1. Узнать название приложения

2. Описание предметной области

3. Сессия 1.пдф

4. Чесать таблицы

5. Кинуть таблицы в БД

6<ER>. ER диаграмма

7<DD>. Словарь данных

8<use\_case>. Поменять use\_case

9. <project> запустить приложение

10. Последовательно уничтожать окна

11. Руководство пользователя в конце бы

≠===================

Приложение:

Авторизация

db.Moders

db.Jury

db.Members

db.Sponsors

Окно гостя:

db.Events

db.EventDirection

+pics

Окно организатора

db.Sponsors

Окно просмотра инфы:

db.Jury

db.Members

db.Moders

Окно регистрации жюри/модератора:

db.Jury

db.Events

db.Moders

db.EventDirection

db.EventType

========================

ТОЧКА ЗРЕНИЯ БД ТАБЛИЦ И ПРОГРАММЫ(В скобках связи для диаграммы):

Активности(+Жюри, +модератор):

Ставим id

Спускаем инфу вниз

Удаляем дубликаты

Переименовать столбцы

СОЗДАТЬ В БД И НЕ ЮЗАТЬ:

EventDirection направ

Id

Event direction

EventType мероприят

Id

eventType

Мероприятия(+города, +EventDirection):

Добавить id

Переименовать столбцы

Кинуть EventDirection

Организаторы(+страны)

Ид

Столбцы

Модераторы(+страны +EventDirection +EventType)

Ид

Столбцы

Участники(+страны)

Ид

Столбцы

Жюри(+страна, +EventType)

Ид

Столбцы

Страны

Столбцы

Города

Столбцы

Гербы нахуй

===================================

ЧАСТЬ 1. (Диаграммы, базы данных)

1. Смотрим предметную область и строим диаграмму прецедентов

1.1. Причёсываем таблицы

1.1.2. Строим ER-диаграмму

1.2 Словарь данных на основе таблиц и диаграммы

1.3. Кидаем через import data все таблицы(Нахуй связи нахуй типы данных)

P.S. к 1.3:

-ОБЯЗАТЕЛЬНО ДОЛЖНЫ БЫТЬ PK, ИНАЧЕ ПИЗДА ENTITYFRAMEWORK

-Никаких ключевых слов в названиях. Таблицы желательно называть с больших букв, чтобы классы моделей тоже были по неймингу с больших букв

ЧАСТЬ 2. (Приложение)

2. Создать проект

2.1. Подключиться к БД через EF или ADO . NET:

-Проект > Добавить элемент > Данные > Модель ADONET EDM > EF Designer from Database и т.д

2.2. Создать необходимые формы и подогнать к ним данные из базы данных

2.3. КАПЧА: /

-создаём в пейнте смешную картинку, запихиваем на форму LABEL, в его фон загружаем смешную картинку. ВЫРУБАЕМ AUTOSIZE. TEXT ALIGN - центр. Шрифт: Curlz/Chiller/ComicSans

-Рандомим текст через random, попутно фильтруя символы ASCII. В нашей капче нам нужны:

48-57 - цифры

65-90 - заглавные лат.буквы

97-122 - строчные лат.буквы

Логика проста: если рандомом генерится что-то из 48-122 и оно входит в эти промежутки, то мы добавляем её в конечную строку

2.4. ЗАГРУЗКА ФОТОГРАФИЙ:

-Перевести все форматы картинок в один, иначе придётся костылиться не один час. Тут .jpg.

-Переносим все картинки в ПРОЕКТНЕЙМ/bin/debug/ПАПКАНЕЙМ

-Создаём функцию загрузки фото, которая принимает параметр string name

-Нужно дать абсолютный путь, иначе никак. Пишем код по образцу:

[Image image = Image.FromFile(СОВМЕСТИТЬ ПУТИ(ПУТЬ ДО DEBUG, ПУТЬ ДО ПАПКИ));]

Image image = Image.FromFile(Path.Combine(Application.StartupPath, $"ПАПКАНЕЙМ/{name}.jpg"));

pictureBox1.Image = image

2.5 ВЫВОД ДАТЫ

-Можно обратиться к конкретному времени ПК через DateTimeOffset . Now

Чтобы обратиться к часам, можно обратиться к dateTimeOffset.Hour

2.6 ЗАГРУЗКА ПИКЧИ В ПРОГРАММУ:

-Создаём объект OpenFileDialog

-Фильтруем чисто по расширениям фото

-Сохраняем путь

-Пихаем его в picBox1.ImageLocation

OpenFileDialog ofd = new OpenFileDialog();

ofd.Filter = "Изображения |\*.jpg;\*.jpeg;\*.png";

if (ofd.ShowDialog() == DialogResult.OK)

{

string selectedFilePath = ofd.FileName;

pictureBox1.ImageLocation = selectedFilePath;

}