerekenMaliyet = AppHelper.CalMisssinging(tarifId); // Eksik malzeme maliyetini hesapla
        row.Cells["GerekenMaliyet"].Value = gerekenMaliyet;
        row.DefaultCellStyle.BackColor = Color.Red; // Yapılamayan tarif kırmızıya boyanır
    }
    else
    {
        row.Cells["GerekenMaliyet"].Value = DBNull.Value; // Yapılabilen tarifler için eksik maliyet yok
        row.DefaultCellStyle.BackColor = Color.LightGreen; // Yapılabilen tarif yeşile boyanır
    }

    // Checkbox false ise tüm renkleri varsayılan beyaz olarak ayarla
    if (!cbshowSomeRecipes.Checked)
    {
        row.DefaultCellStyle.BackColor = Color.White;
    }
}

// Gerektirse malzeme listesi ve diğer detayları ekleyebilirsiniz
1 reference
private void DisplayRecipeDetails()
{
    string connectionString = "Server=DESKTOP-B511VU1\\SQLEXPRESS;Database=RecipesApp;Integrated Security=True;";
    string recipeQuery = "SELECT TarifAdi, Kategori, HazirlamaSuresi, Talimatlar FROM Tarifs WHERE TarifID = @TarifID";
    string ingredientQuery = "SELECT m.MalzemeAdi, m.MalzemeMiktar FROM TarifMalzemes ta JOIN Malzemes m ON ta.MalzemeID = m.MalzemeID";

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlCommand recipeCommand = new SqlCommand(recipeQuery, connection);
        recipeCommand.Parameters.AddWithValue("@TarifID", _tarif.TarifID);

        SqlCommand ingredientsCommand = new SqlCommand(ingredientQuery, connection);
        ingredientsCommand.Parameters.AddWithValue("@TarifID", _tarif.TarifID);

        try
        {
            connection.Open();
            using (SqlDataAdapter reader = recipeCommand.ExecuteReader())
            {
                if (reader.Read())
                {
                    lblTarifAdi.Text = reader["TarifAdi"].ToString();
                    lblKategori.Text = reader["Kategori"].ToString();
                    lblHazirlamaSuresi.Text = reader["HazirlamaSuresi"].ToString();
                    lblMaliyet.Text = AppHelper.CalCost(_tarif.TarifID).ToString("N2");
                    txtTalimatlar.Text = reader["Talimatlar"].ToString();
                }
            }
        }
        catch { }
    }
}
```

Tarfı Araması

Tarfı İsmi

Kategori

Maliyet

Malzeme Sayısı

▼

▼

0

Malzemeye göre arama

dbMalzeme

1. Kullanılan Algoritmalar:

2. **Arama Algoritması:** Kullanıcının tarif adı veya malzemelere göre arama yapabilmesi için SQL sorguları optimize edilmiştir. Her aramada, tariflerin malzemeleri ile girilen malzemeler karşılaştırılır ve eşleşme yüzdesi hesaplanarak listelenir.
3. **Tarif Öneri Algoritması:** Veritabanındaki tariflerin malzeme stoklarına göre önerilmesi işlemi gerçekleştirilir. Eğer tarif için gerekli malzemeler mevcut değilse, eksik malzemelerin maliyeti hesaplanır ve kullanıcıya gösterilir.
4. **Filtreleme ve Sıralama Algoritması:** Kullanıcı, tarifleri belirli kriterlere göre sıralayabilir ve filtreleyebilir. SQL sorguları, kullanıcının seçtiği kriterlere göre veritabanından verileri çekmek üzere dinamik olarak oluşturulur.

```
using (SqlConnection connection = new SqlConnection(connectionString))
{
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@SearchText", "*" + searchText + "*");

    for (int i = 0; i < selectedIngredients.Count; i++)
    {
        command.Parameters.AddWithValue("@ingredient" + i, selectedIngredients[i]);
    }

    if (!string.IsNullOrEmpty(selectedCategory) && selectedCategory != "Seçiniz")
    {
        command.Parameters.AddWithValue("@SelectedCategory", selectedCategory);
    }

    if (minCost > 0 && maxCost > 0)
    {
        command.Parameters.AddWithValue("@minCost", minCost);
        command.Parameters.AddWithValue("@maxCost", maxCost);
    }
}
```

```

// Filter out popular (common, maligned) and fit for use filters, variable filter()
public static List<Recipe> ApplyFilters(List<Recipe> searchResult, List<Recipe> selectedIngredients, bool isOnlyProducible, string selectedCategory, int minCount, int maxCount)
{
    List<Recipe> filteredRecipes = new List<Recipe>();
    string query = "
        SELECT RecipeID, t.TariffID, t.MariaLacSuroci, t.Category
        FROM Tariffs t
        JOIN TariffsMalness ON t.TariffID = m.TariffID
        JOIN Malness ON m.MalnessID = n.MalnessID
        WHERE t.TariffID LIKE @searchFilter";

    if (selectedIngredients.Count > 0)
    {
        query += " AND MalnessID IN (" + string.Join(", ", selectedIngredients.Select(x, i) => $"Ingredient" + i)) + ")";
    }

    if (string.IsNullOrEmpty(selectedCategory) && selectedCategory != "Begin")
    {
        query += " AND t.Category = @selectedCategory";
    }

    if (minCount > 0 && maxCount > 0)
    {
        query += " AND (SELECT SUM(m.MalnessMltzr + m.StrangMtzr) FROM TariffsMalness t JOIN Malness ON m.MalnessID WHERE m.TariffID = t.TariffID) BETWEEN @minCount AND @maxCount";
    }

    query += " GROUP BY t.TariffID, t.TariffID, t.MariaLacSuroci, t.Category";
}

```

Tarif A/B

Kategori

Maksimalisasi Ruang

Tarif

Maksimalisasi Ruang

0

Tarif

Tarif Adı : Makama	Malzemeler
Kategori : Ana Yemek	Malzeme: Domates
Hazırlanma Süresi (dk) : 15	Miktar: 0.2
Talimatlar :	Sil
Makama kaynar suda haşlanır ve kızdırılır. Hazırlanan sos ile karıştırılır ve sıcak servis edilir.	Malzeme: Un
	Miktar: 0.4
	Sil
total	Kaydet

```

        }
        public static decimal CalMissingIng(int tarifId)
        {
            decimal missingCost = 0;

            string query = @"
SELECT SUM((tm.MalzemeMiktari - COALESCE(m.ToplamMiktari, 0)) * m.BirimFiyat) AS MissingCost
FROM TarifMalzemes tm
JOIN Malzemes m ON tm.MalzemeId = m.MalzemeId
WHERE tm.TarifId = @TarifId
AND m.ToplamMiktari < tm.MalzemeMiktari";

            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                SqlCommand command = new SqlCommand(query, connection);
                command.Parameters.AddWithValue("@TarifID", tarifId);

                try
                {
                    connection.Open();
                    var result = command.ExecuteScalar();

                    missingCost = result != null ? Math.Round(Convert.ToDecimal(result), 2) : 0;
                }
                catch (Exception ex) { Console.WriteLine(ex.Message); }
            }

            return missingCost; //Hep sıfır oluyor koda tekrar bak
        }
    }
}

```

```

        }
        else
        {
            MessageBox.Show("Tarif bulunamadı.");
        }
    }

    using (SqlDataReader ingredientReader = ingredientsCommand.ExecuteReader())
    {
        listBoxIngredients.Items.Clear();
        while (ingredientReader.Read())
        {
            string malzemeAdi = ingredientReader["MalzemeAdi"].ToString();
            string malzemeMiktar = ingredientReader["MalzemeMiktar"].ToString();
            listBoxIngredients.Items.Add($"{malzemeAdi} - {malzemeMiktar}");
        }
    }

    catch (Exception ex)
    {
        MessageBox.Show("Veritabanı bağlantısı hatası: " + ex.Message);
    }
}

```

Menu

Tarif Ekle

Tarif Güncelle

Tarif Sil

	Gereken Maliyet	Tarif Adı	Hazırlama Süresi	Maliyet	Eğleme Yüzdesi
▶		Kamıyank	60	974,5	100
		Mantı	120	47	100

IV. DENEYSEL SONUÇ

Uygulama, çeşitli senaryolarda test edilmiştir. Kullanıcılar, mevcut tarifleri ve malzemeleri yönetirken arama ve filtreleme işlemlerinde hızlı geri dönüş almıştır. Eksik malzemeli tarifler kırmızı renkle işaretlenmiş ve kullanıcıların tarif önerileri dinamik olarak güncellenmiştir. Kullanıcı arayüzü, sezgisel tasarımıyla kullanıcılar tarafından kolaylıkla anlaşılmıştır.

V. SONUÇ

Bu çalışma, yemek tarifleri yönetimi ve öneri sistemi konularında pratik bir masaüstü çözümü sunmaktadır. Dinamik filtreleme ve arama özellikleri, kullanıcıların tarifleri daha kolay bulmalarını sağlamıştır. Uygulamanın gelecekte daha fazla veri işleme kapasitesi eklenerek genişletilmesi mümkündür.

KAYNAKLAR

- [1] 1. Udemy. Microsoft SQL Server for Beginners [Online Video Eğitimi]. Udemy Platformu, erişim: <https://www.udemy.com/>
- [2] 2. YouTube. C# Programming for Beginners - Full Course [Video Serisi]. Erişim: <https://www.youtube.com/>
- [3] 3. YouTube. SQL Tutorial - Full Database Course for Beginners [Video Serisi]. The Net Ninja Kanalı, erişim: <https://www.youtube.com/>
- [4] 4. Microsoft Learn. Entity Framework Core ile Veritabanı İşlemleri [Çevrimiçi Eğitim]. Microsoft, erişim: <https://learn.microsoft.com/>
- [5] 5. YouTube. Creating a Database in SQL Server [Video Eğitim]. Programming with Mosh Kanalı, erişim: <https://www.youtube.com/>
- [6] 6. W3Schools. SQL Tutorial. Erişim: <https://www.w3schools.com/sql/>
- [7] 7. YouTube. C# Windows Forms Application Tutorial for Beginners [Video Serisi]. Erişim:

