IEGULDĪJUMS TAVĀ NĀKOTNĒ!

Eiropas Reģionālās attīstības fonds

Prioritāte: 2.1. Zinātne un inovācijas

Pasākums: 2.1.1. Zinātne, pētniecība un attīstība

Aktivitāte: 2.1.1.1. Atbalsts zinātnei un pētniecībai

Projekts: ”Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem”

Projekta sākuma datums: 2014.gada 1.janvāris.

Projekta beigu datums: 2015.gada 30.jūnijs.

Līguma Nr. 2013/0031/2DP/2.1.1.1.0/13/APIA/VIAA/010

ESF finansējuma saņēmējs: SIA, SWH SETS

Sadarbības partneris: Elektronikas un datorzinātņu institūts (EDI)

Projekta aktivitātes Nr. 3.9.4 "Lietotāju tiesību modelēšanas risinājumu izpēte" progresa pārskats

Pārskats Nr. 31 par periodu no 2015.gada 1.janvāra līdz 2015.gada 30.jūnijam.

SATURS

1. Kopsavilkums

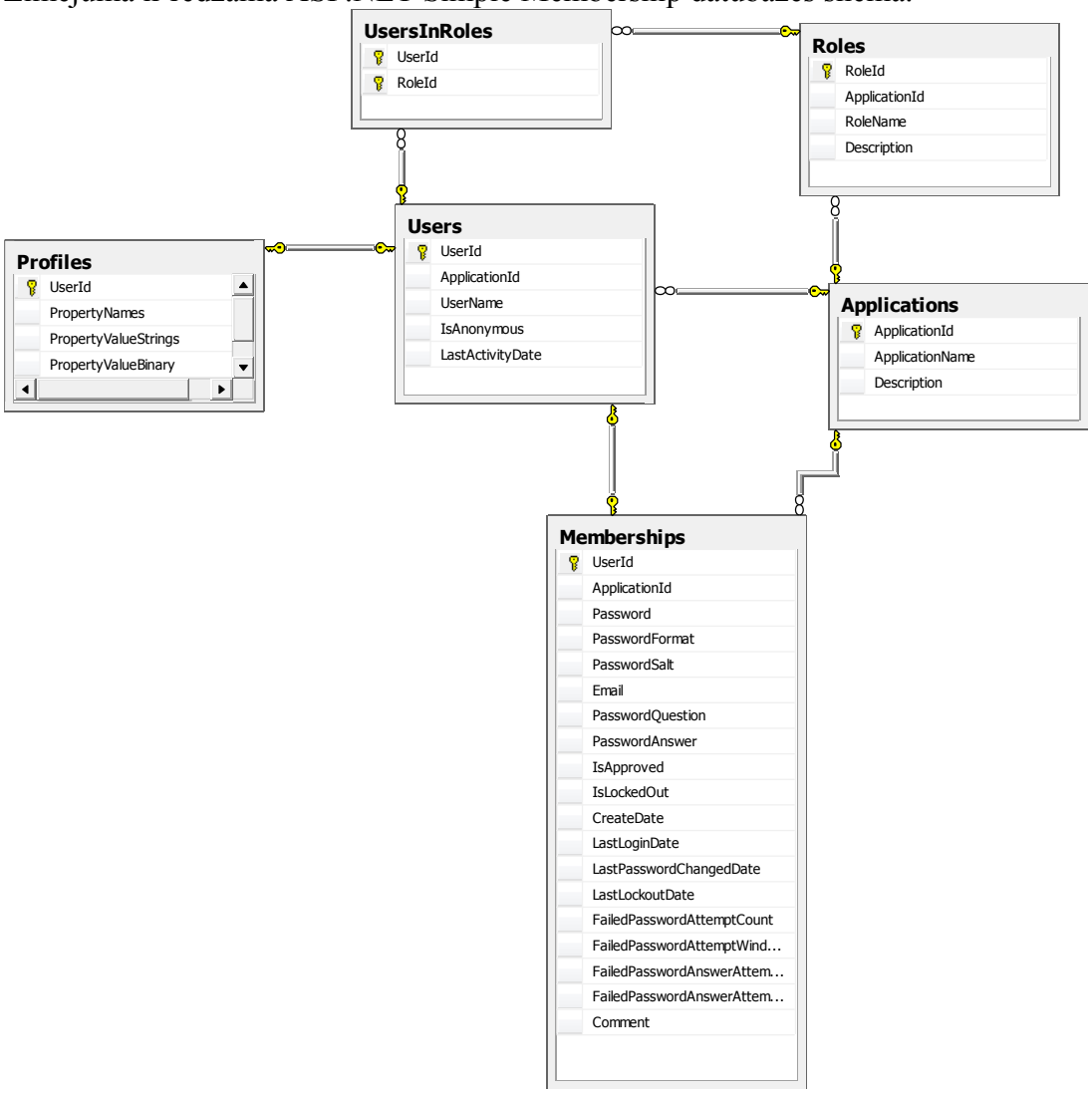
Pārskata periodā (2015-01-01 – 2015-06-30) projekta „Multi - modeļu izstrādes tehnoloģija .NET pielietojumu projektiem” aktivitātes "Lietotāju tiesību modelēšanas risinājumu izpēte" ietvaros veikti šādi darbi:

1. ASP .NET Membership un Simple Membership izpēte.
2. ASP.NET Identity izpēte.
3. ER-modeļa lietotāju datu glabāšanai datubāzē izstrāde.
4. Autorizācijas un tiesību pārbaudes interfeisa analīze un izstrāde.
5. Tiesību modeļa definēšanas iespēju analīze.
6. Aktivitātes pētnieciskā darbība apspriesta ik nedēļas projekta semināros.

2. Lietotāju tiesību atbalsts .NET lietojumprogrammās

2.1. ASP.NET Membership un Simple Membership

ASP.NET Membership parādījās kopā ar ASP.NET 2.0 2005. gadā. Kopš tā laika ir daudz kas izmainījies tipiskajos autorizācijas un autentifikācijas scenārijos, kādus izmanto lietojumprogrammas. Datubāzes struktūra, ar kuru strādā ASP.NET Membership, ir stingri fiksēta un nav maināma. Tā atbalsta uz lomām bāzētas lietotāju tiesības tikai lomu līmenī. Tāpat tā neatbalsta OWIN tehnoloģiju.
ASP.NET Simple Membership ir nākošā Microsoft lietotāju tiesību realizācija. Tā parādījās kopā ar Microsoft VisualStudio 2010 SP1. Salīdzinot ar ASP.NET Membership, ASP.NET Simple Membership ir vieglāk darboties ar lietotāju profiliem, taču tajā ir saglabājušies minētie ASP.NET Membership trūkumi.
Zīmējumā ir redzama ASP.NET Simple Membership datubāzes shēma.

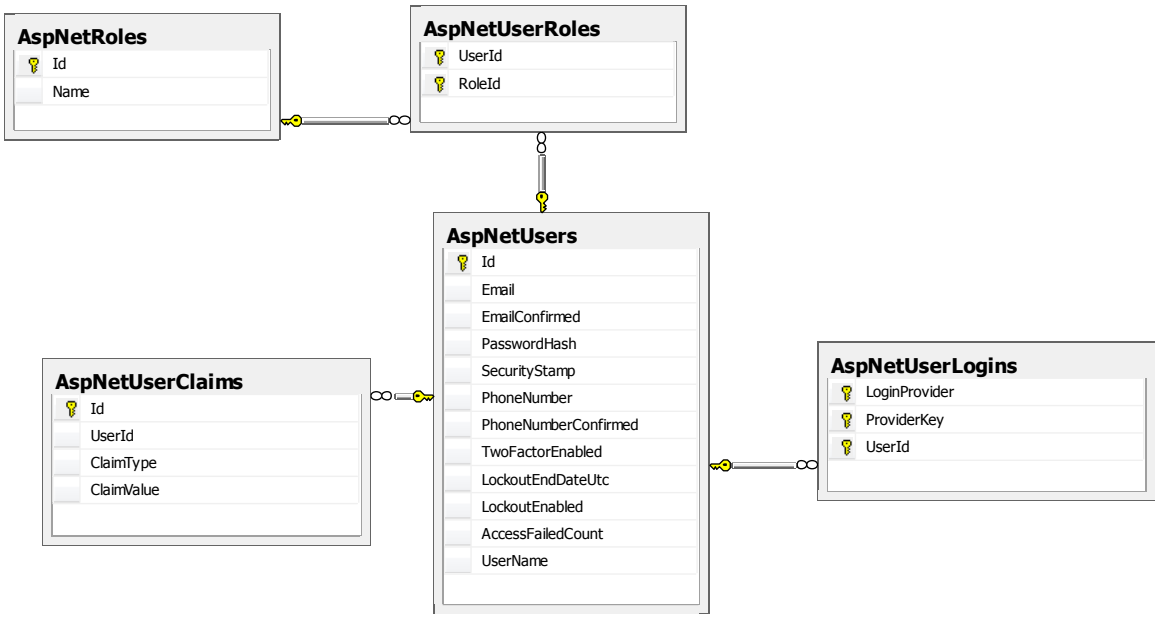


Zīmējums 1

Var teikt, ka ASP.NET Membership un ASP.NET SimpleMembership ir novecojušas un neatbilst mūsdienu prasībām.

2.2. ASP.NET Identity

ASP.NET Identity ir pašlaik jaunākā Microsoft lietotāju un to tiesību sistēma, kura atbalsta visus ASP.NET ietvarus (framework) [1]. ASP.NET Identity ir viegli paplašināma. Pēc noklusēšanas ASP.NET Identity izmanto Entity Framework Code First pieeju, lai darbotos ar lietotājiem un lietotāja tiesību datiem. ASP.NET Identity atbalsta gan RBAC, gan CBAC.
Zīmējumā ir redzama ASP.NET Identity noklusētā datubāzes shēma. [2]

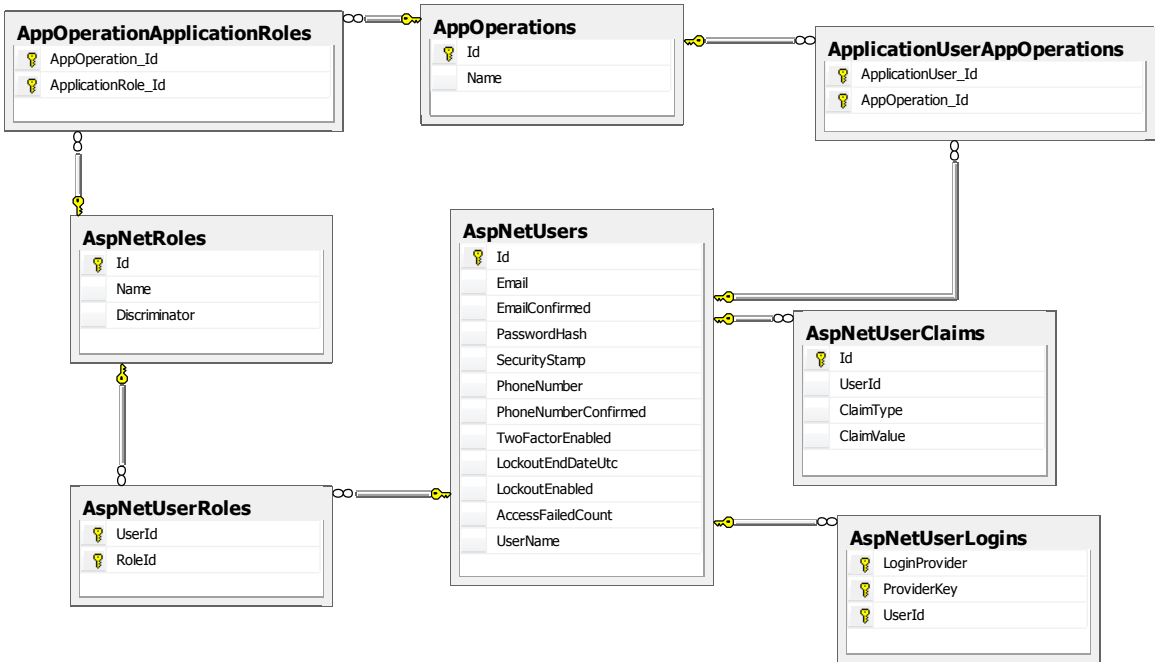


Zīmējums 2

Sīkāku ASP.NET Membership, ASP.NET Simple Membership un ASP.NET salīdzinājumu var atrast rakstā [3].

3. ER-modelis lietotāju datu glabāšanai datubāzē

Lai varētu lietot ASP.NET Identity ar mūsu izvēlēto lietotāju tiesību modeli, papildināsim ASP.NET Identity ER-Modeli, pievienojot AppOperations tabulu, kas atbildīs operācijām un rezolūciju tabulas AppOperationApplicationRoles un ApplicationUserAppRoles, ar kuru palīdzību var pievienot lomai vai lietotājam tiesības izpildīt noteiktu operāciju.



Zīmējums 3

Tabula AspNetUserLogin ir paredzēta ārēju lietotāju (piemēram, Twitter, Facebook) pieslēgšanās nodrošināšanai, AspNetUserClaims – Claims based authentification nodrošināšanai. Šīs iespējas mūsu izvēlētajā lietotāju tiesību risinājumā pagaidām netiek izmantotas, tāpēc tālāk nebūs aprakstītas. Tālāk sīkāk apskatīsim tabulas, kuras tiek izmantotas lietotāju un to tiesību datu glabāšanai

3.1. AspNetUsers

AspNetUsers apraksta lietotāju pamatinformāciju.

Lauka vārds	Datu tips	Apraksts
Id	String(128)	Unikāls identifikators (GUID)
Name	String(256)	Lomas vārds
ForceToChangePassword	bool	Pazīme, ka lietotājam pieslēdzoties ir jāmaina parole
Email	String(256)	Lietotāja e-pasta adrese
EmailConfirmed	bool	E-pasta adrese ir apstiprināta
PasswordHash	String(max)	Paroles hash funkcijas vērtība

SecurityStamp	String(max)	
PhoneNumber	String(max)	Telefona numurs
PhoneNumberConfirmed	bool	Telefona numurs ir apstiprināts
TwoFactorEnabled	bool	Pieejama 2FA (Two-Factor Authentication)
LockoutEndDateUtc	Datetime	Lietotāja bloķēšanas pēc neveiksmīgiem pieslēgšanās mēģinājumiem beigu laiks
LockoutEnabled	bool	Lietotājs bloķēts pēc neveiksmīgiem pieslēgšanās mēģinājumiem
AccessFailedCount	int	Neveiksmīgo pieslēgšanās mēģinājumu skaits
UserName	String(256)	Lietotāja vārds

3.2. AspNetRoles

AspNetRoles apraksta pieejamās lietotāja lomas.

Lauka vārds	**Datu tips**	**Apraksts**
Id	String(128)	Unikāls identifikators (GUID)
Name	String(256)	Lomas vārds

3.3. AspNetOperations

AspNetRoles apraksta pieejamās lietotāja lomas.

Lauka vārds	**Datu tips**	**Apraksts**
Id	int	Unikāls identifikators
Name	String(256)	Operācijas vārds

3.4. AspNetUserRoles

ApplicationRoleAppOperations – piesaista lomas lietotājiem.

Lauka vārds	**Datu tips**	**Apraksts**
UserId	String(128)	Ārējā atslēga – lietotāja Id
RoleId	String(128)	Ārējā atslēga – lomas Id

3.5. ApplicationRoleAppOperations

ApplicationRoleAppOperations – piesaista operācijas lomām.

Lauka vārds	**Datu tips**	**Apraksts**
ApplicationRole_Id	String(128)	Ārējā atslēga – lomas Id
AppOperation_Id	int	Ārējā atslēga – operācijas Id

3.6. ApplicationUsertAppOperations

ApplicationAppOperations piesaista operācijas lietotājiem.

Lauka vārds	Datu tips	Apraksts
ApplicationUser_Id	String(128)	Ārējā atslēga – lietotāja Id
AppOperation_Id	int	Ārējā atslēga – operācijas Id

4. Autorizācijas un tiesību pārbaudes interfeiss

Lai būtu iespēja izmantot dažādas autorizācijas un lietotāju tiesību pārbaudes realizācijas, tika definēts sekojošs interfeiss:

4.1. Login

```
string Login(string username, string password, bool ispersistent);
```

Funkcija identificē lietotāju. Ja lietotājs identificēts veiksmīgi, atgriež null, pretējā gadījumā kļūdu paziņojumu.
Parametri:
username – lietotāja vārds
password - lietotāja parole

4.2. Login

```
string Logout();
```

Funkcija pārtrauc lietotāja sesiju.

4.3. IsLogged

```
string IsLogged();
```

Funkcija pārbauda, vai lietotājs ir autorizējies. Ja lietotājs ir autorizējies, atgriež null, pretējā gadījumā kļūdu paziņojumu.

4.4. ChangePassword

```
string ChangePassword(string oldPassword, string newPassword);
```

Funkcija nomaina lietotāja paroli. Ja parole nomainīta veiksmīgi, atgriež null, pretējā gadījumā kļūdu paziņojumu.
Parametri:
oldPassword – vecā parole
newPassword - jaunā parole

4.5. ResetPassword

```
string ResetPassword(string user, string password);
```

Funkcija uzstāda lietotājam jaunu paroli. Ja parole uzstādīta veiksmīgi, atgriež null, pretējā gadījumā kļūdu paziņojumu.
Parametri:
username – lietotāja vārds
password - lietotāja parole

4.6. getCurrentUserName

```
string getCurrentUserName();
```

Funkcija atgriež identificētā lietotāja vārdu.

4.7. getCurrentUserId

```
string getCurrentUserId();
```

Funkcija atgriež identificētā lietotāja Id.

4.8. IsInRole

```
bool IsInRole(string roleName);
```

Funkcija pārbauda, vai lietotājs pieder lomai.
Parametri:
roleName – lomas vārds

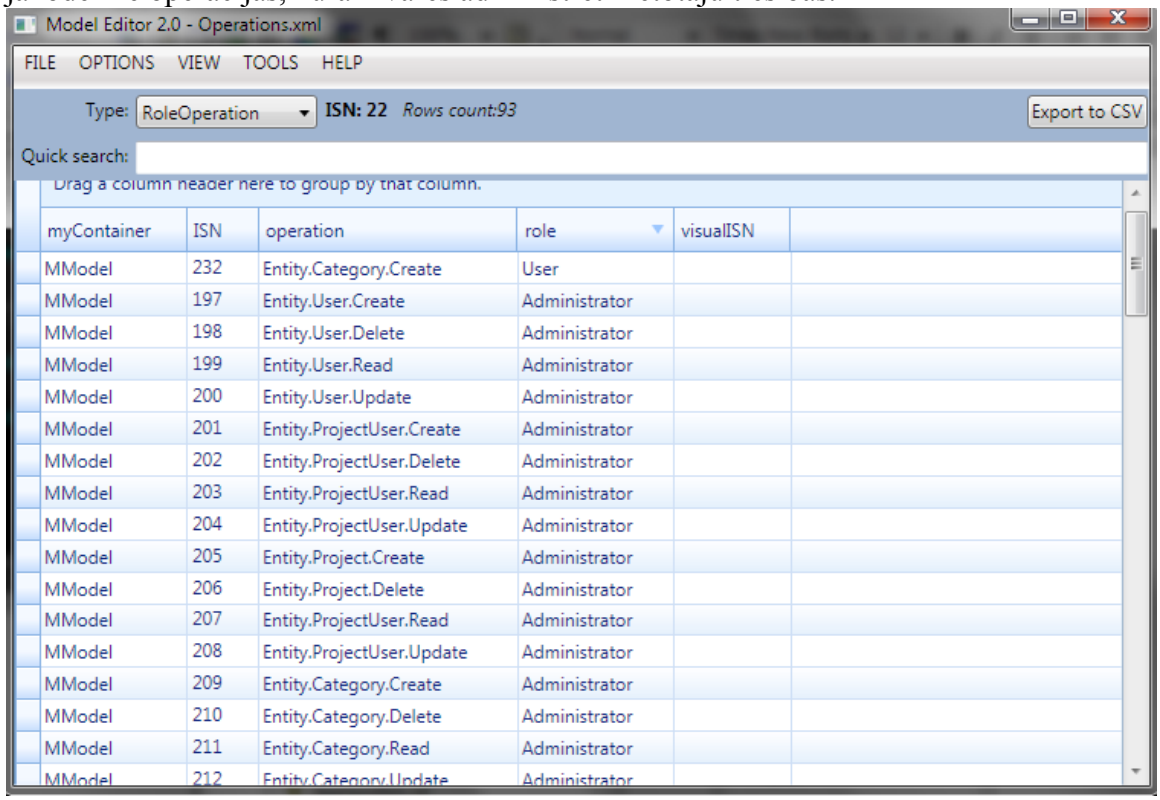
4.9. CanExecute

```
bool CanExecute(string objectType,string name,string operation);
```

Funkcija pārbauda, vai lietotājs var izpildīt operāciju.
Parametri:
objectType – objekta tips
name - objekta vārds
operation - operācija

5. Tiesību modeļa definēšana

Tālāk apskatīsim, kādā veidā var definēt lietotāju tiesību modeli.
Lai definētu lietojumprogrammas tiesību modeli [4], ar modeļa redaktora [5] palīdzību ir jānodefinē operācijas, kurām varēs administrēt lietotāju tiesības.



Zīmējums 4

Parasti visām entītijām tiek definētas 4 operācijas: Izveidot (Create), Lasīt (Read), Labot (Update) un Dzēst (Delete).
Dažādās lietojumprogrammās var būt vēlme gala lietotājam(lietojumprogrammas administratoram) administrēt lietotāja tiesības lomu līmenī, vai administrēt lietotāja tiesības operāciju līmenī. Tiesību modelis atļauj noteikt lietotāju tiesības abos augstāk minētajos veidos., atšķirsies tikai no modeļa ģenerētais kods. Apskatīsim abus šos variantus

5.1. Lietotāju tiesību administrēšana lomu līmenī

Lietotāju tiesību administrēšana lomu līmenī nozīmē, ka gala lietotājs neredz operācijas, kuru tiesības var administrēt un saite starp operācijām un lomām tiek definēta programmas kodā.

5.2. Lietotāju tiesību administrēšana operāciju līmenī

Lietotāju tiesību administrēšana operāciju līmenī nozīmē, ka gala lietotājs var noteikt, kādas operācijas tiek atļautas katrai lomai. Parasti šajā pieejā gala lietotājs var definēt arī jaunas lomas. Dotā pieeja ir elastīgāka, bet tajā pašā laikā prasa lielākas zināšanas un lielāku administrēšanas darbu lietojumprogrammas gala lietotājam.

6. Rezultāti

Aktivitātes ietvaros izpētīts lietotāju tiesību atbalsts .NET lietojumprogrammās, izveidots ER-modelis lietotāju datu glabāšanai datubāzē, izstrādāts autorizācijas un tiesību pārbaudes interfeiss un apzinātas tiesību modeļa definēšanas iespējas.

7. Literatūras saraksts

[1] ASP.NET Identity https://aspnetidentity.codeplex.com/
[2] ASP.NET Identity Framework http://www.teamscs.com/2014/11/asp-net-identity-framework/
[3] ASP.NET Identity, Membership and SimpleMembership Comparison http://www.beansoftware.com/ASP.NET-Tutorials/identity-membership-simplemembership.aspx
[4] 3.8.2 "Funkciju un objektorientētu lietotāju tiesību modelis" progresa pārskats
[5] 2.2 "Universālais Modeļu Redaktors" progresa pārskats