Министерство образования, науки и молодежной политики Нижегородской области

Государственное бюджетное профессиональное образовательное учреждение "Нижегородский Губернский колледж"

ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ

**ПМ.01.**

Специальность **09.02.07 Информационные системы и программирование**

|  |  |
| --- | --- |
| Выполнил |  |
| Студент | И.Е. Громов |
| подпись |  |
| Группа 41П курс 4 |  |
| Руководитель | Е.П.Голубева |
| подпись |  |
|  |  |
|  |  |

Нижний Новгород 2025

Оглавление

[День 1 – Подготовка и создание БД 3](#_Toc192261921)

[День 2 - Data Dictionary, диаграмма прецедентов 5](#_Toc192261922)

[День 3 - Настольное приложение 9](#_Toc192261923)

[День 5 - DLL, Тестирование 15](#_Toc192261924)

[Тестовый пример #1: 18](#_Toc192261925)

[Тестовый пример #2: 18](#_Toc192261926)

[Тестовый пример #3: 19](#_Toc192261927)

[Тестовый пример #4: 20](#_Toc192261928)

[Test case #5: 21](#_Toc192261929)

[День 6 – проверка чужой работы и написание итогового отчета 22](#_Toc192261930)

[Приложение А (Таблицы для импорта) 23](#_Toc192261931)

[Приложение Б (Руководство) 28](#_Toc192261932)

[Приложение В (критерии) 34](#_Toc192261933)

[Листинг 40](#_Toc192261934)

# День 1 – Подготовка и создание БД

Первым делом были подготовлены Excel-таблицы для корректного импорта данных (см. [приложение](#_Приложение_А:) А):

1)таблица «Города»

2) таблица «Жюри»

3) таблица «Событие»

4) таблица «Направление»

5) таблица «Модераторы»

6) таблица «Организаторы»

7) таблица «Страны»

8) таблица «Участники»

9) таблица «Пол»

10) таблица «Мероприятия»

11) таблица «Активности»

12) таблица для связи жюри и активностей

13) таблица «Победитель»

После подготовки таблиц, создал базу данных (скрипт приложен в [листинге](#_Листинг)) и заполняем соответствующими данными. В конечном итоге получаем следующую er-диаграмму:

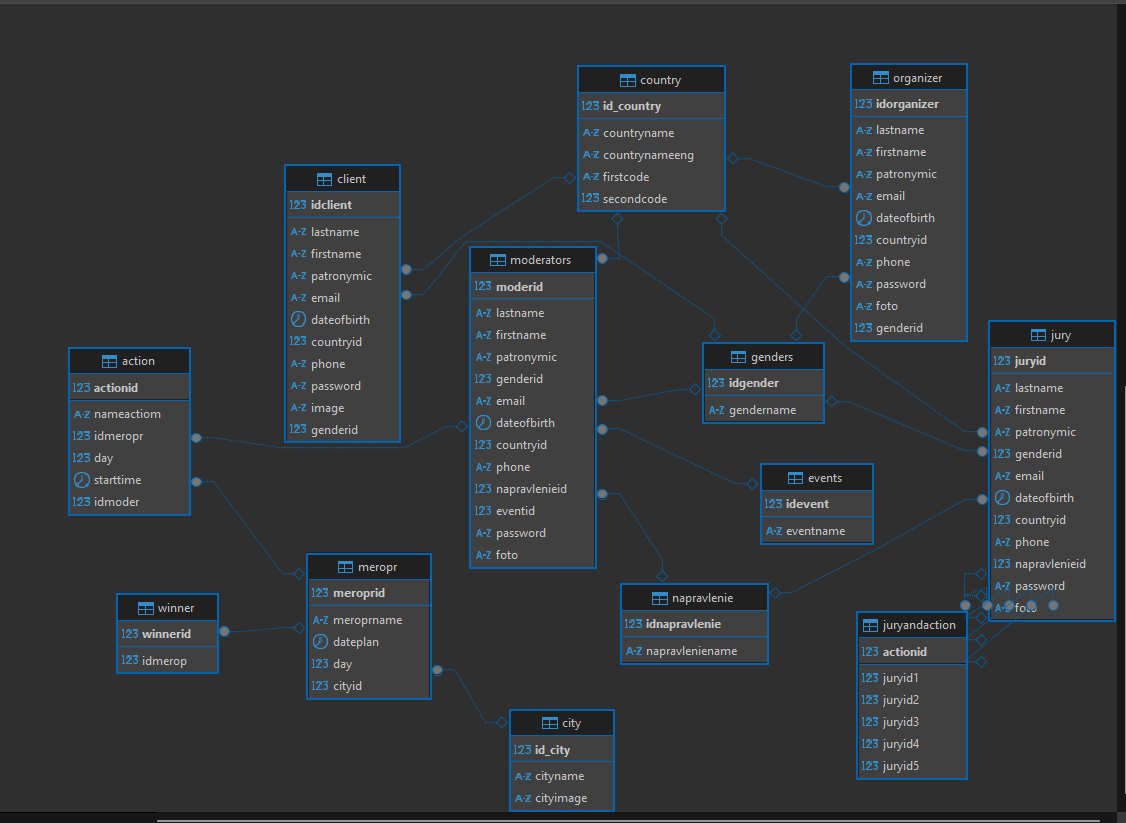


Рис 1 – Er-диаграмма

# День 2 - Data Dictionary, диаграмма прецедентов

Изначально была реализована таблица Data Dictionary, которая представляет собой словарь данных

Таблица - Genders

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Genders** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **idGender** | serial | Y | Auto increment |
|  | **genderName** | VARCHAR (10) | Y |  |
|  |  |  |  |  |
|  |  |  |  | Таблица - Client |
|  |  |  |  |  |
| **Client** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **idClient** | serial | Y | Auto increment |
|  | **lastName** | VARCHAR (100) | Y |  |
|  | **firstName** | VARCHAR (50) | Y |  |
|  | **patronymic** | varchar(10) | Y |  |
|  | **Email** | text | Y |  |
|  | **dateOfBirth** | timestamp | Y |  |
|  | **phone** | varchar(20) | Y |  |
|  | **password** | text | Y |  |
|  | **image** | text | Y |  |
| FK | **genderId** | int | Y | вторичный ключ к таблице "Gender" |
|  |  |  |  | Таблица - Organizer |
| **Organizer** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | idOrganizer | serial | Y | Auto increment |
|  | lastName | varchar(100) | Y |  |
|  | firstName | varchar(50) | Y |  |
|  | patronymic | varchar(50) | Y |  |
|  | email | text | Y |  |
|  | dateOfBirth | timestamp | Y |  |
| FK | countryId | int | Y | вторичный ключ к таблице "Country" |
|  | phone | varchar(20) | Y |  |
|  | **password** | text, | Y |  |
|  | foto | text | Y |  |
| FK | genderId | int | Y | вторичный ключ к таблице "Gender" |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Napravlenie** |
| **Napravlenie** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | idNapravlenie | serial | Y | Auto increment |
|  | NapravlenieName | text | Y |  |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Events** |
| **Events** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | idEvent | serial | Y | Auto increment |
|  | eventName | text | Y |  |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Jury** |
| **Jury** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | juryId | serial | Y | Auto increment |
|  | lastName | varchar(100) | Y |  |
|  | firstName | varchar(50) | Y |  |
|  | patronymic | varchar(50) | Y |  |
| FK | genderId | int | Y | вторичный ключ к таблице "Gender" |
|  | email | text | Y |  |
|  | dateOfBirth | timestamp | Y |  |
| FK | countryId | int | Y | вторичный ключ к таблице "Country" |
|  | phone | varchar(20) | Y |  |
| FK | NapravlenieId | int | Y | вторичный ключ к таблице "Napravlenie" |
|  | **password** | text | Y |  |
|  | foto | text | Y |  |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Moderators** |
| **Moderators** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **moderId** | serial | Y | Auto increment |
|  | **lastName** | varchar(100) | Y |  |
|  | **firstName** | varchar(50) | Y |  |
|  | **patronymic** | varchar(50) | Y |  |
| FK | **genderId** | int | Y | вторичный ключ к таблице "Gender" |
|  | **email** | text | Y |  |
|  | **dateOfBirth** | timestamp | Y |  |
| FK | **countryId** | int | Y | вторичный ключ к таблице "Country" |
|  | **phone** | varchar(20) | Y |  |
| FK | **NapravlenieId** | int | Y | вторичный ключ к таблице "Napravlenie" |
| FK | **eventId** | int | Y | вторичный ключ к таблице "Events" |
|  | **password** | text | Y |  |
|  | **foto** | text |  |  |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Jury** |
| **Meropr** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **MeroprId** | serial | Y | Auto increment |
|  | **MeroprName** | varchar(100) | Y |  |
|  | **datePlan** | varchar(50) | Y |  |
|  | **day** | varchar(50) | Y |  |
| FK | **cityId** | int | Y | вторичный ключ к таблице "City" |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Action** |
| **Action** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | MeroprId | serial | Y | Auto increment |
|  | MeroprName | varchar(100) | Y |  |
|  | datePlan | varchar(50) | Y |  |
|  | **day** | varchar(50) | Y |  |
| FK | cityId | int | Y | вторичный ключ к таблице "City" |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  | Таблица - JuryAnd**Action** |
| **JuryAndAction** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | ActionId | serial | Y | Auto increment |
| FK | JuryId1 | int | Y | вторичный ключ к таблице "Jury" |
| FK | JuryId2 | int | Y | вторичный ключ к таблице "Jury" |
| FK | JuryId3 | int | Y | вторичный ключ к таблице "Jury" |
| FK | JuryId4 | int | Y | вторичный ключ к таблице "Jury" |
| FK | JuryId5 | int | Y | вторичный ключ к таблице "Jury" |
|  |  |  |  |  |
|  |  |  |  | Таблица - **Winner** |
| **Winner** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | WinnerId | serial | Y | Auto increment |
| FK | IdMerop | int | Y | вторичный ключ к таблице "Meropr" |

Далее была реализована диаграмма прецедентов (диаграмма вариантов использования)

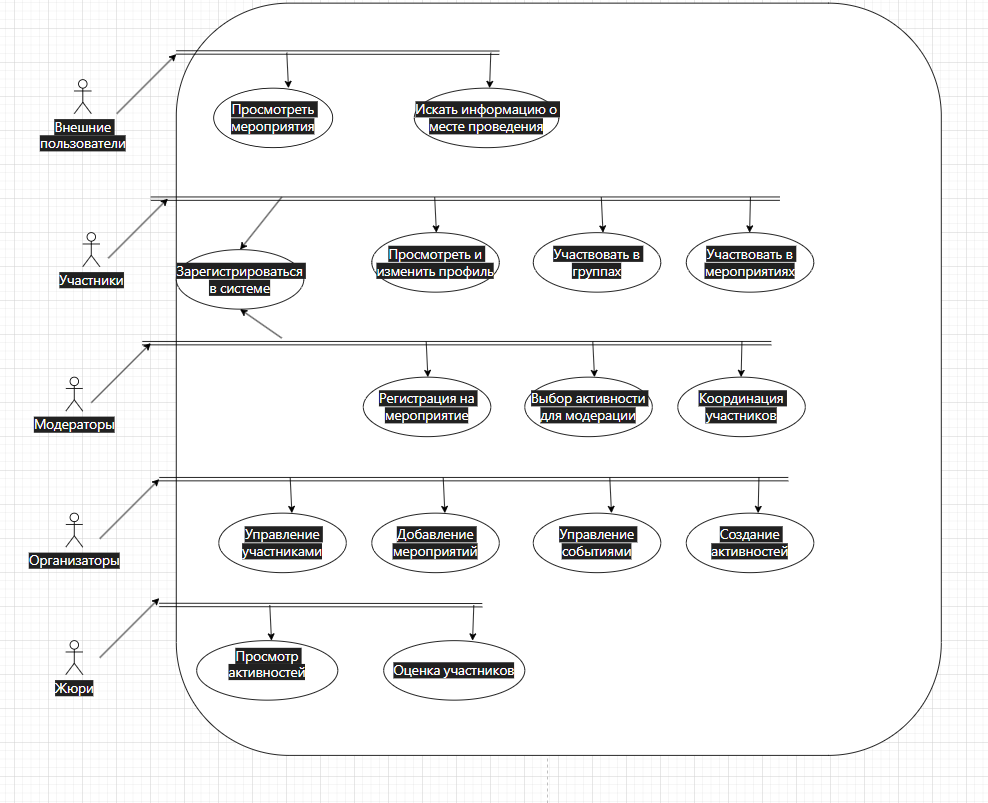


Рис 2 – Диаграмма прецедентов

# День 3 - Настольное приложение

Поскольку в течение учебного года у меня возникали трудности с Avalonia, большую часть времени я потратил на заполнение пропусков знаний.

В целом у меня получилось подключить БД, настроить навигацию, грубо разработать макеты окон и к некоторым из них прописал функционал:

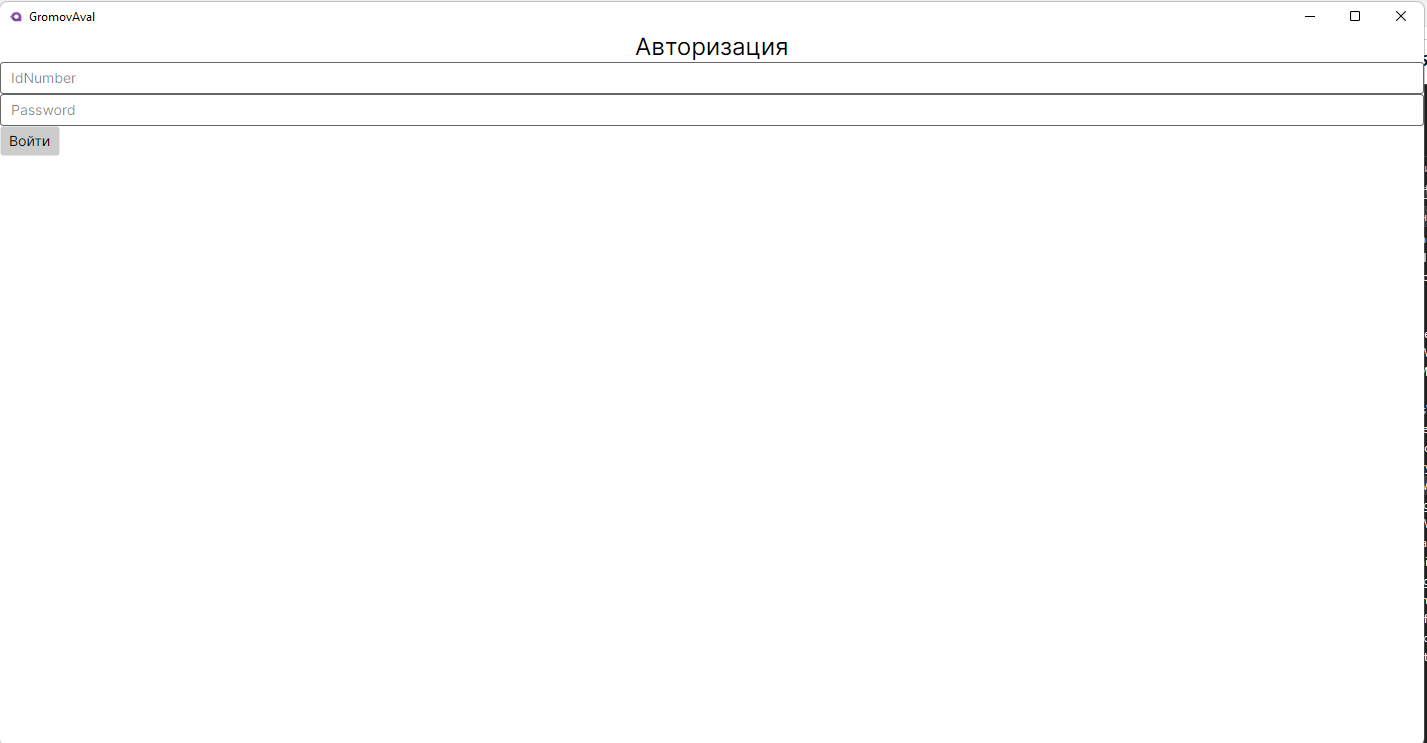


Рис 3 – окно авторизации



Рис 4– окно организатора

Также смог вывести список мероприятий

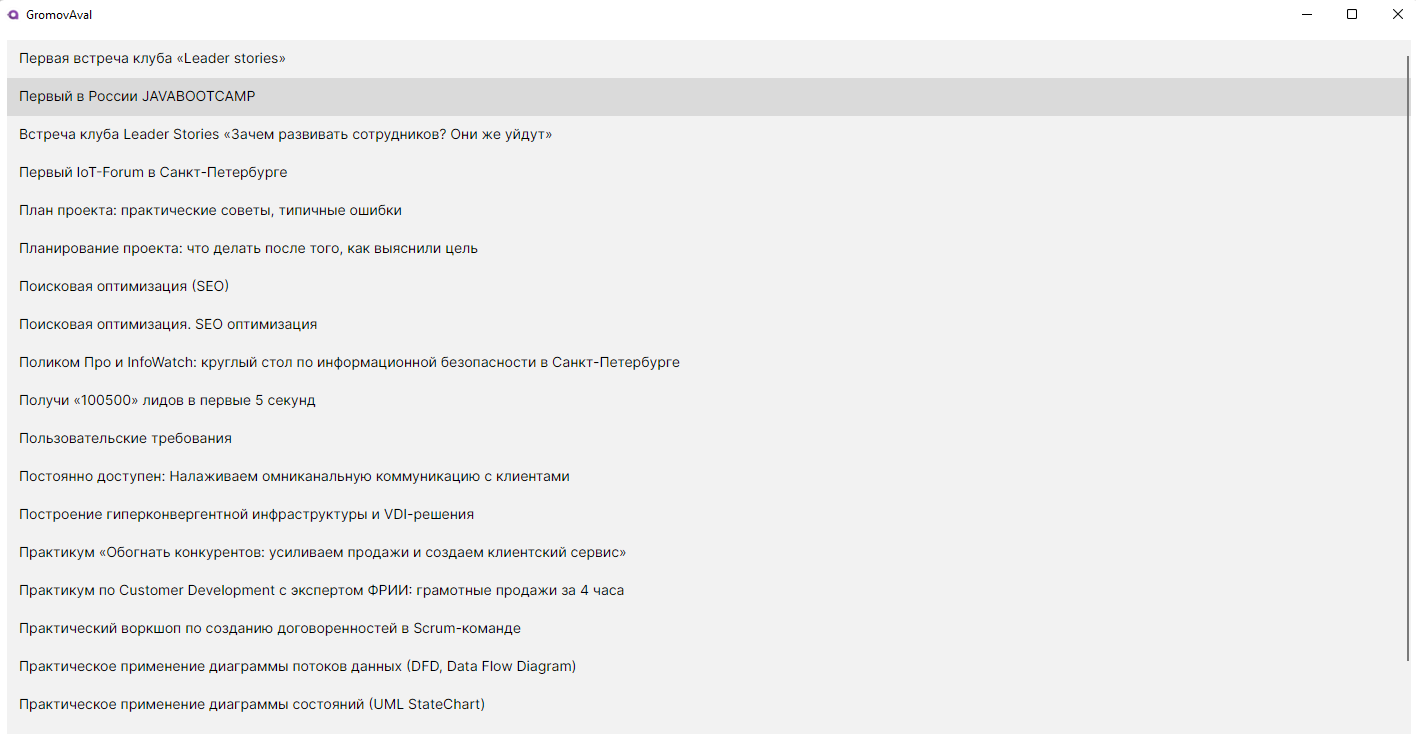


Рис 5– раскрытие окна мероприятий

День 4 – Капча

Решил начать разработку с самого начала, поскольку решил сменить Community ToolKit на Reactive UI.

В конечном итоге были созданы:

начальный экран с выводом данных о мероприятиях и возможностью перехода на экран авторизации

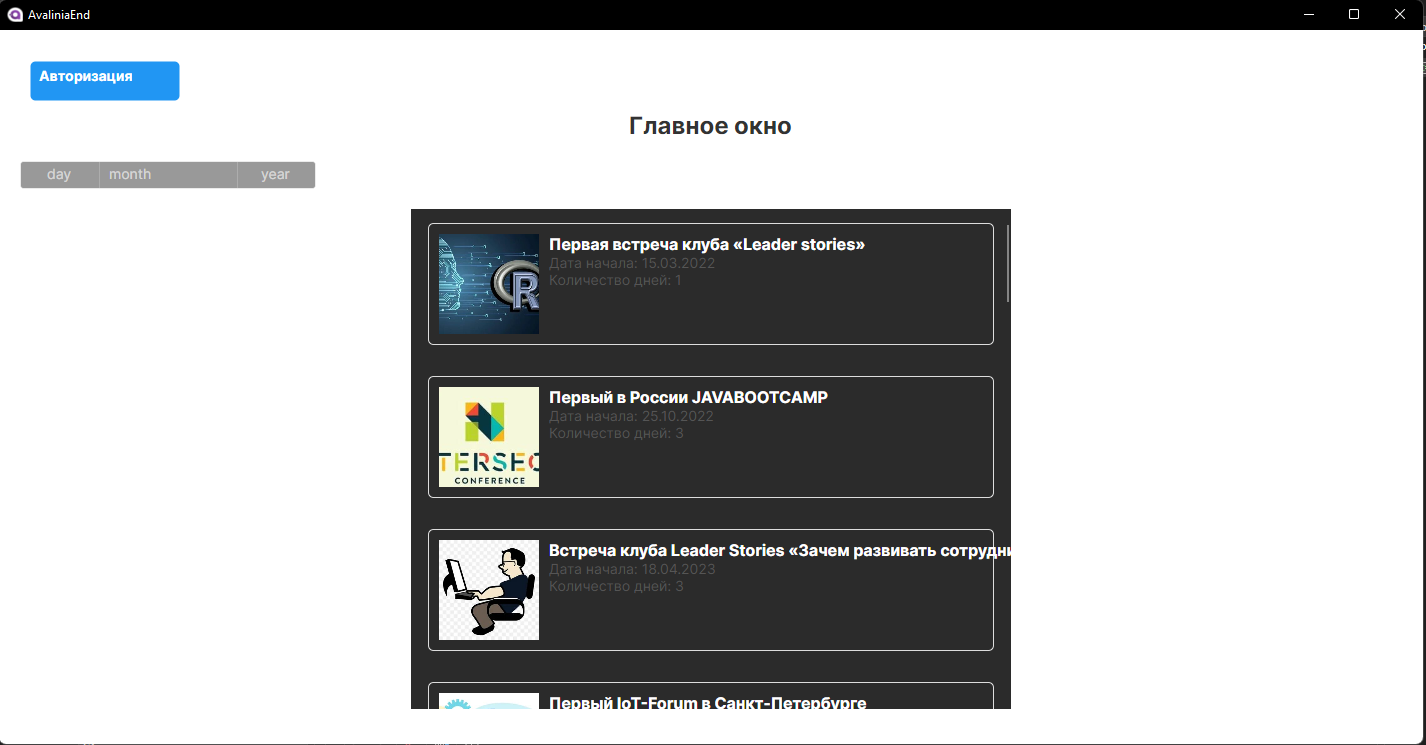


Рис 6 – начальный экран

Далее разработал экран авторизации с реализацией функции авторизации (пользователь вводит свой айди, как это указано в задании, и, соответственно пароль), также создана функционирующая простая капча

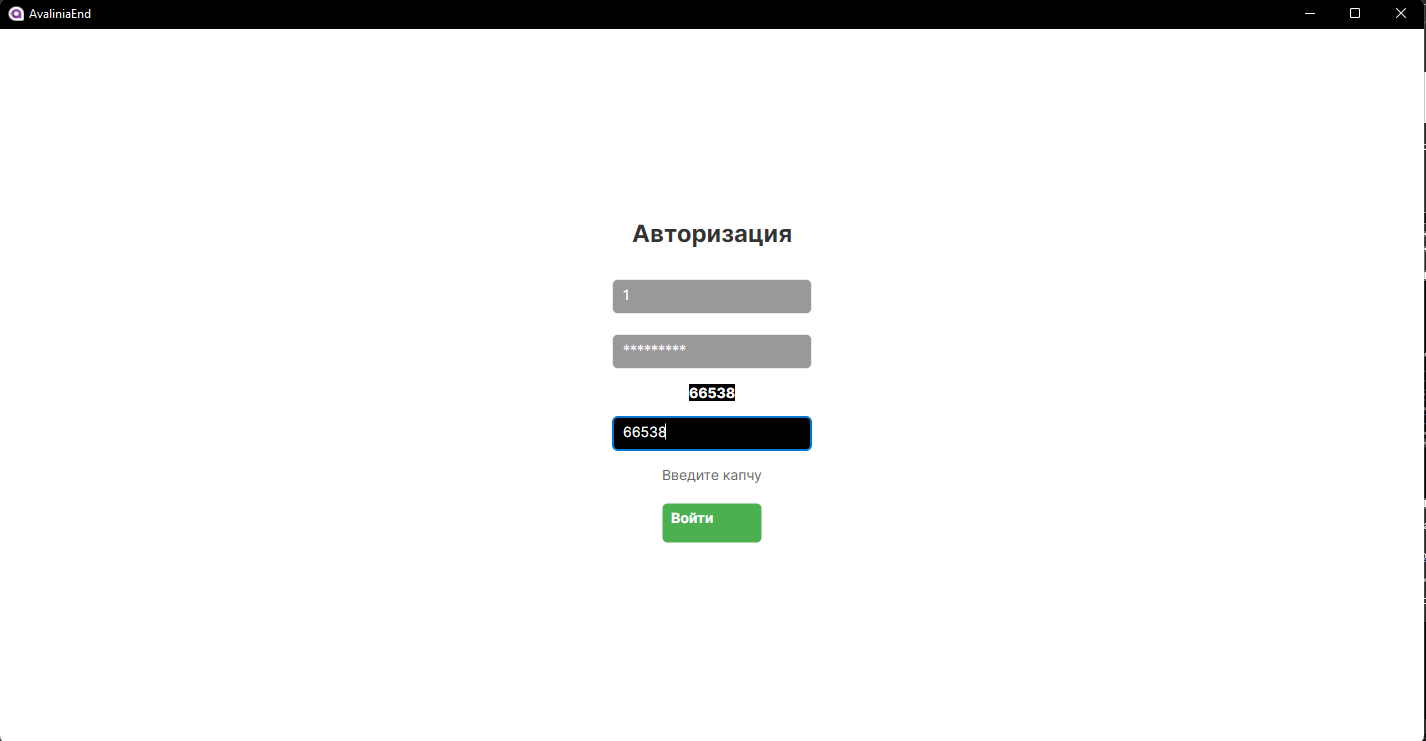


Рис 7– экран авторизации

После попадаем на экран организатора, в котором реализован грамотный вывод приветствия (корректное время суток и имя с отчеством), фото пользователя и разделы, с которыми может работать организатор

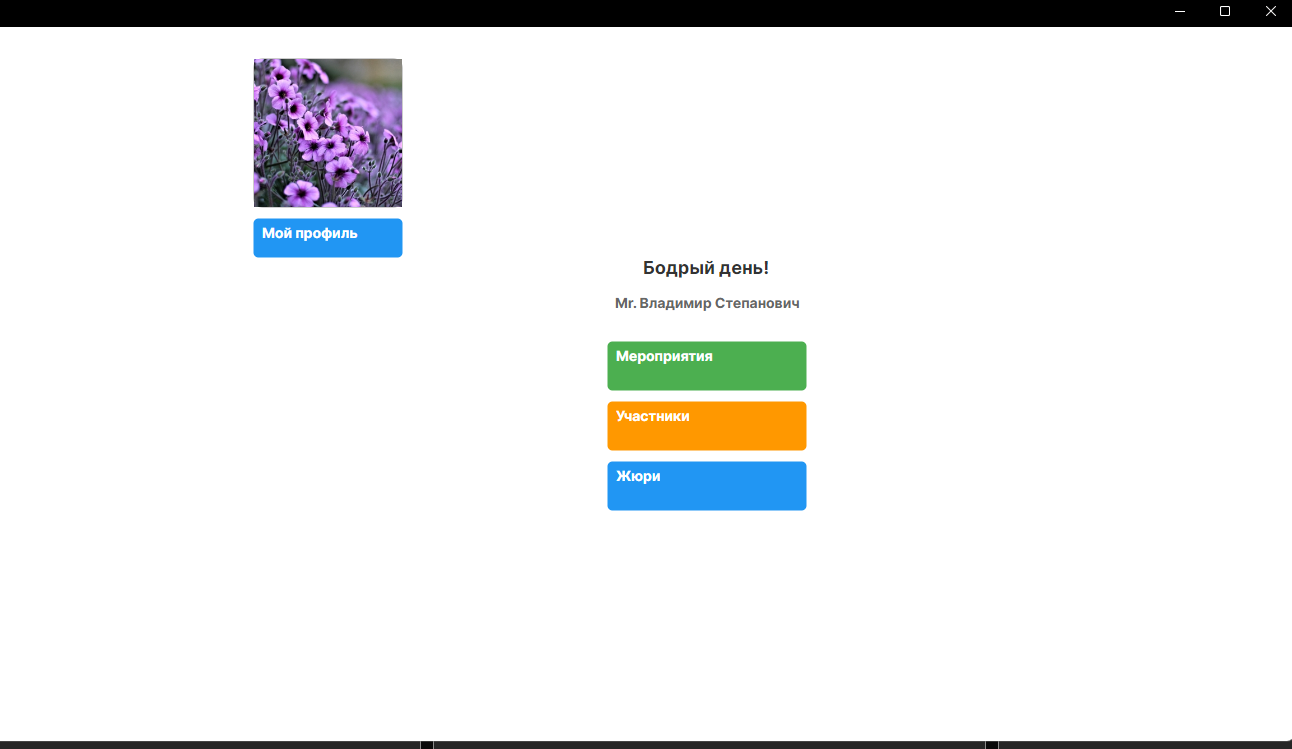


Рис 8– окно организатора

И последнее, что я успел это вход на окно мероприятий (и то не до конца)

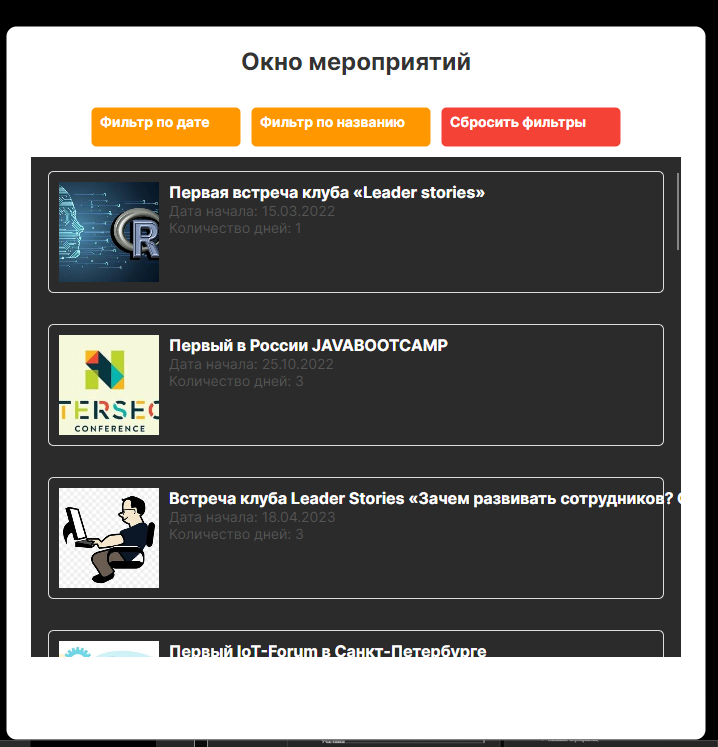


Рис 9– окно мероприятий

Также разработал руководство пользователя (полное руководство приложено в [приложении](#_Приложение_Б) Б):

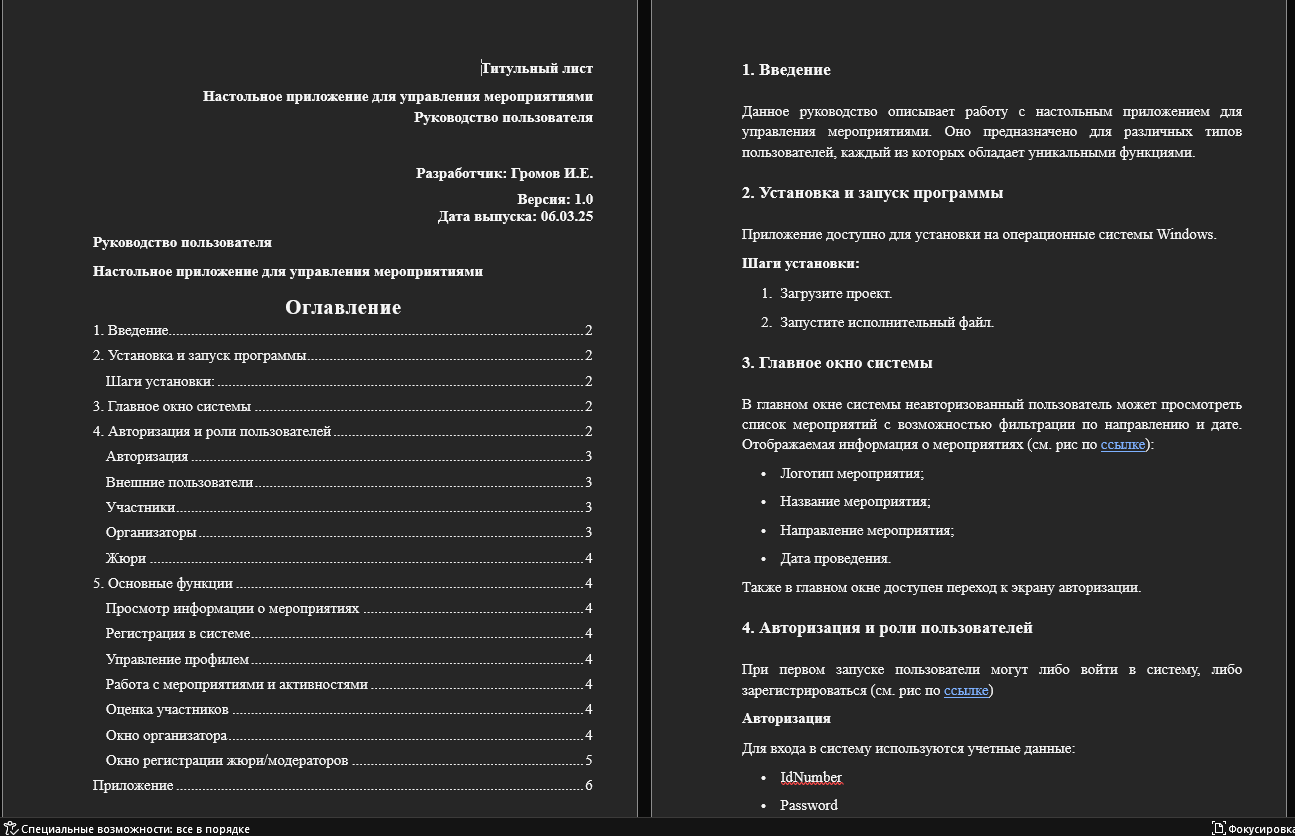


Рис 10 – руководство использования

# День 5 - DLL, Тестирование

Моей задачей являлось создание библиотеки REG\_MARK\_LIB.dll или REG\_MARK\_LIB.jar, которая имеет следующие методы(см. [листинг](#_Листинг_1) 2):

1. **Boolean CheckMark(String mark)** – данный метод проверяет переданный номерной знак в формате a999aa999 (латинскими буквами) и возвращает true или false в зависимости от правильности номерного знака. Метод должен учитывать также и существующие номера регионов.
2. **String GetNextMarkAfter(String mark)** – данный метод принимает номерной знак в формате a999aa999 (латинскими буквами) и выдает следующий номер в данной серии или создает следующую серию.
3. **String GetNextMarkAfterInRange(String prevMark, String rangeStart, String rangeEnd)** – данный метод принимает номерной знак в формате a999aa999 (латинскими буквами) и выдает следующий номер в данной данном промежутке номеров rangeStart до rangeEnd (включая обе границы). Если нет возможности выдать следующий номер, необходимо вернуть сообщение “out of stock”.
4. **Integer GetCombinationsCountInRange(String mark1, String mark2)** – данный метод принимает два номера в формате a999aa999 (латинскими буквами) и возвращает количество возможных номеров между ними (включая обе границы). Метод необходим, чтобы рассчитать оставшиеся свободные номера для региона.

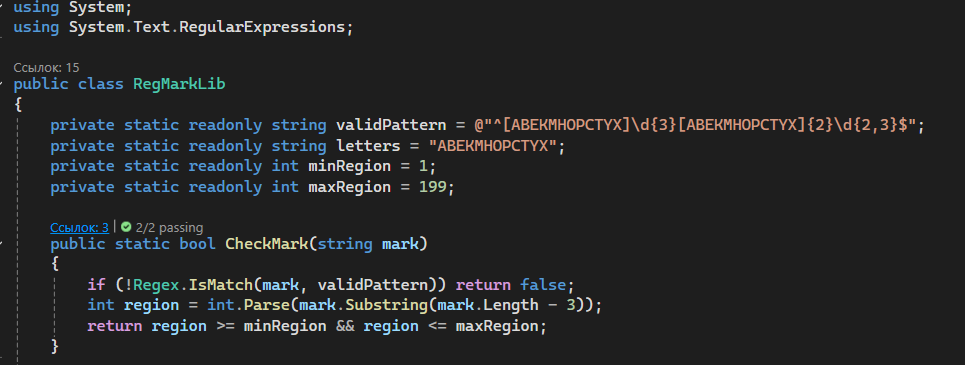


Рис 11 – объявление переменных и метод проверки корректности

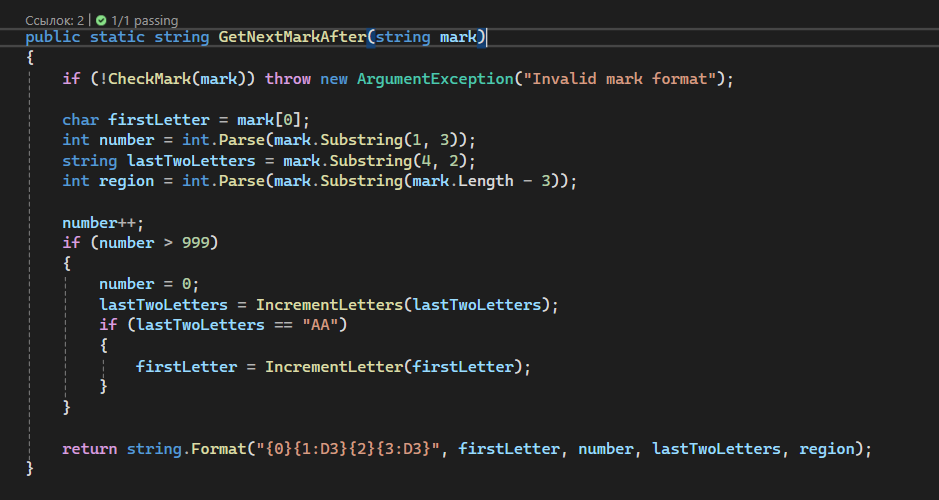


Рис 12– метод создания следующего номера

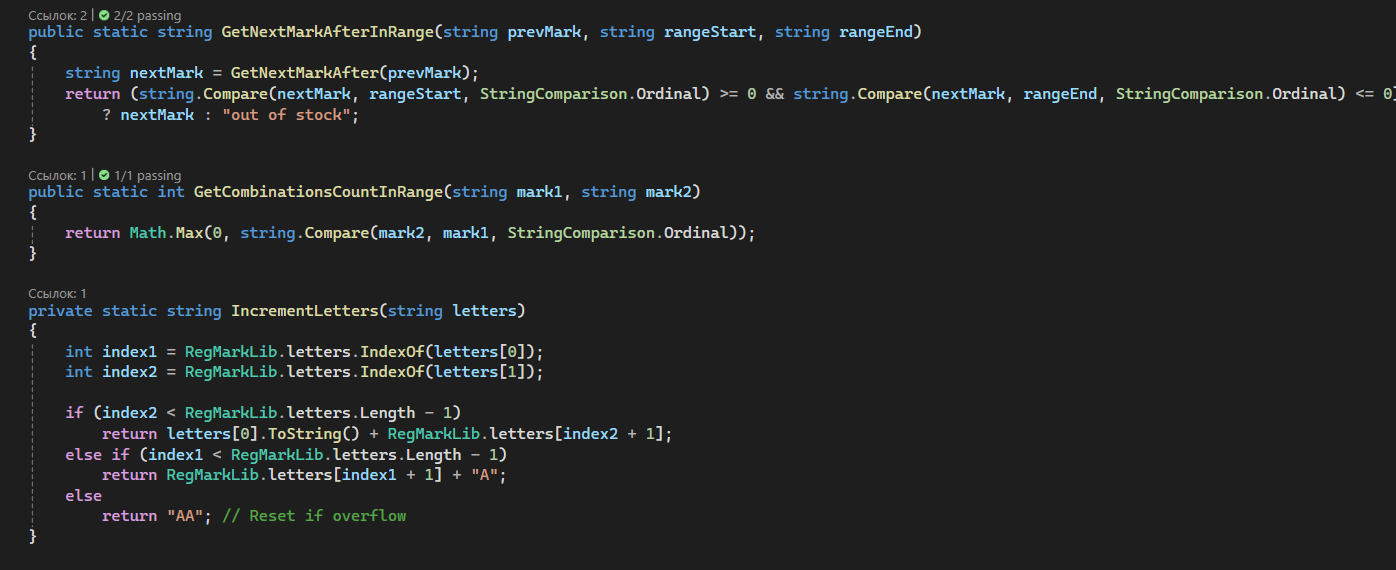


Рис. 13– методы «Создание нового номера в диапазоне» и «Получение количества номеров в диапазоне»

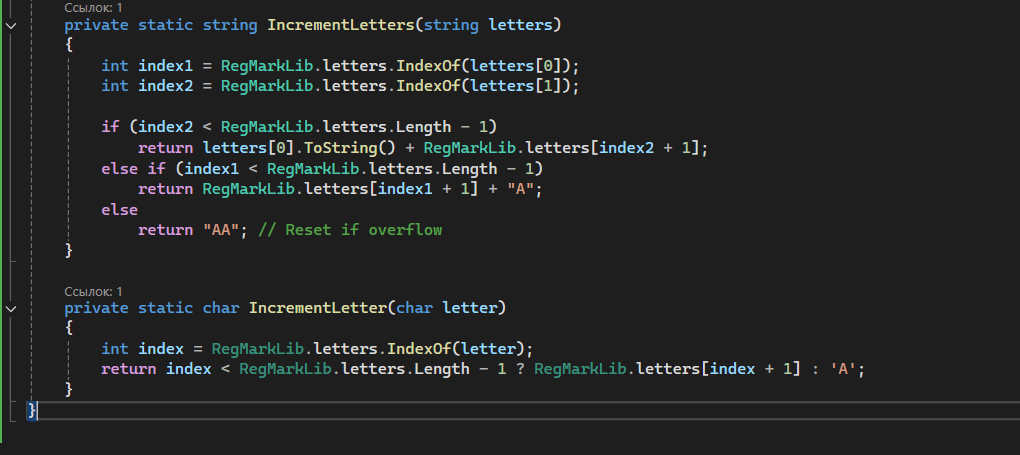


Рис 14– вспомогательные методы для увеличения буквенной последовательности

Далее были разработаны юнит-тесты, просмотреть которые можно в [листинге](#_Листинг_1)3

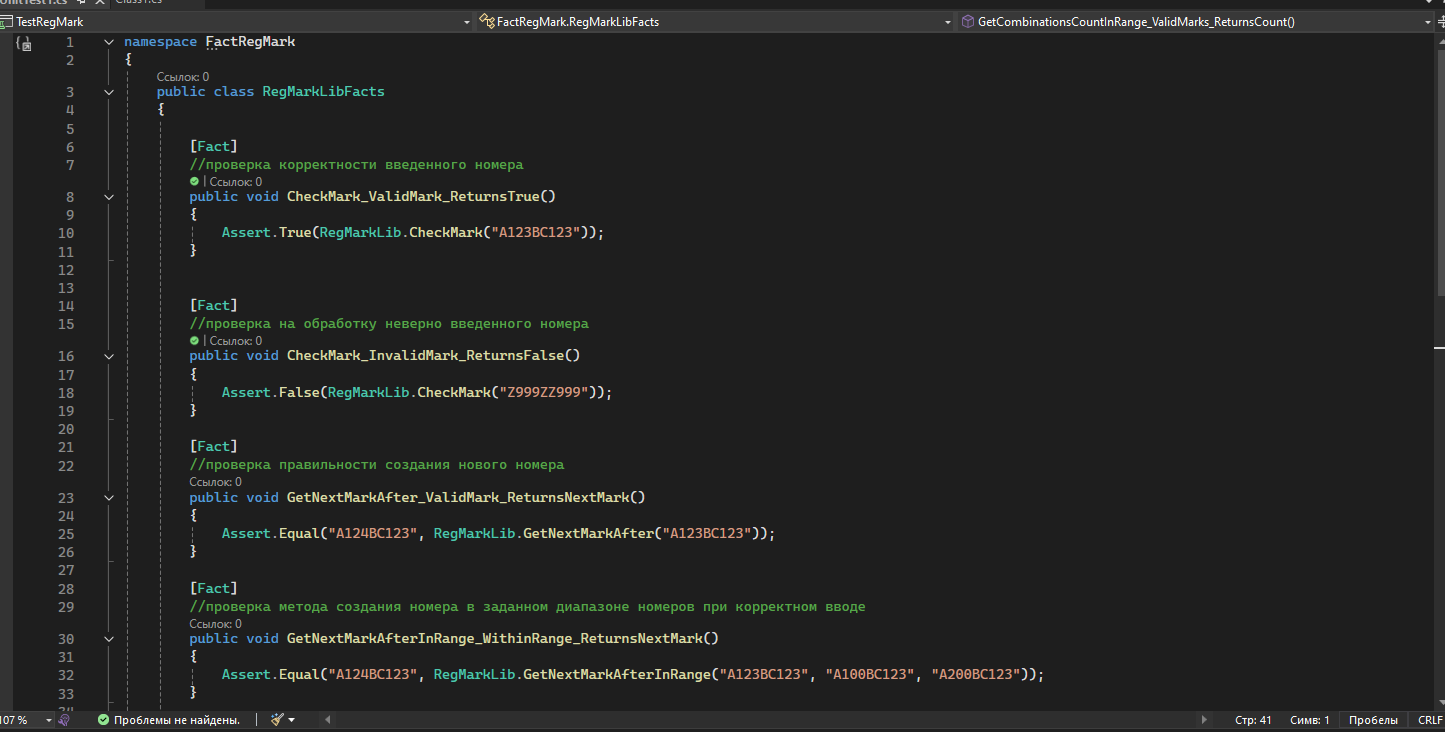


Рис 15– юнит-тесты

Затем были написаны тест-кейсы

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Тестовый пример #1:  |  |  | | --- | --- | | **Тестовый пример #** | 1 | | **Приоритет тестирования** | Низкий | | **Заголовок/название теста** | Добавление без фото | | **Краткое изложение теста** | Попытка добавления жюри/модератора без загрузки фотографии | | **Этапы теста** | - Открыть окно регистрации  - Заполнение соответствующих полей  - Добавление пользователя | | **Тестовые данные** | ФИО; пол (выбор из списка); роль (выбор из списка)’ e-mail; телефон в формате +7(999) - 099-90-90; направление (выбор из выпадающего списка или ручной ввод; при ручном вводе данные сохраняются в базе данных); • мероприятие (в случае выбора функции «Прикрепить к мероприятию», выбор из списка); фото (необязательное поле); пароль (с повтором), соответствующий требованиям | | **Ожидаемый результат** | Программа создаст нового жюри/модератора без фото | | **Фактический результат** | Программа позволяет создать нового жюри/модератора без добавления фотографии | | **Статус** | Успешно | | **Предварительное условие** | -Программа запущена  - Пройдена аутентификация под учетной записью организатора  -Запущено окно регистрации | | **Постусловие** | -Пользователь успешно добавлен  - Приложение работает в штатном режиме | | **Примечания/комментарии** | - |  Тестовый пример #2:  |  |  | | --- | --- | | **Тестовый пример #** | 2 | | **Приоритет тестирования** | Высокий | | **Заголовок/название теста** | Добавление без ФИО | | **Краткое изложение теста** | Добавление жюри/модератора без указания ФИО | | **Этапы теста** | - Открыть окно регистрации  - Заполнение соответствующих полей  - Добавление пользователя | | **Тестовые данные** | ФИО; пол (выбор из списка); роль (выбор из списка)’ e-mail; телефон в формате +7(999) - 099-90-90; направление (выбор из выпадающего списка или ручной ввод; при ручном вводе данные сохраняются в базе данных); • мероприятие (в случае выбора функции «Прикрепить к мероприятию», выбор из списка); фото (необязательное поле); пароль (с повтором), соответствующий требованиям | | **Ожидаемый результат** | Программа не создаст нового жюри/модератора без ввода ФИО | | **Фактический результат** | Программа выдает предупреждение о невозможности создания нового жюри/модератора без ввода ФИО | | **Статус** | Успешно | | **Предварительное условие** | -Программа запущена  - Пройдена аутентификация под учетной записью организатора  -Запущено окно регистрации | | **Постусловие** | -Программа выдала ошибку  - Приложение работает в штатном режиме | | **Примечания/комментарии** | - |  Тестовый пример #3:  |  |  | | --- | --- | | Тестовый пример **#** | 3 | | **Приоритет тестирования** | Высокий | | **Заголовок/название теста** | Добавление без указания направления | | **Краткое изложение теста** | Добавление жюри/модератора без указания направления | | **Этапы теста** | - Открыть окно регистрации  - Заполнение соответствующих полей  - Добавление пользователя | | **Тестовые данные** | ФИО; пол (выбор из списка); роль (выбор из списка)’ e-mail; телефон в формате +7(999) - 099-90-90; направление (выбор из выпадающего списка или ручной ввод; при ручном вводе данные сохраняются в базе данных); • мероприятие (в случае выбора функции «Прикрепить к мероприятию», выбор из списка); фото (необязательное поле); пароль (с повтором), соответствующий требованиям | | **Ожидаемый результат** | Программа не даст создать пользователя без указания направления | | **Фактический результат** | Программа запрещает создание пользователя без указания направления | | **Статус** | Успешно | | **Предварительное условие** | -Программа запущена  - Пройдена аутентификация под учетной записью организатора  -Запущено окно регистрации | | **Постусловие** | -Программа выдала ошибку  - Приложение работает в штатном режиме | | **Примечания/комментарии** | - |  Тестовый пример #4:  |  |  | | --- | --- | | Тестовый пример **#** | 4 | | **Приоритет тестирования** | Средний | | **Заголовок/название теста** | Попытка создания кривого пароля | | **Краткое изложение теста** | Создание пользователя с паролем, не соответствующим требованиям | | **Этапы теста** | - Открыть окно регистрации  - Заполнение соответствующих полей  - Добавление пользователя | | **Тестовые данные** | пароль (с повтором), не соответствующий требованиям: • не менее 6 символов; • заглавные и строчные буквы; • не менее одного спецсимвола; • не менее одной цифры. | | **Ожидаемый результат** | Программа уведомит о неверном формате пароля | | **Фактический результат** | Программа уведомляет о неверном формате пароля и просит ввести новый пароль | | **Статус** | Успешно | | **Предварительное условие** | -Программа запущена  - Пройдена аутентификация под учетной записью организатора  -Запущено окно регистрации | | **Постусловие** | -Программа выдала ошибку  - Приложение работает в штатном режиме | | **Примечания/комментарии** | - |  Test case #5:  |  |  | | --- | --- | | **Test Case #** | 5 | | **Приоритет тестирования** | Средний | | **Заголовок/название теста** | Создание без телефона | | **Краткое изложение теста** | Создание пользователя без указания номера или ввода его в неверном формате | | **Этапы теста** | - Открыть окно регистрации  - Заполнение соответствующих полей  - Добавление пользователя | | **Тестовые данные** | ФИО; пол (выбор из списка); роль (выбор из списка)’ e-mail; телефон в формате +7(999) - 099-90-90; направление (выбор из выпадающего списка или ручной ввод; при ручном вводе данные сохраняются в базе данных); • мероприятие (в случае выбора функции «Прикрепить к мероприятию», выбор из списка); фото (необязательное поле); пароль (с повтором), соответствующий требованиям | | **Ожидаемый результат** | Программа не даст создать пользователя без указания номера | | **Фактический результат** | Программа запрещает создание пользователя без указания номера | | **Статус** | Успешно | | **Предварительное условие** | -Программа запущена  - Пройдена аутентификация под учетной записью организатора  -Запущено окно регистрации | | **Постусловие** | -Программа выдала ошибку  - Приложение работает в штатном режиме | | **Примечания/комментарии** | - | |

# День 6 – проверка чужой работы и написание итогового отчета

Проверка чужой работы по критериям, изложенным в [приложении](#_Приложение_В_(критерии)) В

|  |  |  |
| --- | --- | --- |
|  | Criteria | Mark |
| A | Системный анализ и проектирование | 12,00 |
| B | Разработка программного обеспечения | 23,78 |
| C | Стандарты разработки программного обеспечения | 7,00 |
| D | Документирование программных решений | 5,00 |
| E |  | 0,00 |
| F |  | 0,00 |
| G |  | 0,00 |
| H |  | 0,00 |
| I |  | 0,00 |
| Total |  | 47,58 |

# Далее разрабатывал данный отчетПриложение А (Таблицы для импорта)

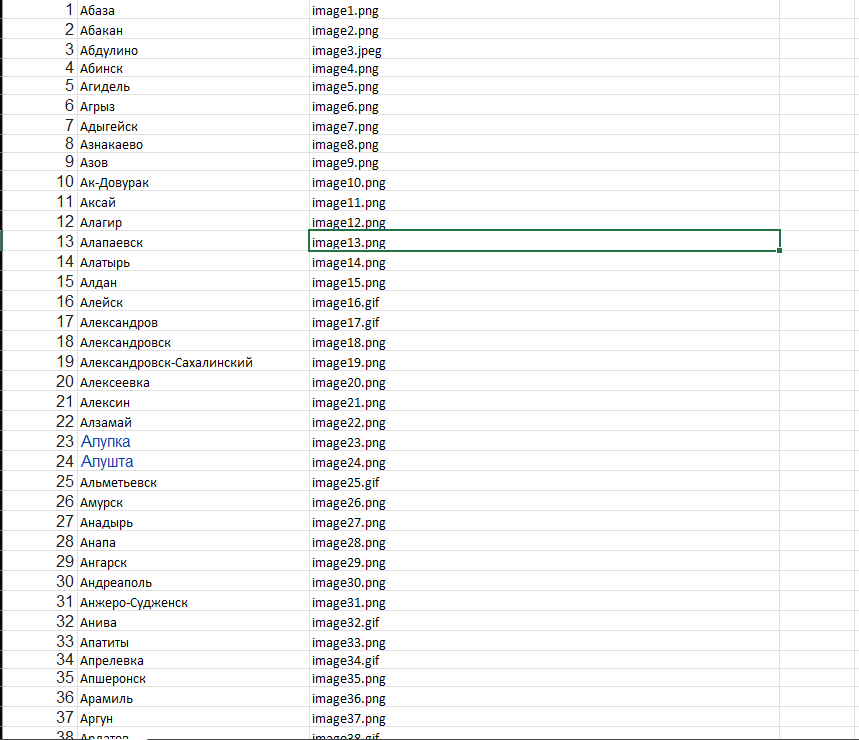


Рис – таблица «Города»

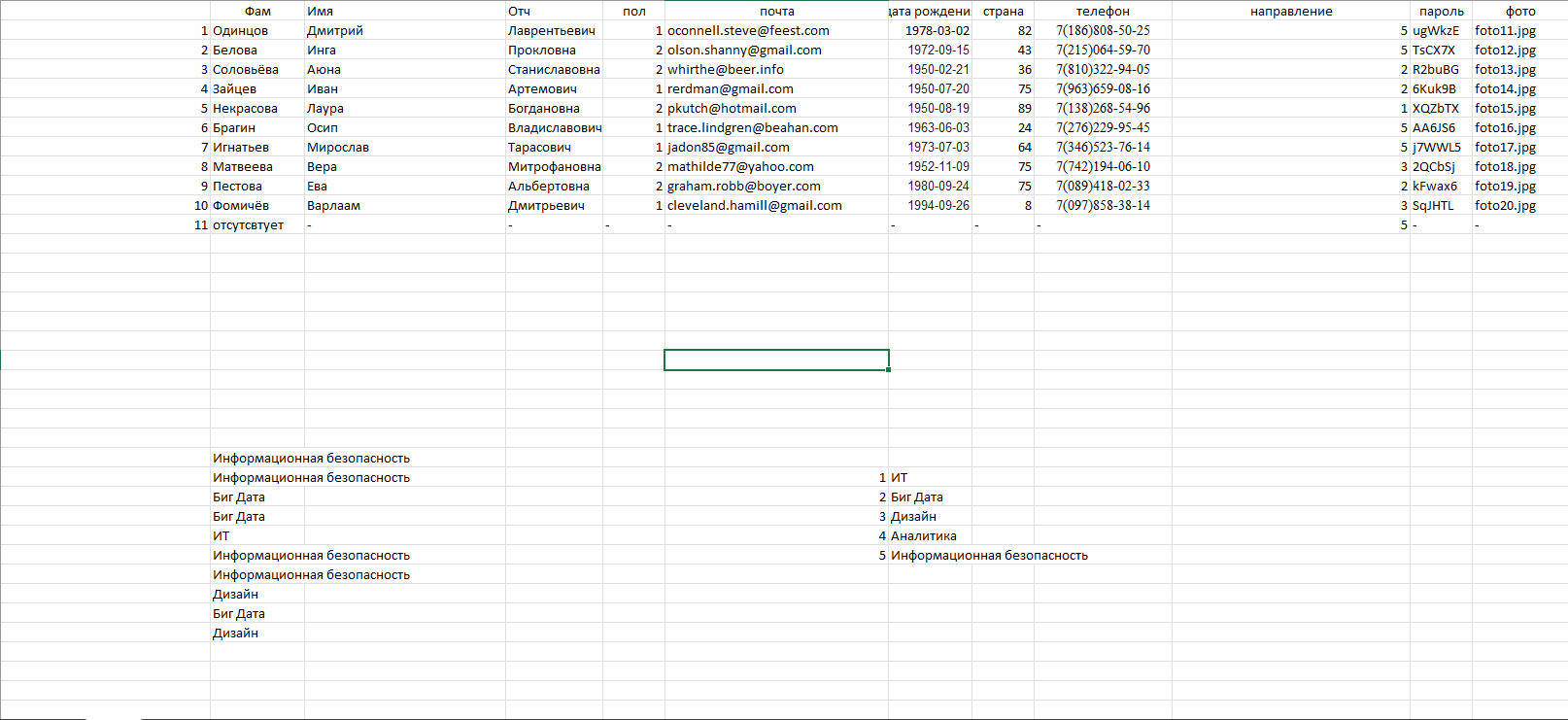


Рис – таблица «Жюри»

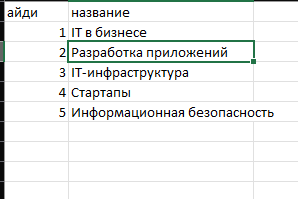


Рис – таблица «Событие»

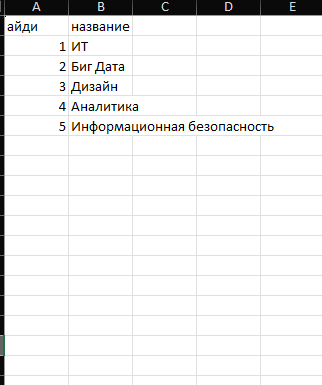


Рис – таблица «Направление»

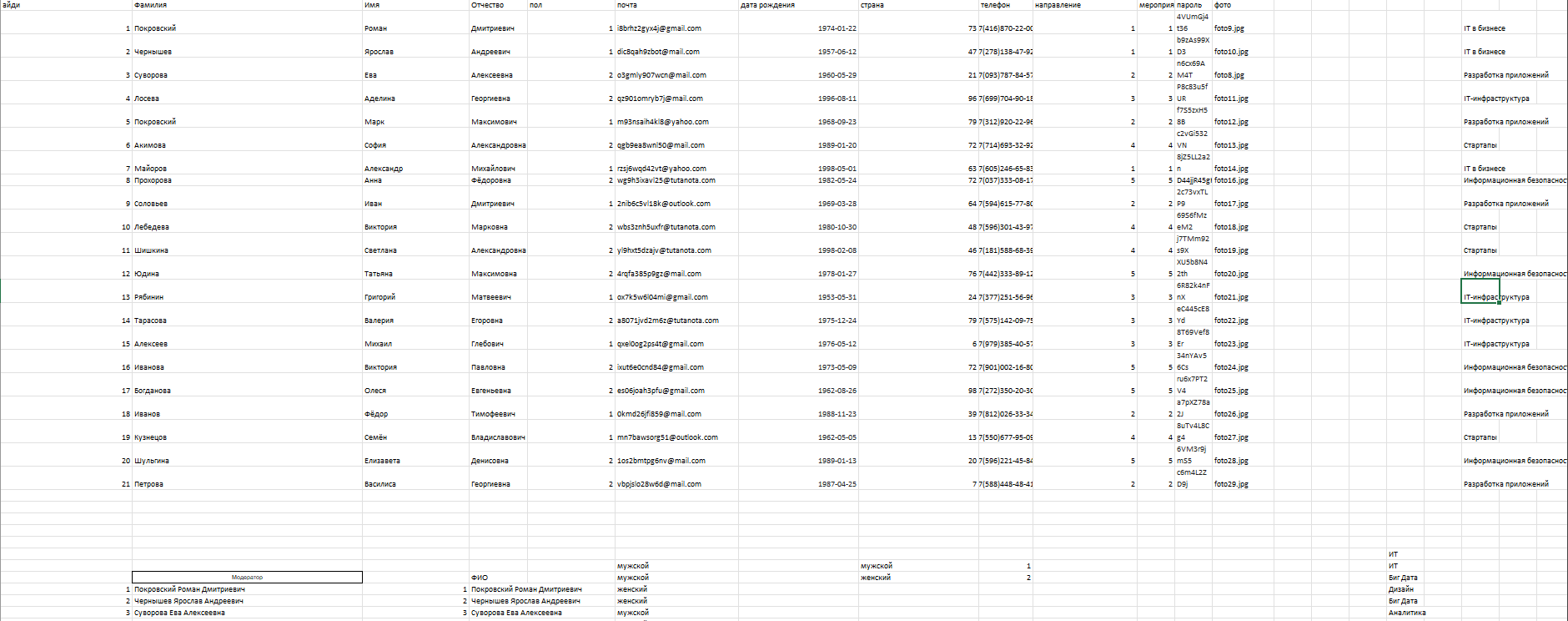


Рис – таблица «Модераторы»

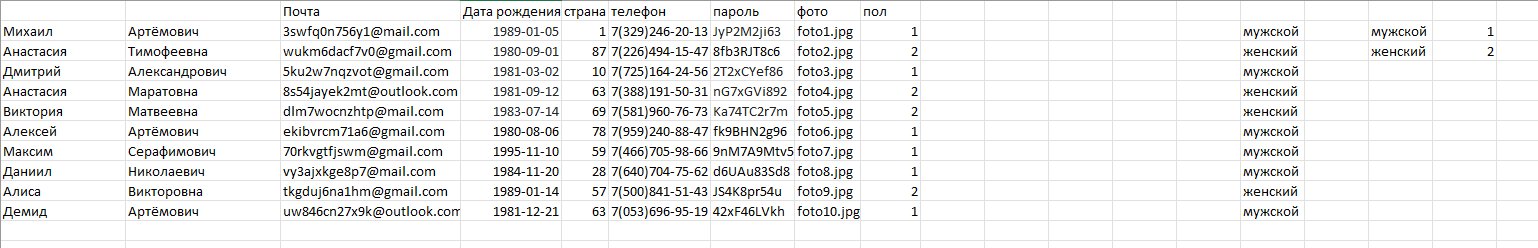


Рис – таблица «Организаторы»

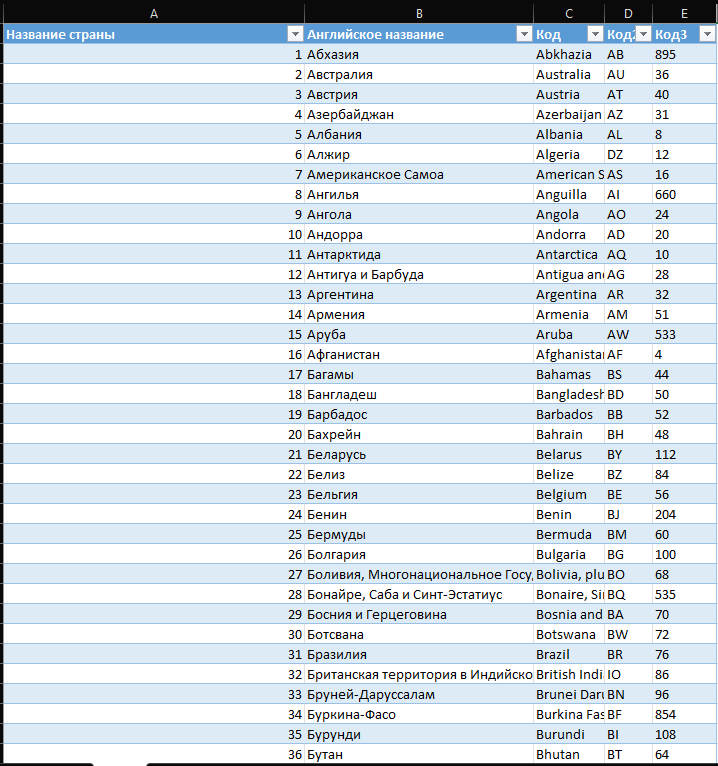


Рис – таблица «Страны»



Рис – таблица «Участники»

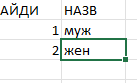


Рис – таблица «Пол»

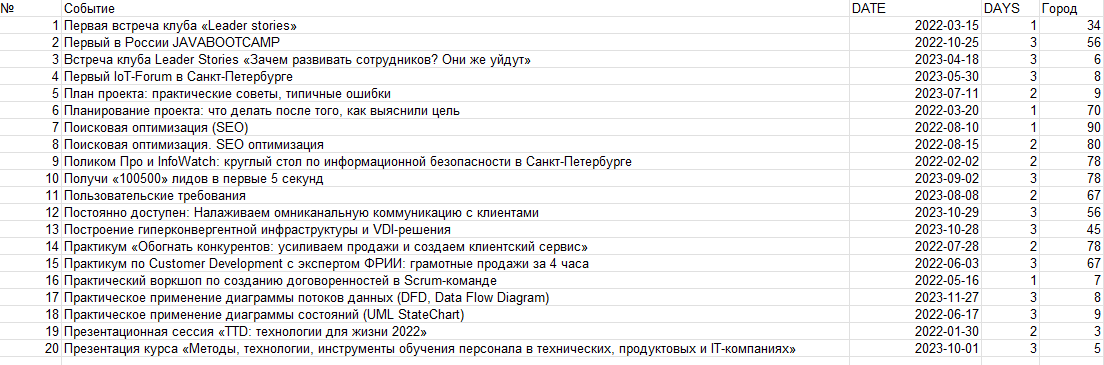


Рис – таблица «Мероприятия»

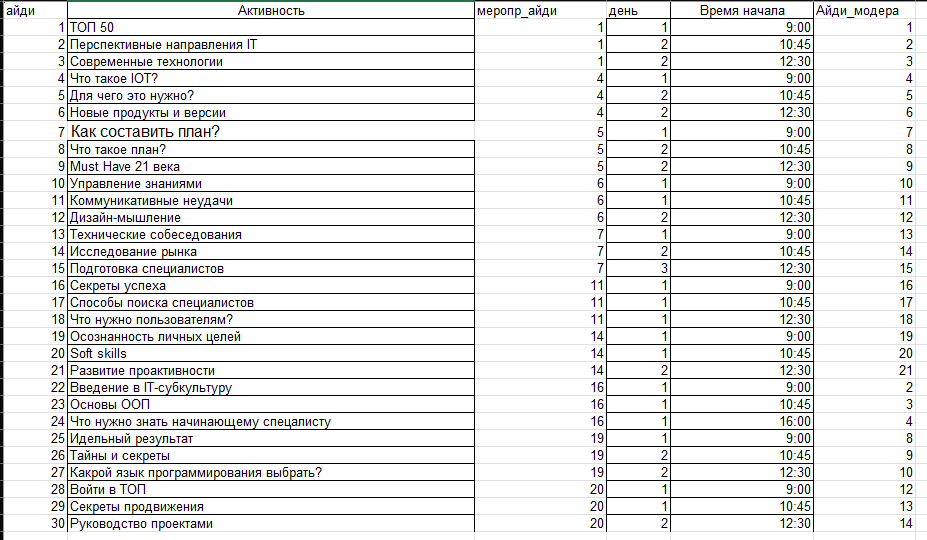


Рис – таблица «Активности»

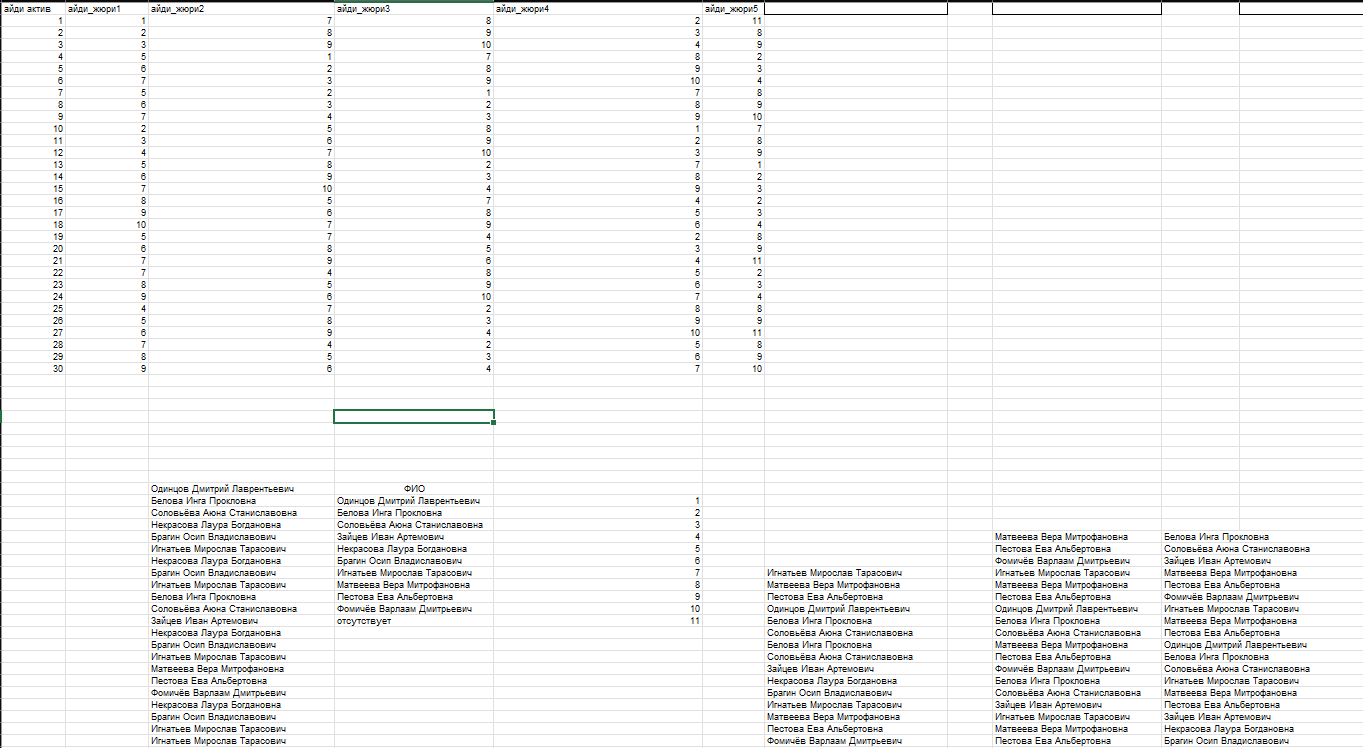


Рис – таблица для связи жюри и активностей

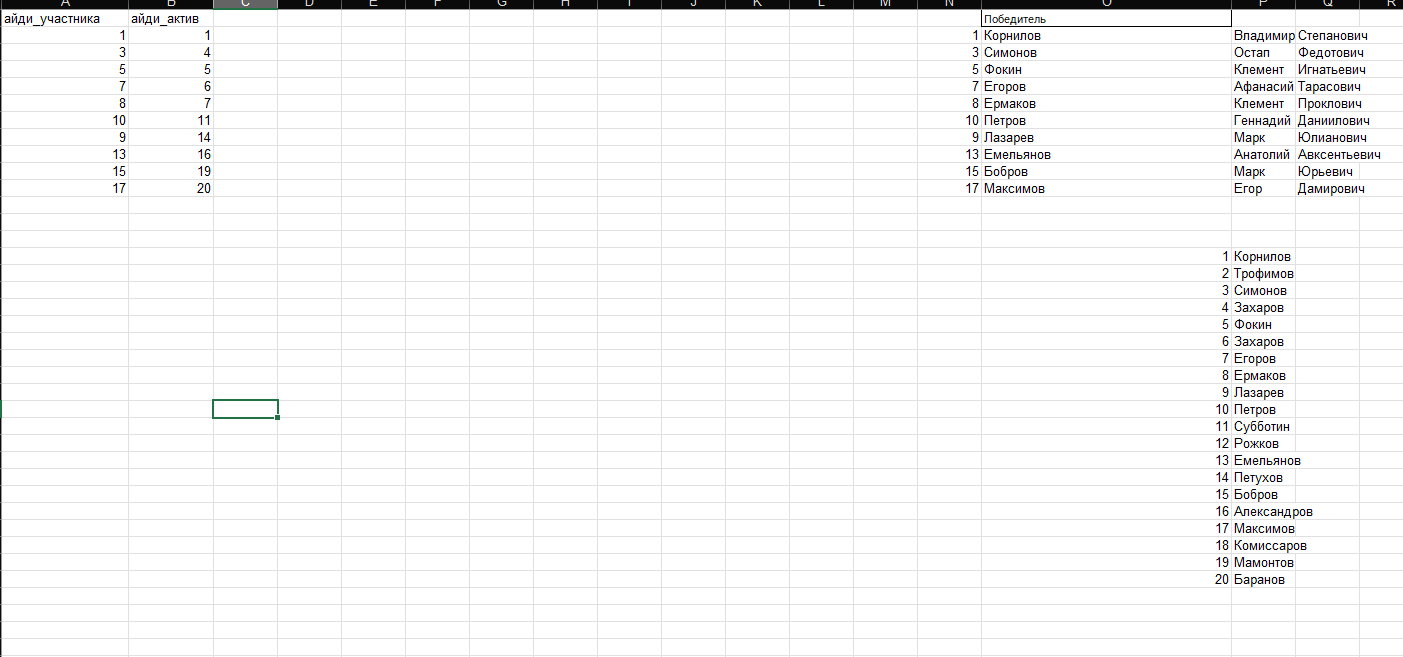


Рис – таблица «Победитель»

# Приложение Б (Руководство)

|  |
| --- |
| Титульный лист  Настольное приложение для управления мероприятиями Руководство пользователя  Разработчик: Громов И.Е.  Версия: 1.0 Дата выпуска: 06.03.25  Руководство пользователя  Настольное приложение для управления мероприятиями  1. Введение  Данное руководство описывает работу с настольным приложением для управления мероприятиями. Оно предназначено для различных типов пользователей, каждый из которых обладает уникальными функциями.  2. Установка и запуск программы  Приложение доступно для установки на операционные системы Windows.  Шаги установки:  Загрузите проект.  Запустите исполнительный файл.  3. Главное окно системы  В главном окне системы неавторизованный пользователь может просмотреть список мероприятий с возможностью фильтрации по направлению и дате. Отображаемая информация о мероприятиях (см. рис по [ссылке](#_Рис_–_начальный)):  Логотип мероприятия;  Название мероприятия;  Направление мероприятия;  Дата проведения.  Также в главном окне доступен переход к экрану авторизации.  4. Авторизация и роли пользователей  При первом запуске пользователи могут либо войти в систему, либо зарегистрироваться (см. рис по [ссылке](#_Рис_–_экран))  Авторизация  Для входа в систему используются учетные данные:  IdNumber  Password  Дополнительно реализована защита от несанкционированного доступа:  CAPTCHA (4 символа и графический шум);  Блокировка входа на 10 секунд после трех неудачных попыток;  Возможность запоминания учетных данных пользователя.  После успешного входа пользователи попадают в соответствующее окно:  Участник → Окно участника;  Модератор → Окно модератора;  Организатор → Окно организатора;  Жюри → Окно жюри.  Внешние пользователи  Могут просматривать общую информацию о мероприятиях.  Не могут вносить изменения.  Участники  Регистрируются в системе и получают личный кабинет.  Просматривают и редактируют профиль.  Работают в группе в рамках активностей мероприятия.  Модераторы  Регистрируются в системе.  Выбирают активность для модерации.  Координируют участников.  Организаторы  Управляют мероприятиями и участниками.  Добавляют новые события и активности.  Регистрируют жюри и модераторов.  Жюри  Просматривают участников активностей.  Выставляют оценки.  5. Основные функции  Просмотр информации о мероприятиях  В меню "Мероприятия" доступен список всех мероприятий с фильтрацией по дате и категории.  Для получения подробной информации нажмите на название мероприятия.  Регистрация в системе  Нажмите "Регистрация" на главном экране.  Введите личные данные и выберите роль.  Подтвердите регистрацию через e-mail.  Управление профилем  В личном кабинете доступны редактирование данных, загрузка аватара и смена пароля.  Работа с мероприятиями и активностями  Организаторы могут добавлять мероприятия через меню "Создать мероприятие".  Участники записываются на мероприятия, выбирают активности.  Модераторы управляют активностями.  Оценка участников  Жюри в разделе "Оценки" выбирают активность и выставляют баллы участникам.  Окно организатора  См. рис. по [ссылке](#_Рис_–_окно)  При входе система приветствует пользователя по имени и отчеству.  Приветствие зависит от времени суток:  9:00 - 11:00 – Утро;  11:01 - 18:00 – День;  18:01 - 24:00 – Вечер.  Подгружается фото пользователя.  Окно регистрации жюри/модераторов  См. рис. по [ссылке](#_Рис_–_страница)  Регистрация доступна только организаторам.  При открытии окна система генерирует уникальный IdNumber, который нельзя изменить.  Необходимо заполнить следующие данные:  ФИО;  Пол (выбор из списка);  Роль (жюри/модератор);  Email;  Телефон (+7(999)-099-90-90);  Направление (выбор из списка или ручной ввод с сохранением в БД);  Прикрепление к мероприятию (если требуется);  Фото (необязательное поле);  Пароль (с повтором), соответствующий требованиям:  Не менее 6 символов;  Заглавные и строчные буквы;  Минимум 1 спецсимвол;  Минимум 1 цифра.  Приложение    Рис – начальный экран    Рис – экран авторизации    Рис – окно организатора    Рис – окно мероприятий    Рис – страница регистрации |

# Приложение В (критерии)

|  |  |  |  |
| --- | --- | --- | --- |
| **Sub  Criteria  ID** | **Sub Criteria    Name or Description** | **Aspect - Description** | **Extra Aspect Description (Obj or Subj)  OR  Judgement Score Description (Judg only)** |
| A1 | Диаграмма прецедентов |  |  |
|  |  | Актеры определены верно | Минус 40% за каждую ошибку |
|  |  | Варианты использования определены верно | Минус 40% за каждую ошибку |
|  |  | Виды взаимодействия определены верно | Минус 40% за каждую ошибку |
| A2 | Дизайн базы данных (ERD) - проверка по итогам двух сессий |  |  |
|  |  | Основные сущности определены | Минус 0,5 балла за ошибку или не определенный элемент |
|  |  | Отношения определены правильно (с учётом отсутствующих объектов) | Минус 0,3 балла за ошибку или не определенный элемент |
|  |  | Все атрибуты (поля) рассмотрены и определены | Минус 0,4 балла за ошибку или не определенный элемент |
|  |  | Созданы ограничения на связи между сущностями, отражающие характер предметной области | Минус 0,25 балла за ошибку или не определенный элемент |
|  |  | Идентификатор в таблице присутствует | Минус 0,1 балла за ошибку или не определенный элемент |
|  |  | Разработанная база данных находится в 3НФ (при наличии минимального необходимого набора сущностей и связей) | Минус 0,3 балла за НФ |
|  |  | Таблицы и поля названы в соответствии с индустриальными стандартами |  |
|  |  |  | Названия таблиц и полей содержат орфографические ошибки и/или записаны в разном стиле |
|  |  |  | Названия таблиц и полей позволяют однозначно идентифицировать их содержимое |
|  |  |  | Названия таблиц и полей отождествляются с сущностями и атрибутами предметной области однозначно |
|  |  |  | Названия таблиц и полей в едином стиле, падеже, числе и дают полное и очевидное понимание хранимых данных |
| A3 | Словарь данных |  |  |
|  |  | Словарь данных сохранен согласно требованиям |  |
|  |  | Словарь данных соответствует предоставленной ERD | Минус 20% за каждую ошибку |
|  |  | Все типы данных, размеры полей являются подходящими | Минус 10% за каждую ошибку |
|  |  | Обязательность полей является подходящей | Минус 20% за каждую ошибку |
|  |  | Комментарии содержит полезную информацию для администраторов БД |  |
|  |  |  | Словарь данных отсутствует |
|  |  |  | Словарь данных заполнен информацией, не несущей дополнительного смысла, о таблицах, полях, типах данных и ключах |
|  |  |  | Комментарии, указанные дополнительно к основной информации в словаре данных, помогают ориентироваться в системе |
|  |  |  | С помощью словаря данных легко разобраться в структуре любому специалисту, не имеющему отношения к данной предметной области |
| A7 | Оценка структуры проекта - проверка по итогам всех сессий |  |  |
|  |  | Название приложения соответствует названию компании |  |
|  |  | Логика приложения не перемешана | Минус 40% за каждую ошибку |
|  |  |  |  |
| Sub  Criteria  ID | Sub Criteria    Name or Description | Aspect - Description | Extra Aspect Description (Obj or Subj)  OR  Judgement Score Description (Judg only) |
| B1 | Разработка ПО - Главное окно системы |  |  |
|  |  | Список мероприятий отображается верно | Минус 40% за каждую ошибку |
|  |  | Реализован фильтр по направлению | Минус 40% за каждую ошибку |
|  |  | Реализован поиск по дате | Минус 40% за каждую ошибку |
|  |  | Переход на авторизацию |  |
| B2 | Разработка ПО- Авторизация |  |  |
|  |  | Окно авторизации реализовано |  |
|  |  | Авторизация работает на основе БД |  |
|  |  | Переход осуществляется в соответствии с ролью пользователя | Минус 40% за каждую ошибку или отсутствующее окно учетной записи (хотя бы заголовок окна) |
| B3 | Разработка ПО - Окно организатора |  |  |
|  |  | Окно создано и соответствует макету | Минус 40% за каждую ошибку |
|  |  | Система приветствует пользователя по имени и отчеству |  |
|  |  | Система приветствует пользователя  с указанием времени работы |  |
|  |  | При входе в систему подгружается фото пользователя |  |
| B4 | Разработка ПО-Регистрация жюри и модераторов |  |  |
|  |  | Окно создано и соответствует макету | Минус 40% за каждую ошибку |
|  |  | Окно доступно только организаторам |  |
|  |  | Происходит генерирация уникального ID Number, изменить его нельзя |  |
|  |  | Выбор пола из списка |  |
|  |  | Маска на поле телефон |  |
|  |  | Выбор роли из списка |  |
|  |  | Ввод корректного e-mail |  |
|  |  | Требования к паролю учтены | Минус 40% за каждую ошибку |
|  |  | Изображение загружается |  |
|  |  | Повтор пароля реализован с указанием видимости | Минус 40% за каждую ошибку |
| B5 | Обработка ошибок/исключений - по итогам всех сессий |  |  |
|  |  | Приложение корректно обрабатывает или запрещает несоответствие введенной информации типу данных атрибута. |  |
|  |  | Приложение корректно отображает изображения при перемещении папки с исполняемым файлом |  |
|  |  | Фатальные ошибки не возникают и приложение аварийно не завершает работу, реализована программная обработка исключительных ситуаций в приложении | Минус 25% за каждое аварийное завершение работы |
| B6 | Импорт данных |  |  |
|  |  | Данные об активностях импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные о городах импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные о жюри импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные о мероприятиях импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные о странах импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные об участниках импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные о модераторах импортированы | Минус 20 % за каждую ошибку |
|  |  | Данные об организаторах импортированы | Минус 20 % за каждую ошибку |
|  |  |  |  |
| Sub  Criteria  ID | Sub Criteria    Name or Description | Aspect - Description | Extra Aspect Description (Obj or Subj)  OR  Judgement Score Description (Judg only) |
| C1 | Согласованный внешний вид приложения в соответствии с требованиями к макету - проверка по итогам всех сессий |  |  |
|  |  | Присутствует возможность изменения размеров окна, где это необходимо |  |
|  |  | Присутствует ограничение на минимальный размер окна |  |
|  |  | Верстка масштабируемая: при увеличении размера окна увеличивается размер контентной части |  |
|  |  | Использованы соответствующие элементы управления (например, выпадающие списки, поля для выбора дат с календарем и тп) |  |
|  |  | У всех компонентов на всех экранах настольного приложения одинаковое выравнивание, размерность, отступы |  |
|  |  | Переход фокуса ввода происходит последовательно по элементам интерфейса |  |
|  |  | Разработан последовательный пользовательский интерфейс: есть возможность перемещаться между существующими окнами (переходы между окнами, кнопка назад) |  |
|  |  | На каждом окне приложения присутствует заголовок |  |
| C2 | Исходный код - проверка по итогам всех сессий |  |  |
|  |  | Исходный код приложения соответствует гайдлайну | Минус 25%, если идентификаторы не соответствуют стилю CamelCase (для C# и Java) и snake\_case (для Python) |
|  |  | Четкость, смысл именования очевидны | Минус 25% за каждую ошибку |
|  |  | Использовано не более одной команды в строке | Минус 25% за каждую ошибку |
|  |  | Код понятен без комментариев |  |
|  |  |  | Код нечитаемый |
|  |  |  | Код позволяет минимально понять реализованный функционал |
|  |  |  | Код хорошо организован, позволяет с легкостью понять реализованный функционал |
|  |  |  | Идеальный код |
| C3 | Предоставление результатов работы (git) - проверка по итогам всех сессий |  |  |
|  |  | Все необходимые результаты размещены в репозитории системы контроля версий | Минус 25% за отсутствующий результат |
|  |  | Исходные коды представлены не в виде архива |  |
| C4 | Сообщения обратной связи с пользователем |  |  |
|  |  | Выводится информация об отсутствии результатов поиска |  |
|  |  | Перед удалением объекта запрашивается подтверждение пользователя |  |
|  |  | Сообщения обратной связи (подсказки, ошибки) полезны для пользователей при работе системы |  |
|  |  |  | Обратная связь для пользователя отсутствует |
|  |  |  | Сообщения для пользователя присутствуют, но по ним сложно определить, что именно заполнено неверно и как исправить ошибку |
|  |  |  | Сообщения полезные и информативные, содержат полную информацию о совершенных ошибках пользователя и подсказки для их исправления, ясно дают понять порядок дальнейших действий |
|  |  |  | Присутствуют визуальные подсказки для пользователя при вводе данных; Сообщения полезные и информативные, содержат полную информацию о совершенных ошибках пользователя и подсказки для их исправления; Сообщения разделены на типы (ошибки, предупреждения, информация) |
| C5 | Комментарии по коду - проверка по итогам всех сессий |  |  |
|  |  | Отсутствуют закомментированные блоки кода |  |
|  |  | Использованный тип комментариев позволяет сгенерировать XML-документацию | Используются соответствующие теги: param, return(s), summary и др. |
|  |  | Исходный код приложения прокомментирован с очевидным смыслом |  |
|  |  |  | Комментарии отсутствуют |
|  |  |  | Комментарии дают минимальное понимание кода |
|  |  |  | Комментарии доступно поясняют код |
|  |  |  | Комментарии дают полное и очевидное понимание кода |
|  |  |  |  |
| Sub  Criteria  ID | Sub Criteria    Name or Description | Aspect - Description | Extra Aspect Description (Obj or Subj)  OR  Judgement Score Description (Judg only) |
| D1 | Руководство пользователя |  |  |
|  |  | Название файла с руководством пользователя соответствует требованиям |  |
|  |  | Документ оформлен в соответствии с требованиями | Минус 0,1 за каждую ошибку |
|  |  | Документация составлена профессионально |  |
|  |  |  | Руководство пользователя не предоставлено или невозможно определить, что документ предоставленный документ является руководством пользователя |
|  |  |  | Руководство пользователя составлено посредственно, неинформативно, без пояснений и снимков экрана |
|  |  |  | Все возможности системы описаны в руководстве пользователя понятным образом, присутствуют понятные пояснения, примеры и снимки экрана |
|  |  |  | Потрясающее руководство пользователя! |
|  |  | Документация охватывает всю систему |  |
|  |  |  | Документация отсутствует |
|  |  |  | В документации представлены некоторые (не все) возможности системы |
|  |  |  | Все возможности системы представлены в документации |
|  |  |  | Все возможности системы для указанных ролей представлены в документации, выполнено превосходно |

# Листинг

Листинг 1 – скрипт БД

|  |
| --- |
| **create** **table** Country(  id\_Country serial **primary** **key** **not** **null**,  countryName **varchar**(50) **not** **null**,  countryNameEng **varchar**(50) **not** **null**,  firstCode **varchar**(10) **not** **null**,  secondCode **int** **not** **null**  )  **create** **table** Genders(  idGender serial **primary** **key** **not** **null**,  genderName **text** **not** **null**  )  **create** **table** City(  id\_City serial **primary** **key** **not** **null**,  cityName **char**(30) **not** **null**,  cityimage **text** **not** **null**  )  **create** **table** Client(  idClient serial **primary** **key** **not** **null**,  lastName **varchar**(100) **not** **null**,  firstName **varchar**(100) **not** **null**,  patronymic **varchar**(100) **not** **null**,  Email **text** **not** **null**,  dateOfBirth **timestamp** **not** **null**,  countryId **int** **references** Country(id\_Country),  phone **varchar**(20) **not** **null**,  **password** **text** **not** **null**,  image **text** **not** **null**,  genderId **int** **references** Genders(idGender)  )  **create** **table** Organizer(  idOrganizer serial **primary** **key** **not** **null**,  lastName **varchar**(100) **not** **null**,  firstName **varchar**(50) **not** **null**,  patronymic **varchar**(50) **not** **null**,  email **text** **not** **null**,  dateOfBirth **timestamp** **not** **null**,  countryId **int** **references** Country(id\_Country),  phone **varchar**(20) **not** **null**,  **password** **text**,  foto **text** **not** **null**,  genderId **int** **references** Genders(idGender)  )  **create** **table** Napravlenie(  idNapravlenie serial **primary** **key** **not** **null**,  NapravlenieName **text** **not** **null**  )  **create** **table** Events(  idEvent serial **primary** **key** **not** **null**,  eventName **text** **not** **null**  )  **create** **table** Jury(  juryId serial **primary** **key** **not** **null**,  lastName **varchar**(100) **not** **null**,  firstName **varchar**(50) **not** **null**,  patronymic **varchar**(50) **not** **null**,  genderId **int** **references** Genders(idGender),  email **text** **not** **null**,  dateOfBirth **timestamp** **not** **null**,  countryId **int** **references** Country(id\_Country),  phone **varchar**(20) **not** **null**,  NapravlenieId **int** **references** Napravlenie(idNapravlenie),  **password** **text** **not** **null**,  foto **text** **not** **null**  )  **create** **table** Moderators(  moderId serial **primary** **key** **not** **null**,  lastName **varchar**(100) **not** **null**,  firstName **varchar**(50) **not** **null**,  patronymic **varchar**(50) **not** **null**,  genderId **int** **references** Genders(idGender),  email **text** **not** **null**,  dateOfBirth **timestamp** **not** **null**,  countryId **int** **references** Country(id\_Country),  phone **varchar**(20) **not** **null**,  NapravlenieId **int** **references** Napravlenie(idNapravlenie),  eventId **int** **references** Events(idEvent),  **password** **text** **not** **null**,  foto **text** **not** **null**  )  **create** **table** Meropr(  MeroprId serial **primary** **key** **not** **null**,  MeroprName **text**,  datePlan **timestamp** **not** **null**,  **day** **int** **not** **null**,  cityId **int** **references** City(id\_City)  )  **create** **table** **Action**(  ActionId serial **primary** **key** **not** **null**,  IdMeropr **int** **references** Meropr(MeroprId),  **day** **int** **not** **null**,  StartTime **time** **not** **null**,  IdModer **int** **references** Moderators(moderId)  )  **create** **table** JuryAndAction(  ActionId serial **primary** **key** **not** **null**,  JuryId1 **int** **references** Jury (juryId),  JuryId2 **int** **references** Jury (juryId),  JuryId3 **int** **references** Jury (juryId),  JuryId4 **int** **references** Jury (juryId),  JuryId5 **int** **references** Jury (juryId)  )  **create** **table** Winner (  WinnerId serial **primary** **key** **not** **null**,  IdMerop **int** **references** Meropr(MeroprId)  ) |

Листинг 2 – реализация dll-библиотеки

|  |
| --- |
| using System;  using System.Text.RegularExpressions;  public class RegMarkLib  {  private static readonly string validPattern = @"^[ABEKMHOPCTYX]\d{3}[ABEKMHOPCTYX]{2}\d{2,3}$";  private static readonly string letters = "ABEKMHOPCTYX";  private static readonly int minRegion = 1;  private static readonly int maxRegion = 199;  public static bool CheckMark(string mark)  {  if (!Regex.IsMatch(mark, validPattern)) return false;  int region = int.Parse(mark.Substring(mark.Length - 3));  return region >= minRegion && region <= maxRegion;  }  public static string GetNextMarkAfter(string mark)  {  if (!CheckMark(mark)) throw new ArgumentException("Invalid mark format");  char firstLetter = mark[0];  int number = int.Parse(mark.Substring(1, 3));  string lastTwoLetters = mark.Substring(4, 2);  int region = int.Parse(mark.Substring(mark.Length - 3));  number++;  if (number > 999)  {  number = 0;  lastTwoLetters = IncrementLetters(lastTwoLetters);  if (lastTwoLetters == "AA")  {  firstLetter = IncrementLetter(firstLetter);  }  }  return string.Format("{0}{1:D3}{2}{3:D3}", firstLetter, number, lastTwoLetters, region);  }  public static string GetNextMarkAfterInRange(string prevMark, string rangeStart, string rangeEnd)  {  string nextMark = GetNextMarkAfter(prevMark);  return (string.Compare(nextMark, rangeStart, StringComparison.Ordinal) >= 0 && string.Compare(nextMark, rangeEnd, StringComparison.Ordinal) <= 0)  ? nextMark : "out of stock";  }  public static int GetCombinationsCountInRange(string mark1, string mark2)  {  return Math.Max(0, string.Compare(mark2, mark1, StringComparison.Ordinal));  }  private static string IncrementLetters(string letters)  {  int index1 = RegMarkLib.letters.IndexOf(letters[0]);  int index2 = RegMarkLib.letters.IndexOf(letters[1]);  if (index2 < RegMarkLib.letters.Length - 1)  return letters[0].ToString() + RegMarkLib.letters[index2 + 1];  else if (index1 < RegMarkLib.letters.Length - 1)  return RegMarkLib.letters[index1 + 1] + "A";  else  return "AA"; // Reset if overflow  }  private static char IncrementLetter(char letter)  {  int index = RegMarkLib.letters.IndexOf(letter);  return index < RegMarkLib.letters.Length - 1 ? RegMarkLib.letters[index + 1] : 'A';  }  } |

Листинг 3 – юнит-тесты

|  |
| --- |
| namespace FactRegMark  {  public class RegMarkLibFacts  {  [Fact]  //проверка корректности введенного номера  public void CheckMark\_ValidMark\_ReturnsTrue()  {  Assert.True(RegMarkLib.CheckMark("A123BC123"));  }  [Fact]  //проверка на обработку неверно введенного номера  public void CheckMark\_InvalidMark\_ReturnsFalse()  {  Assert.False(RegMarkLib.CheckMark("Z999ZZ999"));  }  [Fact]  //проверка правильности создания нового номера  public void GetNextMarkAfter\_ValidMark\_ReturnsNextMark()  {  Assert.Equal("A124BC123", RegMarkLib.GetNextMarkAfter("A123BC123"));  }  [Fact]  //проверка метода создания номера в заданном диапазоне номеров при корректном вводе  public void GetNextMarkAfterInRange\_WithinRange\_ReturnsNextMark()  {  Assert.Equal("A124BC123", RegMarkLib.GetNextMarkAfterInRange("A123BC123", "A100BC123", "A200BC123"));  }  [Fact]  //проверка метода создания номера в заданном диапазоне номеров при некорректном вводе  public void GetNextMarkAfterInRange\_OutOfStock\_ReturnsMessage()  {  Assert.Equal("out of stock", RegMarkLib.GetNextMarkAfterInRange("A999BC123", "A100BC123", "A999BC123"));  }  [Fact]  //проверка метода, выводящего количество номеров в заданном диапазоне  public void GetCombinationsCountInRange\_ValidMarks\_ReturnsCount()  {  Assert.Equal(1, RegMarkLib.GetCombinationsCountInRange("A123BC123", "A124BC123"));  }  }  } |