

## **Özgünlük Bildirisi**

1. Bu çalışmada, başka kaynaklardan yapılan tüm alıntıların, 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

Mehmet Cambaz

# HATA TAKİP YAZILIMI

## ( ÖZET )

Bu proje, hata takibi sorununda çoğu durumda taraflar arasında iletişim problemi olması nedeniyle, taraflar arasında anlaşmazlık doğmasını önlemek; her yerden hata iletimi ve takibi yapılabilmesi amacıyla gerçekleştirilmiştir.

İleticinin anlatmak istediği ile sorunu çözmek ile yükümlü kişinin anladığı çoğunlukla aynı olmadığından, sorunun anlaşılması çok büyük bir problem oluşturmaktadır. Bundan dolayı sorumlu kişinin daha önceden kendi belirlemiş olduğu soruların cevaplanma sürecinden sonra hata hakkında ayrıntılı bilgi gönderilmesinin sağlanması; sorumlu kişinin hatanın nerede, hangi modülde, nelerle ilgili olabileceği hakkında maksimum bilgiyi edinmesi açısından çok yardımcı olacaktır.

Projenin bir başka amacı ise hayatın her anında ve her yerde kişinin yanında bulundurduğu akıllı cihazlar ile işlerini takip etmesi ve iletişim kurabilmesidir. Herhangi bir proje ile ilgili hata veya eksiklik gönderebilmesi için hem cep telefonları hem de cep bilgisayarları için ek uygulamalar geliştirilmiştir.

Projenin ölçeklenebilir olması için üç katmanlı mimari yapısı kullanılmıştır. Ayrıca bu mimari ile yazılımın parçalarının değiştirilmesi, güncellenmesi kolaylaşmış ve başka bir projede yeniden kullanılabilir parçalar elde edilmiştir.

Projenin veri katmanında SQL Server 2005 Express Edition, saklı yordamlar ve SQL Server'a özel bir tercüman nesnesi bulunmaktadır. İş katmanında ise ilgili varlıklarla alakalı isteklere karşılık veren nesneler ve akıllı cihazların işlem yapabilmesi için gerçekleştirilmiş olan web servisleri vardır. En son olarak sunum katmanında ise ana uygulama olan ASP.NET sayfaları, cep bilgisayarlarından sisteme erişim için gerçekleştirilmiş PDA Yazılımı ve aynı şekilde cep telefonlarından sisteme erişim için gerçekleştirilmiş bir J2ME Yazılımı mevcuttur. Ayrıca projenin sunum katmanındaki bu üç ayrı uygulama, farklı sistemlerin birlikte çalışabildiğini göstermiştir.

# **ERROR & BUG TRACKING SOFTWARE**

## **( SUMMARY )**

At present, errors, bugs and similar defects for projects are hard to track and fix. Usually, someone who wants to report a bug does not give enough information about the problem or can not describe the situation properly. Because of this problem, the responsible people who are supposed to take care of defects are unable to accomplish their missions.

Sometimes senders give the information properly but the explanation is not enough, so that responsible can misunderstand the problem. In general there are always communication problems between senders and responsible.

This project's goal is to minimize the risks of misunderstanding and to establish a good communication protocol for sending defects that responsible can really understand properly what is wrong about the project, where is the trouble, which module or part is dysfunctional.

A good communication protocol will help the responsible fully comprehend the situation. It can be done by asking questions to the sender (before getting the error explanation) that has been prepared exactly for the project that helps to get the critical information about the project. The questions can be answered as yes or no so that responsible can estimate where is the actual problem.

Due to developing technology and cheapening internet costs, most of the pocket pcs and cellular phones are able to reach world wide web and this is not a luxury for today, it is necessary for people who travel.

Another aims of the project are enabling senders to send defects, bugs or errors from their smart devices such as pocket pcs or cellular phones and enabling responsible to control whether there are unsolved errors at critical projects.

The project is implemented with three tier application model, sometimes known as n tier application architecture. This software architecture enables the project to be easy-to-change and reusable for future projects. Because of all parts are implemented as separate as possible; while changing or removing a part it is simpler to maintain the order for the project again.

There are three main layers at three tier application model: Data Tier, Business Tier and Presentation Tier. The explanation for each tier is given below.

**Data Tier:** This is the layer in which the data is kept and can be reached from. It can be a database, file etc. Built-in compiled queries such as Stored Procedures are at this tier. Also there is a Data Access Tier (which can be included as a part of Data Tier) between Business Tier and Data Tier which is responsible for data transfer between two tiers.

**Business Tier:** This is the layer in which the 'business' logic is implemented. It is responsible for taking data from Data Tier, making necessary calculations and giving the meaningful data to the Presentation Tier applications. Also it is responsible for taking data from Presentation Tier and giving the data meaningful for Data Tier. Presentation Tier requests are answered at this tier.

**Presentation Tier:** This is the outer layer which the user sees and makes operations. It is responsible for user input/output. It transmits the requests and data to Business Tier. Smart devices, web pages, windows applications etc. can be at this layer.

The project is implemented with this kind of application architecture with .NET 2.0. Although the base is implemented with .NET, there is a Java 2 Micro Edition application at Presentation Tier for cellular phones. This shows that interoperability with different systems is possible.

For the implemented project the tiers and their contents are listed.

**Data Tier:** SQL Server 2005 Express Edition is the main part of the Data Tier, where all data are kept. There are also Stored Procedures which return identity value of last inserted row, they are kept compiled at database. There is an adapter class which is used as Data Access Tier specially written by myself using SQL Server connection and query classes.

**Business Tier:** There are classes at this layer who are responsible for requests of their related entities. They are at App\_Code folder of ASP.NET web site project. Also there are three web services which take care of smart device related operations.

The web services are:

**kullaniciWS:** The web service responsible for user operations on smart devices, has web methods for logging in.

**projeWS:** The web service responsible for project operations on smart devices, has web methods for listing all available projects and getting the number of unsolved bugs of a project.

**hataWS:** The web service responsible for bug operations on smart devices, has web methods for sending bugs and getting the questions of a project before sending bugs.

**Presentation Tier:** There are four different applications at this tier which are bug fixer's web-based application, bug sender's web-based application, pocket pc application and cellular phone application. Web-based applications contain ASP.NET web pages specially coded for bug fixers or bug senders. All applications use the classes at Business Tier for their requests.

Bug fixer's web-based application consists pages such as: project adding/deleting, question(s) adding/editing/removing, bug summary analyzing, bug detail examining, express and share opinions, talk about the bug, adding a solution to bug, determining a responsible from a project or a bug, list all responsables from projects and bugs in detail, adding an user whose user type responsible. Also fixers can be able to do all tasks senders can.

Bug sender's web-based application consists pages such as: sending a bug answering questions for the project one by one until no question is left or sending bugs starting to answer from his/her wanted question, looking at both summary and detailed information of his/her sent bugs, talking and sharing opinions about sent bug, approving a solution of his/her sent bug, examining; talking about bugs that are fixed or solved, editing user preferences such as password, e-mail address or computer properties.

All classes and ASP.NET web pages' behind codes are written by C#, all pages except logging out page has a master page named `MasterPage.master`. This provides a consistent look for all of the web site.

Pocket PC application is implemented for Windows Mobile 5.0 with C# at Visual Studio 2005 and controlled with Windows Mobile 5.0 Pocket PC emulator. This application can be named as PDA Software of the project. This application is connected to web services with 'Adding a Web Reference' operation at .NET. With a internet-connected Pocket PC which uses Windows Mobile 5.0 operating system, users can be able to send bugs from this smart device. If the user logs in as a responsible, he/she can get the number of unsolved bugs from a project that is critical.

Cellular phone application is implemented with Java 2 Micro Edition known as J2ME. It is developed under 'netBeans 5.0' with the help of 'netBeans Mobility Pack' which is an add-in for netBeans to develop J2ME MIDlets easily. This application's web service connection classes are generated with 'J2ME Web Service Client' class stub which can be added to project at netBeans. The forms, objects and work algorithm is implemented in one MIDlet named 'HataTakip'. However, there are four packages at this application: three packages for web service client classes and a main package for MIDlet that makes all operations. The same operations can be done with this application as Pocket PC application offers.

# İÇİNDEKİLER

ÖZGÜNLÜK BİLDİRİSİ.....	I
ÖZET.....	II
SUMMARY.....	III
İÇİNDEKİLER.....	VI
ŞEKİL LİSTESİ.....	VII
TABLO LİSTESİ.....	VIII
1 GİRİŞ.....	1
2 PROJE TANIMI VE PLANI.....	3
3 KURAMSAL BİLGİLER.....	5
3.1 SQL Server 2005 (Express Edition).....	5
3.2 .NET.....	5
3.2.1 Katmanlı Mimari.....	5
3.2.2 ASP.NET 2.0.....	7
3.2.3 Windows Mobile 5.0.....	7
3.2.4 Web Servisleri.....	8
3.3 J2ME.....	8
3.3.1 netBeans 5.0.....	8
3.3.2 netBeans Mobility Pack.....	8
4 ANALİZ VE MODELLEME.....	9
4.1 Analiz.....	9
4.2 Modelleme.....	9
5 TASARIM, GERÇEKLEME VE TEST.....	10
5.1 Veri Katmanı.....	10
5.1.1 Veritabanı.....	10
5.1.2 Saklı Yordamlar.....	12
5.1.3 Tercüman (Adapter) Sınıfı.....	12
5.2 İş Katmanı.....	13
5.2.1 proje.cs.....	13
5.2.2 soru.cs.....	13
5.2.3 hata.cs.....	14
5.2.4 kullanıcı.cs.....	15
5.2.5 sorumlu.cs.....	16
5.2.6 soruBilgi.cs.....	17
5.2.7 hataBilgi.cs.....	18
5.2.8 Web servisleri sınıfları.....	19
5.3 Sunum Katmanı.....	21
5.3.1 Hata giderici web uygulaması.....	21
5.3.2 Hata iletici web uygulaması.....	23
5.3.3 PDA uygulaması.....	24
5.3.4 Cep telefonu uygulaması.....	26
5.4 Test.....	29
6 DENEYSEL SONUÇLAR.....	30
7 SONUÇ ve ÖNERİLER.....	31
8 KAYNAKLAR.....	33

## ŞEKİL LİSTESİ

Şekil 2.1: Zaman çizelgesi (Gantt Chart).....	4
Şekil 3.1: Üç katmanlı mimariye genel bir bakış. [6]'dan alınmıştır.....	6
Şekil 3.2: ASP.NET'te üç katmanlı mimari ([7]'den esinlenilmiştir).....	7
Şekil 5.1: Veritabanı tercüman sınıfı, sınıf diyagramı).....	12
Şekil 5.2: Veritabanı tercüman sınıfı, UML diyagramı).....	12
Şekil 5.3: Proje işlemlerinden sorumlu sınıf, sınıf diyagramı).....	13
Şekil 5.4: Proje işlemlerinden sorumlu sınıf, UML diyagramı.....	13
Şekil 5.5: Soru işlemlerinden sorumlu sınıf, sınıf diyagramı).....	14
Şekil 5.6: Soru işlemlerinden sorumlu sınıf, UML diyagramı).....	14
Şekil 5.7: Hata işlemlerinden sorumlu sınıf, sınıf diyagramı).....	15
Şekil 5.8: Hata işlemlerinden sorumlu sınıf, UML diyagramı.....	15
Şekil 5.9: Kullanıcı işlemlerinden sorumlu sınıf, sınıf diyagramı.....	16
Şekil 5.10: Kullanıcı işlemlerinden sorumlu sınıf, UML diyagramı.....	16
Şekil 5.11: Sorumlu kişi işlemlerinden sorumlu sınıf, sınıf diyagramı.....	17
Şekil 5.12: Sorumlu kişi işlemlerinden sorumlu sınıf, UML diyagramı.....	17
Şekil 5.13: Soruları temsil eden sınıf, sınıf diyagramı.....	18
Şekil 5.14: Soruları temsil eden sınıf, UML diyagramı.....	18
Şekil 5.15: Hataları temsil eden sınıf, sınıf diyagramı.....	19
Şekil 5.16: Hataları temsil eden sınıf, UML diyagramı.....	19
Şekil 5.17: Web servisleri sınıfları, UML diyagramı.....	19
Şekil 5.18:App_Code tüm sınıflar özet, UML diyagramı.....	20
Şekil 5.19: PDA Yazılımı işlem formu.....	24
Şekil 5.20: PDA Yazılımı UML sınıf diyagramı.....	25
Şekil 5.21: Cep telefonu uygulaması soru sorma ekranı örnek çıktı.....	27
Şekil 5.22.a: main paketi özet UML sınıf diyagramı.....	27
Şekil 5.22.b: kullanıcıWS paketi özet UML sınıf diyagramı.....	28
Şekil 5.22.c: projeWS paketi özet UML sınıf diyagramı.....	28
Şekil 5.22.d: hataWS paketi özet UML sınıf diyagramı.....	28

# TABLO LİSTESİ

Tablo 2.1: Zaman Çizelgesi iş tablosu.....	4
--	---