

RESTORAN YEMEK TAKİP PROJESİ (REYETAP)

C programlama dilini kullanarak yemek siparişi ve yönetimi sağlayabilecek bir Restoran Yemek Takip Projesi (ReYeTaP) gerçekleştirilmiştir. ReYeTaP projesi 2 farklı kısımdan oluşmaktadır.

1. Müşteri.exe

1.1. Giriş

Bu kısım kullanıcıların yemek siparişi vermelerine, mevcut sipariş durumlarını kontrol etmelerine ve önceki siparişlerini görüntülemelerine olanak tanımaktadır. Bu program "stdio.h", "stdlib.h", "time.h", "string.h" ve "dirent.h" kütüphaneleri kullanılarak gerçekleştirilmiştir.

1.3. Programın Kodu ve Açıklamaları

```
7  typedef struct {
8      int fiyat, hazirlamaSuresi;
9      char yemekAdi[15];
10     char kullanıcı[20];
11     char siparisZamaniStr[25]; // Siparis zamanini tutacak dizi
12     char hazirlanmaZamaniStr[25]; // Hazirlanma zamanini tutacak dizi
13     char siparisID[25];
14     char Mevcut[15];
15     char Durum[10];
16 } siparis;
```

Bu yapı, bir siparişin fiyat, hazırlama süresi, yemek adı, kullanıcı adı, sipariş zamanı, hazırlık zamanı, sipariş ID'si, mevcut durumu ve sipariş durumu gibi bilgilerini saklar.

```
30 int main() {
31
32     int secim = anaMenu();
33
34     while (secim != 0) {
35         switch (secim) {
36             case 1: yeniSiparis("yemeklistesi.txt"); break;
37             case 2: mevcutSiparis("siparisler.txt"); break;
38             case 3: oncekiSiparis("siparisler.txt"); break;
39             case 0: break;
40             default: printf("Hatali secim!\n"); break;
41         }
42         secim = anaMenu();
43     }
44     return 0;
}
```

Ana fonksiyon, kullanıcının menüden bir seçim yapmasını sağlar ve seçime göre ilgili fonksiyonu çağırır. Kullanıcı 0 girene kadar döngü devam eder.

1.3.1. Ana Menü Fonksiyonu

Ana menü fonksiyonu, kullanıcının yapabileceği işlemleri listeler ve kullanıcıdan bir seçim yapmasını ister. Aşağıda ana menüyü kullanıcıya gösteren fonksiyon ve çıktısı verilmiştir.

```

44 int anaMenu() {
45     int secim;
46     printf("\nRestoran Yemek Takip Projesi (ReYeTaP)\n");
47     printf("1. Yeni Siparis\n");
48     printf("2. Mevcut Siparis Durumu\n");
49     printf("3. Daha Onceki Siparislerim\n");
50     printf("0. Cikis\n");
51
52     printf("Seciminizi yapiniz: ");
53     scanf("%d", &secim);
54
55     system("cls");
56
57     return secim;
58 }

```

```

Restoran Yemek Takip Projesi (ReYeTaP)
1. Yeni Siparis
2. Mevcut Siparis Durumu
3. Daha Onceki Siparislerim
0. Cikis
Seciminizi yapiniz:

```

1.3.2. Yeni Sipariş Fonksiyonu

Bu fonksiyon, kullanıcıdan bir yemek seçmesini ve siparişini onaylamasını sağlar. Sipariş onaylandıktan sonra, sipariş detayları "siparişler.txt" adlı dosyaya yazılır.

```

60 void yeniSiparis(const char *dosyaYolu) {
61     int numara = 0;
62     char kullanıcı[20];
63     char onay;
64     int yemekbulundu = 0;
65     int sıra = 1;
66     int mevcutSiparis = 0;
67     int hazirlamaSuresi = 0;

```

Burada fonksiyon içinde kullanılacak değişkenler tanımlanıyor ve başlangıç değerleri atanıyor. Bu kısımda kullanıcıdan siparişi alınır ve kullanıcıdan bir kullanıcı adı istenir. Kullanıcı girdiği bu ad ile mevcut siparişlerini ve geçmiş siparişlerini kontrol edebilir. Bu fonksiyonun çıktısı aşağıda verilmiştir.

```

Yemek listesi:
NO      YEMEK ADI      FIYAT      HAZIRLANMA SURESI
1        Lahmacun      75TL
2          Manti      170TL
3          Tost        60TL
4    TavukDurum      80TL
5        EtDurum      160TL
6      menemen        15TL
7      makarna        130TL
Kullanici adinizi giriniz: elif
Siparis numarasini giriniz, bitirmek icin -1 tuslayiniz : 1
Siparisi onayliyor musunuz? (E/H): e
Siparis onaylandi.
Siparis numarasini giriniz, bitirmek icin -1 tuslayiniz : |

```

```

190 void olusturSiparisID(char *siparisID, int siparisNumarasi) {
191     time_t simdiki_zaman;
192     struct tm *zaman_bilgisi;
193     char tarih_dizisi[10];
194     char numara_dizisi[4];
195
196     // Sı anki zamanı al
197     time(&simdiki_zaman);
198     zaman_bilgisi = localtime(&simdiki_zaman);
199
200     // Tarih bilgisini biçimlendir
201     strftime(tarih_dizisi, sizeof(tarih_dizisi), "%d%m%y", zaman_bilgisi);
202
203     // Siparis numarasına dizeye dönüştür
204     sprintf(numara_dizisi, "%03d", siparisNumarasi);
205
206     // Siparis ID'sini olustur
207     strcpy(siparisID, "SIP");
208     strcat(siparisID, tarih_dizisi);
209     strcat(siparisID, "_");
210     strcat(siparisID, numara_dizisi);
211 }

```

Bu fonksiyon, mevcut tarih ve sipariş numarasını kullanarak bir sipariş ID'si oluşturur. Bu id numarası ile sipariş edilen her yemeğe ait bilgiler (yemek adı, fiyatı, yemek sipariş edilme zamanı, yemek hazırlanma zamanı, kullanıcı bilgisi) tutulacaktır. Bu, siparişlerin benzersiz şekilde tanımlanmasını ve takip edilmesini sağlar.

```

213 int kullanıcıAdiKontrolu(const char *dosyaYolu, const char *kullanici) {
214     FILE *dosya = fopen(dosyaYolu, "r");
215     if (dosya != NULL) {
216         while (fscanf(dosya, "%s %s %d %s %s %s", s1.siparisID, s1.yemekAdi, &s1.fiyat, s1.siparisZamaniStr, s1.hazirlanmaZamaniStr, s1.Durum, s1.kullanici) == 7) {
217             if (strcmp(kullanici, s1.kullanici) == 0) {
218                 fclose(dosya);
219                 return 1; // Kullanıcı adı zaten kullanılmış
220             }
221         }
222         fclose(dosya);
223         return 0; // Dosyada kullanıcı adı bulunamadı
224     } else {
225         perror("Dosya açilamadi!");
226         return -1; // Dosya açilamadi
227     }
228 }

```

Yukarıda verilen fonksiyon ile kullanıcı girdiği kullanıcı adı ile işlemlerini gerçekleştirebilecektir. Girilen kullanıcı adı daha önce kullanılmışsa kullanıcıdan yeni bir kullanıcı adı istenecektir.

```

224 // Kullanıcı adı ve parola kontrol fonksiyonu
225 int parolaKontrolu(const char *dosyaYolu, const char *kullanici, const char *parola) {
226     FILE *dosya = fopen(dosyaYolu, "r");
227     if (dosya != NULL) {
228         char kayitliKullanici[20];
229         char kayitliParola[20];
230         while (fscanf(dosya, "%s %s", kayitliKullanici, kayitliParola) == 2) {
231             if (strcmp(kayitliKullanici, kullanici) == 0 && strcmp(kayitliParola, parola) == 0) {
232                 fclose(dosya);
233                 return 1;
234             }
235         }
236         fclose(dosya);
237     }
238     return 0;
239 }

```

Bu kısımda her bir kullanıcıdan sipariş alırken parola istenir ve kullanıcı bu parola ile işlemlerini gerçekleştirir. Fonksiyonun çıktısı aşağıda verilmiştir.

```

Yemek listesi:
NO      YEMEK ADI      FIYAT      HAZIRLANMA SURESI
1       Lahmacun      75TL       20dk
2       Manti         170TL      45dk
3       Tost          60TL       10dk
4       TavukDurum     80TL       15dk
5       EtDurum       160TL      15dk
6       menemen       15TL       60dk
7       makarna       130TL      35dk
Kullanici adinizi giriniz: elif
Parolanizi giriniz: 414877
Siparis numarasini giriniz, bitirmek icin -1 tuslayiniz : 1
Siparisi onayliyor musunuz? (E/H): e
Siparis onaylandi.
Siparis numarasini giriniz, bitirmek icin -1 tuslayiniz :

```

1.3.3. Mevcut Siparişler

Kullanıcı mevcut siparişlerini görüntülemek isterse oluşacak çıktı ve ilgili fonksiyon aşağıda verilmiştir.

```
230 void mevcutSiparis(const char *dosyaYolu) {
231     char kullanıcı[20];
232     int toplam=0;
233     printf("Kullanıcı adınızı giriniz: ");
234     scanf("%15s", kullanıcı);
235
236     FILE *siparisDosyasi = fopen(dosyaYolu, "r");
237     if (siparisDosyasi != NULL) {
238         time_t simdikiZaman;
239         time(&simdikiZaman);
240         int kullanıcıBulundu = 0;
241         struct tm *zaman_bilgisi = localtime(&simdikiZaman);
242
243         printf(" yemek adi\tfiyat \t tahmini teslim durum\n");
244         while (fscanf(siparisDosyasi, "%s %s %d %s %s %s", s1.siparisID, s1.yemekAdi, &s1.fiyat, s1.siparisZamaniStr, s1.hazirlanmaZamaniStr, s1.Durum, s1.kullanici) == 7) {
245             if (strcmp(kullanici, s1.kullanici) == 0) {
246                 int yil, ay, gun, saat, dakika, saniye;
247                 if (sscanf(s1.hazirlanmaZamaniStr, "%d.%d.%d-%d.%d.%d", &yil, &ay, &gun, &saat, &dakika) == 5) {
248                     struct tm hazirlanmaZamani = {0};
249                     hazirlanmaZamani.tm_year = yil - 1900;
250                     hazirlanmaZamani.tm_mon = ay - 1;
251                     hazirlanmaZamani.tm_mday = gun;
252                     hazirlanmaZamani.tm_hour = saat;
253                     hazirlanmaZamani.tm_min = dakika;
254                     hazirlanmaZamani.tm_sec = 0;
255
256                     time_t hazirlanmaZamaniEpoch = mktime(&hazirlanmaZamani);
257
258                     if (hazirlanmaZamaniEpoch != -1) {
259                         double zamanFarki = difftime(hazirlanmaZamaniEpoch, simdikiZaman);
260
261                         if (zamanFarki > 0) {
262                             printf("%10s %8dTL %20s %10s\n", s1.yemekAdi, s1.fiyat, s1.hazirlanmaZamaniStr, s1.Durum);
263                             kullanıcıBulundu = 1;
264                             toplam+=s1.fiyat;
265                         }
266                     }
267                 }
268             }
269         }
270     }
271 }
```

```
Kullanıcı adınızı giriniz: elif
yemek adi      fiyat    tahmini teslim   durum
Lahmacun       75TL      2024.05.25-12.47 beklemeye
sepet tutarı: 75
```

siparisinizin hazirlanma suresi restoranimizin yogunluguna gore degisiklik gosterebilir.

Restoran Yemek Takip Projesi (ReYeTaP)

1. Yeni Siparis
 2. Mevcut Siparis Durumu
 3. Daha Onceki Siparislerim
 0. Cikis
- Seciminizi yapiniz: |

1.3.4. Önceki Siparişler

Kullanıcı önceki siparişlerini görüntülemek isterse oluşacak çıktı ve ilgili fonksiyon aşağıda verilmiştir.

```
281 void oncesiSiparis(const char *dosyaYolu) {
282     char kullanıcı[20];
283     int kullanıcıBulundu = 0;
284     printf("Kullanıcı adınızı giriniz: "); scanf("%s", kullanıcı);
285
286     FILE *siparisDosyasi = fopen(dosyaYolu, "r");
287     if (siparisDosyasi != NULL) {
288         time_t simdikiZaman;
289         time(&simdikiZaman);
290         while (fscanf(siparisDosyasi, "%s %s %d %s %s %s", s1.siparisID, s1.yemekAdi, &s1.fiyat, s1.siparisZamaniStr, s1.hazirlanmaZamaniStr, s1.Durum, s1.kullanici) == 7) {
291             if (strcmp(kullanici, s1.kullanici) == 0) {
292                 int yil, ay, gun, saat, dakika, saniye;
293                 if (sscanf(s1.hazirlanmaZamaniStr, "%d.%d.%d-%d.%d.%d", &yil, &ay, &gun, &saat, &dakika) == 5) {
294                     struct tm hazirlanmaZamani = {0};
295                     hazirlanmaZamani.tm_year = yil - 1900;
296                     hazirlanmaZamani.tm_mon = ay - 1;
297                     hazirlanmaZamani.tm_mday = gun;
298                     hazirlanmaZamani.tm_hour = saat;
299                     hazirlanmaZamani.tm_min = dakika;
300                     hazirlanmaZamani.tm_sec = 0;
301
302                     time_t hazirlanmaZamaniEpoch = mktime(&hazirlanmaZamani);
303
304                     if (hazirlanmaZamaniEpoch != -1) {
305                         double zamanFarki = difftime(simdikiZaman, hazirlanmaZamaniEpoch);
306                         if (zamanFarki > 0) {
307                             printf("%15s %8dTL %20s\n", s1.yemekAdi, s1.fiyat, s1.siparisZamaniStr);
308                             kullanıcıBulundu = 1;
309                         }
310                     }
311                 }
312             }
313         }
314         if (!kullaniciBulundu) {
315             printf("Gecmis siparisiniz bulunmamaktadir!\n");
316         }
317         fclose(siparisDosyasi);
318     } else {
319         perror("Siparis dosyasi acilamadi!");
320     }
321 }
```

```
Kullanici adinizi giriniz: elif
Lahmacun      75TL      2024.05.25-12.27

Restoran Yemek Takip Projesi (ReYeTaP)
1. Yeni Siparis
2. Mevcut Siparis Durumu
3. Daha Onceki Siparislerim
0. Cikis
Seciminizi yapiniz: |
```

2. Restoran.exe

2.1. Giriş

Bu kısım, bir restoranın yönetim sistemine ilişkin kapsamlı bir analiz sunmaktadır. Restoran yönetim sistemi; yemeklerin eklenmesi, güncellenmesi, silinmesi, siparişlerin onaylanması veya reddedilmesi, günlük ve aylık kazançların hesaplanması, müşteri ve yemek analizleri gibi birçok fonksiyonu içermektedir. Bu kısımda, sistemin işleyişi ve kullanıcıya sağladığı avantajlar üzerinde durulacaktır. Bu program “stdio.h”, “stdlib.h”, “time.h”, “string.h” ve “dirent.h” kütüphaneleri kullanılarak gerçekleştirilmiştir.

2.2 Programın Kodu ve Açıklamaları

2.2.1 Ana Menü Fonksiyonu

Ana menü fonksiyonu restoran yönetimine yapabileceği işlemleri listeler ve kullanıcıdan bir seçim yapmasını ister. Aşağıda ana menü fonksiyonu ve çıktısı verilmiştir.

```
140 int anaMenu() {
141     int secim;
142     printf("Restoran Yonetim Sistemi\n");
143     printf("1. Yemek Ekle-Guncelle-Sil\n");
144     printf("2. Yeni Siparis ONAY/RED\n");
145     printf("3. Rapor Al\n");
146     printf("4. Rapor Goruntule\n");
147     printf("5. Analizler\n");
148     printf("6. Mutfak\n");
149     printf("0. Cikis\n");
150     printf("Seciminizi giriniz: "); scanf("%d", &secim);
151     system("cls");
152     return secim;
153 }
```

```
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

2.2.2 Yemek Ekle-Güncelle-Sil Fonksiyonu

Restoran yönetimi istediği zaman yemek listesine yeni bir yemek ekleyebilir, mevcut yemek bilgilerini güncelleyebilir veya listede yer alan bir yemeği silebilir.

```
154 int ekleguncellesil(){
155     int secim_;
156     printf("1. Yemek Ekle\n");
157     printf("2. Yemek Guncelle\n");
158     printf("3. Yemek Sil\n");
159     printf("0. Anamenu\n");
160     printf("Seciminizi giriniz: "); scanf("%d", &secim_);
161     system("cls");
162     return secim_;
163 }
164 }
```

```
1. Yemek Ekle
2. Yemek Guncelle
3. Yemek Sil
0. Anamenu
Seciminizi giriniz:
```

“Yemek Ekle” seçeneği seçildiğinde ilgili fonksiyon ve oluşan çıktı aşağıda verilmiştir.

```
176 void yemekEkle(const char *dosyaYolu) {
177
178     printf("Yemek adini giriniz: ");
179     scanf("%49s", sl.yemekAdi);
180     printf("Yemek fiyatini giriniz: ");
181     scanf("%d", &sl.fiyat);
182     printf("Yemek hazirlama suresini giriniz (dakika): ");
183     scanf("%d", &sl.hazirlamaSuresi);
184     printf("Yemek durumunu giriniz (mevcut/mevcut degil): ");
185     scanf("%15s", sl.Mevcut);
186
187     // Dosyaya yazma islemi
188     FILE *dosya = fopen(dosyaYolu, "a+");
189     if (dosya == NULL) {
190         perror("Dosya acilamadi!");
191         exit(EXIT_FAILURE);
192     }
193     fprintf(dosya, "%s %d %d %s\n", sl.yemekAdi, sl.fiyat, sl.hazirlamaSuresi, sl.Mevcut);
194     fclose(dosya);
195     printf("Yemek basariyla eklendi.\n");
196 }
197 }
```

```
Yemek adini giriniz: pide
Yemek fiyatini giriniz: 90
Yemek hazirlama suresini giriniz (dakika): 20
Yemek durumunu giriniz (mevcut/mevcut degil): mevcut
Yemek basariyla eklendi.
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz:
```

“Yemek Güncelle” seçeneğini seçildiğinde kullanıcı mevcut yemek bilgilerini (yemek adı, fiyatı, hazırlanma süresi, durumu) güncelleyebilecektir. İlgili fonksiyon ve çıktı aşağıda verilmiştir.

```

199 void yemekGuncelle(const char *dosyaYolu) {
200     char arananYemek[50];
201     int bulundu = 0;
202     int sira = 1;
203
204     printf("Yemek listesi:\n");
205     printf("%2s %16s %15s %22s %10s\n", "NO", "YEMEK ADI", "FIYAT", "HAZIRLANMA SURESI", "DURUM");
206
207     FILE *dosya = fopen(dosyaYolu, "r");
208     if (dosya != NULL) {
209         while (fscanf(dosya, "%49s %d %d %15s", sl.yemekAdi, &sl.fiyat, &sl.hazirlamaSuresi, sl.Mevcut) == 4) {
210             printf("%2d %16s %13dTL %18ddk %13s\n", sira, sl.yemekAdi, sl.fiyat, sl.hazirlamaSuresi, sl.Mevcut);
211             sira++;
212         }
213         fclose(dosya);
214     } else {
215         perror("Dosya acilamadi!");
216         return;
217     }
218
219     printf("Guncellenecek yemek adini giriniz: ");
220     scanf("%49s", arananYemek);
221
222     dosya = fopen(dosyaYolu, "r");
223     if (dosya == NULL) {
224         perror("Dosya acilamadi!");
225         exit(EXIT_FAILURE);
226     }
227
228     FILE *tempDosya = fopen("temp.txt", "w");
229     if (tempDosya == NULL) {
230         perror("Gecici dosya acilamadi!");
231         fclose(dosya);
232         exit(EXIT_FAILURE);
233     }
234
235     while (fscanf(dosya, "%49s %d %d %15s", sl.yemekAdi, &sl.fiyat, &sl.hazirlamaSuresi, sl.Mevcut) == 4) {
236         if (strcmp(sl.yemekAdi, arananYemek) == 0) {
237             printf("Yeni fiyatini giriniz: ");
238             scanf("%d", &sl.fiyat);
239             printf("Yeni hazirlama suresini giriniz (dakika): ");
240             scanf("%d", &sl.hazirlamaSuresi);
241             printf("Yeni durumunu giriniz (mevcut/mevcutdegil): ");
242             scanf("%15s", sl.Mevcut);
243             bulundu = 1;
244         }
245     }

```

```

Yemek listesi:
NO      YEMEK ADI      FİYAT      HAZIRLANMA SURESI      DURUM
1       Lahmacun      75TL       20dk      mevcut
2       Hamburger     250TL      30dk      mevcutdegil
3       Manti         170TL      45dk      mevcut
4       Tost          60TL       10dk      mevcut
5       TavukDurum    80TL       15dk      mevcut
6       EtDurum       160TL      15dk      mevcut
7       menemen       15TL       60dk      mevcut
8       makarna       130TL      35dk      mevcut
9       pide          90TL       20dk      mevcut

Guncellenecek yemek adini giriniz: pide
Yeni fiyatini giriniz: 100
Yeni hazirlama suresini giriniz (dakika): 25
Yeni durumunu giriniz (mevcut/mevcutdegil): mevcut

```

“Yemek Sil” seçeneğinde ise kullanıcı istediği yemeği yemek listesinden silebilecektir.

```

264 void yemekSil(const char *dosyaYolu) {
265     char silinecekYemek[50];
266     int sira = 1;
267
268     printf("Yemek listesi:\n");
269     printf("%2s %16s %15s %22s %10s\n", "NO", "YEMEK ADI", "FIYAT", "HAZIRLANMA SURESI", "DURUM");
270
271     FILE *dosya = fopen(dosyaYolu, "r");
272     if (dosya != NULL) {
273         while (fscanf(dosya, "%49s %d %d %15s", sl.yemekAdi, &sl.fiyat, &sl.hazirlamaSuresi, sl.Mevcut) == 4) {
274             printf("%2d %16s %13dTL %18ddk %13s\n", sira, sl.yemekAdi, sl.fiyat, sl.hazirlamaSuresi, sl.Mevcut);
275             sira++;
276         }
277         fclose(dosya);
278     } else {
279         perror("Dosya acilamadi!");
280         return;
281     }
282
283     printf("Silinecek yemek adini giriniz: ");
284     scanf("%49s", silinecekYemek);
285
286     dosya = fopen(dosyaYolu, "r");
287     if (dosya == NULL) {
288         perror("Dosya acilamadi!");
289         exit(EXIT_FAILURE);
290     }
291
292     FILE *tempDosya = fopen("temp.txt", "w");
293     if (tempDosya == NULL) {
294         perror("Gecici dosya acilamadi!");
295         fclose(dosya);
296         exit(EXIT_FAILURE);
297     }
298
299     int bulundu = 0;
300     while (fscanf(dosya, "%49s %d %d %15s", sl.yemekAdi, &sl.fiyat, &sl.hazirlamaSuresi, sl.Mevcut) == 4) {
301         if (strcmp(sl.yemekAdi, silinecekYemek) != 0) {
302             fprintf(tempDosya, "%s %d %d %s\n", sl.yemekAdi, sl.fiyat, sl.hazirlamaSuresi, sl.Mevcut);
303         } else {
304             bulundu = 1;
305         }
306     }

```

```

Yemek listesi:
NO          YEMEK ADI          FIYAT          HAZIRLANMA SURESI          DURUM
1           Lahmacun           75TL           20dk           mevcut
2           Hamburger          250TL          30dk           mevcutdegil
3           Manti               170TL          45dk           mevcut
4           Tost                60TL           10dk           mevcut
5           TavukDurum          80TL           15dk           mevcut
6           EtDurum             160TL          15dk           mevcut
7           menemen             15TL           60dk           mevcut
8           makarna             130TL          35dk           mevcut
9           pide                100TL          25dk           mevcut
Silinecek yemek adini giriniz: pide
Yemek basariyla silindi.
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |

```

2.2.3 Yeni Sipariş ONAY/RED Fonksiyonu

Müşteriler tarafından verilen yeni siparişler kontrol edilebilecek ve onay/red olarak durumu güncellenebilecektir. Durumu güncellenen yeni siparişler Mutfak uygulamasına gönderilecektir.

```

321 void siparisOnayRed(const char *dosyaYolu) {
322     FILE *dosya = fopen(dosyaYolu, "r");
323     if (dosya == NULL) {
324         perror("Dosya acilamadi!");
325         exit(EXIT_FAILURE);
326     }
327
328     FILE *tempDosya = fopen("temp.txt", "w");
329     if (tempDosya == NULL) {
330         perror("Gecici dosya acilamadi!");
331         fclose(dosya);
332         exit(EXIT_FAILURE);
333     }
334     int onay = 0;
335
336     // Siparisleri oku ve listele
337     while (fscanf(dosya, "%s %s %d %s %s %s", s1.siparisID, s1.yemekAdi, &s1.fiyat, s1.siparisZamaniStr, s1.hazirlanmaZamaniStr, s1.Durum, s1.kullanici) == 7)
338     {
339         if (strcmp(s1.Durum, "beklemede") == 0) {
340             printf("Siparis ID: %s, Yemek Adi: %s, Fiyat: %d, Durum: %s, Kullanici: %s\n", s1.siparisID, s1.yemekAdi, s1.fiyat, s1.Durum, s1.kullanici);
341             printf("Onaylamak icin 1, reddetmek icin 0, islemi bitirmek icin -1 giriniz: ");
342             scanf("%d", &onay);
343
344             if (onay == 1) {
345                 strcpy(s1.Durum, "onaylandi");
346             } else if (onay == 0) {
347                 strcpy(s1.Durum, "reddedildi");
348             } else if (onay == -1) {
349                 break;
350             }
351         }
352     }
353 }

```

```

Siparis ID: SIP250524_001, Yemek Adi: Lahmacun, Fiyat: 75, Durum: beklemede, Kullanici: elif
Onaylamak icin 1, reddetmek icin 0, islemi bitirmek icin -1 giriniz: 1
Siparislerin durumu guncellendi.
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |

```


2.2.4 Rapor Al Fonksiyonu

Her günün sonunda ilgili güne ait (00:00 – 23:59 saatleri arası) sipariş kayıtları “siparişler.txt” dosyasından alınıp yeni bir dosyaya kaydedilmektedir. Yeni kayıt yapılacak dosyanın ismi ilgili tarihle ilişkilendirilmiştir. Günlük rapor görüntülemek isteyen kullanıcı, sadece tarih bilgisi girerek ilgili tarihe ait siparişler bilgilerini ekranda görüntüleyebilecektir.

```
372 void gunsonu(const char *kaynakDosyaYolu) {
373     // Sistem tarihini al
374     time t simdikiZaman = time(NULL);
375     struct tm *zamanBilgisi = localtime(&simdikiZaman);
376     char tarih[11];
377     strftime(tarih, sizeof(tarih), "%Y.%m.%d", zamanBilgisi); // Saat bilgisi olmadan tarih formatı
378
379     // Kaynak dosyayı aç
380     FILE *kaynakDosya = fopen(kaynakDosyaYolu, "r");
381     if (kaynakDosya == NULL) {
382         perror("Kaynak dosya acilamadi!");
383         exit(EXIT_FAILURE);
384     }
385
386     // Hedef dosya adını olustur ve aç
387     char hedefDosyaAdi[40];
388     sprintf(hedefDosyaAdi, "gunsonurapor\\Rapor_%s.txt", tarih);
389     FILE *hedefDosya = fopen(hedefDosyaAdi, "w");
390     if (hedefDosya == NULL) {
391         perror("Hedef dosya acilamadi!");
392         exit(EXIT_FAILURE);
393     }
394 }
```

```
Gunluk rapor 'gunsonurapor\Rapor_2024.05.25.txt' dosyasina basariyla kaydedildi.
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

2.2.5. Rapor Görüntüle Fonksiyonu

```
419 void raporGoruntule() {
420     char tarih[11];
421     printf("Goruntulemek istediginiz raporun tarihini YYYY.MM.DD formatinda giriniz: ");
422     scanf("%10s", tarih);
423
424     char raporDosyaAdi[40];
425     sprintf(raporDosyaAdi, "gunsonurapor\\Rapor_%s.txt", tarih);
426
427     FILE *raporDosyasi = fopen(raporDosyaAdi, "r");
428     if (raporDosyasi == NULL) {
429         printf("Belirtilen tarihe ait rapor bulunamadi veya dosya acilamadi.\n");
430         return;
431     }
432
433     // Rapor dosyasinin iceridini oku ve ekrana yaz
434     char satir[256];
435     while (fgets(satir, sizeof(satir), raporDosyasi) != NULL) {
436         printf("%s", satir);
437     }
438
439     // Dosyayi kapat
440     fclose(raporDosyasi);
441 }
```

```
Goruntulemek istediginiz raporun tarihini YYYY.MM.DD formatinda giriniz: 2024.05.15
SIP150524_001 Manti 170 2024.05.15-00.13.07 2024.05.15-00.58.07 onaylandi furkan
SIP150524_001 TavukDurum 80 2024.05.15-00.13.08 2024.05.15-00.28.08 onaylandi furkan
SIP150524_001 makarna 120 2024.05.15-00.13.11 2024.05.15-00.38.11 onaylandi furkan
SIP150524_002 Lahmacun 75 2024.05.15-00.13.17 2024.05.15-00.33.17 onaylandi cansu
SIP150524_002 Manti 170 2024.05.15-00.13.18 2024.05.15-00.58.18 onaylandi cansu
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

2.2.6 Analizler Fonksiyonu

Bu kisminda kullanicinin isteğine bağı olarak müşterilerin vermiş olduğu siparişlerin analizleri gerçekleştirilmiştir.

```
165 int analiz() {
166     int _secim;
167     printf("1. Gunluk Kazanc\n");
168     printf("2. Aylik Kazanc\n");
169     printf("3. Donem kazanci\n");
170     printf("4. En Cok Tuketim\n");
171     printf("0. Cikis\n");
172     printf("Seciminizi giriniz: "); scanf("%d", &_secim);
173     system("cls");
174     return _secim;
}
```

```
1. Gunluk Kazanc
2. Aylik Kazanc
3. Donem kazanci
4. En Cok Tuketim
0. Cikis
Seciminizi giriniz: |
```

“Günlük Kazanç” seçeneği seçildiğinde kullanıcı gün içindeki toplam kazancı görebilecektir.

```
443 void gunlukKazanc(const char *klasorYolu) {
444     char tarih[11];
445     printf("Gunluk kazanci gormek istediginiz tarihi YYYY.MM.DD formatinda giriniz: ");
446     scanf("%10s", tarih);
447
448     char dosyaAdi[256];
449     sprintf(dosyaAdi, "%s\\Rapor_%s.txt", klasorYolu, tarih);
450
451     FILE *dosya = fopen(dosyaAdi, "r");
452     if (dosya == NULL) {
453         perror("Dosya acilamadi!");
454         return;
455     }
456
457     int toplamKazanc = 0;
458     while (fscanf(dosya, "%s %s %d %s %s %s %s",
459         s1.siparisID, s1.yemekAdi, &s1.fiyat,
460         s1.siparisZamaniStr,
461         s1.hazirlanmaZamaniStr, s1.Durum, s1.kullanici) == 7) {
462         if (strcmp(s1.Durum, "onaylandi")==0) {
463             toplamKazanc += s1.fiyat;
464         }
465     }
466
467     fclose(dosya);
468     printf("%10s tarihine ait Toplam kazanc: %d TL\n", tarih, toplamKazanc);
469 }
```

```
Gunluk kazanci gormek istediginiz tarihi YYYY.MM.DD formatinda giriniz: 2024.05.25
2024.05.25 tarihine ait Toplam kazanc: 75 TL
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

“Aylık Kazanç” seçeneği kullanıcıya girilen yıl ve aya göre o aydaki toplam kazancı gösterecektir.

```
471 void aylıkKazanc(const char *kaynakKlasor) {
472     // Ayın gün sayısını hesapla
473     int gunSayisi;
474     int yil, ay;
475
476     printf("Lutfen hesaplamak istediginiz yili ve ayi giriniz (YYYY.MM): "); scanf("%d.%d", &yil, &ay);
477     if (ay == 2) {
478         if ((yil % 4 == 0 && yil % 100 != 0) || yil % 400 == 0) {
479             gunSayisi = 29; // Artık yıl kontrolü
480         } else {
481             gunSayisi = 28;
482         }
483     } else if (ay == 4 || ay == 6 || ay == 9 || ay == 11) {
484         gunSayisi = 30;
485     } else {
486         gunSayisi = 31;
487     }
488
489     // Aylık kazanç
490     int aylıkkazanc = 0;
491
492     // Dosya işlemleri için gerekli değişkenler
493     char dosyaAdi[100];
494     char raporYolu[100];
495
496     // Dosyaları tarayarak aylık kazancı hesapla
497     for (int gun = 1; gun <= gunSayisi; gun++) {
498         sprintf(dosyaAdi, "Rapor_%04d.%02d.%02d.txt", yil, ay, gun);
499         sprintf(raporYolu, "%s/%s", kaynakKlasor, dosyaAdi);
500
501         // Dosyayı aç
502         FILE *dosya = fopen(raporYolu, "r");
503         if (dosya == NULL) {
504             continue; // Dosya yoksa bir sonraki güne geç
505         }
506     }
```

```
Lutfen hesaplamak istediginiz yili ve ayi giriniz (YYYY.MM): 2024.04
Aylık Kazanc (2024.04): 320 TL
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

“Dönem Kazancı” seçeneği belirtilen iki tarih arasındaki siparişler sonucu toplam kazancı kullanıcıya gösterecektir.

```
525 void donemKazanci(const char *kaynakKlasor) {
526     char baslangicTarihi[11], bitisTarihi[11];
527
528     // Kullanıcıdan baslangic ve bitis tarihlerini al
529     printf("Lutfen donemin baslangic tarihini YYYY.MM.DD formatinda giriniz: ");
530     scanf("%10s", baslangicTarihi);
531     printf("Lutfen donemin bitis tarihini YYYY.MM.DD formatinda giriniz: ");
532     scanf("%10s", bitisTarihi);
533
534     struct tm baslangic = {0}, bitis = {0};
535     sscanf(baslangicTarihi, "%d.%d.%d", &baslangic.tm_year, &baslangic.tm_mon, &baslangic.tm_mday);
536     sscanf(bitisTarihi, "%d.%d.%d", &bitis.tm_year, &bitis.tm_mon, &bitis.tm_mday);
537     baslangic.tm_year -= 1900; // tm_year, 1900'den itibaren yılları sayar
538     baslangic.tm_mon -= 1; // tm_mon, 0'dan itibaren ayları sayar
539     bitis.tm_year -= 1900;
540     bitis.tm_mon -= 1;
541
542     int toplamKazanc = 0;
543     siparis sl;
544     char dosyaAdi[100];
545     char raporYolu[100];
546     time_t baslangicZaman = mktime(&baslangic);
547     time_t bitisZaman = mktime(&bitis);
548
549     // Baslangic ve bitis zamanlari arasindaki tüm günleri dolas
550     for (time_t zaman = baslangicZaman; zaman <= bitisZaman; zaman += 86400) { // Her gün için 86400 saniye eklenir
551         struct tm *tm = localtime(&zaman);
552         strftime(dosyaAdi, sizeof(dosyaAdi), "Rapor_%Y.%m.%d.txt", tm);
553         sprintf(raporYolu, "%s/%s", kaynakKlasor, dosyaAdi);
554
555         FILE *dosya = fopen(raporYolu, "r");
556         if (dosya == NULL) {
557             continue; // Dosya yoksa bir sonraki güne geç
558         }
559     }
```

```
Lutfen donemin baslangic tarihini YYYY.MM.DD formatinda giriniz: 2024.04.01
Lutfen donemin bitis tarihini YYYY.MM.DD formatinda giriniz: 2024.05.25
2024.04.01 - 2024.05.25 tarihleri arasi toplam kazanc: 2340 TL
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

“En Çok Tüketim” seçeneği seçildiğinde en baştan itibaren tüm dosyalar üzerinden yapılacak inceleme sonucunda en çok tüketilen yemek (tüketim miktarı ile birlikte), en kazançlı gün (kazanç miktarı ile birlikte), en çok sipariş veren kullanıcı (sipariş miktarı ile birlikte) ve toplam yemek hazırlama süresi bilgileri kullanıcıya gösterilecektir.

```
576 void enCokTuketim(const char *kaynakKlasor) {
577     DIR *klasor = opendir(kaynakKlasor);
578     struct dirent *dosya;
579     YemekTuketim yemekler[MAX_YEMEK_SAYISI] = {0};
580     int yemekSayisi = 0;
581     GunlukKazanc gunlukKazanclar[365] = {0}; // Yılın her günü için bir kayıt varsayımıyla
582     int gunSayisi = 0;
583     KullaniciSiparis kullaniciSiparisler[MAX_YEMEK_SAYISI] = {0};
584     int kullaniciSayisi = 0;
585     double toplamHazirlamaSuresi = 0.0;
586
587     if (klasor == NULL) {
588         perror("Klasor acilamadi!");
589         return;
590     }
```

```
En Cok Tuketilen Yemek: Manti (8 adet)
En Kazancli Gun: 2024.05.16 (840 TL)
En Cok Siparis Veren Kullanici: furkan (11 siparis)
Toplam Yemek Hazirlama Suresi: 9.2 saat
Restoran Yonetim Sistemi
1. Yemek Ekle-Guncelle-Sil
2. Yeni Siparis ONAY/RED
3. Rapor Al
4. Rapor Goruntule
5. Analizler
6. Mutfak
0. Cikis
Seciminizi giriniz: |
```

2.2.7 Mutfak Fonksiyonu

Müşteri tarafından verilmiş ve yönetim tarafından onaylanmış yemekler için süre üretilacaktır. Restoran uygulaması tarafından belirlenen aşçı sayısına istinaden verilen siparişteki yemeklerin hazırlanma süresini belirlenecektir.

```
710 void mutfak(const char *siparisDosyaAdi, const char *yemekListesiDosyaAdi) {
711     siparis yemekler[MAX_YEMEK_SAYISI] = {0};
712     siparis s;
713     int yemekSayisi = 0;
714
715     yemekleriOku(yemekListesiDosyaAdi, yemekler, &yemekSayisi);
716
717     FILE *siparisDosyasi = fopen(siparisDosyaAdi, "r");
718     if (siparisDosyasi == NULL) {
719         perror("Siparis dosyasi acilamadi");
720         return;
721     }
722
723     FILE *mutfakDosyasi = fopen("mutfak.txt", "w");
724     if (mutfakDosyasi == NULL) {
725         perror("Mutfak dosyasi acilamadi");
726         fclose(siparisDosyasi);
727         return;
728     }
729
730     siparisKuyragu kuyruk = {{0}, 0, 0, 0};
731     time_t asciBitisZamanlari[3] = {0};
732
733     while (fscanf(siparisDosyasi, "%s %s %d %s %s %s", s.siparisID, s.yemekAdi, &s.fiyat,
734         s.siparisZamaniStr, s.hazirlanmaZamaniStr, s.Durum, s.kullanici) == 7) {
735         if (strcmp(s.Durum, "onaylandi") == 0) {
736             s.hazirlamaSuresi = hazirlanmaSuresiBul(yemekler, yemekSayisi, s.yemekAdi);
737             if (s.hazirlamaSuresi != -1) {
738                 struct tm tmSiparis = {0}, tmHazirlanma = {0};
739                 int gunSiparis, aySiparis, yilSiparis, saatSiparis, dakikaSiparis;
740                 int gunHazirlanma, ayHazirlanma, yilHazirlanma, saatHazirlanma, dakikaHazirlanma;
741
742                 if (sscanf(s.siparisZamaniStr, "%d.%d.%d-%d", &yilSiparis, &aySiparis, &gunSiparis, &saatSiparis, &dakikaSiparis) == 5 &&
743                     sscanf(s.hazirlanmaZamaniStr, "%d.%d.%d-%d", &yilHazirlanma, &ayHazirlanma, &gunHazirlanma, &saatHazirlanma, &dakikaHazirlanma) == 5) {
744
745                     tmSiparis.tm_year = yilSiparis - 1900;
746                     tmSiparis.tm_mon = aySiparis - 1;
747                     tmSiparis.tm_mday = gunSiparis;
748                     tmSiparis.tm_hour = saatSiparis;
749                     tmSiparis.tm_min = dakikaSiparis;
750                     tmSiparis.tm_sec = 0;
751                     tmSiparis.tm_isdst = -1;
752
753                     tmHazirlanma.tm_year = yilHazirlanma - 1900;
754                     tmHazirlanma.tm_mon = ayHazirlanma - 1;
755                     tmHazirlanma.tm_mday = gunHazirlanma;
756                     tmHazirlanma.tm_hour = saatHazirlanma;
757                     tmHazirlanma.tm_min = dakikaHazirlanma;
758                     tmHazirlanma.tm_sec = 0;
759                     tmHazirlanma.tm_isdst = -1;
760
761                     time_t siparisZamaniEpoch = mktime(&tmSiparis);
762                     time_t hazirlanmaZamaniEpoch = mktime(&tmHazirlanma);
763
764                     if (difftime(hazirlanmaZamaniEpoch, siparisZamaniEpoch) >= 0) {
765                         if (asciBitisZamanlari[0] == 0) {
766                             for (int i = 0; i < 3; i++) {
767                                 asciBitisZamanlari[i] = siparisZamaniEpoch;
768                             }
769                         }
770
771                         int enErkenBitenAsciIndex = 0;
772                         time_t enErkenBitisZamani = asciBitisZamanlari[0];
773                         for (int i = 1; i < 3; i++) {
774                             if (asciBitisZamanlari[i] < enErkenBitisZamani) {
775                                 enErkenBitenAsciIndex = i;
776                                 enErkenBitisZamani = asciBitisZamanlari[i];
777                             }
778                         }
779                     }
780                 }
781             }
782         }
783     }
```

Aşağıda verilebilecek siparişlerin adını ve bilgisini tutan “yemeklistesi.txt” dosyasının, müşteriler tarafından verilen ve restoran tarafından onaylanan siparişleri tutan “siparisler.txt” dosyasının ve restoran tarafından onaylanan siparişlere atanan aşçıları ve sipariş bilgilerini ayrıntılı gösteren “mutfak.txt” dosyasının birer örneği verilmiştir.

yemeklistesi.txt

Yemek Adı	Fiyat	Hazırlama Süresi	Yemek Durumu
Lahmacun	75	20	mevcut
Hamburger	250	30	mevcutdegil
Manti	170	45	mevcut
Tost	60	10	mevcut
TavukDurum	80	15	mevcut
EtDurum	160	15	mevcut
Menemen	15	60	mevcut
makarna	130	35	mevcut

siparişler.txt

SIP250524_001 Lahmacun 75 2024.05.25-18.05 2024.05.25-18.25 onaylandi furkan

SIP250524_001 Manti 170 2024.05.25-18.05 2024.05.25-18.50 onaylandi furkan

SIP250524_001 TavukDurum 80 2024.05.25-18.05 2024.05.25-18.20 onaylandi furkan

SIP250524_001 EtDurum 160 2024.05.25-18.05 2024.05.25-18.20 onaylandi furkan

SIP250524_002 Lahmacun 75 2024.05.25-18.05 2024.05.25-18.25 onaylandi feo

SIP250524_002 EtDurum 160 2024.05.25-18.05 2024.05.25-18.20 onaylandi feo

mutfak.txt

SIP250524_001 Lahmacun 75 20 2024.05.25-18.05 2024.05.25-18.25 onaylandi furkan A1

SIP250524_001 Manti 170 45 2024.05.25-18.05 2024.05.25-18.50 onaylandi furkan A2

SIP250524_001 TavukDurum 80 15 2024.05.25-18.05 2024.05.25-18.20 onaylandi furkan A3

SIP250524_001 EtDurum 160 15 2024.05.25-18.05 2024.05.25-18.35
onaylandi furkan A3

Ödev Sorumluları:

Elif İclal YİĞİT 414877

Cansu KAYALAR 425493

Furkan BARIŞ 415576