|  |
| --- |
| UML CLASS DIAGRAM  Append here the UML class diagram you have created as an SVG image.  SVG formatında oluşturduğunuz UML diyagramını bu bölüme ekleyiniz. |
| **Şekil 1** Örnek UML diyagramı |

|  |
| --- |
| MANUAL for USER MODE  Append here the screenshots and instructions for the user of the program.  Programı kullanacak kişi için ekran görüntülerini ve talimatları bu bölüme ekleyiniz. |
| Yukarıda Şekil1’de UML diyagramında görüldüğü üzere kullanıcının bir haftada yediği öğünlerin büyüklükleri ile toplam aldığı kalori hesaplanıp, bir haftada yaptığı sporları seçip toplamda kaç kalori harcadığını hesaplayan bir proje oluşturulmuştur. Proje üç header file, üç header fileların cpp dosyaları ve main.cpp dosyası olmak üzere toplamda yedi dosyadan oluşmaktadır.  **1.User Mode**    **Şekil 1.1.** Meals sınıfları **Şekil 1.2.** Sports sınıfları  **Şekil 1.3.** Users sınıfı  Yukarıda Şekil 1.1, 1.2, 1.3’te görüldüğü üzere projede toplam sekiz header file bulunmaktadır. Raporda bu kısımdan itibaren ekran çıktısı ile aşama aşama anlatılacak olup ekran çıktısıyla beraber hangi fonksiyon ile ekran çıktısındaki işlem gerçekleştirildiği yan yana verilecektir.  Program çalıştırıldığında Şekil 2.1’de görüldüğü üzere programı kullanan kişiye menü gözükmektedir ve bu menüde Developer modu ya da User modu seçmesi istenmektedir.    **Şekil 2.1.** Ana menü ekran çıktısı **Şekil 2.2.** Ekran çıktısını sağlayan kod  User mode seçildiğinde composition kavramı ile diğer header file dosyalarında bulunan Breakfast, Lunch, Dinner, Basketball, Football, Tennis ve Swimming sınıfları User sınıfının içinde tanımlanmış olan nesneler ile çağırılıp işlemler yapılmaktadır. User mode seçildiğinde ilk işlem olarak operator>> ile user bilgilerini kullanıcıdan alırken operator<< ile kullanıcı bilgileri ekrana basılmaktadır.    **Şekil 3.1.** user mode ekran çıktısı **3.2.** User operator<< ve operator>> fonksiyonları  Kullanıcıdan User sınıfında bulunan name, surname, id, kilo, yas ve numberofweeks değişkenleri alındıktan ve ekrana bastırıldıktan sonra aşağıda Şekil 4.1-4.2 ve 4.3’te görüldüğü üzere kullanıcıdan kahvaltıda, öğle yemeğinde ve akşam yemeğinde yediği öğünlerin bilgisi seçilmesi istenilmektedir.    **Şekil 4.1.** User classı içindeki breakfast nesnesi için ekran çıktısı    **Şekil 4.2.** User classı içindeki breakfast nesnesinin içine girilen öğün bilgilerini kaydeden kod  Şekil 4.2.’de görüldüğü üzere kullanıcıdan kahvaltıda yaptığı öğünlerin büyüklüğünü ogun\_bilgisi\_alma() fonksiyonu ile gün gün girmesi istenilip breakfast classı içinde tanımlanmış set\_breakfast(string[], int) fonksiyonu ile User classı içinde bulunan breakfast nesnesinin string arrayine kaydedilmektedir.void showB () fonksiyonu ile Basketball classındaki string array yazdırılmaktadır. Haftada yedi gün bulunduğundan dolayı int değeri 7 olarak girilmiştir. 4.1 ekran çıktısında bulunan kahvaltı için yapılan işlem aynı şekilde main.cpp dosyasında Lunch ve Dinner classları içindeki string arrayi için de yapılmıştır.    **Şekil 5.1.** haftada kaç öğün sorgusu ekran çıktısı    **Şekil 5.2.** haftada kaç öğün sorgusu işlemini gerçekleştiren kod  Şekil 5.1.’de görülen ekran çıktısında 4.2’de kahvalti için yapılan işlem öğle yemeği ve akşam yemeği için de yapılmış olup Meals.h header dosyasında bulunan tüm classların string arrayleri için tek tek kaydedilip Şekil 5.2.’de görüldüğü üzere ogun\_adet\_sorgula() fonksiyonu ile her bir class için yapılmış olup main dosyası içinde tanımlanmış olan secilen\_ogun değişkenine eşitlenmiştir. Classlarda tanımlanmış olan örneğin kahvaltı için adet\_sayisi\_hesaplama\_breakfast(string) fonksiyonu ile kullanıcı Small, Medium ya da Large birini seçip fonksiyonun içinde string arrayinde kullanıcının seçtiği öğün boyutundan kaç defa yendiğini hesaplamaktadır. Aynı işlem Lunch ve Dinner classı için de yapılıp ogun\_toplam ile ekrana çıktı alınmaktadır.    **Şekil 6.1.** Toplam karloriyi hesaplayan kod    **Şekil 6.2.** Toplam kalori ekran çıktısı  Şekil 6.2.’de görüldüğü üzere Şekil 6.1.’de bulunan kodlar kahvaltı, öğle yemeği ve akşam yemeği için alınan toplam kaloriyi hesaplamaktadır. Meals.h header dosyasında bulunan her class için ayrı ayrı toplam kalori hesaplayan fonksiyon tanımlanmıştır ve main dosyası içerisinde toplam\_kalori değişkeni tanımlanmıştır. Main dosyası içerisinde tanımlanan bu değişken toplam öğünlerden alınan kaloriyi tutuyor olup cout ile ekran çıktısı alınmaktadır.    **Şekil 7.1.** User mode da haftalık yapılan sporların kullanıcıdan alınması kaç dk yaptığı bilgisi alınması ve ekrana yazdırılması    **Şekil 7.2.** Şekil 7.1.’daki ekran çıktısı için gerekli olan kod  Şekil 7.1.’da görüldüğü üzere kullanıcıdan basketball,football,tennis ve swimming sporlarından birini seçmesi istenilmektedir ve global değişken olarak static string sporlar[7] içine kaydedilmektedir. Main içinde tanımlanmış olan string spor\_bilgisi değişkeni Basketball stringini tutmaktadır ve Basketball classı içinde o hafta kaç defa basketball oynadığı bilgisini tutacak olan b\_adet için set\_b\_adet(int) fonksiyonu ile User classı içinde tanımlanmış olan Basketball b1 nesnesine veri kaydedilmektedir. Haftada\_kac\_gun\_spor(string) fonksiyonu ile set\_b\_adet fonksiyonuna haftada kaç defa basketball oynandıysa return edilmektedir. Şekil 7.1.’de görüldüğü üzere kullanıcı haftada 2 defa basketball oynadığı bilgisini girmiştir ve ekrana 1 haftada 2 defa Basketbol oynadiniz çıktısı alınmıştır. Hangi gün oynadığı ile kaç dk basketbol oynadığı kullanıcıya soruluyor olup basketbol için dakikada 6 kalori yakılmaktadır ve kullanıcının girdiği dk sayısı Basketball classında tanımlanmış set\_basektPtr(int [],int) fonksiyonu ile bilgi tutulmaktadır. Statik olarak tanımlanmış string sporlar[7] değişkeni 7 elemanlı olduğu için basketPtr, futbolPtr, tenisPtr ve yuzmePtr yedi elemanlı olmaktadır. set\_basketPtr() gibi classlarda tanımlanmış int arraylerde eğer set edilen classın adı değilse o bilgi 0 girilmektedir. Örneğin Pazartesi ve Salı günü basketbol oynandıysa Pazartesi ve Salı günü için bilgi alıp diziye eklerken diğer günler için sıfır bilgisi girilmektedir. Basketball classı için yapılan bu işlem Football, Tennis, Swimming classları için de yapılmıştır.    **Şekil 8.1.** Toplam harcanan kaloriyi ekrana basan kod    **Şekil 8.2.** spor ile harcanan toplam kalori ve son çıktı ekranı  Şekil 8.1.’de görüldüğü üzere main içinde tanımlanmış olan spor\_toplam\_kalori değişkeninde basketbol, futbol, tenis ve yüzme için toplam harcanan kalori bilgisi eşitlenmiştir. Spors.h file dosyasında bulunan Class yapılarının herbiri için toplam\_harcanan\_kalori() fonksiyonu yazılmış olup nesnenin içinde bulunan toplam\_harcanan\_kalori özelliğini döndürmektedirler. User mod sonucu olarak main dosyası içinde sonuc değişkeni tanımlanmış olup öğünlerden alınan toplam kalori bilgisini tutan toplam\_kalori değişkeninden, spor yaparak harcanan kalori bilgisi tutan spor\_toplam\_kalori bilgisini çıkartmaktadır ve user mod için son çıktı ekrana basılmaktadır.  **2. Developer Mode**  Developer mode seçildiğinde aşağıda Şekil 9.1’de görüldüğü üzere kullanıcının karşısına menü gelmektedir. Menüde kullanıcı 1. İşlemi seçerse User sınıfına ait işlemleri, 2. İşlemi seçerse Meals.h header file içindeki classlara ait işlemleri, 3. İşlemi seçerse Sports.h header file içindeki classlara ait işlemleri gerçekletirmektedir.    **Şekil 9.1.** Developer Mode seçildiğinde ekran çıktısı    **Şekil 9.2.** Developer mode seçildiğinde çıkan ekran çıktısını sağlayan kod  Kullanıcı User işlemleri seçtiğinde kullanıcıdan User classı içindeki değişkenleri girmesi istenmektedir ve ekran çıktısı vermektedir.    **Şekil 10.1.** Developer Mode ve user işlemleri için seçildiğinde gerekli işlemleri yapan kod    **Şekil 10.2.** User işlemleri için seçildiğinde ekran çıktısı    **Şekil 11.** Developer mode ve öğün işlemleri için seçildiğinde gerekli işlemleri yapan kod  Şekil 11’de görüldüğü üzere kullanıcıdan kahvaltı,öğle yemeği ve akşam yemeği için öğün boyutları istenmektedir ve bu öğün bilgileri ekran çıktısı operator<< ile gerçekleştirilmektedir. Aynı şekilde user mode için yazılan kahvalti\_toplam\_kalori() fonksiyonu ile ve get\_kahvalti\_toplam\_kalori\_tutma() fonksiyonu ile de ekrana yazdırılmaktadır. Aynı işlem öğle yemeği ve akşam yemeği için de yapılmıştır. Ekran çıktıları aşağıdaki gibidir.    **Şekil 12.1.** Developer mode için öğün işlemlerinde **Şekil 12.2.** Developer mode için öğün işlemlerinde öğle  Kahvaltı için ekran çıktısı yemeği için ekran çıktısı    **Şekil 12.3.** Developer mode için öğün işlemlerinde akşam yemeği için ekran çıktısı  Şekil 12.1., 12.2. , 12.3. ‘te görüldüğü üzere User mode için kullanılan fonksiyon ile de get fonksiyonu ile de ekran çıktısı alınmaktadır.  Developer mode ve Spor işlemleri için seçildiğinde dört farklı class yapısı içinden örnek olması amacıyla swimming örnek alınmıştır. Geri kalan Basketball, Football, Tennis classları için Swimming clası için yapılan işlemlerin aynısı gerçekleştirilmiştir.    **Şekil 13.1.** Swimming classı için işlemler gerçekletirildiğinde ekran çıktısı    **Şekil 13.2**. Şekil 13.1’de görülen ekran çıktısı için gerekli olan kod  Şekil 13.2’de görüldüğü üzere kullanıcıdan main içinde tanımlanmış olan int a değişkeni için bilgi alınmaktadır. Alınan int a bilgisi Swimming classı içinde tanımlanmış olanset\_s\_adet(int)’ a parametre olarak yollanıp kullanıcının haftada kaç defa yüzdüğü öğrenilmektedir. get\_s\_adet() fonksiyonu ile s\_adet return edilmektedir ve ekrana bastırılması için çağırılmaktadır. For döngüsü içinde kullanıcının girdiği kaç defa yüzdüğü bilgisi ile kaç dakika yüzdüğü öğrenilmektedir. Kullanıcı 3 kere yüzdüğü bilgisini girdiyse for döngüsü üç kere tekrar etmektedir ve main içinde tanımlanmış olan b değişkeninde tutulmaktadır. Set\_s\_minutes() fonksiyonu ile b değişkeninde tutulan bilgi s\_minutes’e eşitlenmektedir. get\_s\_minutes() fonksiyonu ile ekran çıktısı alınabilmek için return edilmektedir. Yüzerken harcanan toplam kaloriyi bulmak için set\_s\_toplam\_kalori(int) fonksiyonu kullanılmaktadır. Siwmming class nesnesi içinde toplam kaloriyi tutan s\_toplam\_kalori değişkeni vardır ve yüzme için yakılan kalori dakika başı 7 kcal olduğu için 7 ile çarpılarak parametre olarak gönderilip ekrana yazdırılmaktadır. İşlemin en sonunda kullanıcıdan başka bir işlem yapmak isteyip istemediği sorulmaktadır ve kullanıcının girdiği bilgiye göre ana menüye tekrar dönüp ya da programı sonlandırmaktadır. |

|  |
| --- |
| REFERANSLAR  Write the sources you used for this project as separately numbered.  Projeyi yaparken faydalandığınız kaynakları (kitap, web siteleri, videolar vb.) bu bölümde maddeler halinde birinci maddede gösterildiği gibi yazınız. |
| **1.** https://www.mustafayemural.com/uml-egitimi (erişim tarihi: 18.05.2021)  **2.** https://programmingdigest.com/c-composition-with-programming-example/#:~:text=In%20C%2B%2B%20Composition%2C%20an%20object,are%20also%20destroyed%20with%20it.  **3.** <https://www.geeksforgeeks.org/convert-string-char-array-cpp/>  **4.** <https://www.studytonight.com/cpp/function-overloading.php#:~:text=Function%20Overloading%20in%20C%2B%2B,functions%20in%20the%20same%20class>.  **5.** https://www.learncpp.com/cpp-tutorial/composition/  **6.** INF212 laboratuvar deney föyleri |