**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc132402225)

[1 Требования к программному обеспечению 5](#_Toc132402226)

[2 Использование методов для получения кода с заданной функциональностью 10](#_Toc132402227)

[3 Отладка программного модуля с использованием специализированных программных средств 23](#_Toc132402228)

[4 Моделирование в программных системах 28](#_Toc132402229)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 32](#_Toc132402230)

[ПРИЛОЖЕНИЕ А 34](#_Toc132402231)

# ВВЕДЕНИЕ

Производственная практика пройдена в ЦУКС Главного управления МЧС по Саратовской области.

Производственная практика направлена на формирование у обучающихся общих и профессиональных компетенций, приобретение практического опыта и реализуется в рамках профессионального модуля ПМ.02 «Осуществление интеграции программных модулей».

**Планируемые результаты освоения практики**

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования.

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Осуществлять поиск, анализ и интерпретацию информации, необходимой для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Работать в коллективе и команде, эффективно взаимодействовать с коллегами, руководством, клиентами.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных общечеловеческих ценностей, применять стандарты антикоррупционного поведения.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, эффективно действовать в чрезвычайных ситуациях.

ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности.

ОК 09. Использовать информационные технологии в профессиональной деятельности.

ОК 10. Пользоваться профессиональной документацией на государственном и иностранном языках.

ОК 11. Использовать знания по финансовой грамотности, планировать предпринимательскую деятельность в профессиональной сфере.

В ходе освоения практики Обучающийся должен:

**иметь практический опыт в:**

* интеграции модулей в программное обеспечение;
* отладке программных модулей.

**уметь:**

* использовать выбранную систему контроля версий;
* использовать методы для получения кода с заданной функциональностью и степенью качества.

# 1 Требования к программному обеспечению

Данная система была разработана для управления МЧС по Саратовской области. Система позволит отслеживать сотрудников, погоду в городе Саратове, добавлять, изменять и удалять пользователей, звания и должности. А также создавать документы формата .docx. В приложении предусмотрена авторизация пользователей, которых добавил администратор, с целью создания более защищенной системы.

Программа соответвует следующим требованиям:

1. Система должна показывать дату входа и выхода пользователей в систему.
2. У администратора должна быть возможность добавлять, удалять, изменять данные в базе данных.
3. Программа должна быть защищена от ввода некорректных данных везде, где присутсвует возможность вводить информацию, при попытке ввода некорректных данных программа должна выводить ошибку.
4. Пользовательский интерфейс программы должен быть удобным и простым для понимания пользователя, а также для работы с приложением.

Для разработки системы будет использоваться интегрированная среда разработки от компании JetBrains – Rider 2023.3.2, для работы с СУБД будет использоваться хостинг RegRu и БД – MySQL. Основой проекта послужит WPF .NET Framework и библиотека для дизайна пользовательского интерфейса MaterialDesign. Данные средства разработки были выбраны, потому что выполняют все поставленные требования, предъявляемые на производственной практике.

График требований к системе можно представить в виде Контекстной диаграммы (Рисунок 1):

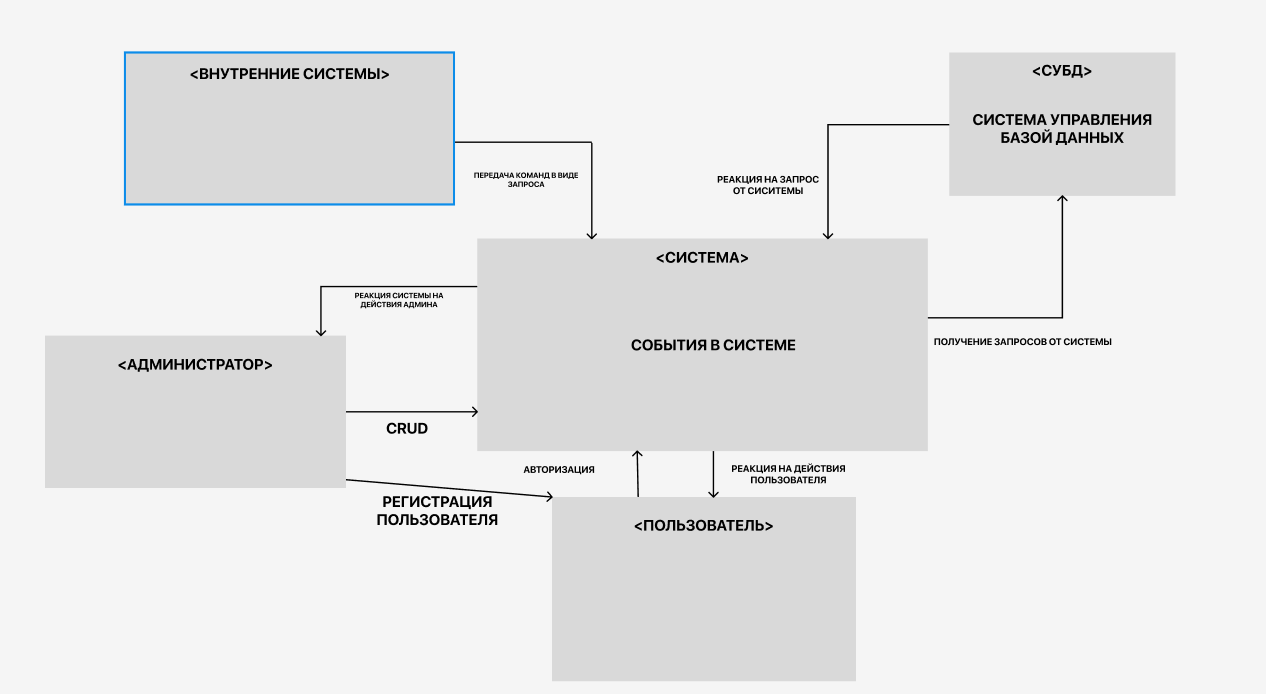


Рисунок 1 – Контекстная диаграмма

Далее будет представлена диаграмма декомпозиции (Рисунок 2). Диаграмма декомпозиции позволяет разбить функциональную модель, которая отображает структуру и функции системы, а также потоки информации и физических объектов, связывающих эти функции.

Суть самой декомпозиции заключается в последовательном разбиении функций до более простого уровня детализиции.

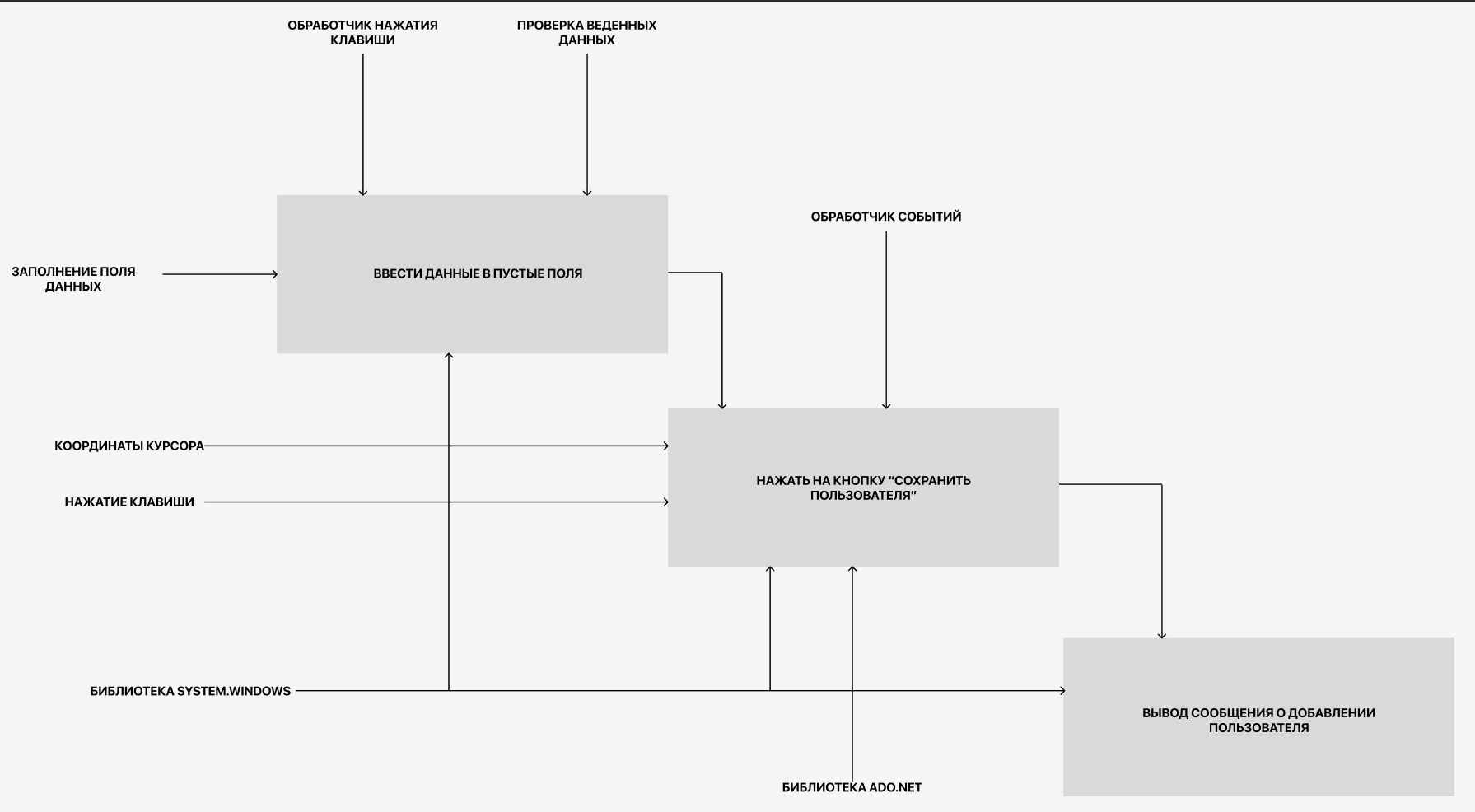


Рисунок 2 – Диаграмма декомпозиции на добавление пользователя

Составление диаграммы USE-CASE заключается в описании основных групп пользователей системы и варианты её использования.

Данная диаграмма представлена на (Рисунок 3).

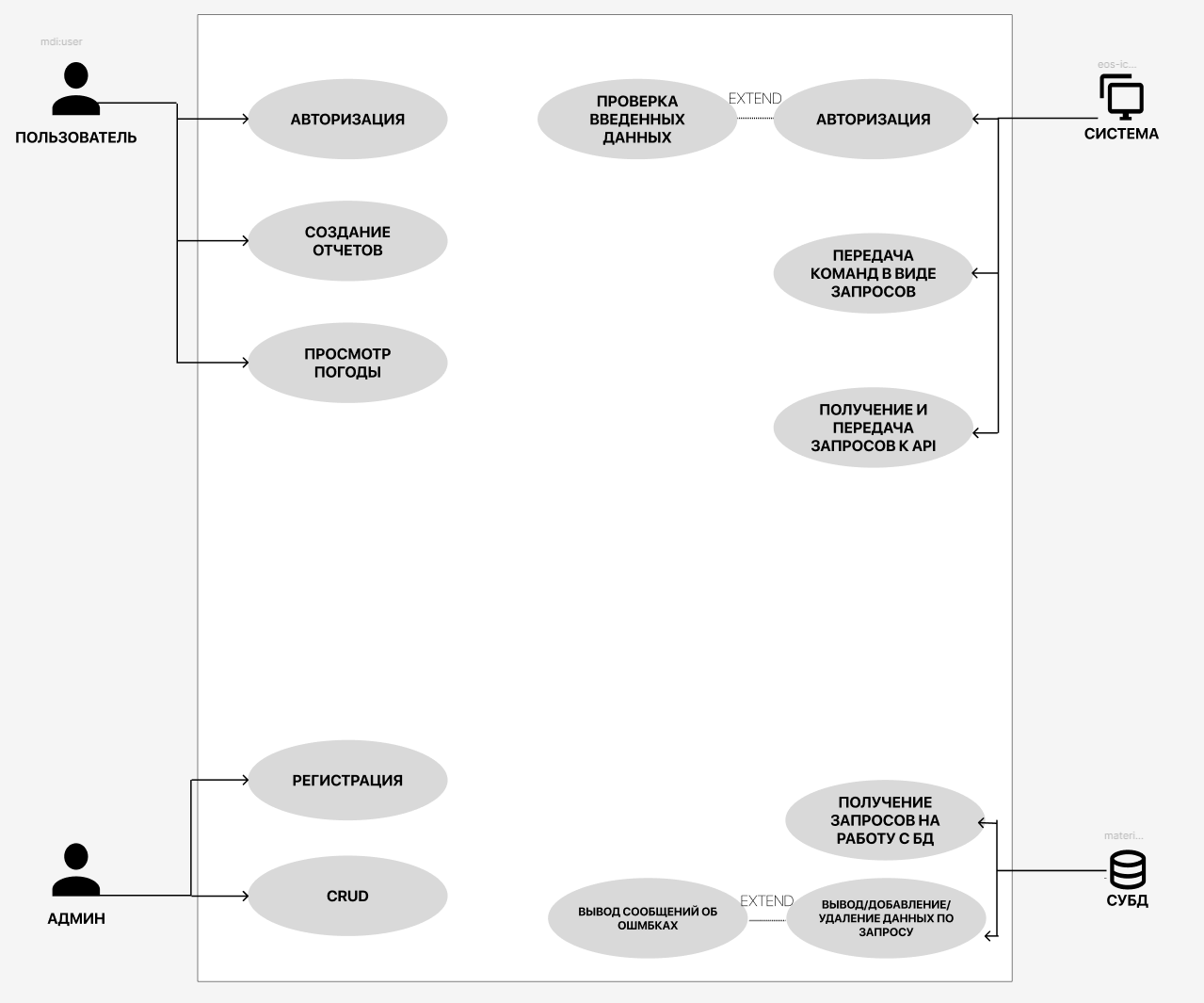


Рисунок 3 –Диаграмма USE-CASE

На данной диаграмме видно, что в данной предметной области есть несколько участников – Пользователь, Система, Админ, СУБД. Каждый из участников выполняет определенный прецедент. Так же на диаграмме присутвуют связи «extend». Данные связи расширяют определенные прецеденты.

Результат проектирования информационной системы можно представить в виде диаграммы классов (Рисунок 4).

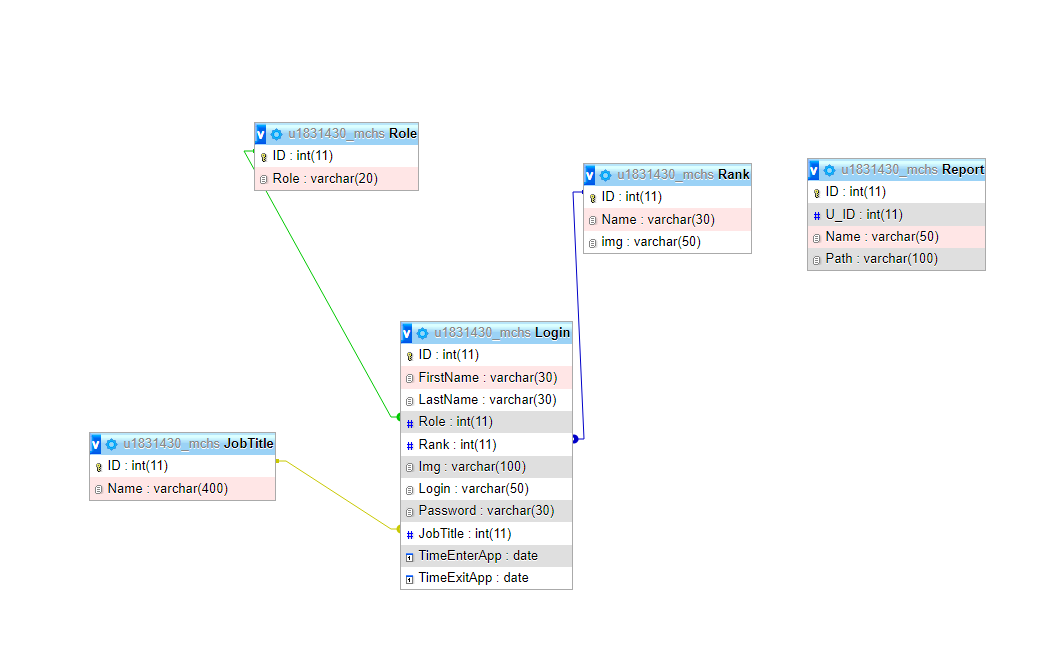


Рисунок 4 – Диаграмма классов

# 2 Использование методов для получения кода с заданной функциональностью

Для разработки системы будет использоваться IDE JetBrains Rider, а для работы с БД и её создания хостинг RegRu и СУБД MySQL.

Первым модулем системы является окно авторизации – данное окно первое, что увидит пользователь, запуская информационную систему. В данной информационной системе не предусмотрена регистрация пользователя в целях безопасности и ограничения информации от сторонних пользователей, не входящих в штат МЧС. Далее будет продемонстрирован дизайн и код логики окна авторизации (Рисунок 5-6).

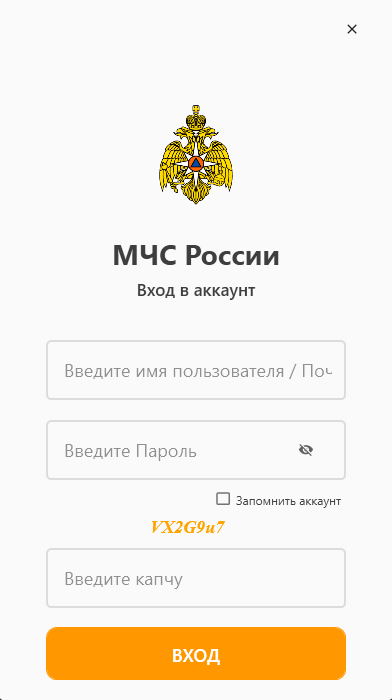


Рисунок 5 – Дизайн окна авторизации



Рисунок 6 – Код окна авторизации

Первоначально при запуске окна происходит генерация случайных букв и цифр для капчи. Следом идет событие нажатия на кнопку «Войти», где происходит проверка на заполненность полей, следом проверка на то ввел ли пользователь капчу, если пользователь ввел неправильно капчу, то кнопка блокируется на 10 секунд. Если пользователь ввел все данные правильно, система сверяет их из БД и основываясь на роли открывает окно «Пользователь» или «Администратор». В конце при вызове окон происходит фиксация времени входа в систему.

Далее будет рассмотрен экраны пользователя (Рисунок 7-12).

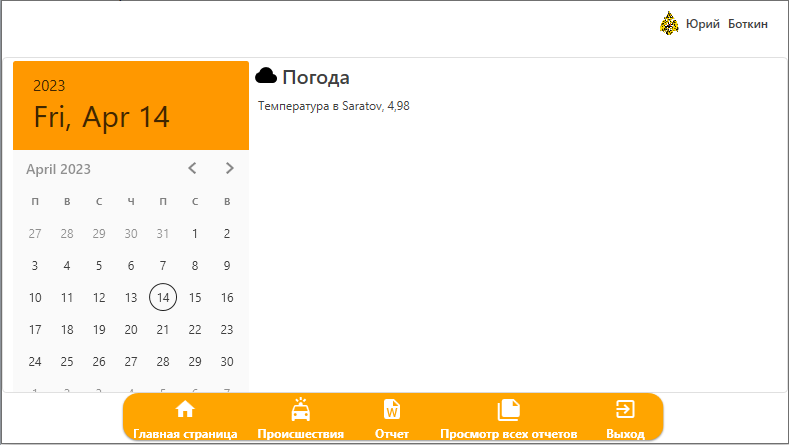


Рисунок 7 – Дизайн стартового экрана пользователя



Рисунок 8 – Код стартового экрана

Стартовый экран пользователя отображает текущую дату и погоду, а также позволяет переходить между другими экранами и заходить на страницу пользователя. В коде отображения погоды происходит следующее: первоначально создается uri сайта откуда берется API, следом идет запрос на получение данных с uri, после получаем и записываем в JSON ответ от сервера, в конце считываются данные, создается объект полученный из байт и заполняется данными Label.

Далее будет продемонстрирован дизайн и код экрана «Профиль»

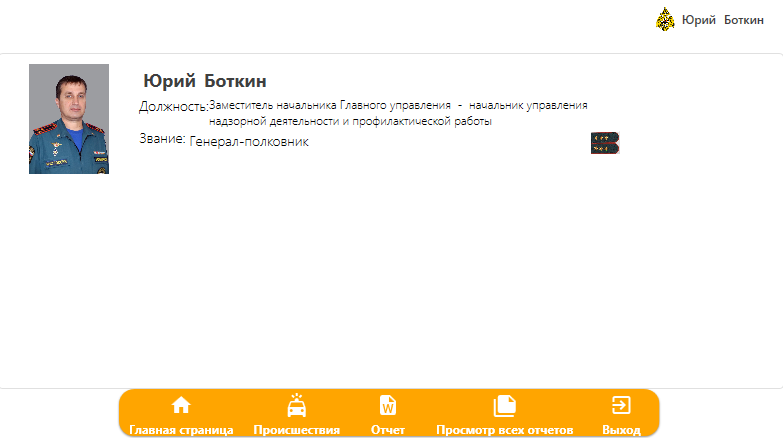


Рисунок 9 – Дизайн профиля пользователя



Рисунок 10 – Код экрана профиля

Первоначально создается модель, где будут записаны данные из БД. Происходит подключение к БД, пишется запрос на выборку данных из нескольких таблиц по идентификатору авторизованного пользователя. В конце с помощью цикла wile происходит запись данных в модель и дальнейшая передача этих данных в дизайн.

Последний экран у пользователя – это экран создания отчетов в формате docx.

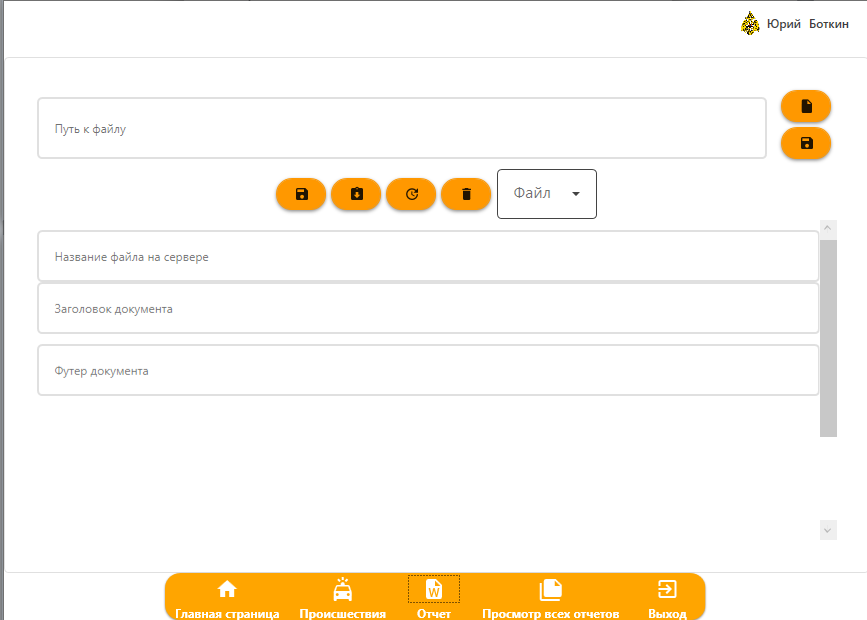


Рисунок 11 – Дизайн экрана создания отчета

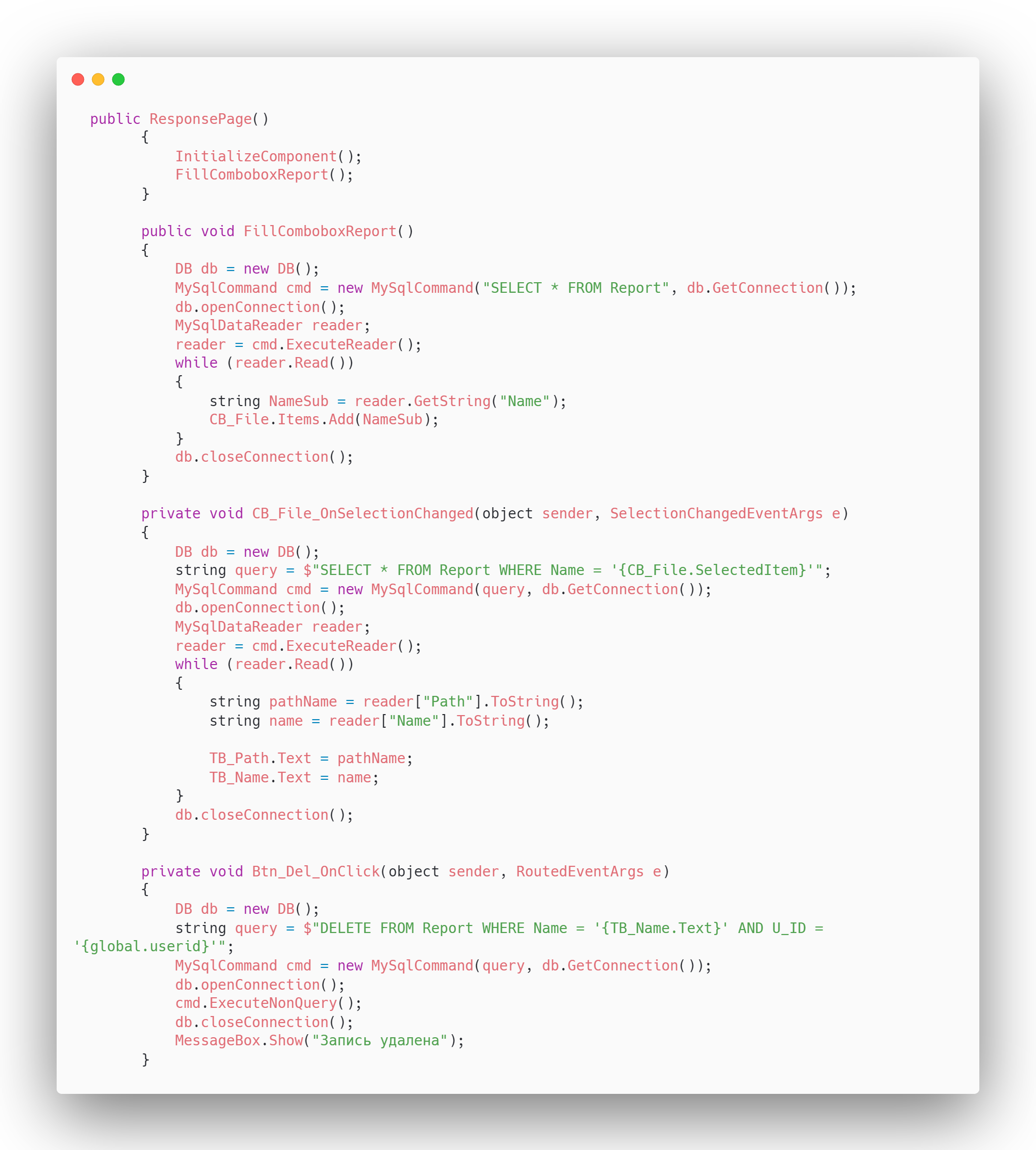
 

Рисунок 12 – Код экрана отчета

Первоначально происходит заполнение данными комбобокса из таблицы «Отчеты», после этого создается обработчик событий на изменение состояния ComboBox. В нем происходит выборка данных из таблицы «Отчеты» по выбранному элементу из выпадающего списка, следом идет создание команды для сервера, открывается подключение, выполняется команда, считываются данные из таблицы и ими заполняется TextBox, в конце закрывается подключение. Далее создается кнопка удаления и обновления данных в БД с выборкой по ID пользователя. Следом идет кнопка отображения данных из файла в TextBox.

Последняя сложная кнопка – это кнопка создания DOCX файла по шаблону с помощью библиотеки Spire.Doc. Первоначально выделяется память для создания документа, создаются секции разметки файла, настраивается заголовок файла, параграфы с настройкой шрифта и в конце создается футер для файла.

Далее будет рассмотрен один экран у администратора, так как все его экраны написаны по одному и тому же шаблону и различаются лишь количеством полей для ввода данных. Дизайн и код данного экрана представлен на (Рисунок 13-14).

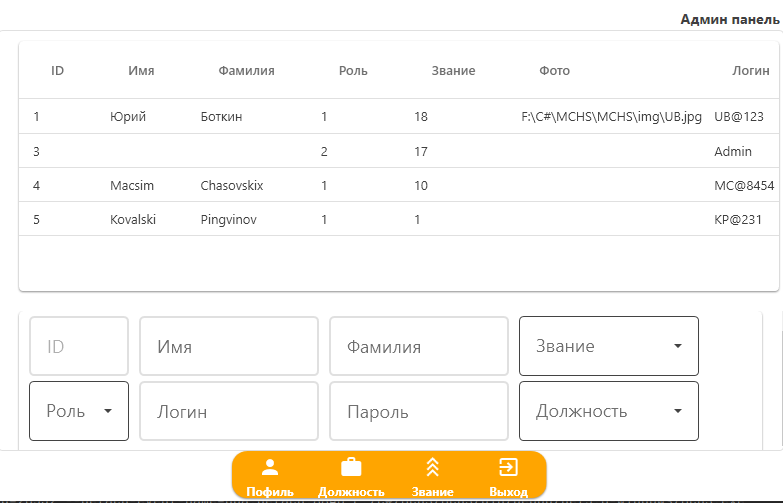


Рисунок 13 – Дизайн экрана добавления пользователя

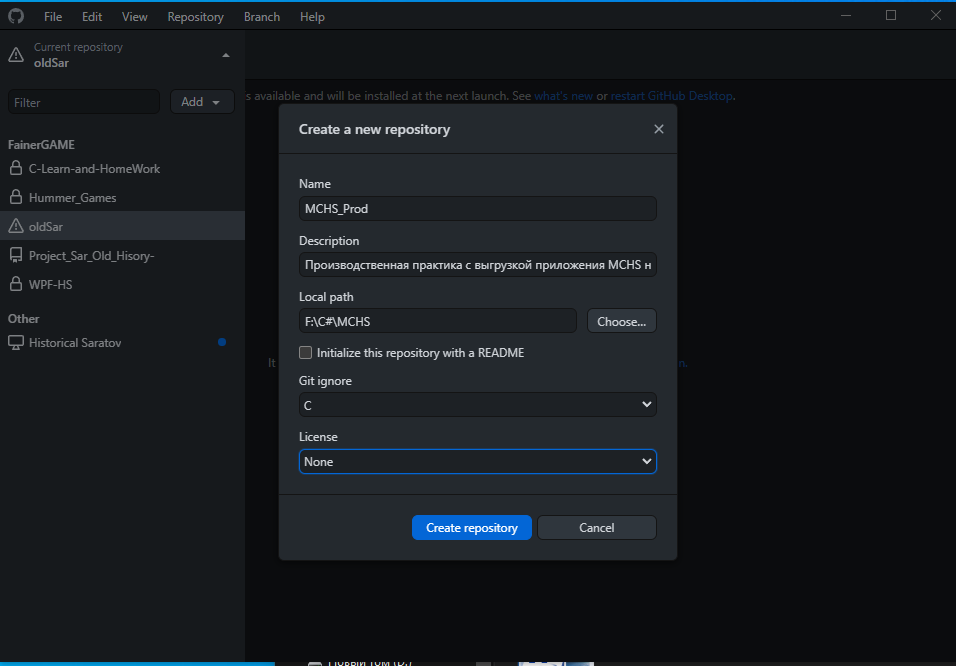
Рисунок 14 – Код экрана добавления пользователя

При запуске экрана из модели данных подгружается информация, находящаяся в таблице «Логин», следом заполняется ComboBox. Каждое заполнение ComboBox находится в своем методе, в котором происходит создание запроса к БД, подключение, выполнение считывания и записи данных и отключение от неё. Далее идет кнопка сохранения, в ней происходит проверка заполнения всех полей, если какое-то поле не заполнено выводится ошибка, если же пользователь ввел все правильно, то происходит подключение к БД, создание запроса на добавление данных с присвоением каждому атрибуту соответствующего поля, выполняется команда, закрывается соединение, выводится MessageBox об успешном сохранении и в конце обновляется модель для отображение новой информации в ListView.

Кнопка обновления работает аналогично как и сохранение, но только в запросе к БД сверяются данные ID в TextBox и таблице Login.

Кнопка удаления первоначально подключается к БД, создается запрос на удаление записи по ID, открывается подключение, выполняется запрос, закрывается подключение, выводится MessageBox об успешном удалении и в конце обновляется модель.

В конце данной темы будет продемонстрирована работа с системой контроля версий и загрузка проекта на GitHub (Рисунок 15 - 19).

 Рисунок 15 – Создание репозитория на GitHub

Для создания репозитория через GitHub Desktop, нужно:

1. Открыть приложение GitHub Desktop
2. Авторизироваться
3. Нажать на кнопку add
4. Выбрать Create New Repository
5. Дать имя, описание репозиторию
6. Выбрать путь к папке с проектом
7. Выбрать язык программирования
8. Нажать на клавишу Create Repository

После создания репозитория нужно опубликовать репозиторий, нажав на кнопку «Publish repository».

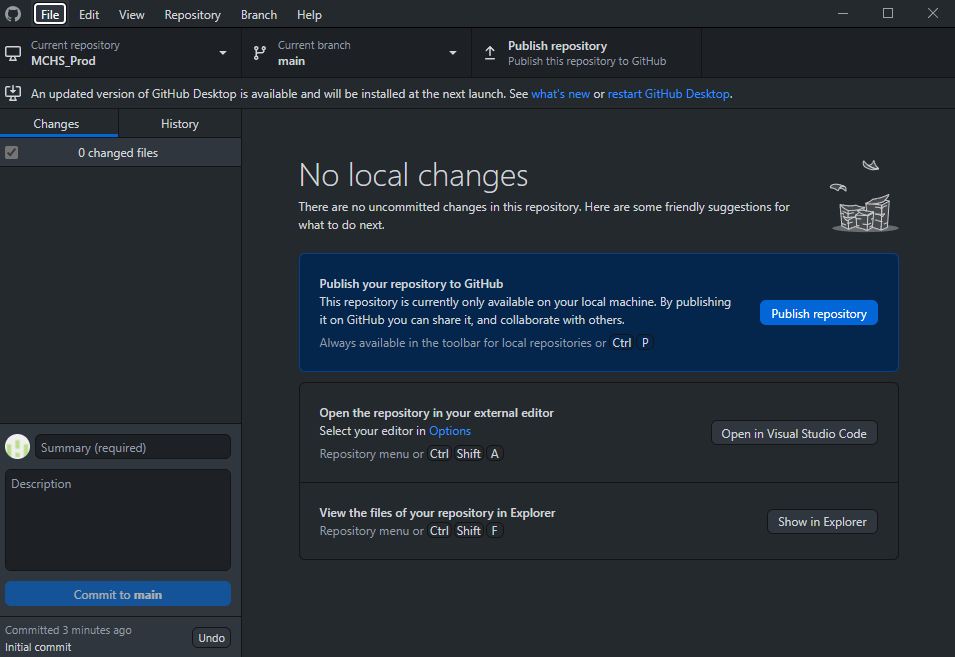


Рисунок 16 – Публикация репозитория

После того как создали репозиторий нужно зайти в папку данного репозитория на устройстве и скопировать в неё файлы проекта, следом переходим в GitHub Desktop, даем имя коммиту и описание, после этого нажимаем на кнопку «Commit to main».

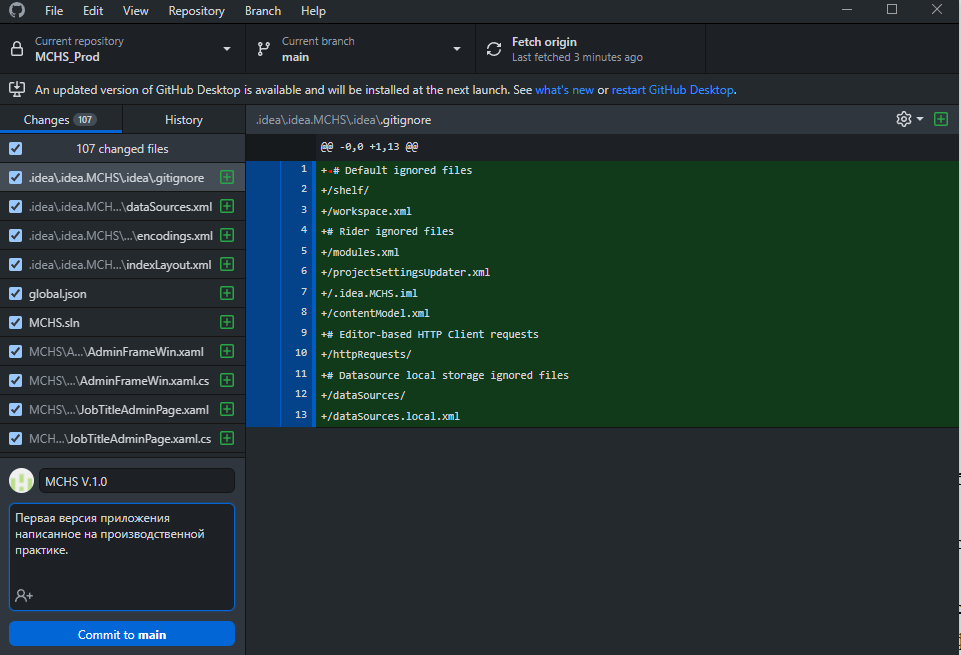


Рисунок 17 – Коммит проекта

В конце нужно нажать на клавишу «Push origin» для сохранения данных на серверах GitHub.

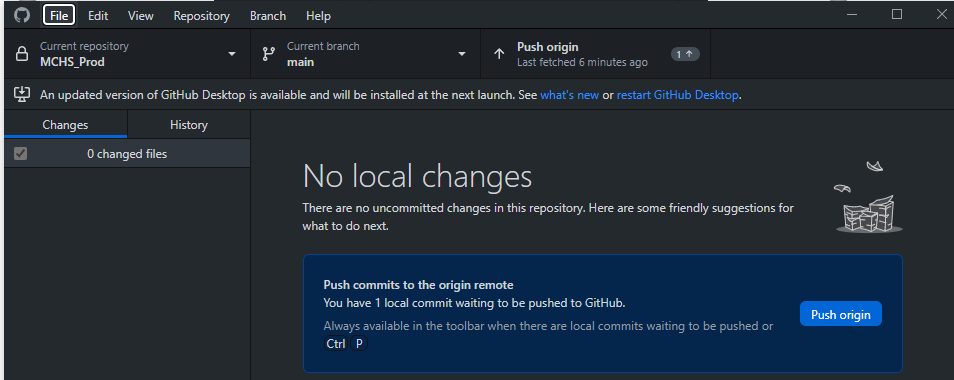


Рисунок 18 – Отправка данных на сервера GitHub



Рисунок 19 – Файлы на GitHub

# 3 Отладка программного модуля с использованием специализированных программных средств

При разработке информационной системы возникали ошибки, требующие исправления. Это были как логические ошибки, в результате которых система выдавала неверные значения или система не запускалась вовсе, так и синтаксические из-за которых система не могла передать значения или же выдавала неверную верстку.

Неверный SQL запрос, в следствии чего при обновлении базы данных обновляются все поля вместо нужного (Рисунок 20).



Рисунок 20 – Ошибка запроса

Для исправления данной ошибки нужно добавить WHERE с условием проверки ID пользователя с TextBox (Рисунок 21).

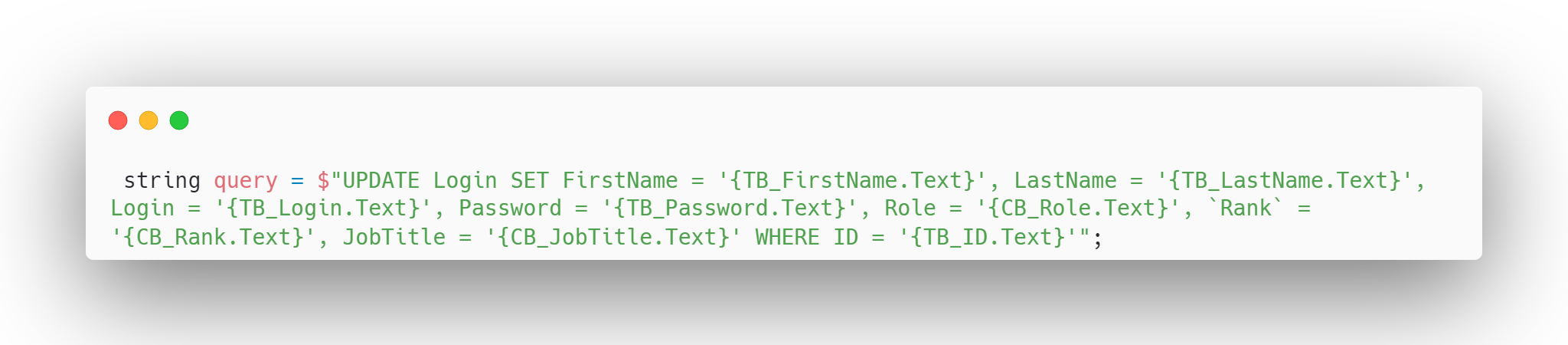


Рисунок 21 – Правильный запрос к БД

Неверный SQL запрос при котором приложение падало и не могло добавить новую запись (Рисунок 22).



Рисунок 22 – Ошибка запроса на добавление времени входа

Для исправления данной ошибки нужно заменить INSERT на UPDATE (Рисунок 23).



Рисунок 23 – Правильный запрос на добавление записи в Таблицу «Login»

Ошибка отображения верстки, некорректно отображается «Должность» и пропадает «Звание» (Рисунок 24).

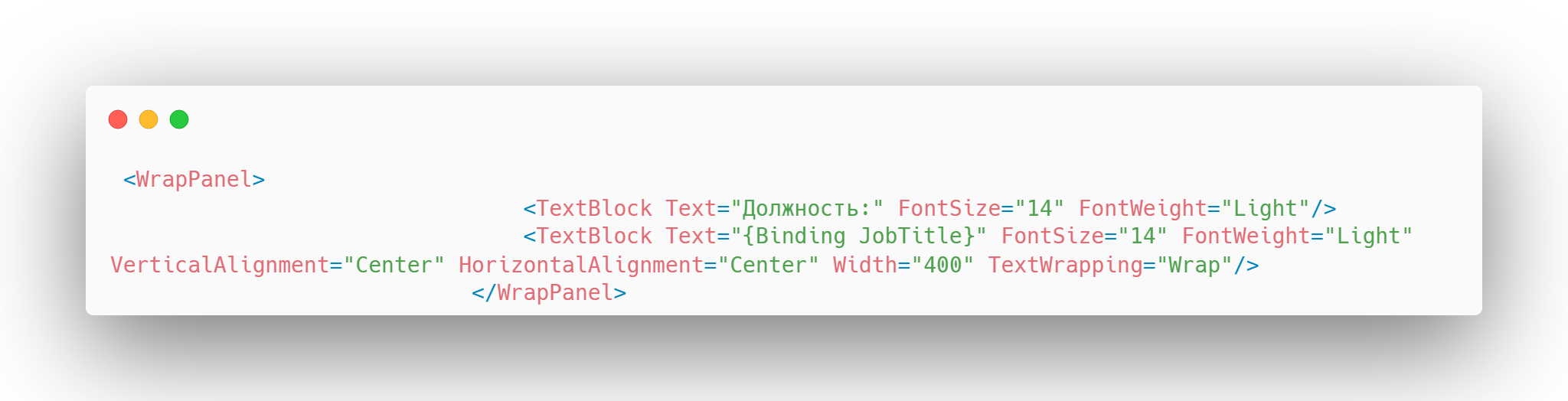


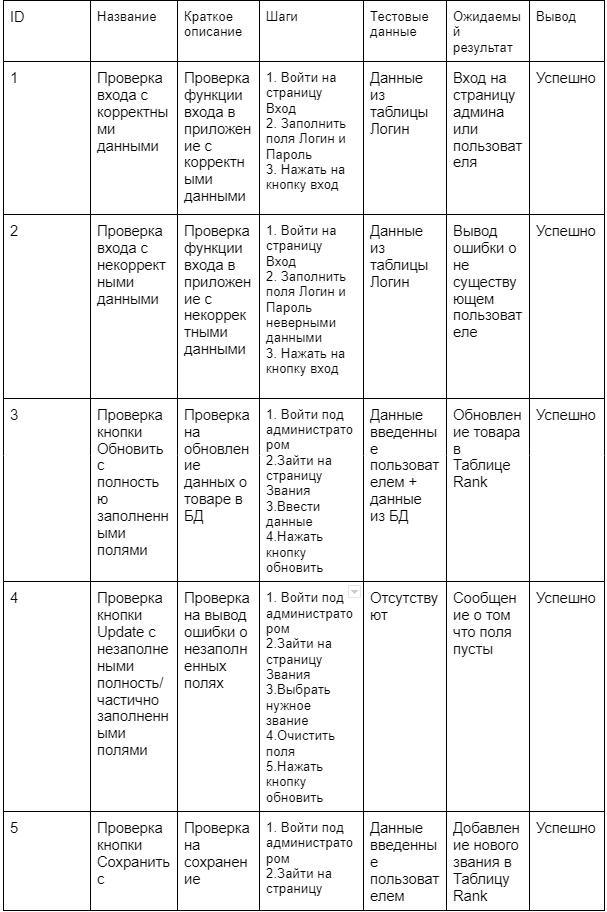
Рисунок 24 – Некорректная верстка

Для исправления данной ошибки нужно изменить размер текста с 14 на 12 (Рисунок 25).



Рисунок 25 – Верный XAML код

Для проверки работоспособности информационной системы был разработан пакет контрольных тестов (Рисунок 26).



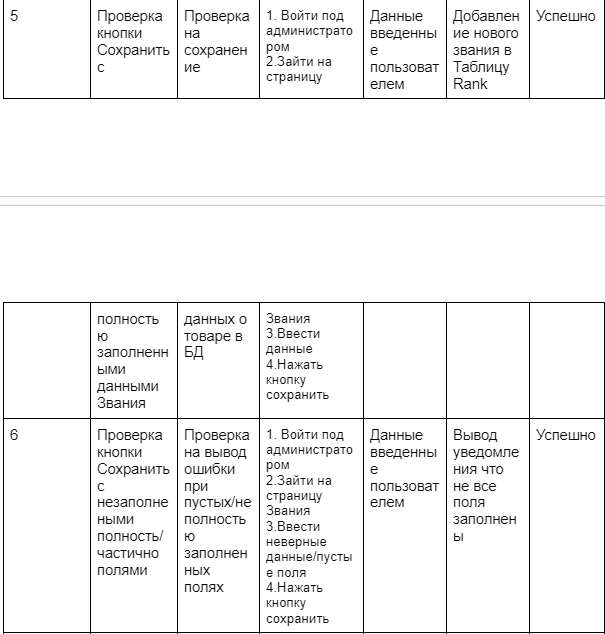


Рисунок 26 – Таблица проведенных тестов

# 4 Моделирование в программных системах

Стандарты кодирования являются важной частью разработки программного обеспечения на C#. Они помогают обеспечить единообразие и читаемость кода, улучшить его качество и облегчить его сопровождение. Проверка кода на соответствие стандартам кодирования является неотъемлемой частью процесса разработки программного обеспечения. В данном задании необходимо провести проверку кода на соответствие стандартам кодирования С# и составить перечень учтенных правил с примерами. Некоторые из правил, которые могут быть учтены при проверке кода на соответствие стандартам кодирования С#, включают в себя именование переменных, использование пробелов и отступов, использование комментариев, форматирование кода и т.д

Пример использования комментариев в C# (Рисунок 27).

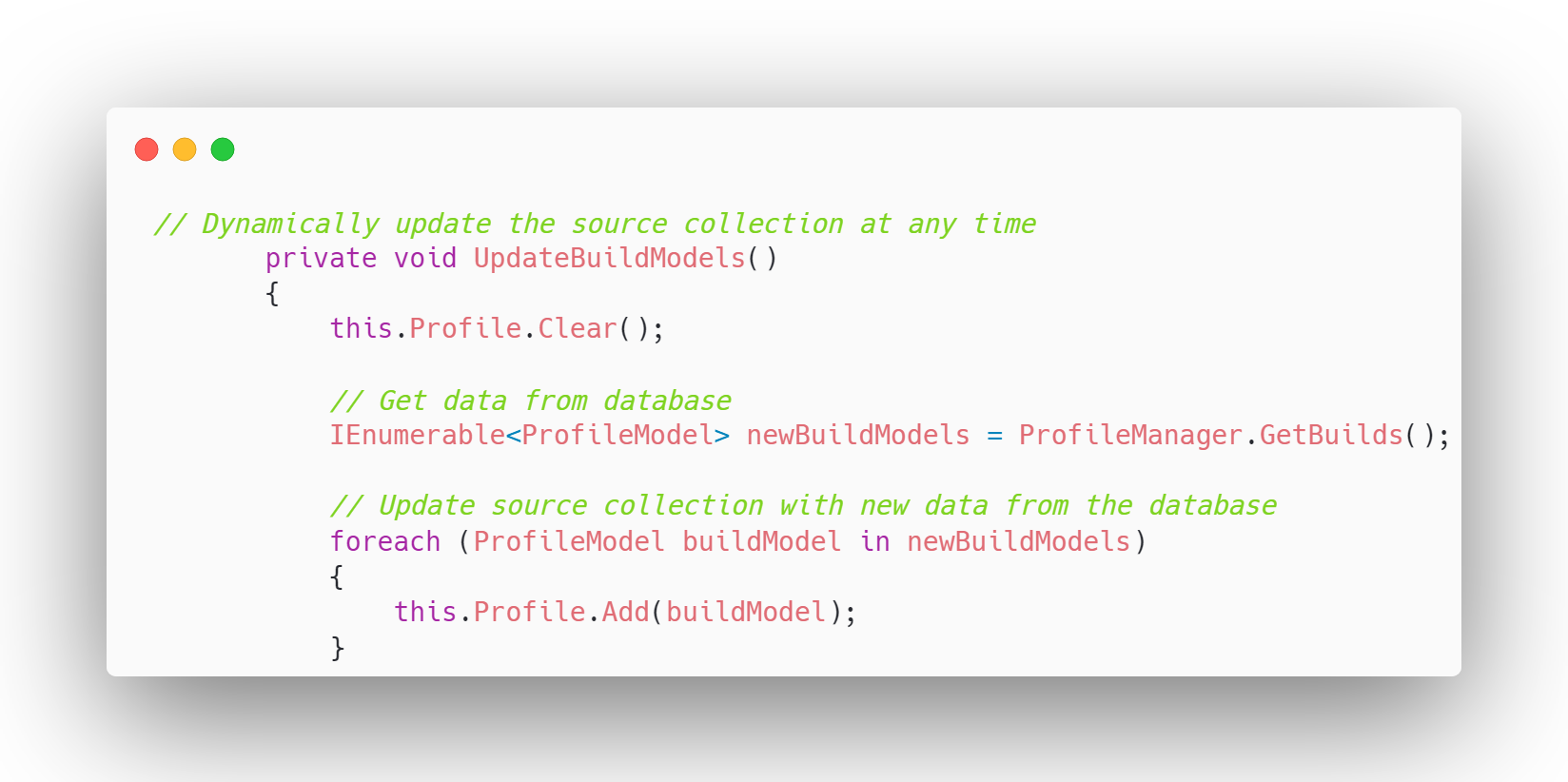


Рисунок 27 – Пример использования комментариев в C#

Пример наименования переменных в стиле PascalCase (CamelCase) (Рисунок 28).

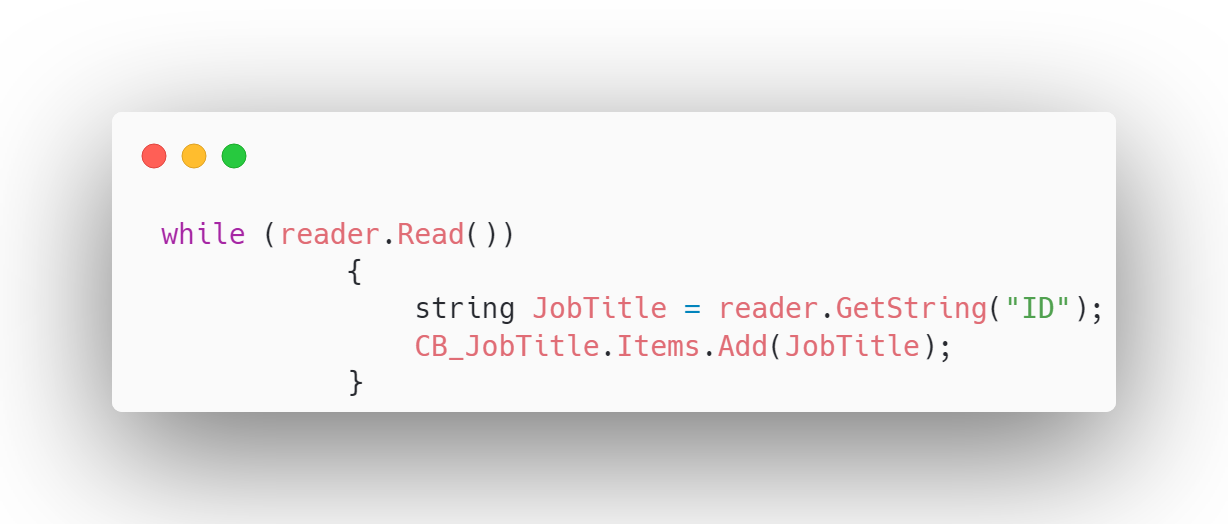


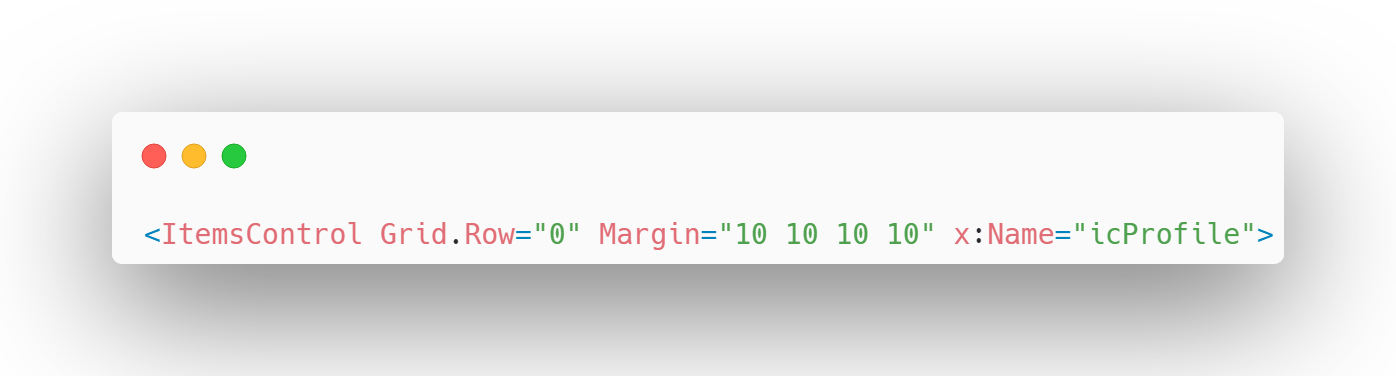
Рисунок 28 – Нотация PascalCase (CamelCase) в объявлении переменной

Пример правильных пробелов и отступов в C# (Рисунок 29).



Рисунок 29 – Пример стандартных отступов и пробелов в C#

Пример camelCase в названии элементов разметки XAML (Рисунок 30).

Рисунок 30 – Нотация camelCase в названии элементов разметки XAML

**ЗАКЛЮЧЕНИЕ**

В результате прохождения производственной практики цель практики была достигнута, все задачи выполнены в полном объеме.

В ходе прохождения производственной практики в Главном управлении МЧС России по Саратовской области были построены диаграммы предметной области, сформулированы назначение и цели создания системы, функциональные требования, требования к интерфейсу, безопасности и надежности. Были выделены и реализованы основные модули системы, создан интерфейс для приложения. Также были выявлены, описаны и устранены ошибки, возникшие в ходе разработки программы. Был разработан пакет контрольных тестов для проверки работоспособности функций программы и проведено тестирование кода.

В конце работы была проведена проверка кода на соответствие стандартам кодирования языка C#.

Эти действия позволили разработать и реализовать программу, которая соответствует требованиям и стандартам, а также работает корректно и без ошибок.

Прохождение производственной практики в Главном управлении МЧС России по Саратовской области дало возможность получить практический опыт в разработке программного обеспечения и применении современных технологий.

**СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ**

**Учебники/Учебные пособия**

1. Герберт Шилдт. C# 8.0. Полное руководство. [Текст]: Книга, написанная Гербертом Шилдтом, которая является полным руководством по языку программирования C# версии 8.0. Книга содержит подробное описание основных концепций языка C#, включая типы данных, операторы, выражения, управляющие конструкции, методы, классы, интерфейсы, делегаты, события, обработку исключений, многопоточность, LINQ и многое другое. М.: ДМК Пресс, 2019. - 864 с.
2. Джон Скит. C# 8.0. Карманный справочник. [Текст]: Книга, написанная Джоном Скитом, которая является кратким справочником по языку программирования C# версии 8.0. В этой книге автор предоставляет информацию о ключевых особенностях языка, включая новые возможности, которые были добавлены в версии 8.0. Книга содержит описание основных конструкций языка, таких как типы данных, операторы, управляющие конструкции, классы, интерфейсы, делегаты, события, лямбда-выражения и т.д. Кроме того, книга содержит примеры кода, которые помогут читателю лучше понять, как использовать язык C# для решения различных задач. М.: ДМК Пресс, 2019. - 240 с.
3. Федорова Г. Н. Разработка программных модулей программного обеспечения для компьютерных систем (3-е изд., стер.) учебник.- М.: Академия, 2020. – 384 с.

**Интернет ресурсы**

1. "draw.io" [Интернет-инструмент] // Свободный, кроссплатформеный сервис, с открытым исходным кодом для создания UML диаграмм, сетевых диаграмм и прочих. – Режим доступа: https://draw.io
2. «Документация по MaterialDesign» [Электронный справочник] // Официальная документация по MaterialDesign. – Режим доступа https://m3.material.io
3. «Документация по MySQl» [Электронный справочник] // Официальная документация по MySql. – Режим доступа: https://dev.mysql.com
4. «Документация по работе с phpMyAdmin» [Электронный справочник] // Статья о том как работать с сервисом phpMyAdmin. – Режим доступа: https://help.reg.ru/support/hosting/bazy-dannykh/rabota-v-phpmyadmin
5. «Документация по C#» [Электронный справочник] // Официальная документация по языку программирования C#. – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/csharp/
6. «Документация по API OpenWeather» [Электронный справочник] // Официальная документация по языку программирования C#. – Режим доступа: https://openweathermap.org/api

# ПРИЛОЖЕНИЕ А

**Характеристика объекта практики**

|  |  |
| --- | --- |
| **Название характеристики** | **Характеристика** |
| Название | ЦУКС Главного управления МЧС по Саратовской области |
| Год создания | 31.12.2011 |
| Юридический адрес | Саратовская область, г. Саратов, пл. Соборная, д. 7 |
| Отраслевая принадлежность | Обработка информации и оповещение людей и МЧС по ЧС |
| Основные этапы развития | В мае 2001 года комитет по делам ГО и ЧС при Правительстве Саратовской области был преобразован в министерство по делам ГО и ЧС Саратовской области. 1 января 2002 года государственная противопожарная служба вошла в состав МЧС Саратовской области. С 1 января 2005 года был создан территориальный орган МЧС России – Главное управление МЧС России по Саратовской области, призванный решать вопросы гражданской обороны, чрезвычайных ситуаций и ликвидации последствий стихийных бедствий. |
| Специализация | МЧС России |
| Организационно-правовая форма | Федеральное государственное казенное учреждение |
| Форма собственности | Государственная |
| Цели | Основной целью ЦУКС является устойчивое и непрерывное управление силами и средствами МЧС в ходе выполнения мероприятий по защите населения и территорий от чрезвычайных ситуаций, по обеспечению пожарной безопасности и безопасности людей на водных объектах, а также по гражданской обороне. |
| Задачи | Информирование органов управления и сил МЧС России, функциональных и территориальных подсистем РСЧС, ГО о ЧС мирного и военного времени на территории Саратовской области |
| Миссия | Деятельность по обеспечению безопасности в чрезвычайных ситуациях. ЦУКС осуществляет устойчивое и непрерывное управление силами и средствами МЧС в ходе выполнения мероприятий по защите населения и территорий от чрезвычайных ситуаций, по обеспечению пожарной безопасности и безопасности людей на водных объектах, а также по гражданской обороне. |