МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ

ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ДОНЕЦКИЙ НАЦИОНАЛЬНЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра ПИ

Факультет КНТ

Отчет по лабораторной работе №1

Тема: «Базовые принципы работы с системами контроля версий».

Курс: «Профессиональная практика программной инженерии»

Выполнил

ст. гр. ПИ-18в

Настеко И.Р.

Донецк – 2021

Вариант 2

Форум с ветками-обсуждениями, с обязательной реализацией групповых ролей: гость, посетитель, модератор, администратор.

Краткое описание компонентов модулируемого проекта

**Модели**

Роли

У всех ролей кроме гостя есть обязательные поля: Айди ,Имя, Логин, Пароль, Почта, Права

Администратор

public class Admin : IUser

{

public Guid Id { get; }

public string Name { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public string Email { get; set; }

public List<UserRightsEnum> Rights { get; set; }

}

Модератор

public class Moderator : IUser

{

public Guid Id { get; }

public string Name { get; set; }

public string Login { get; set; }

public string Password { get; set; }

public string Email { get; set; }

public List<UserRightsEnum> Rights { get; set; }

}

Посетитель

public class Visitor : IUser

{

public Guid Id { get; }

public string Name { get; set; }

public string Login { get;set; }

public string Password { get; set; }

public string Email { get; set; }

public List<UserRightsEnum> Rights { get; set; }

}

Гость

public class Guest

{

public string Name { get; set; }

public List<UserRightsEnum> Rights { get; set; }

}

Перечисление для прав пользователей

public enum UserRightsEnum

{

CreateBranch,

DeleteBranch,

BanUser,

UnBanUser,

AddMessage,

DeleteMessage,

DeleteUserMessage,

ReadMessages

}

CreateBranch, – создание ветки

DeleteBranch, – удаления ветки

BanUser, – забанить пользователя

UnBanUser, – разбанить пользователя

AddMessage, – отправить сообщение

DeleteMessage, – удалить сообщение

DeleteUserMessage, – удалить сообщение определённого юзера

ReadMessages – чтение сообщений

Также созданы модели Сообщение, Ветка обсуждений

public class Message

{

public Guid Id { get; set; }

public string MessageText { get; set; }

public Guid UserId { get; set; }

public DateTime CreatedDate { get; set; }

public Guid BranchId { get; set; }

}

public class Branch

{

public Guid Id { get; set; }

public string Name { get; set; }

public int MessageCount { get; set; }

}

**Сервисы**

Для взаимодействия с моделями созданы интерфейсы сервисов

Функции для администратора

internal interface IAdminService

{

bool BanUser(Guid userId);

bool AppointModerator(Guid userId);

void SetRights(Guid userId, UserRightsEnum[] rights);

}

BanUser ­– забанить пользователя

AppointModerator – назначить пользователю права модератора

Функции веток обсуждения

public interface IBranchService

{

void CreateBranch(string branchName);

void DeleteBranch(Guid branchId);

}

CreateBranch/DeleteBranch – создать/удалить ветку

Функции иденификации пользователей

public interface IIdentityService

{

bool ResgisterNewUser(IUser user);

bool CheckIsUserExist(Guid userID);

bool AuthorizeUser(string userName, string password);

}

RegisterNewUser – зарегистрировать нового пользователя

CheckIsUserExist – проверить существование пользователя

AuthorizeUser – авторизовать пользователя

Функции сообщений

public interface IMessageService

{

void SendMessage(Guid branchId, Message message);

void DeleteMessage(Guid messageId);

}

SendMessage – отправить сообщение в ветку

DeleteMessage – удалить сообщение

Шаги разрабоки

1. Проектирование архитектуры приложения

2. Проектирование базы данных и создание моделей для всех сущностей

4. Создание таблиц и полей для всех сущностей в базе данных.

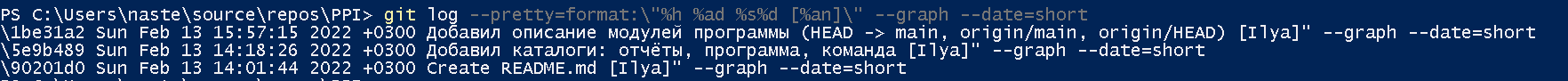
3. Реализовать функции интерфейса сервиса идентификации IIdentityService

4. Разработку сервисов IBranchService IMessageService IAdminService можно вести параллельно.

5. Для всех функций написать Unit тесты.

6. Спроектировать и разработать фронтэнд часть проекта.

Вывод команды git log --pretty=format:\"%h %ad | %s%d [%an]\" --graph --date=short.



Вывод команды git diff для одной из ревизий.

