**Информация о пользователе.**

Разработана страница портала *Информация о пользователе,* которая необходима для ознакомительных целей, для просмотра информации о сотруднике. Для администратора и пользователя, чьи данные отображены на странице, предусмотрена возможность редактирования отдельной информации (добавление/удаление фотографии, контактной информации).

На странице отображается:

* Фотография
* Фамилия
* Имя
* Отчество
* Дата рождения
* Контакты
* Список способностей; технологии, которыми владеет сотрудник
* Список проектов
* Текущие проекты
* В которых участвовал ранее
* Для администратора доступна ссылка *редактирование* (редактирование информации пользователя в административном режиме)



Общая информация.

Фамилия, имя, отчество, пол, дата рождения загружаются из таблицы dbo.Person стандартным методом Load(userID) класса Person.

Контакты.

Контактная информация храниться в таблице dbo.PersonAttributes*.*

PersonID - ID сотрудника

KeyWord - ключевое слово *Contact*

StringField - информация о контакте, храниться в таблице следующим образом

*CotnactName + “* [*\n*](\n) *” + ContactInfo* (например, *“ICQ* [*\n*](file:///\\n) *\*\*\*\*\*\*\*”*)

Загружается из базы с помощью метода GetPersonAttributesByKeyword (int personID, string keyword) класса PersonAttributes, сохраняется в базу данных с помощью метода AddStandardStringAttribute (PersonAttributeTypes type, string StringField) класса Person, удаляется из базы данных с помощью метода Delete() класса BasePlainObject.

Фотографии.

Информация об имена фотографий храниться в таблице dbo.PersonAttributes*.*

PersonID - ID сотрудника

KeyWord - ключевое слово Photo

StringField - уникальное имя файла.

Информация обо всех фотографиях данного пользователя храниться на самой странице

/// индекс отображаемой фото

private int index

/// все фото данного пользователя

private PersonAttribute[] PhotoList;

При работе с базой данных добавление информации о фотографии происходит при помощи AddStandardStringAttribute (PersonAttributeTypes type, string StringField) класса Person, удаляется из базы данных с помощью метода Delete() класса BasePlainObject. Сама фотография закачивается на WebFileService при добавлении фотографии и удаляется с него при стирани информации из базы данных.

Вся работа с фотографиями (загрузка/удаление с WebFileService, занесение/удаление информации о фотографии из базы данных) реализована в классе PhotosAttachmentManager.

public class PhotosAttachmentManager : FileManager

{

/// Saving new file attachments.

/// <param name="fileCollection">Collection of posted files.</param>

/// <param name="person"></param>

public void SavePhotoAttachmentFiles(HttpFileCollection fileCollection, Person person)

/// Delete marked files from attachment collection.

/// <param name="collection">PhotosAttachmentCollection.</param>

public void DeleteAttachPhoto(int userID, string nameOfPhoto)

/// Delete attach file.

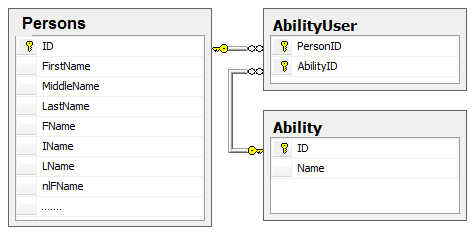
/// <param name="fileName">File to delete.</param>

public void DeleteAttachFile(string fileName)

}

Способности.

Для хранения различных способностей создана таблица dbo.Ability, для связи способностей и сотрудников создана таблица dbo.AbilityUser.



dbo.Ability

ID - ID способности.

Name - название способности, ее описание.

dbo.AbilityUser

PersonID - ID сотрудника, которому приписана способность.

UserID - ID способности, которую приписывают сотруднику.

Для работы со способностями создан класс Ability.

public class Ability

{

private string m\_AbilityName; //название способности

…

}

В классе Person добавлено свойство, возвращающее список способностей данного сотрудника, public IList<Ability> Abilities. Для реализации данной свойства на DAL-уровне реализован класс AbilityProvider.

public abstract class AbilityProvider : DataAccess

{

…

/// Creates new ability in database.

/// <param name="abilityname">Name of ability.</param>

/// <returns>ID of new database record.</returns>

public abstract int CreateAbility(string abilityname);

/// Returns all abilities for current user.

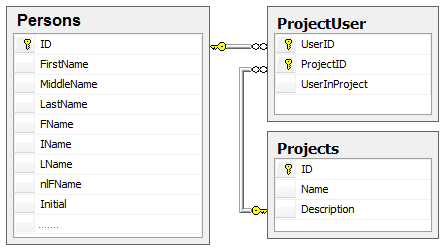
/// <param name="userID">User ID.</param>

public abstract IList<Ability> GetAllUserAbilities(int userID);

}

Проекты.

Для хранения информации о проектах используется таблица dbo.Projects, для связи проектов и сотрудников создана таблица dbo.ProjectUser.



dbo.Projects

ID - ID проекта.

Name - название проекта.

dbo.ProjectUser

UserID - ID сотрудника, который участвует в проекте.

ProjectID - ID проекта, в котором участвует сотрудник.

UserInProject - принимает значения 0 или 1 (0 – участие в разработке проекта завершено, 1 – сотрудник участвует в разработке проекта).

Для работы с проектами создан класс Project.

public class Project

{

private string m\_ ProjectName; //название проекта

…

}

На DAL-уровне реализован класс ProjectProvider.

public abstract class ProjectProvider : DataAccess

{

…

/// Returns all projects for current user.

/// <param name="userID">User ID.</param>

public abstract IList<Project> GetAllUserProjects(int userID);

/// Returns complete/current projects for current user.

/// <param name="userID">User ID.</param>

/// <param name="returnCompleteProjects">True – return current projects, /// false – return complete projects </param>

public abstract IList<Project> GetUserProjects(int userID, bool returnCompleteProjects);

}