Debreceni Egyetem
Informatikai Kar

KÖZBESZERZÉSI FOLYAMATOK INFORMATIKAI LEKÉPEZÉSE .NET KÖRNYEZETBEN

Témavezető:
Dr. Rutkovszky Edéné
Ügyvivő szakértő

Készítette:
Madarász István
Programtervező informatikus (B.Sc.)

Külső konzulens:
Dr. Pázmányi Sándor
Vezérigazgató, eKÖZIG Zrt.

Debrecen
2010

TARTALOMJEGYZÉK

1. BEVEZETÉS 4

2. CÉLOK 7

3. ANYAG ÉS MÓDSZER 9

3.1. A .NET KERETRENDSZER 9

3.2. AZ ASP.NET 12

3.3. AZ ASP.NET TÖRTÉNETE 12

3.4. AZ ASP.NET LAPJAI 12

3.4.1. INLINE MODELL 13

3.4.2. CODE-BEHIND MODELL 13

3.5. MICROSOFT VISUAL STUDIO 2008 15

3.6. MICROSOFT SQL SERVER 2008 EXPRESS 15

3.6.1. SQL SERVER MANAGEMENT STUDIO EXPRESS 16

3.7. AZ ADO.NET 16

3.8. AZ INTERNET INFORMATION SERVICES (IIS) 17

4. EREDMÉNYEK 18

4.1. A FEJLESZTÉSI FOLYAMAT ÉS AZ ELKÉSZÜLT ALKALMAZÁS ÁTTEKINTÉSE 18

4.2. SPECIFIKÁLÁS 19

4.3. TERVEZÉS 20

4.3.1. USE CASE ÉS ACTIVITY DIAGRAM 20

4.3.2. AZ ADATBÁZIS FELÉPÍTÉSE 24

4.3.3. AZ ADATBÁZIS TÁBLÁK FELÉPÍTÉSE 25

4.4. IMPLEMENTÁLÁS 28

4.5. VALIDÁLÁS ÉS VERIFIKÁLÁS 29

4.6. AZ ALKALMAZÁS FUTTATÁSÁHOZ SZÜKSÉGES KÖRNYEZET 30

4.6.1. INTERNETES HASZNÁLAT 30

4.6.2.INTRANETES HASZNÁLAT 31

4.7. AZ ELKÉSZÜLT ALKALMAZÁS BEMUTATÁSA 31

4.7.1. A BEJELENTKEZŐ KÉPERNYŐ 31

4.7.2. AZ ADMINISZTRÁTORI FELÜLET 32

4.7.2.1. A KÖZBESZERZÉSI TERV KARBANTARTÁSA 32

4.7.2.2. KÖZBESZERZÉSI ELJÁRÁSOK INDÍTÁSA 35
4.7.2.3. FELHASZNÁLÓK KEZELÉSE ÉS JOGOSULTSÁGUK MEGADÁSA..... 35
4.7.2.4. AZ AJÁNLATKÉRŐ ADATAINAK KARBANTARTÁSA 37
4.7.2.5. IRATMINTÁK, SAJÁT DOKUMENTUM SABLONOK KARBANTARTÁSA 38
5. ÖSSZEFOGLALÁS 39
6. KÖSZÖNETNYILVÁNÍTÁS 40
7. IRODALOMJEGYZÉK 41

1. BEVEZETÉS

Napjainkban elengedhetetlenné vált, hogy az üzleti, közigazgatási és magánszférában egyaránt a kapott információkat költséghatékonyan és minél előbb feldolgozzuk, értelmezzük, azokból további információkat állítsunk elő, avagy továbbítsuk, válaszokat készítsünk belőlük. Az informatika és ez által az információs társadalom oly nagy léptékű fejlődésen megy keresztül, amely rohamosan csökkenti az egyes információk megszerzéséhez, előállításához, valamint a folyamatok elvégzéséhez szükséges időt. Mindezeket figyelembe véve egyre inkább előtérbe kerülnek az on-line, illetve webes szolgáltatások. Ezek lényegesen lerövidítenek bizonyos ügyintézési folyamatokat és nem utolsó sorban sokkal kényelmesebb lehet a használatuk. Gondoljunk akár csak egy termék interneten történő megvásárlására, élelmiszerrendelésre, banki folyamatok otthonról történő intézésre vagy az e-ügyintézésre. Ezek mind időt és ez által pénzt spórolnak meg és nem utolsó sorban kevesebb papírmunkával járnak. Ezen folyamatok ily drasztikus fejlődése lehetőséget, illetve igényt támaszt a programozók, programfejlesztők számára, hogy új technikákat, új módszereket alkalmazva olyan webes vagy akár asztali alkalmazásokat fejlesszenek, melyek kellően összetettek ahhoz, hogy bármilyen folyamatot le tudjanak modellezni, meg tudjanak valósítani. Ugyanakkor elvárás, hogy kellően felhasználóbarátok legyenek és jelentős mértékben segítsék, gyorsítsák a mindennapjaink ügyeit, folyamatait.

A manapság oly jellemző web alkalmazások használatához elegendő egy web-böngésző és egy internetkapcsolat vagy egy helyi hálózat (intranet) megléte. Népszerűségük oka, hogy gyorsak, könnyen kezelhetőek és nem igényelnek semmilyen szakértői háttértudást (például nem kell telepíteni azokat a saját gépünkre) valamint nincs nagy hardver követelményük sem és nem utolsó sorban a karbantartásuk központilag történik, melyből a felhasználó szinte semmit nem lát.

Fontos szempont, hogy illetéktelen személyek más adatait ne lássák, ne tudjanak hozzáférni, így ne tudják azokat módosítani, törölni vagy újat hozzáadni, esetleg illetéktelenül felhasználni azokat. Különösen igaz ez a banki folyamatokra, valamint a közigazgatásra ahol kritikus folyamatok irányítását, lebonyolítását kell megoldani.

Az ilyen kritikus folyamatok megvalósítását a magánszférában egyre jobban igénylik, míg a hivatali és üzleti szférában egyre inkább kötelezővé teszik.

Valós igényeken alapszik a közigazgatás elektronizálása azaz, hogy minél több konkrét eljárási cselekmény, közigazgatási ügy és szolgáltatás elérhető legyen elektronikus formában. Ez a gyorsabb és pontosabb ügyintézést, az esetleges ügyintézési hibák kiküszöbölését, valamint a folyamatok támogatását és kevesebb papírmunkával járó intézését teszi lehetővé. Az elektronikus közigazgatásnak nincs egységes definíciója, de fő jellemzői, hogy elektronikus információkkal foglalkozik, az állam működésével kapcsolatos, jogintézményi sajátosságai vannak, speciális szabályozás jellemezheti és célja az elektronikus ügyintézés megvalósítása.

Jogszabályi megfogalmazása: *1126/2003. (XII. 12.) Korm. határozat a Magyar Információs Társadalom Stratégiáról és annak végrehajtásáról.*

Szakdolgozatomban kiemelten foglalkozok a közigazgatásban értelmezett folyamatok közül a közbeszerzéssel. A közbeszerzésről általánosságban:

„A közbeszerzés – többek között - az államigazgatási és egyéb költségvetési szervek közszolgáltató tevékenységükkel közvetlenül összefüggő árubeszerzéseinek, építési beruházásainak és szolgáltatási megrendeléseinek külön törvényben meghatározott köre, amelynek értékhatárát Magyarországon az éves költségvetési törvény állapítja meg.

A közbeszerzés célja – egy lehetséges megfogalmazásban - az államháztartás kiadásainak ésszerűsítése, a közpénzek felhasználásának átláthatóvá tétele, széles körű nyilvános ellenőrizhetősége, továbbá a közbeszerzések során a verseny tisztaságának biztosítása.

A központosított közbeszerzés célja, hogy az állami igényeket nagy tételben jelentős árengedményekkel tudják beszerezni, azonban a központosított kiírás esetleg lehetőséget ad a "testre szabott feladatkiírásra", illetve egyéb visszaélésekre.”(wikipédia, Közbeszerzés)

Jogszabályi megfogalmazása, maga a Közbeszerzési törvény (továbbiakban Kbt.): *2003. évi CXXIX. törvény A közbeszerzésekről.*

A közbeszerzés széles körben érint minden, a közbeszerzés hatálya alá tartozó szervezetet többek között iskolákat, óvodákat, önkormányzatokat, valamint helyi önkormányzati költségvetési szerveket. Az elektronikus közbeszerzés elterjedésével jelentősen csökkenthető vagy akár papírmentessé tehető a közbeszerzési folyamat, automatizálni és hatékonyabbá

lehet tenni az adott esetben igencsak bonyolult folyamatokat, így jelentős megtakarítások érhetők el és kontrolálhatóbbak lesznek a folyamatok.

Jogszabályi megfogalmazása: *257/2007. (X. 4.) Korm. rendelet a közbeszerzési eljárásokban elektronikusan gyakorolható eljárási cselekmények szabályairól, valamint az elektronikus árlejtés alkalmazásáról.*

2. CÉLOK

Átgondolva, hogy mennyi lehetőség és kihívás rejlik az elektronikus közigazgatásban és így a közbeszerzés még nem elektronizált folyamataiban a dolgozatom témájának a közbeszerzési folyamatok elektronikus implementációját választottam. További szempont volt, hogy aktuális, valós alapokon nyugvó problémára keressek megoldást.

A papíron folytatott közbeszerzési eljárások, ha kis mennyiségben még nem is, de nagyobb számú eljárás lefolytatása esetén igencsak bonyolulttá válhatnak mind tárolás, mind katalogizálás és szervezés terén, ezért olyan alkalmazás kifejlesztését tűztem ki célul, mely:

- átláthatóbbá teszi a közbeszerzési folyamatokat,
- segíti a rendszerezést és archiválást,
- megfelel a mindenkori törvényi előírásoknak,
- automatizmusokkal segíti a folyamat végigvitelét,
- korszerű és szabványos technikákat alkalmaz,
- webes alkalmazásként elérhető,
- intézményi mérettől függetlenül érdemben segít minden a közbeszerzési eljárás lefolytatásában érintett személyt és nem utolsó sorban
- felhasználóbarát.

Az alábbi táblázatban ismertetem a fő problémákat, hogy a közbeszerzés jelenlegi formájában miért is okoz nehézségeket összevetve az új elektronizált elgondolásokkal (**1. táblázat**):

1. táblázat

Jelenleg	Új elgondolás
• A dokumentumok, melyek egy eljárás lefolytatása során keletkeznek papír alapon tárolódnak.	• A dokumentumok elektronikus formában keletkeznek és tárolódnak az alkalmazásban.
• A határidők figyelése, melyek betartása elengedhetetlen, gyakran átláthatatlanná válnak főleg az egy évnél tovább elhúzódó eljárásoknál.	• A határidőket az alkalmazás automatikusan figyeli és előre értesít azok közeledtéről.
• Az eljárásokat az indításukkor aktuális Kbt. alapján kell lefolytatni, így mindig tudni kell az akkori jogszabályokat.	• Az alkalmazás mindig az eljárás indításához tartozó jogszabályi környezetet alkalmazza.
• Az ajánlattevőktől beérkező és az azoknak küldött iratokat külön iktatni kell.	• Az iratokat az alkalmazás iktatja, lehetőséget biztosít külön erre a célra készült iktató programmal való együttműködésre.
• Az eljárás szakaszaiban létrejövő tájékoztatókat, eredményeket, hiánypótlások kérését ki kell küldeni minden az eljárásban résztvevő ajánlattevőnek.	• Az eljárás szakaszaiban létrejövő tájékoztatók, eredmények, hiánypótlások kiküldésére az alkalmazás lehetőséget biztosít elektronikusan és faxon keresztül.
• A közbeszerzési dokumentumokat 5 évig kell megőrizni, mely rengeteg helyet foglal.	• Az alkalmazásban az iratok és dokumentációk rendszerezve, több évre visszamenőleg tárolódnak.

3. ANYAG ÉS MÓDSZER

Ez a fejezet áttekintést nyújt a használt eszközök, a futtató- és fejlesztő környezetek funkcionalitásáról, tulajdonságairól a dolgozat terjedelmi korlátait figyelembe véve.

Az alkalmazás fejlesztése során, teljes mértékben Microsoft technikákra és fejlesztői eszközökre támaszkodtam. Azért a Microsoft technológiáját választottam, mert így alkalmazni tudtam a legújabb programozási technológiákat, valamint kiemelhető, hogy a Microsoft által fejlesztett professzionális eszközök mögött rendkívül jó támogatás áll.

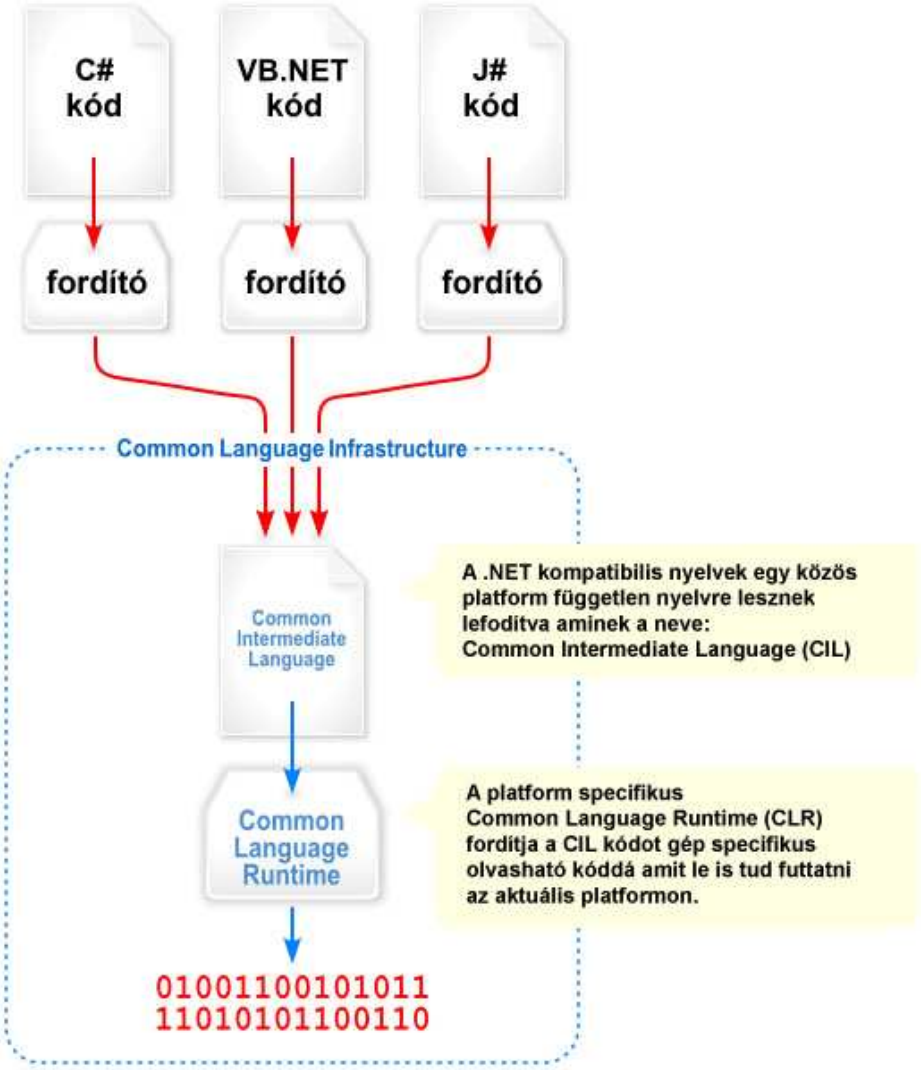
Az alkalmazás web alapú alkalmazás, jellemzően ASP-s környezetben, de használható lokálisan is. Mivel web alapú, így egy háromrétegű: kliens - web szerver - adatbázis alapú alkalmazás készült. Kliens oldalon bármilyen web-böngésző használható, web szerverként a Microsoft IIS 6.0-s változata üzemel rajta .NET 3.5 keretrendszerrel, míg adatbázisként a Microsoft SQL Express 2008 szervere szolgál. A fejlesztés Microsoft Visual Studio 2008 és SQL Server Management Studio Express fejlesztői szoftverek segítségével valósult meg.

A használt szoftvereket, a keretrendszert és az eszközöket és azok tulajdonságait külön részletezni fogom a dolgozatomban, valamint a megfelelő szakmai színvonalra törekedve, ámde közérthető módon tárgyalni fogom az alkalmazás és annak környezetének fő funkcióit tekintve fejlesztői és felhasználói szemszögből.

3.1. A .NET KERETRENDSZER

A .NET keretrendszer a Microsoft által készített és forgalmazott a gyors alkalmazásfejlesztést (Rapid Application Development), a platformfüggetlenséget és a hálózati átlátszóságot támogató szoftverfejlesztői platform. A korábban használt COM platform leváltására tervezték. A .NET keretrendszer eszköztára szinte lefedi a szoftverfejlesztés minden területét úgy, mint kliens és szerveroldali megoldások, adatbázisok kezelése, játékfejlesztés, stb.

A .NET Framework alapjául a Common Language Infrastructure (CLI) szolgál (**1. ábra**). Ez nem más, mint azon szabályok halmaza, amelyek leírnak egy nyelvfüggetlen fejlesztői környezetet, a futtatókörnyezetet, típusrendszert stb. A .NET Framework implementációját Common Language Runtime-nak (CLR) hívják. A CLI maga is négy fő részre oszlik.



1. ábra - Common Language Infrastructure

- Common Language Specification (CLS): A CLI kompatibilis nyelvek által betartandó szabályokat írja le.
- Common Type System (CTS): A CLI azon része, amely a típusokat, azok memóriabeli megjelenését illetve egymással való interakcióját specifikálja.
- Common Language Runtime (CLR): A CLR felelős a CLI nyelven megírt programok betöltéséért valamint futtatásáért, a memóriahasználatért, a kivételkezelésért továbbá a kódbiztonságért is.

- Common Intermediate Language (CIL): A CIL egy úgynevezett köztes kód. Bármely a CLI nyelvben megírt program erre a köztes kódra fog lefordulni. A tényleges futtatáskor a kódot az úgynevezett jitter (Just In Time Compiler) fordítja le a processzor által már értelmezhető natív kódra.

A .NET keretrendszer számtalan programozási nyelvet támogat, melyek mindegyike megfelel a következő programozási paradigmáknak:

- objektum-orientáltság: a világ, mint kis objektumokkal való modellezése, minden objektum saját állapottal és operációkkal rendelkezik, melyeket más objektumok meghívhatnak
- imperatív/deklaratív programozás: imperatív nyelvek esetén úgymond elmagyarázzuk a számítógépnek mit is kell csinálni a cél elérése érdekében, míg deklaratív nyelvek esetén megadjuk a célt és a fordító, futtató rendszerre bízzuk, hogyan jut el oda. Imperatív nyelvek a C#, Java, Delphi, Ada stb. Deklaratív nyelvek: SQL, ASP.NET, stb.
- eseményvezéreltség: bizonyos események bekövetkezése maga után vonhatja más események bekövetkezését és/vagy a hozzájuk rendelt kódrészlet lefutását
- reflexív programozás: a program futás időben képes vizsgálni és módosítani a saját struktúráját és viselkedését.

Néhány támogatott nyelv:

- C#, a C++-hoz és a Java nyelvéhez hasonló objektum-orientált nyelv
- JScript .NET, Microsoft JScript nyelvének fordított verziója
- J#, a Java és a J++ programozók átmeneti nyelve a .NET keretrendszer felé
- Managed C++, a C++ egy változata a .NET platformra
- Visual Basic .NET, a klasszikus Visual Basic egy továbbfejlesztett, objektum-orientált, többszálas verziója

A .NET keretrendszer főbb kiadásai az 1.0, 2.0, 3.0 és jelenleg a legelterjedtebb a 3.5, de hamarosan kiadásra kerül a végleges 4.0-s változat is.

3.2. Az ASP.NET

Az ASP az aktív szerver oldalak (Active Server Pages) rövidítése, a Microsoft szerveroldali keretrendszere, mely dinamikus weboldalak generálására alkalmas. Továbbá fejleszthetőek vele web alkalmazások és web szolgáltatások is. 2002. januárjában jelent meg a .NET 1.0-s verziójával. Alapvető komponense a Common Language Runtime (CLR), mely lehetővé teszi, hogy bármely támogatott .NET programnyelven írhassanak kódot a programozók.

3.3. Az ASP.NET története

Az Internet Information Services 4.0 1997-es kiadása utána Microsoft elkezdte keresni annak a lehetőségét, hogy hogyan tudná kiváltani a problémáit az alap ASP-nek, valamint különös hangsúlyt fektettek arra, hogy elkülönítsék a megjelenést és a tartalmat ezáltal „tiszta” kódot tudjanak írni a programozók. A prototípust XSP-nek hívták és Java nyelven íródott, ám hamarosan áttértek az új CLR-re épülő platformra mely többek között objektum orientált környezetet és szemét gyűjtögetést (garbage collection) is biztosított, melyet a Microsoft COM (Component Object Model) nem. A CLR-re való áttéréssel egy időben az XSP-t újraimplementálták C# nyelven és átkeresztelték ASP+-ra. 4 év fejlesztés és egy sor beta kiadás után 2002. január 5-én jelent meg az ASP.NET 1.0, a .NET keretrendszer 1.0 tagjaként.

3.4. Az ASP.NET LAPJAI

A .NET lapok általában, mint web formok vagy mint webes űrlapok ismertek, melyek a webes alkalmazásfejlesztés „alapelemei”. Ezen űrlapok .aspx kiterjesztésű fájlokban tárolódnak, melyek tipikusan statikus (X)HTML kódot, külön jelölésű szerver oldali kódot vagy felhasználók által definiált vezérlőket tartalmaznak. Ezekben helyezhetik el a fejlesztők a weboldalhoz szükséges dinamikus vagy statikus információkat. A dinamikus kód, mely a szerveren fut két helyen helyezhető el: inline kódként a web form kódjában vagy egy külön helyen a code-behind fájlban.

3.4.1. INLINE MODELL

Amennyiben a szerveren futó kódunkat a web form kódjában akarjuk elhelyezni, úgy ezt megtehetjük egy erre szánt külön blokkban jelölve azt. Ezen technika hasonló más web fejlesztési technikákhoz úgy, mint PHP, JSP és ASP viszont nem ajánlott a használata két okból: a kód egy idő után átláthatatlan lesz, másrészt több hívást igényel a lap a legenerálásakor.

Példa az inline kód használatára (a vastag betűvel jelölt rész az inline modell használatához szükséges):

```
<%@ Page Language="C#" %>

  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<script runat="server">
 protected void Page_Load(object sender, EventArgs e)
    {
       LabelTime.Text = DateTime.Now.ToLongTimeString();
    }
</script>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title>Sample page</title>
</head>
<body>
    <form id="form1" runat="server">
    <div>
       The current time is: <asp:Label runat="server" id=" LabelTime" />
    </div>
    </form>

</body>
</html>
```

3.4.2. CODE-BEHIND MODELL

A Microsoft ajánlása a dinamikus kóddal kapcsolatban, hogy helyezzük el azt egy külön fájlban vagy egy erre a célra kijelölt script részben. A code-behind fájlok tipikus elnevezése MyPage.aspx.cs vagy MyPage.aspx.vb attól függően, hogy milyen nyelvet használ a

programozó. A hozzá tartozó web form elnevezése pedig MyPage.aspx. Ez a technika automatizálva van a Microsoft Visual Studioban és más fejlesztői környezetekben is.

Példa a code-behind modellre:

A web form kódjában elhelyezett részlet (a vastag betűvel jelölt rész a code-behind fájlra való hivatkozást mutatja):

<%@ Page Language="C#" **CodeFile="SampleCodeBehind.aspx.cs" Inherits="Website.SampleCodeBehind" AutoEventWireup="true"** %>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
   <title>Sample page</title>
</head>
<body>
   <form id="form1" runat="server">
   <div>
      The current time is: <asp:Label runat="server" id=" LabelTime" />
   </div>
   </form>

</body>
</html>
```

A code-behind fájlban elhelyezett részlet:

```
using System;

namespace Website
{
        public partial class SampleCodeBehind : System.Web.UI.Page
        {
                protected void Page_Load(object sender, EventArgs e)
                {
                        LabelTime.Text = DateTime.Now.ToLongTimeString();
                }
        }
}
```

Funkciójában mind az inline mind a code-behind modell ugyanazt hajtja végre, viszont utóbbi modell esetén élesen elkülönül a dinamikus és statikus kód, ezáltal megvalósul a „tiszta” kód írásának lehetősége.

3.5. MICROSOFT VISUAL STUDIO 2008

A Microsoft alkalmazások fejlesztésére szánt hivatalos fejlesztői környezet, IDE (Inegrated Development Enviroment). 2007 végén jelent meg a Visual Studio 2008-as verziója, mely manapság a legelterjedtebb. Segítségével tetszőleges Microsoft alapú szoftver, asztali-, webes-, adatbázison alapuló vagy akár mobil alkalmazás fejleszthető. Támogatást és segítséget nyújt, hogy a fejlesztők és cégek minél könnyebben, hatékonyabban tudjanak alkalmazásokat készíteni.

Lehetőséget biztosít a programok megtervezésére, azok futtatására és tesztelésére. Alkalmazva a legújabb technikákat, a mindennapi elvárásoknak megfelelő és azokon túlmutató programok készíthetők vele, mind a felhasználói élményt, a használhatóságot és a biztonságot tekintve. Gyorsítható vele a programozás ugyanis előre megírt, már sokszor tesztelt, leggyorsabban futó rutinok (például sorba rendezések) vagy akár komplexebb modulok (például felhasználói bejelentkezést segítő modul) is találhatóak benne.

Többféle kiadása és változata létezik, így mindenki számára hasznos segítséget adhat. Kielégíti mind az egyéni fejlesztők és mind a csoportos munkában dolgozók igényeit is. Jelenlegi változata a Visual Studio 2010.

3.6. MICROSOFT SQL SERVER 2008 EXPRESS

A Microsoft reláció-modell alapú SQL szervere. Elsődleges nyelve a T-SQL és az ANSI SQL. Jelenlegi legújabb verziója a 2008, mely 2008. augusztus 6 –án jelent meg. Ezen verzió a korábbiakhoz képest már támogatást nyújt strukturált és fél- strukturált adatok tárolására is úgy, mint digitális média formátumok, például: kép, videó és audió.

Az SQL Server Express Edition a Microsoft ingyenes adatbázis kezelő programja, mely tartalmazza a teljes adatbázis kezelő motort viszont, korlátozások találhatóak benne. Az adatbázisok számát vagy a felhasználókat tekintve nincs korlátozás, ugyanakkor fizikai megkötések vannak: a futtató szerverben maximum 1GB memória lehet és 1 processzor,

továbbá az adatbázisok mérete nem haladhatja meg a 4 GB-ot. Az alap adatbázis egy .mdf kiterjesztésű fájlban tárolódik.

3.6.1. SQL SERVER MANAGEMENT STUDIO EXPRESS

Az SQL Server Management Studio egy grafikus adatbázis menedzselő eszköz a Microsoft SQL szerveréhez. Szkript szerkesztőket, valamint grafikus eszközöket biztosít melyekkel az adatbázis objektumokon, műveletek végezhetőek. Ez az elsődleges interfész az SQL 2005 óta, mellyel az SQL szerverek menedzselhetőek. Az SQL Server Management Studio-nak van az SQL Express szerverhez készült verziója is melyet, mint SQL Server Management Studio Express (SSMSE) ismernek.

Az Management Studio legfőbb jellegzetessége az objektum böngésző (Object Browser), mely segítségével a felhasználó böngészheti (browse), szelektálhatja (select) vagy befolyásolhatja szerver bármely objektumát. Többek között használható arra, hogy vizuálisan felügyeljük vagy analizáljuk a végrehajtási terveket és optimalizáljuk az adatbázis teljesítményét. Természetesen létrehozhatunk vele új adatbázisokat vagy módosíthatunk bármely már meglévő sémát, hozzáadhatunk, elvehetünk táblákat vagy indexeket. Tartalmaz továbbá egy lekérdező ablakot (Query window) amely egy grafikus alapú interfész, mellyel írhatunk vagy akár végre is hajthatunk lekérdezéseket.

3.7. Az ADO.NET

A .NET Framework az adatbázisok kezelését az ADO.NET-ben (ActiveX Data Objects) definiált osztályok segítségével támogatja. A programozók legtöbb esetben ezt használják, hogy adatokat nyerjenek ki vagy manipuláljanak adatbázisokban. Lehetőséget biztosít a fejlesztők számára, hogy a nélkül érjék el az adatbázisokat, hogy ismernék azok implementációját, egyedül a kapcsolódáshoz szükséges információkkal kell rendelkezniük.
Két féle módon érhetjük el az adatbázist:

- Hagyományosan SQL kódokat írva, úgynevezett DataProvider-t használva lekérdezzük az adatokat, melyeket a DataAdapter –k segítségével úgynevezett DataSet –ben tárolunk. Ez által nincs szükség folyamatos adatbázis kapcsolatra és a módosítások sem egyből az éles adatbázisban történnek. A DataProvider az

alkalmazás és az adatbázis közötti kapcsolat létesítésére szolgál, a DataAdapter pedig arra megfelelő parancsok segítségével hidat képez a DataSet és a kinyert adatok között.

- LINQ (Language Integrated Query)–t használva. Ezen technológia lehetőséget biztosít a .NET nyelveknek natív adatelérési kódok használatára. A mai fejlesztők az objektumorientált gondolkodást és tervezést részesítik előnyben. A LINQ ebben segít, hogy a nem objektumorientált adatforrásokkal való műveletek, mint például az XML, vagy a relációs adatbázisok kezelése egyszerűbb legyen, jobban igazodjon ehhez a szemlélethez. Segítségével az adott nyelv eszközeivel SQL szintaxis szerűen írhatunk lekérdezéseket. Ebből az SQL szerű lekérdezésből viszont a gép generál egy valódi SQL parancsot, mely a lehető legjobban optimalizált és majd ezt hajtja végre.

3.8. Az Internet Information Services (IIS)

Az Internet Information Services (IIS) a Microsoft által fejlesztett, a Windows-platformon futó internet-alapú szolgáltatásokat összefogó termék neve. Az Apache HTTP Server után a leggyakrabban használt webkiszolgáló (a weboldalak száma alapján). Az IIS különböző szolgáltatásokat nyújt úgy, mint FTP, FTPS, SMTP, HTTP és HTTPS.

Első megjelenése a Windows NT 3.51-el történt, mint egyszerű webkiszolgáló. Későbbiekben, ahogy egyre nőttek az igények a web szolgáltatásokra úgy fejlődött az IIS is. Az IIS 3.0–ban került bevezetésre az alap ASP szolgáltatás. Idővel minden egyes Windows NT kiadással jött egy újabb IIS is, míg nem belekerült a leghasználtabb operációs rendszerbe is a Windows XP-be. Az XP-vel az 5.1 es IIS került kiadásra, mely ugyan ingyenes és bárki számára telepíthető, ugyanakkor korlátozza a kapcsolatok számát, egy időben egyszerre 10 kapcsolatot szolgál ki és egy weblapot. Jelenleg a Windows Server 2008 és a Windows 7–hez már az IIS 7.0 –t szolgáltatják. A 7.0 rengeteg újítást tartalmaz elődjeihez képest. Moduláris felépítésű lett, sokkal biztonságosabb, jobban skálázható és rugalmas.

4. EREDMÉNYEK

4.1. A FEJLESZTÉSI FOLYAMAT ÉS AZ ELKÉSZÜLT ALKALMAZÁS ÁTTEKINTÉSE

A dolgozatom keretében létrejött alkalmazás, a kitűzött céloknak megfelelően törekszik a törvényi és jogszabályi előírások betartására, könnyen kezelhető és felhasználóbarát, valamint a használói számára biztosított a közbeszerzési eljárások egységes rendszerszerben történő kezelése és teljes körű, rendszerezett nyilvántartása és dokumentálása.

A kifejlesztett alkalmazás létrehozása több lépcsőben történt:

- Specifikálás: meghatároztam, hogy az alkalmazásnak milyen megszorítások mellett milyen elvárásoknak kell megfelelnie mind folyamat, mind a környezet és a kezelhetőség szempontjából.
- Tervezés: a specifikációt figyelembe véve fizikai és logika tervezés következett, mely magában foglalta az adatbázis szerkezetének valamint a program struktúrájának, logikájának megtervezését.
- Implementálás: a fizikai és logikai tervek alapján megvalósítottam, leprogramoztam az alkalmazást.
- Validálás és verifikálás: ellenőriztem, hogy az alkalmazás megfelel-e az elvártaknak, vagyis a specifikációban foglaltaknak és hogy kielégíti-e a felhasználók által meghatározott igényeket.

Tekintettel az alkalmazás méretére, mely rendkívül nagy és összetett – több mint 50 formból és több mint 13000 kódsorból áll -, csak bizonyos mélységben, a dolgozat terjedelmi korlátait figyelembe véve mutatom be a főbb tervezési, programozási és általam fontosnak vélt momentumokat.

4.2. SPECIFIKÁLÁS

Az alkalmazásnak bárhonnan könnyen elérhetőnek és kezelhetőnek kellett lennie, ugyanakkor számos, egyidejű felhasználó igényeit kell kiszolgálnia, megfelelő válaszidőkkel. Az alkalmazás törvényi megfelelésének szempontjából teljes mértékben eleget kellett tennie a „2003. évi CXXIX. törvény a közbeszerzésekről” törvénynek és e törvény aktuális módosulásainak.

Ezen szempontokat szem előtt tartva többek között az alábbi igények fogalmazódtak meg:

- A teljes közbeszerzési folyamatot modellezze, azaz a hirdetmény megjelenésétől kezdve, az eljárás lefolytatásán át az eredményhirdetésig valamennyi lépést.
- A program elérhető legyen interneten adott esetben intraneten.
- Több felhasználó különböző jogosultságokkal használhassa a programot, azaz valamennyi a közbeszerzési eljárás lefolytatásában érintett szereplő munkáját támogassa (pl.: közbeszerzési referensek, döntéshozók, az eljárást lebonyolítók, bírálóbizottsági tagok, stb.).
- Minden egyes a Kbt.-ben meghatározott eljárás fajtához biztosítsa azon eszközöket, melyek segítségével egy eljárás a megjelenésétől számítva végigvihető a rendszerben, dokumentálható annak minden egyes momentuma.
- A rendszer nyújtson segítséget az egyes eljárások egymást követő lépéseinek végigvitelében.
- Közbeszerzési tervet lehessen vele készíteni és az abban foglalt eljárások archiválása is legyen megoldott.
- Automatizmusok segítségével előre figyelmeztessen bizonyos határidőkre, melyek az adott eljárás életciklusában fontos szerepet játszanak.

Ezen fő szempontokat és még sok más apróbbat szem előtt tartva került megalkotásra az alkalmazás ASP.NET környezetben.

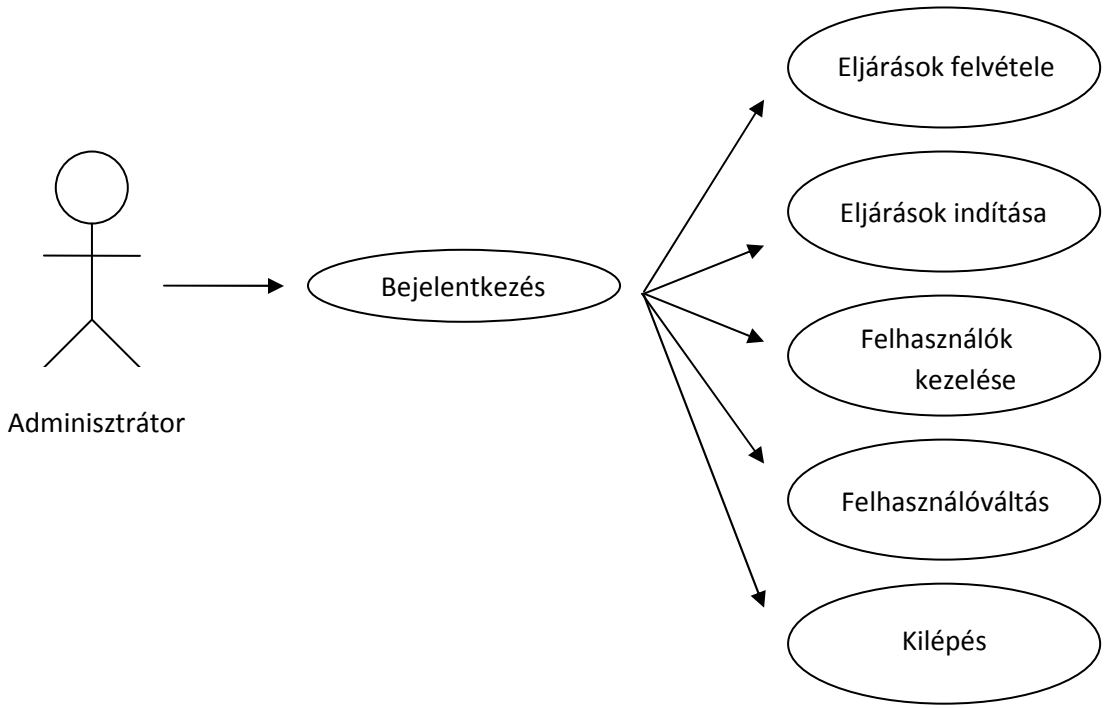
4.3. TERVEZÉS

4.3.1. USE CASE ÉS ACTIVITY DIAGRAM

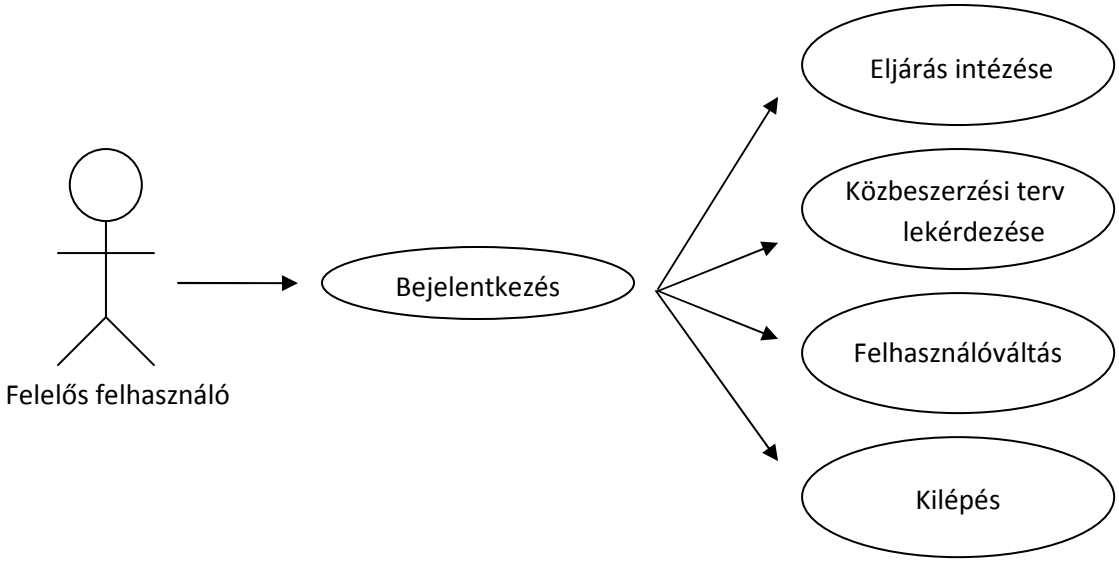
A specifikációs és felhasználói szempontokat szem előtt tartva UML (Unified Modeling Language) diagramok segítségével terveztem meg az alkalmazást alkotó algoritmusokat, osztályokat valamint a folyamatokat. Az UML egy egységesített modellező nyelv melynek segítségével egy rendszer különböző nézeteit tudjuk modellezni, tervezni és dokumentálni. Az UML modelleket és diagramokat használ a különböző nézetek leírására.

Az UML használat eset (Use Case) diagramja segítségével lehet leírni, hogy a felhasználók milyen interakciókat végezhetnek a rendszeren belül, mihez is van joguk. A funkcionalitás megtervezésében jelentős segítséget nyújt. Ezen diagramot használva szemléltetem a felhasználók jogosultságait. Három féle felhasználói típust definiáltam:

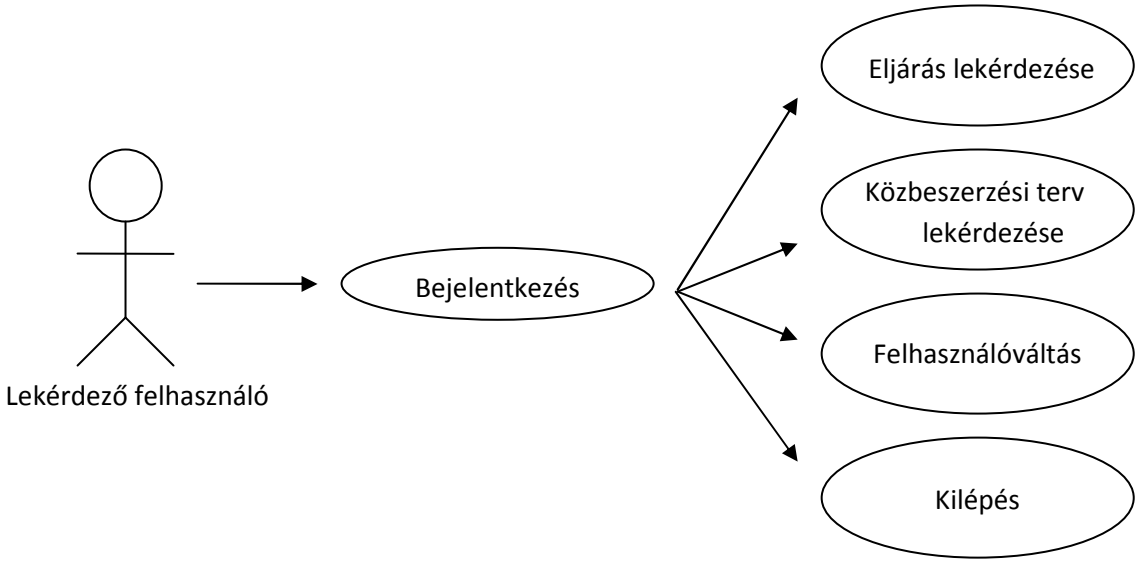
- Adminisztrátor: ő a felelőse a rendszert használók karbantartásáért, új eljárások felvételéért a közbeszerzési tervbe, valamint a már felvitt eljárások elindításáért. (**2. ábra**)
- Felelős felhasználó: az adminisztrátor által rászabott eljárások lefolytatásért és csak azért felelős személy. (**3. ábra**)
- Lekérdező felhasználó: hasonló a felelős felhasználóhoz, annyi különbséggel, hogy a hozzárendelt eljárásokban módosítást nem tud végrehajtani, csupán betekintést nyer annak aktuális állapotába, megtekintheti annak dokumentumait, hozzá kapcsolódó eseményeit. (**4. ábra**)



2. ábra – Adminisztrátor felhasználó jogosultságai



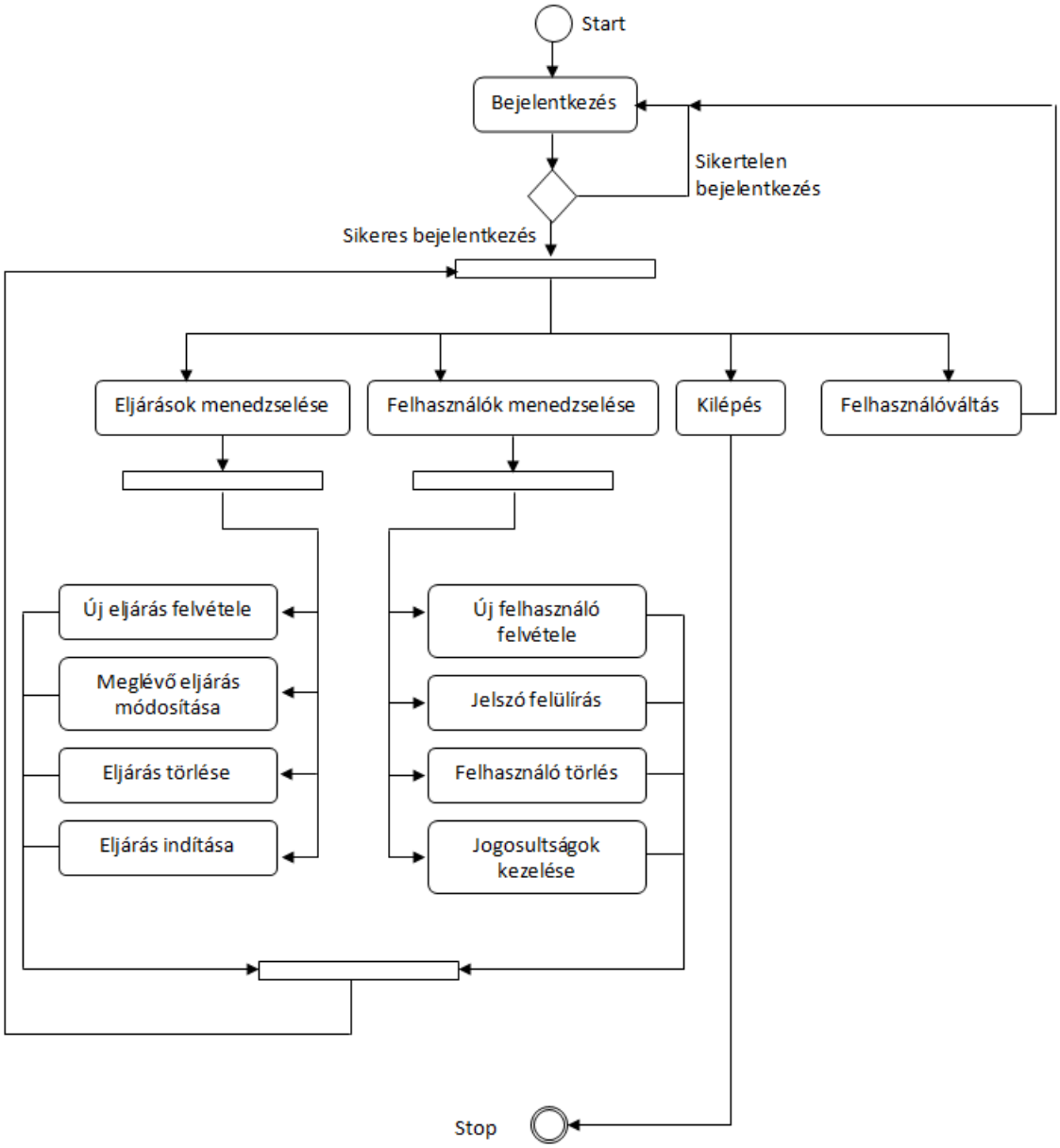
3. ábra – Felelős felhasználó jogosultságai



4. ábra – Lekérdező felhasználó jogosultságai

Az UML aktivitás diagramja (Activity Diagram) segítségével lehet grafikusan, lépéseiben ábrázolni egy munkafolyamatot, szemléltetni a párhuzamos folyamatokat, az elágazásokat illetve a ciklusokat. Leírhatunk vele egy üzleti folyamatot vagy akár egy lépésről-lépésre haladó munkafolyamatot is.

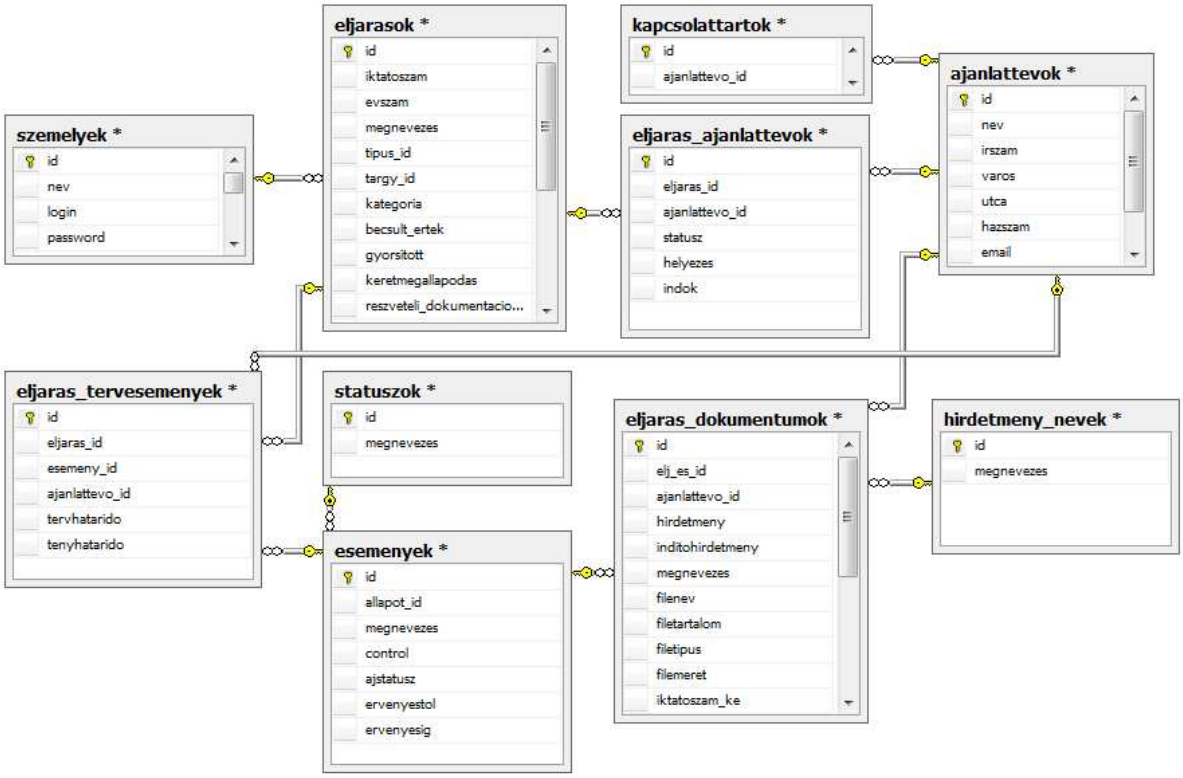
Ezen aktivitás diagramot felhasználva ábrázolom a program fő működési elvét az adminisztrátor szemszögéből. (**5. ábra**)



5. ábra – A munkafolyamat az adminisztrátor szemszögéből

4.3.2. AZ ADATBÁZIS FELÉPÍTÉSE

A folyamatok és az alkalmazás összetettségének köszönhetően egy jól átgondolt adatbázis megtervezésére volt szükség. A szoftvert sokan használhatják egy időben, akár több különböző intézmény is, ezen okból kifolyólag külön kezelni kellett, hogy az egyes intézményekhez az adatbázis mely részeit (tábláit) kell külön, önállóan intézményfüggően karbantartani. A többi „közös” tábla karbantartását pedig egy külön szinkronizációs program végzi. Az adatbázis méretére való tekintettel, mely több mint 25 táblából áll, egy részletet mutatok be a következőekben. (**6. ábra**)



6. ábra – Az elkészült adatbázis egy részlete

A fenti ábrán látható adatbázisrészlet az a minimum, mely segítségével már megvalósítható az eljárások tárolása, az automatizmusok alkalmazása, határidők figyelése és a felhasználók jogosultságkezelése. Továbbiakban részletesebben kifejtem a **6. ábrán** látható adatbázis táblák szerepét, tulajdonságait.

4.3.3. Az ADATBÁZIS TÁBLÁK FELÉPÍTÉSE

- Esemenyek: azt tárolja, hogy egy eljárás lefolytatása közben milyen események történhetnek vele.
    - Id: szám típusú, az egyes események megkülönböztetésére szolgál.
    - Megnevezes: szöveg típusú, az esemény nevét tartalmazza.
    - Control: szöveg típusú, az esemény bekövetkezésének hatására mely aspx lapot kell betölteni.
    - Ajstatusz: szám típusú, az eseményhez milyen ajánlattevői státusz szükséges.
    - Ervenyestol: dátum típusú, behatárolja, hogy mely aktuális Kbt. törvényt kell rá alkalmazni.
    - Ervenyesig: dátum típusú, behatárolja, hogy mely aktuális Kbt. törvényt kell rá alkalmazni.
- Kapcsolattartok: az egyes az eljárásra pályázatot benyújtókhoz (továbbiakban ajánlattevők) tartozó kapcsolattartó személyét tárolja.
    - Ajanlattevo_id: szám típusú, mely ajánlattevőhöz tartozik a kapcsolattartó.
    - Név: szöveg típusú, a kapcsolattartó neve.
    - Cím: szöveg típusú, a kapcsolattartó címe.
    - Telefon: szám típusú, a kapcsolattartó telefonszáma.
    - E-mail: szöveg típusú, a kapcsolattartó e-mail címe.
- Ajanlattevok: az egyes az eljárásokra pályázatot benyújtók tárolása.
    - Id: szám típusú. az egyes ajánlattevők megkülönböztetésére szolgál.
    - Név: szöveg típusú, az ajánlattevő neve.
    - Irszam: szám típusú, az ajánlattevő irányítószáma

- Varos: szöveg típusú, az ajánlattevő városa.
- Utca: szöveg típusú, az ajánlattevő utcája.
- Hazszam: szöveg típusú, ajánlattevő házszáma.
- Telefon: szám típusú, az ajánlattevő telefonszáma.
- E-mail: szöveg típusú, az ajánlattevő e-mail címe.

- Eljaras_tervesemenyek: milyen események vannak tervbe véve az eljárással kapcsolatban.
  - Id: szám típusú, az egyes tervesemények azonosítására szolgál.
  - Eljaras_id: szám típusú, az eljárás azonosítására szolgál.
  - Esemeny_id, szám típusú, az esemény azonosítására szolgál.
  - Ajanlattevo_id: szám típusú, az ajánlattevő azonosítására szolgál.
  - Tervhatarido: dátum típusú, az esemény tervezett bekövetkezésének időpontja.
  - Tenyhatarido: dátum típusú, az esemény tényleges bekövetkezésének időpontja

- Eljaras_dokumentum: az egyes eljárásokhoz tartozó információk és hozzá kapcsolódó dokumentumok tárolására szolgál.
  - Id: szám típusú, az egyes dokumentumok azonosítására szolgál.
  - Elj_es_id: szám típusú, az eljárás e dokumentumához tartozó esemény id -ja.
  - Ajanlattevo_id: szám típusú, a dokumentumhoz tartozó ajánlattevő id –ja.
  - Hirdetmeny: szám típusú, azt jelöli, hogy a dokumentum hirdetmény-e.
  - Inditohirdetmeny: szám típusú, azt jelöli, hogy a dokumentum indítóhirdetmény-e.
  - Megnevezes: szöveg típusú, a dokumentum elnevezése.
  - Filenev: szöveg típusú, a feltöltött dokumentum neve.

- Filetartalom: binary típusú, maga a feltöltött dokumentum.
- Filetipus: szöveg típusú, a feltöltött fájl típusa.
- Filemeret: bigint típusú, a feltöltött fájl mérete.
- Iktatoszam_ke: szöveg típusú, amennyiben a Közbeszerzési Értesítőben megjelent annak iktatószáma.

- Eljarasok: a rendszerbe felvitt közbeszerzési eljárások tárolódnak ebben a táblában.
  - Id: szám típusú, az egyes eljárások azonosítására szolgál.
  - Iktatoszam: szöveg típusú, az eljáráshoz tartozó belső iktatószám.
  - Evszam: dátum típusú, az eljárás indításának éve.
  - Megnevezes: szöveg típusú, az eljárás neve.
  - Tipus_id: szám típusú, az eljárás fajtájának meghatározására szolgál.
  - Targy_id: szám típusú, az eljárás tárgyának meghatározására szolgál.
  - Kategoria: szöveg típusú, az eljárás kategóriájának tárolása.
  - Becsult_ertek: szám típusú, az eljárás lefolytatásához szükséges becsült összeg.
  - Gyorsitott: szám típusú, az eljárás gyorsított eljárás-e.
  - Keretmegallapodasos: szám típusú, az eljárás keretmegállapodással kerül-e lefolytatásra.
  - Reszveteli_dokumentacio: szám típusú, az ajánlattevőknek kiküldött dokumentáció, mely a részvételi feltételeket tartalmazza.
- Eljaras_ajanlattevok: az egyes eljárásokhoz tartozó ajánlattevők és azok az eljárás lefolytatásában elért státuszuk tárolására szolgáló tábla.
  - Id: szám típusú, az egyes státuszok azonosítására szolgál.
  - Eljaras_id: szám típusú, mely eljáráshoz tartozik ez a státusz.

- Ajanlattevo_id: szám típusú, a státuszt mely ajánlattevő érte el.
- Statusz: szám típusú, az elért státuszt határozza meg.
- Helyezes: szám típusú, amennyiben az eljárás lefolytatásra került és az ajánlattevő elért egy helyezést, az itt tárolódik.
- Indok: szöveg típusú, amennyiben valamely okból kifolyólag az ajánlattevő ajánlata érvénytelen és/vagy kizárásra került, itt található a hozzá tartozó indok.

- Hirdetmeny_nevek: ebben a táblában tárolódnak az egyes eljárásokhoz hozzárendelhető hirdetmény fajták.
  - Id: szám típusú, a hirdetmény azonosítására szolgál.
  - Megnevezes: szöveg típusú, a hirdetmény neve.
- Szemelyek: a rendszerbe felvett személyeket tároló tábla.
  - Id: szám típusú, a személyek azonosítására szolgál.
  - Nev: szöveg típusú, az személy nevét tárolja,
  - Login: szöveg típusú, az adott személy bejelentkezési neve.
  - Password: szöveg típusú, az adott személy bejelentkezési jelszava.
- Statuszok: az eljárás lefolytatása során elérhető státuszokat tárolja a tábla.
  - Id: szám típusú, a státuszok azonosítására szolgál.
  - Megnevezes: szöveg típusú, a státusz elnevezése.

4.4. IMPLEMENTÁLÁS

A tervezést követően az alkalmazás fejlesztését implementációs szakaszokra osztottam. Külön-külön és egymásra épülően, a programrészek és a hozzájuk tartozó adatbázisrészletek egymást kiegészítve alakítottam ki. Alábbiakban ismertetem a főbb szakaszokat:

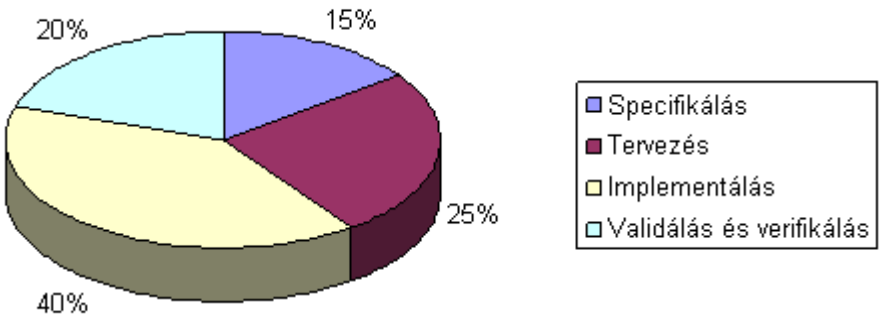
- Bejelentkező képernyő – alap felhasználók és bejelentkezési nevek rögzítése adatbázisban.
- Jogosultságnak megfelelő képernyő, menü betöltése – MasterPage elkészítése.
- A közösen mindenki által használható részek elkészítése – UserControl-ok megalkotása.
- Dokumentumok, adatok adatbázisban történő tárolása, lekérése, módosítása – SQL szkriptek megalkotása.
- A használat során felmerülő hibaüzenetek kezelése (pl.: adatbázis nem elérhető) – Tájékoztató képernyő elkészítése a hibáról.
- A program használat közben felmerülő javaslatok, igények tárolása a továbbfejlesztésre való tekintettel – Űrlapok létrehozása ezen igények bejelentésére.

A fejlesztett alkalmazást ASP.NET környezetben implementáltam. MasterPage segítségével a különböző jogú felhasználók hasonló menüszerkezettel ugyan, de más tartalommal találkoznak. Az egyes eseményekhez (pl.: eljárás felvitele a rendszerbe, dokumentum feltöltése, eljárás intézése, stb.) különböző UserControl-ok töltődnek be. A felhasználói felületet az eljárások intézésének logikáját követve, csoportosítottam bizonyos részekre, így AJAX eszközrendszer használatával különböző tabulátor füleken, táblázatokban, paneleken lettek elhelyezve az egyes interakciókhoz szükséges gombok, illetve értesítési felületek. Mivel ASP-s alkalmazásról van szó esetenként előfordulhat, hogy a szoftver külső okból kifolyólag nem elérhető, ilyen például ha megszakad a felhasználó által előfizetett internet elérés. Ez és hasonló, esetlegesen felmerülő esemény bekövetkezését hibaüzenetben tudatom a felhasználóval.

4.5. VALIDÁLÁS ÉS VERIFIKÁLÁS

Az elkészült programot felül kell vizsgálni, hogy teljes mértékben megfelel-e mind a specifikációnak és mind pedig a felhasználók, megrendelők által elvártaknak. Továbbá ellenőrizni, tesztelni kell a programot, hogy bármilyen a felhasználó által végrehajtható lépés megfelelően működik-e és fel kell készíteni a programot a nem várt események kezelésére is.

Amennyiben a teljes alkalmazás megvalósításához szükséges időt 100% -nak tekintjük, úgy a (**7. ábra)** mutatja, hogy a fejlesztési folyamat egyes lépései mennyi időt vettek igénybe.



7. ábra - A fejlesztési folyamat lépéseinek időigénye százalékban

4.6. AZ ALKALMAZÁS FUTTATÁSÁHOZ SZÜKSÉGES KÖRNYEZET

Alábbiakban megvizsgáljuk, hogy a program helyi, illetőleg távoli használata esetén milyen eszközök, szoftverek szükségesek a működéséhez.

4.6.1. INTERNETES HASZNÁLAT

Amennyiben a közbeszerzésre kötelezett szervezet egy megbízott céget vesz igénybe a fejlesztett alkalmazás kezelésére és tárolására, valamint az igényeik kiszolgálására, úgy felhasználói oldalon mindössze egy működő számítógép és egy bármilyen a felhasználó által preferált internetböngésző szükséges.

A felhasználónak mindössze annyi dolga van, hogy az internetböngésző címsorába beírja az alábbi címet (jelen esetben a www.x7wvay.hu fiktív cím helyettesítendő a választott cég által megadott címmel):

https://www.x7wvay.hu/kozbeszerzes és már el is kezdheti használni az alkalmazást.

4.6.2. INTRANETES HASZNÁLAT

Amennyiben a közbeszerzésre kötelezett szervezet saját eszközein akarja az alkalmazást kezelni és tárolni, úgy a következő szoftverekre lesz szükség az alkalmazást kiszolgáló számítógépen:

- Microsoft Internet Information Services 6.0 vagy újabb verziójú alkalmazás és web szerver.
- Microsoft SQL Server 2005 vagy újabb verziójú adatbázis szerver.
- A .NET keretrendszer 3.5-s verzió telepítése szükséges a futtató számítógépen.
- Az AJAX eszközrendszer telepítése a futtató számítógépen.

A felhasználói oldalt tekintve, továbbra is elegendő egy számítógép és bármely a felhasználó által preferált internetböngésző, melynek címsorába a következő címet beírva megkezdődhet az alkalmazás használata:
https://localhost/kozbeszerzes .

4.7. AZ ELKÉSZÜLT ALKALMAZÁS BEMUTATÁSA

Alábbiakban – a dolgozat terjedelmi korlátaira való tekintettel - az elkészült alkalmazás két részletét, a bejelentkezést és az adminisztrátori funkciókat fogom bemutatni.

4.7.1. A BEJELENTKEZŐ KÉPERNYŐ

Miután a böngésző címsorába beírtuk a megfelelő címet, az alkalmazás a mindenki számára egységes bejelentkező képernyővel fog elindulni (**8. ábra**). Itt lehetőségünk van az alkalmazásba való bejelentkezésre a megfelelő felhasználói név és jelszó páros használatával, valamint a stílus kiválasztásával a program kinézetének megváltoztatására. A belépés gombbal a beírt adatokat az alkalmazás autentikálja.



8. ábra – A bejelntkező képernyő

Amennyiben a bejelentkezésünk sikertelen, úgy az alkalmazás kéri a hibás adatok ismételt, de immár javított bevitelét.

Ha valós adatokat adtunk meg, akkor az alkalmazás a beírt felhasználói név jogkörének megfelelően bejelentkezteti a felhasználót és megjeleníti az általa elérhető menüket, funkciókat. Jelen esetben adminisztrátori jogkörrel jelentkezünk be.

4.7.2 AZ ADMINISZTRÁTORI FELÜLET

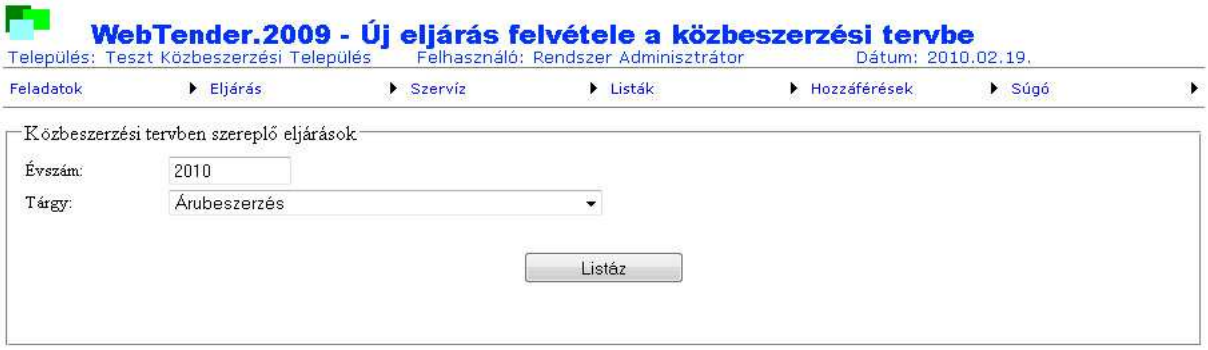
Az adminisztrátor a rendszer előre beépített alap felhasználója. Az alábbi feladatokat láthatja el:

- A közbeszerzési terv karbantartása.
- Közbeszerzési eljárások indítása.
- Felhasználók kezelése, felvétele és jogosultságuk megadása.
- Az ajánlatkérő adatainak karbantartása.
- Iratminták, saját dokumentum sablonok karbantartása.

4.7.2.1. A KÖZBESZERZÉSI TERV KARBANTARTÁSA

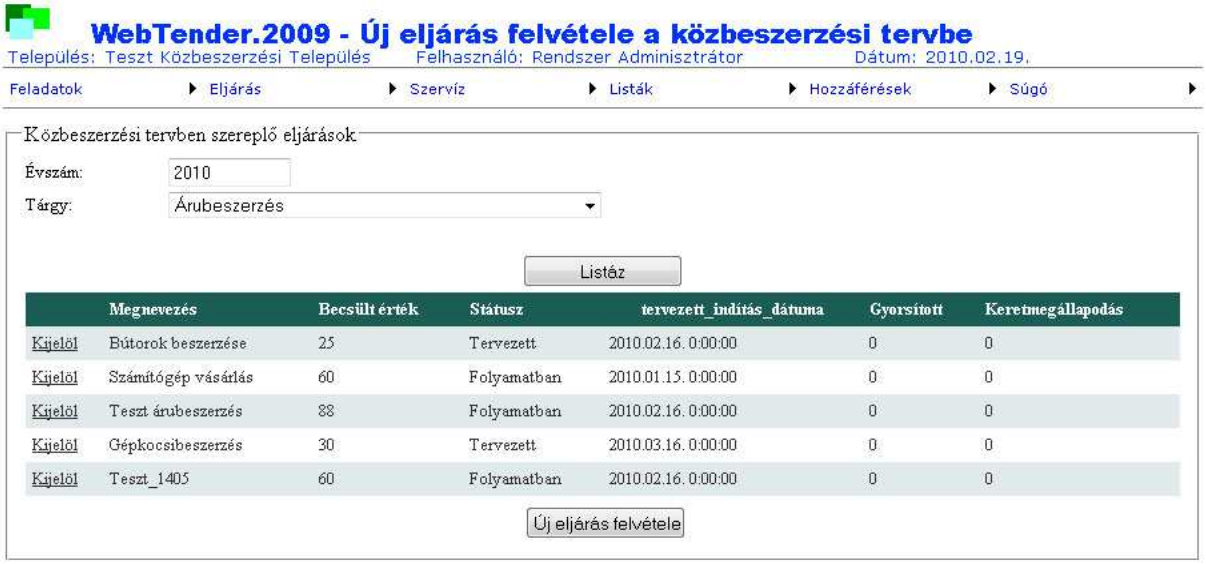
Az Eljárás menüpont *Indítás* alpontját választva az alkalmazás lehetőséget biztosít az éves közbeszerzési terv nyilvántartására, továbbá a megfelelő rezsim és az abban folytatható eljárás fajták kiválasztására (**9. ábra**). A közbeszerzési terv lekérdezésekor az adott eljárás

alapadatait (pl.: hirdetmények, kapcsolódó határidők, kapcsolódó dokumentumok, stb.) megtekinthetőek.



9. ábra – A közbeszerzési terv karbantartása

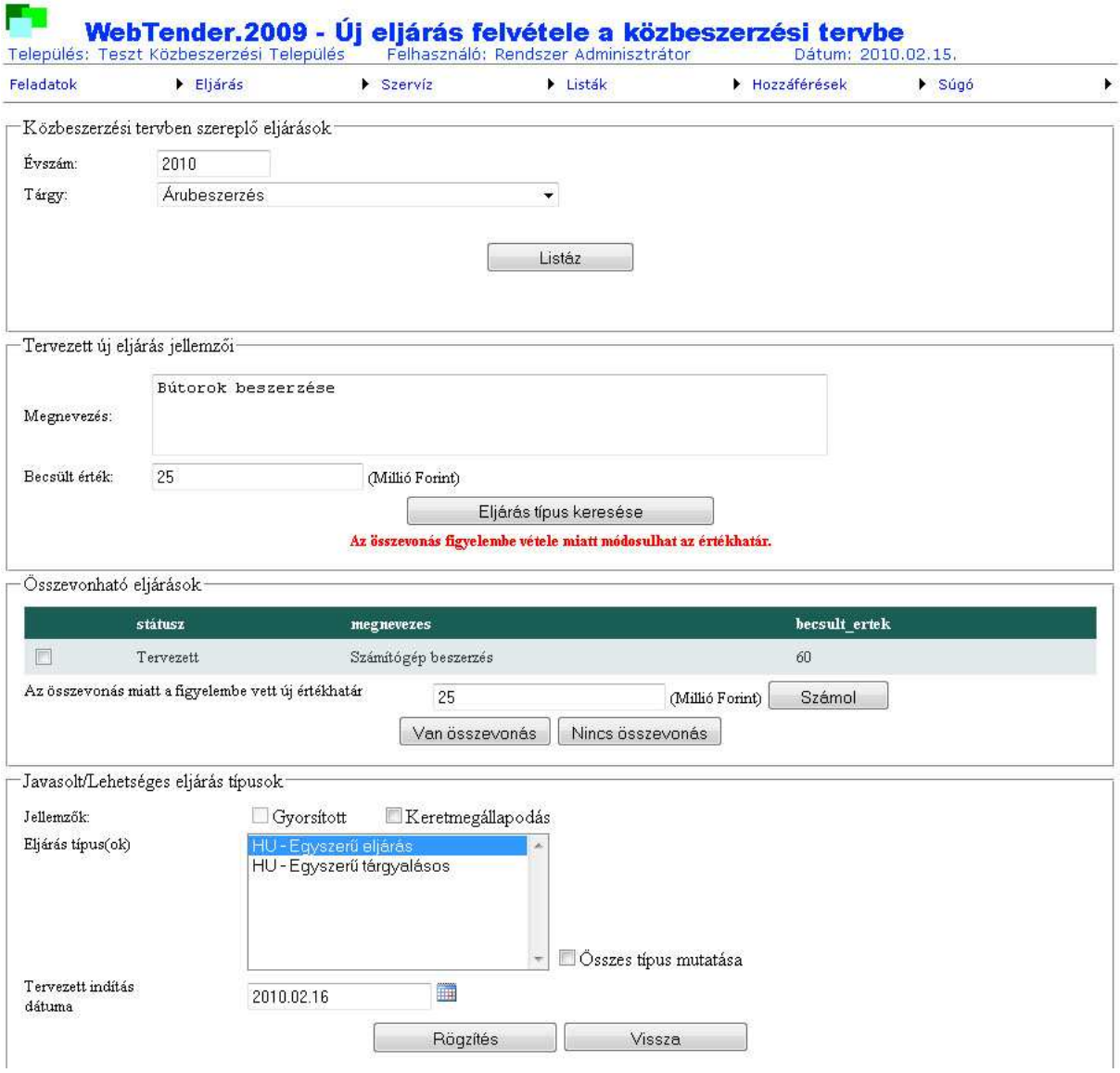
A *Listáz* gomb megnyomása után listázásra kerülnek a feltételeknek megfelelően a közbeszerzési tervben szereplő eljárások (**10. ábra**).



10. ábra - Eljárások listázása

Ahhoz, hogy műveleteket lehessen végezni egy adott eljárással, ki kell jelölni azt. Ez a művelet az eljárás előtt található *Kijelöl* linkkel megtehető. Ezen linket egyszer megnyomva az eljárás kiválasztásra kerül. Ekkor lehetőség nyílik a már felvitt és kijelölt, de még el nem

indított eljárások módosítására, törlésére. Új eljárás felvétele kijelölés nélkül is megtehető, melynek hatására egy új felület fog megjelenni (**11. ábra**).



11. ábra – Új eljárás felvétele

Az *Eljárás típus keresése* gombot megnyomva a program a feltételeknek megfelelő eljárás fajtákat fogja ajánlani. Ezek után a szükséges adatok kitöltésével valamint a *Rögzítés* gomb használatával az adminisztrátor felviheti a közbeszerzési tervbe az eljárást. Ha valamely nélkülözhetetlen információt nem töltött ki a felhasználó – melyet piros csillaggal jelöltem -, akkor az alkalmazás annak pótlásáig nem engedi a tervbe rögzítést.

4.7.2.2. KÖZBESZERZÉSI ELJÁRÁSOK INDÍTÁSA

Ahhoz, hogy egy már felvitt, de még el nem indított eljárást elindítson, az adminisztrátornak elsőként ki kell azt jelölni, majd az ezután megjelenő *Eljárás indítása* gombot kell megnyomni. Ezt követően egy további felület jelenik meg, ahol az eljárás indításához szükséges adatok rögzíthetőek.

4.7.2.3. FELHASZNÁLÓK KEZELÉSE ÉS JOGOSULTSÁGUK MEGADÁSA

A *Szerviz* menüpont *Felhasználók kezelése* alpontját választva az adminisztrátor karbantarthatja a már meg lévő felhasználókat vagy akár újakat vehet föl. Amennyiben már meglévő felhasználó adatait akarja módosítani, úgy elsőként ki kell azt választani a felsoroltak közül, majd a *Módosítás* gombra kattintva adhatja meg az új adatokat és a *Mentés* gombbal véglegesítheti azokat (**12. ábra**). Ugyanezen kijelöléses módszert alkalmazva felhasználók törölhetőek a rendszerből.

Új felhasználó esetén nem kell kijelölni egyik felhasználót sem. Ki kell tölteni az adatokat és ha a beírt azonosítóval megegyező felhasználó még nem létezik, akkor a *Mentés* gomb megnyomásával az új felhasználó bekerül a rendszerbe.



12. ábra - Felhasználók kezelése

Amennyiben egy felhasználó jelszavát kívánja az adminisztrátor felülírni, úgy megteheti azt a *Hozzáférések* menü *Jelszó felülírása* alpontjának választásával (**13. ábra**).



13. ábra – Jelszó felülírása

A megjelenő képernyőn első lépésben ki kell választani a legördülő menüből, hogy melyik felhasználó jelszavát kívánja az adminisztrátor felülírni. Második lépésben meg kell adni az új jelszót majd a *Módosítás* gomb megnyomásával rögzíteni azt.

Az egyes közbeszerzési eljárások lebonyolításban, felülvizsgálatában érintett személyek számára biztosított az adott eljáráshoz való hozzáférés. Azon jogokat, hogy egy személy milyen jogosultsággal bír az egyes eljárások felett a *Szerviz* menüpont *Jogosultságok* alpontjában lehet megadni (**14. ábra**).



14. ábra - Felhasználói jogosultságok kezelése

Elsőként ki kell jelölni a legördülő menüből, hogy mely felhasználónak akar az adminisztrátor jogosultságot adni. Második lépésben hozzárendeli az eljárásokat, melyet az eljárás sorának elején szereplő jelölőnégyzet bejelölésével tehet meg. Végül a *Rögzít* gombbal elmenti a módosításokat.

4.7.2.4. AZ AJÁNLATKÉRŐ ADATAINAK KARBANTARTÁSA

Az ajánlatkérő, vagyis a közbeszerzésre kötelezett szervezet adatainak megadása a *Szerviz* menüpont *Alapadatok* alpontjának kiválasztásával történik (**15. ábra**).

Itt megadható minden szükséges adat úgy, mint név, irányítószám, város, utca, házszám, telefon, fax és e-mail cím. A *Rögzítés* gomb segítségével elmenthetőek a megadott információk.



15. ábra - Az ajánlatkérő alapadatai

4.7.2.5. IRATMINTÁK, SAJÁT DOKUMENTUM SABLONOK KARBANTARTÁSA

A rendszerhez tartozik egy iratminta gyűjtemény, mely eljárás típusonként csoportosítva lehetőségét ad, hogy az eljárás során használt dokumentumok egységes formában készüljenek el. A dokumentumok eljárási adatokkal történő feltöltése az alkalmazás által automatikusan történik, így hatékonyabbá teszi a közbeszerzési feladatokat ellátó szakemberek munkáját. Továbbá biztosított a már megszokott sablonok használatának lehetősége, vagyis egyéni iratminták is feltölthetőek. Ezek az újonnan feltöltött sablonok elérhetőek lesznek az eljárás lebonyolítója számára.

5. ÖSSZEFOGLALÁS

Szakdolgozatom keretein belül egy webes alkalmazást készítettem el, mely egyaránt használható internetes és intranetes környezetben. Az alkalmazás célja a közbeszerzési eljárás elektronizálása volt. A kitűzött céloknak megfelelően rendszert és átláthatóságot visz a közbeszerzésre kötelezett szervezetek folyamataiba, megoldja a keletkezett dokumentumok hosszú távú tárolását valamint azok rendezett, szervezett visszakeresését és a dokumentum napló készítését. Az elektronikus irattár funkció segítségével az alkalmazás fizikailag helyet és papírt spórol meg, valamint egyszerűbben központilag elérhetőek a közbeszerzési folyamatok dokumentumai. Törekszik a törvényi előírásokhoz a lehető legpontosabban igazodni. A beépített automatizmusok segítségével lépésről lépésre segíti a közbeszerzési eljárások lefolytatását, felajánlja a következő és az opcionális lépéseket. Automatikusan figyeli a betartandó határidőket és lehetőséget ad saját határidők definiálására, valamint a határidőket grafikusan is megjeleníti a beépített naptár funkció segítségével. Az említett automatizmusok segítik a közbeszerzési eljárás lefolytatásában résztvevő valamennyi szereplőt. Az eljárást lefolytató személynek egy támaszt nyújt, ha esetleg bizonytalan az eljárás szakaszainak folyamatát vagy a határidőket illetően, továbbá könnyíti a munkáját a beépített sablonok használatával és azok automatikus kitöltésével. Az egyes eljárásokhoz és szakaszokhoz jogi segítséget is ad az alkalmazás. A bírálóbizottsági tagoknak és a döntéshozóknak pedig egyfajta rálátást biztosít az eljárásokra. Láthatják azok állapotát, a hozzá kapcsolódó dokumentumokat anélkül, hogy egy második személy közbenjárása szükséges lenne (pl.: nem kell megkérni egy alkalmazottat, hogy gyűjtse össze papíralapon a szükséges dokumentumokat). Az alkalmazás ergonomikus és könnyen kezelhető. A felhasználói felület egységes, a gyakran használt funkciók könnyen elérhető helyen vannak, a menüpontok nem tagolódnak számtalan alpontra.

Az alkalmazás a legújabb és szabványos technológiákat alkalmazza, szabványos interfészeken keresztül lehetőség van más szabványos interfészeket használó programokkal való interakcióra. Ezen interfészek lévén az alkalmazás mind önállóan mind pedig egy nagyobb vállalati rendszer moduljaként is kiválóan alkalmazható.

Továbbfejlesztési lehetőség többek között az alkalmazás által generált dokumentumok titkosítása vagy akár digitális aláírással való ellátása, valamint SMS értesítés a beérkező dokumentumokról vagy a közeledő határidőkről.

6. KÖSZÖNETNYILVÁNÍTÁS

Ez úton szeretném megköszönni Dr. Rutkovszky Edénének, témavezetőmnek, hogy lehetőséget biztosított a dolgozatom megvalósításához. Dr. Pázmányi Sándornak, külső konzulensemnek, hogy témát biztosított és szakmai segítséget nyújtott a dolgozatom megírásához, valamint gyakorlati tapasztalataim elmélyítéséhez.

7. IRODALOMJEGYZÉK

Bevezetés:

- 1126/2003. (XII. 12.) Korm. határozat a Magyar Információs Társadalom Stratégiáról és annak végrehajtásáról, http://misc.meh.hu/binary/6600_jogi_hatter_1126_2003_kormhat.pdf
- 257/2007. (X. 4.) Korm. rendelet a közbeszerzési eljárásokban elektronikusan gyakorolható eljárási cselekmények szabályairól, valamint az elektronikus árlejtés alkalmazásáról, http://net.jogtar.hu/jr/gen/hjegy_doc.cgi?docid=A0700257.KOR
- http://hu.wikipedia.org/wiki/Közbeszerzés
- *2003. évi CXXIX. törvény a közbeszerzésekről, http://www.febesz.hu/egyeb/2003_CXXIX.pdf*

Anyag és módszer

- http://en.wikipedia.org/wiki/ASP.NET
- http://hu.wikipedia.org/wiki/.NET_keretrendszer
- http://hu.wikipedia.org/wiki/Active_Server_Pages
- http://hu.wikipedia.org/wiki/Internet_Information_Services
- http://www.hwsw.hu/hirek/34709/Microsoft_Visual_Studio_2008_Net_Framework_3.5_MSDN_fejleszto_szoftver.html
- http://www.microsoft.com/sqlserver/2008/en/us/express.aspx

Eredmények:

- Ian Sommerville - Szoftverrendszerek fejlesztése (2007)

Árbajegyzék:

- 1. ábra - Common Language Infrastructure az http://upload.wikimedia.org/wikipedia/commons/8/85/Overview_of_the_Common_Language_Infrastructure.svg alapján
- 2. ábra – Adminisztrátor felhasználó jogosultságai
- 3. ábra – Felelős felhasználó jogosultságai
- 4. ábra – Lekérdező felhasználó jogosultságai

- 5. ábra – A munkafolyamat az adminisztrátor szemszögéből
- 6. ábra – Az elkészült adatbázis egy részlete
- 7. ábra – A fejlesztési folyamat lépéseinek időigénye százalékban
- 8. ábra – A bejelntkező képernyő
- 9. ábra – A közbeszerzési terv karbantartása
- 10. ábra - Eljárások listázása
- 11. ábra – Új eljárás felvétele
- 12. ábra - Felhasználók kezelése
- 13. ábra – Jelszó felülírása
- 14. ábra - Felhasználói jogosultságok kezelése
- 15. ábra - Az ajánlatkérő alapadatai

Táblázatjegyzék:

- 1. táblázat – A közbeszerzési folyamatok jelenlegi és új elgondolása