

**İSTANBUL TEKNİK ÜNİVERSİTESİ  
ELEKTRİK-ELEKTRONİK FAKÜLTESİ**

**HEALTH ASSISTANT**

**Bitirme Ödevi**

**Nevin Özbilgin**

**040010428**

**Balkan Güler**

**040010475**

**Bölüm : Bilgisayar Mühendisliği  
Anabilim Dalı: Bilgisayar Bilimleri**

**Danışman : Yrd.Doç.Dr. Feza Buzluca**

Mayıs 2006

## **Özgünlük Bildirisi**

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntılar, ilgili kaynaklar referans gösterilerek açıkça belirtildiğini,
2. Alıntılar dışındaki bölümlerin, özellikle projenin ana konusunu oluşturan teorik çalışmaların ve yazılım/donanımın benim tarafımdan yapıldığını bildiririm.

İstanbul, 22 Mayıs 2006

Nevin Özbilgin

Balkan Güler

# HEALTH ASSISTANT

## ( ÖZET )

Obesite günümüzün en önemli sağlık sorunlarından biridir. Aşırı kilolu insanların oranı her geçen gün artmaktadır. Bunun yanında obesite ve aşırı kilo, yüksek tansiyon, diyabet, yüksek kolesterol gibi ciddi sağlık sorunlarına da yol açmaktadır. Bu sebeple obesitenin ve aşırı kilodan kaynaklanan bu hastalıkların tedavi giderleri toplam sağlık harcamalarının önemli bir kısmını oluşturmakta bu açıdan obesitenin ekonomik etkisi ciddi olmaktadır. [1]

Bunun yanında obesite tedavileri genellikle başarısızlıkla sonuçlanmaktadır. Bunun en büyük sebebi diyet planlarının kişiye özgü olmaması ve kişilerde bu planı devam ettirmek için yeterli motivasyonun olmamasıdır.

Health Assistant kişinin kendi fiziksel özelliklerine bağlı olarak hazırladığı diyet ve eksersiz planları ile ve kullanıcın ilerlemesini, kilo değişimini takip ederek, kullanıcıda gerekli motivasyonun sürdürülmesini sağlayarak, kişilerin normal kiloya ulaşmalarını hedefleyen bir yazılımdır.

Kullanıcılar sistemi kullanabilmek için üye olmaları gerekmektedir. Üyelikten sonra ilk aşamada kullanıcıların fiziksel özelliklerinin analizi yapılır. Burada kullanıcının obesite seviyesi belirlenir. Kullanıcı durumunu gördükten sonra diyet planı oluşturmak isterse kullanıya seçmesi için bir aktivite listesi çıkarılır. Kullanıcı günlük olarak yapabileceği aktiviteleri seçtikten sonra, hedef kilo ve süreyi belirler. Bu değerlerin tutarlı ve mantıklı olması gerekmektedir. Mümkün olmayan hedef kilo ve süre değerleri sistem tarafından kabul edilmez. Kullanıcının be tercihleri göz önünde bulundurularak veritabanında bulunan yiyecekler arasından kullanıcının kalori ihtiyaçlarına göre bir diyet programı ve yine kullanıcının seçtiği aktiviteler arasından bir aktivite planı oluşturulur.

Kullanıcı plan boyunca kilosundaki değişimleri günlük olarak sisteme girer. Bu değişim ve kullanıcın tükettiği kalori, karbonhidrat, protein gibi bilgilerin sistemin web sayfasından grafiksel olarak gösterilir. Böylece kullanıcı ilerlemesini görerek plana bağlı kalması kolaylaşır.

# HEALTH ASSISTANT

## ( SUMMARY )

Obesity is one of the major health problems of today's people. According to the latest statistics from the third National Health and Nutrition Examination Survey, 97 million Americans are overweight or obese. Overweight also usually causes serious diseases such as high blood pressure, diabetes and heart disease. Overweight, obesity and their associated health problems have a significant economic impact on the world's health care system. Medical costs may involve preventive, diagnostic, and treatment services related to obesity. The cost of treatment of obesity and obesity related diseases is approximately \$100 billion annually in US. [1]

On the other hand, the high percentages of obesity treatments fail. Because diet plans are not patient specific and usually become unsuccessful at weight loss. Also patients have not sufficient motivation for maintain a dietary eating and exercise plans. Conforming to a strict diet plan necessitates high motivation of patient. That can only be achieved by tracking the progress and change in weight and other body measurements.

HealthAssistant is software that aims to generate successful dietary eating and exercise plans by considering the patients both physical and psychological conditions and to make easier for patients to follow the diet plan by assisting and motivating them throughout the plan.

A diet plan consists of two aspects. The first aspect is eating plan. Health Assistant generates an eating plan for patient that includes the daily nutrition consumption by elaborating on each meal in terms of kind, amount and calorie of foods constituting that meal. In second aspect of the plan is to recommending some exercises with indication its duration to the patient. The entire plan is prepared by considering the patient's exercise choice and the daily calorie need of the patient. In addition, to motivate the patient to keep them conforming to the plan, patients progress, weight loss and the calorie consumptions are tracked and displayed in charts.

To use this software, first users should download it, and register to the system by providing personal details, username and password. To open a session, user should log in to the system. After a successful login, user can analyze their current body measurements.

User should provide:

- height,
- weight,
- waist circumference,
- gender.

Software calculates the BMI (Body Mass Index), ideal weight and the obesity level of the user. Body Mass Index (BMI) is a number calculated from a person's weight and height. BMI provides a reliable indicator of body fatness for most people and is used to screen for weight categories that may lead to health problems.

$$\text{BMI} = \frac{\text{weight (kg)}}{\text{height}^2 (\text{m}^2)}$$

While the BMI value between 18.5 and 25 is considered normal, value between 25 and 30 is considered overweight and the bigger values than 30 is considered as an indication of obesity.

Also the waist circumference and the gender determine the disease risk of the patient. For example 88 cm is the upper limit of the normal waist circumference of a female and 102 cm is for a male. As the BMI and the waist circumference increase, the disease risk of the patient increases also. Ideal weight is calculating by using the normal BMI value and the height of the user.

After saving these results in database for later uses and displaying them to the patient, software asks them whether they want to generate a dietary eating and exercise plan. If the patient accepts, a list of activities existing in database displayed to the patient. Patient should choose the exercises that they can do in their daily life conditions. For example, it will not be a reasonable choice, if a patient who does not know swimming, chooses swimming in the list. Every activity has a coefficient that determines the burned calorie when that activity done.

Second prerequisite for the plan generation is the selection of diseases. Software displays a list of disease such as high cholesterol and hyper tension to the patient. Patient should mark the disease that they suffered. Eating plan may differ from a healthy person to a patient with high cholesterol. However, because of the insufficient time, diseases are not considered when generating a diet plan.

Especially, while generating an eating plan, there exist lots of difficulties. These are such as;

- How to prevent the same food selected twice in a meal.
- How to select a food in reasonable amounts. 1 kg honey in a meal is not reasonable.
- How to prevent similar foods selected in a meal.
- How to prevent the same food selected in consecutive days.
- How to establish daily meal plan similar to current meal conventions. For example, today, there are three meals (breakfast, lunch, dinner) and some snacks in a day.
- How to prevent a food selected in an irrelevant meal. It is not reasonable to eat beans in breakfast.

To solve these difficulties, a generic meal model structure was developed. Meal model is a schema that defines

1. The number of a meal and snack in a day.
2. The number of foods in each meal.
3. The food group of the each food in each meal.
4. The maximum and minimum amounts of each food groups.
5. The calorie proportion of each food in a meal.
6. The calorie proportion of each meal in a day.

There is one built-in meal model in the system and persisted as an XML file in database; also it is possible to add new meal models. That makes the system very flexible and extendible in terms of generating useful diet plans. The details of all meal models are displayed to the user and they choose a suitable meal model for their eating preferences.

Finally user should decide the target weight and the duration to access this target weight. Target weight and the duration values should be consistent and reasonable. For instance, for a patient who is 90 kg weight, giving 10 kg in 10 days will not reasonable and possible. Diet experts claim, in a healthy diet plan, a patient can lose 10 percent of their weight in 6 months. Shorter durations are prone to health risks. On the basis of this view, Health Assistant warns patients who don't provide reasonable target weight and duration values.

If all the prerequisites are met, a dietary eating and exercise plan is generated, stored in database and displayed to the user. Also duration of each exercise is calculated in plan. User can select a date and view the plan of that day.

Every day throughout the plan, user should provide their current body measurements. These values are kept in database and used to draw chart when user request. Thus, in the system's web site, user can examine their progress, weight loss, daily calorie, carbohydrate, protein and fat consumption, what should they consume on that day in charts. Awareness of progress will increase the motivation and it will be easier for users to follow the plan.

Health Assistant is designed as an example of service oriented architecture. Server side of the system consists of 3 services. These are security service, analyze service and plan service. Each service has three layers: Web service layer, business logic and the data layer.

System uses Sql Server 2005 enterprise edition database and data layer comprises the data access and entity classes. All the sql scripts are embedded into the stored procedures due to the fact that stored procedures are very secure, more flexible and have high performance.

Patient's body measurements are analyzed by analyze service and the business layer of the analyze service is includes the classes responsible for analyze.

Diet plans are generated by plan service. This service is also responsible for tracking the progress. It is the core service of the system.

Security service is responsible for registering, signing on and authenticating a user. After signing a user on, this service returns a session key to the user. Through this session key, user can consume the other services. All of the transmissions with web service are implemented on SSL to prevent user credentials from captured by unauthorized people. Every request to the plan service and the analyze service are responded after the authenticating the session key by the security service.

In the client side, MVC (Model – View – Controller) design pattern is implemented.

MVC consist of three kinds of objects. The Model is the application object, the View is its screen presentation, and the Controller defines the way the user interface reacts to user input. Before MVC, user interface designs tented to lump these objects together. MVC decouples them to increase flexibility and reuse. [2]

In Health Assistant, there are 3 client applications that each of them targets different platforms. 2 of them, windows application client and the mobile client for PDAs have the same functionalities. Applying MVC is a best way two increase the code reuse under that circumstances. In windows application, all model objects are gathered, and decoupled from the view related codes. Also the controller objects that make the necessary changes and manipulations on the model are separated from view. Thus the same model and controller objects are used in both windows client and mobile client applications.

## İÇİNDEKİLER

1 GİRİŞ.....	9
2 PROJENİN TANIMI VE PLANI.....	9
<b>2.1. Proje Tanımı.....</b>	<b>9</b>
<b>2.2. Proje Planı .....</b>	<b>10</b>
3 Kuramsal Bilgiler.....	11
<b>3.1 Visual Studio Team System.....</b>	<b>11</b>
3.1.1 Team Foundation Server.....	13
3.1.2 İstemciler.....	13
<b>3.2 MVC – Model Görünüm Denetçi (Model View Controller) Ve Gözlemci (Observer) Tasarım Kalıpları.....</b>	<b>16</b>
4 ANALİZ VE MODELLEME .....	21
<b>4.1 Ürün Perspektifi.....</b>	<b>21</b>
<b>4.2 Ürün Nitelikleri.....</b>	<b>21</b>
<b>4.3 Sistem Özellikleri .....</b>	<b>22</b>
<b>4.4 Çalışma Ortamı.....</b>	<b>23</b>
4.4.1 Performans Gereksinimleri .....	23
4.4.2 Güvenlik Gereksinimleri.....	23
<b>4.5 Kullanılan Formüller ve Kurallar .....</b>	<b>24</b>
5 TASARIM, GERÇEKLEME VE TEST .....	26
<b>5.1. Genel Mimari Yapı .....</b>	<b>26</b>
<b>5.2 Sunucu Modülü .....</b>	<b>28</b>
5.2.1 Web Servisleri Katmanı .....	29
5.2.2 İş Mantığı Katmanı .....	33
5.2.3 Veri Erişim Katmanı .....	37
5.2.4 Veri Katmanı.....	37
Tablolar .....	38
Saklı Yordamlar .....	38
<b>5.3 İstemci Uygulamalar.....</b>	<b>38</b>
5.3.1 Model Modülü .....	39
5.3.2 Önyüz Modülü .....	40
5.3.3 Güvenlik (Security) Modülü.....	40
5.3.4 Analiz (Analyze) Modülü .....	42
5.3.5 Plan Modülü.....	47
5.3.6 Mobil İstemci Uygulaması.....	50
5.3.7 Web İstemci Uygulaması.....	50
6 Sonuç ve Öneriler .....	52
7 Kaynaklar.....	53



# 1 GİRİŞ

Obesite, günümüzün en önemli sağlık problemlerinden biridir. Diğer taraftan, başarısız diyet planları ve hastanın motivasyon eksikleri nedenleriyle obesite tedavileri büyük oranda başarısızlığa uğramaktadır.

Health Assistant, kullanıcının fiziksel durumunu göz önünde bulundurarak diyet planları yaratır ve planın uygulama süresince kullanıcının motivasyonuna destek vererek kullanıcının süreci takip etmesini kolaylaştırır.

## 2 PROJENİN TANIMI VE PLANI

### 2.1. Proje Tanımı

Projedeki amaç, her çeşit kullanıcı profiline uygun olarak kişilere sağlıklı yaşama alışkanlıkları kazandıracak ve kullanıcının günlük aktivitelerini düzenleyecek diyet – egzersiz planları önerilmesi ve takibidir.

Diyet planı, kullanıcının günlük alması gereken gıdalardan ve yapılması önerilen egzersizlerden oluşur. Kullanıcının yemek veya egzersiz zamanlarında uyarılması, harcanan kalori miktarının gösterilmesi ve oluşan fiziksel değişimin izlenmesi Sağlık Asistanı yazılımının sağladığı özelliklerdendir. Ayrıca tüm tedavi süreci bir diyetisyen tarafından takip edilir.

Bu sistem, daha önce benzer amaçlara hizmet veren sistemlerden farklı olarak, kullanıcının yalnızca diyet planını takip etmesi hedeflenmez, akıllı diyet planlayıcısı sağlar ve profesyonel destek gözlemlemesine olanak verir.

Sistemde iki çeşit kullanıcı vardır. Hasta sistemin esas kullanıcısıdır. Diyet programı hastanın fiziksel bilgileri göz önünde bulundurularak hazırlanır. Diyetisyen ise, sistemin ürettiği diyet programını denetlemek ve gerektiğinde müdahale etmekle görevlidir.

Sistem gereksinimlerinden sunucu uygulaması için gerekli sunucu konfigürasyonu şöyle olmalıdır: Windows Server 2003, .Net Framework 2.0 ve IIS 6.0, Sql Server 2005. İstemci uygulaması .Net Framework 2.0 yüklü Windows 2000, XP işletim sistemlerinde ve .Net Compact Framework 2.0 yüklü Windows CE işletim sistemlerinde çalışmalıdır. Sistemin web uygulaması IE 5.0 ve üst versiyonlarında ve Netscape 6.0 ve üst versiyonlarında çalışmalıdır.

## 2.2. Proje Planı

Proje adımları aşağıdaki gibidir:

- Veritabanı Tasarımı - Normalizasyonu
- Kullanıcı İstemci tarafı kayıt arayüzü geliştirilmesi
- Haberleşme arayüzü tasarımı
- Sunucu uygulaması olarak diyet planının oluşturulması ve diyetisyene onayı
- Kullanıcı İstemci tarafı takip formu arayüzü geliştirilmesi
- Sunucu tarafı raporlama modülü geliştirilmesi
- Sonuçların istatistiksel analizinin yapılması ve kullanıcı istemci web arayüzü oluşturulması
- Modullerin Entegrasyonu
- Test

Proje zamanlaması aşağıdaki gibidir.

	i	Task Name	Duration	Start	Finish
1		Saklı Yordamlar – Web Servis eğitimi	5 days	Mon 06.02.06	Fri 10.02.06
2		Veritabanı Tasarımı - Normalizasyonu	5 days?	Mon 13.02.06	Fri 17.02.06
3		Kullanıcı İstemci tarafı kayıt arayüzü geliştirilmesi	5 days?	Mon 20.02.06	Fri 24.02.06
4		Haberleşme arayüzü tasarımı	30 days?	Mon 27.02.06	Fri 07.04.06
5		Sunucu uygulaması olarak diyet planının oluşturulması ve diyetisyene onayı	15 days?	Mon 27.02.06	Fri 17.03.06
6		Kullanıcı İstemci tarafı takip formu arayüzü geliştirilmesi	10 days?	Mon 13.03.06	Fri 24.03.06
7		Sunucu tarafı raporlama modülü geliştirilmesi	10 days?	Mon 27.03.06	Fri 07.04.06
8		Sonuçların istatistiksel analizinin yapılması ve kullanıcı istemci web arayüzü oluşturulması	10 days?	Mon 03.04.06	Fri 14.04.06
9		Modullerin Entegrasyonu	5 days?	Mon 17.04.06	Fri 21.04.06
10		Test	5 days?	Mon 24.04.06	Fri 28.04.06

Şekil 2.1 Proje zamanlaması