**ГАПОУ «Казанский авиационно-технический колледж имени**

**П.В. Дементьева»**

|  |
| --- |
| **ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  **к курсовому проекту** |
| **КП 09.02.04.20И1.21.000** |
| (обозначение документа)  **МДК.02.04 Программные решения для бизнеса** |
| Тема: Разработка информационной системы на основе СУБД по теме:  «Конкурс творческих профессий» |

2023

Содержание

[1 Анализ предметной области 5](#_Toc164972838)

[1.1 Описание области деятельности организации 5](#_Toc164972839)

[1.2 Анализ существующих аналогов ПО 5](#_Toc164972840)

[1.3 Определение требований к приложению. 7](#_Toc164972841)

[2 Проектирование приложения 8](#_Toc164972842)

[2.1 Определение сущностей и их характеристик 8](#_Toc164972843)

[2.2 Схема базы данных 8](#_Toc164972844)

[2.3 Словарь данных 9](#_Toc164972845)

[2.4 Сценарий работы приложения 10](#_Toc164972846)

[2.5 Дизайн интерфейса приложения 11](#_Toc164972847)

[2.6 Описание и формат входных данных. Надежность и достоверность данных 13](#_Toc164972848)

[2.7 Описание выходных данных. 14](#_Toc164972849)

[3. Разработка приложения 14](#_Toc164972850)

[3.1 Разработка БД 14](#_Toc164972851)

[3.2.1. Разработка стиля приложения 15](#_Toc164972852)

[3.2.2 Разработка подсистемы регистрации и авторизации 15](#_Toc164972853)

[3.2.3. Разработка подсистем приложения 16](#_Toc164972854)

[4. Тестирование 16](#_Toc164972855)

Введение

В эпоху информатизации, когда ремонт умных домофонов и выполнение гарантийных обязательств становятся критическими аспектами деятельности IT-компаний, задача эффективного управления и организации данных о сделках, клиентах и услугах приобретает особую значимость. Точная, структурированная и актуальная информация - залог гарантированного успеха и удовлетворения потребностей всех участников процесса ремонта умных домофонов.

Однако, IT-компании, специализирующиеся на производстве и ремонте умных домофонов, могут столкнуться с серьезными проблемами в области управления информацией, связанной с предоставлением этих услуг. Традиционные методы ведения данных и автоматизированные процессы могут привести к ошибкам, потере времени и, как результат, к недовольству клиентов.

Для решения указанных проблем и повышения эффективности работы отдела гарантии, целью курсовой работы является проектирование и разработка программного модуля «Отдел гарантий IT-компании» (АИС) для управления сделками по ремонту умного домофона. Данная система, является своего рода надстройкой над системой Битрикс 24 и станет мощным инструментом, способствующим улучшению процессов управления, минимизации ошибок и обеспечению высокого качества услуг.

Для достижения данной цели планируется выполнение следующих задач:

1. Анализ предметной области - детальное изучение и понимание специфики ремонта умных домофонов и особенностей гарантийного обслуживания.
2. Проектирование и разработка базы данных - создание структурированной и эффективной базы данных, являющейся основой для автоматизированной информационной системы.
3. Разработка настольного приложения - создание удобного и функционального приложения, которое будет интегрировано с базой данных и обеспечит удобный интерфейс для работы с информацией.
4. Тестирование
5. Разработка инструкций для работы с приложением

Важность данного проекта не может быть недооценена, учитывая растущую роль умных домофонов в повседневной жизни людей. Умные интеркомы

становятся неотъемлемой частью современных домов и офисов, обеспечивая безопасность и удобство для пользователей. Однако, как и любое сложное устройство, они подвержены поломкам и требуют профессионального обслуживания.

В свете этого, эффективное управление информацией о ремонте и гарантийном обслуживании умных домофонов становится ключевым фактором успеха для IT-компаний. Необходимость в автоматизированной информационной системе (АИС) для управления данными о сделках, клиентах и услугах становится очевидной.

АИС, разрабатываемая в рамках данного проекта, будет способствовать оптимизации рабочих процессов, уменьшению времени на обработку заявок и улучшению качества обслуживания клиентов. Она также поможет упростить процесс управления гарантийными обязательствами, уменьшив вероятность ошибок и улучшив управление статусами ремонта.

В дополнение к уже упомянутым задачам, в рамках проекта также планируется тестирование и внедрение системы. Это включает в себя проверку функциональности и производительности системы, а также ее адаптацию к конкретным условиям работы IT-компании.

В целом, данный проект имеет целью не только разработку АИС, но и внесение вклада в развитие отрасли ремонта умных домофонов, улучшение качества услуг и удовлетворенности клиентов.

# 1 Анализ предметной области

## Описание области деятельности организации

"ООО Спутник" является ведущим игроком на рынке информационных технологий, специализируясь на производстве, поддержке и восстановлении инновационных умных домофонов. Целью компании является интеграция и развитие передовых IT-решений, которые облегчают повседневную жизнь граждан, предоставляя им возможность осуществлять круглосуточный контроль и наблюдение за их жилищами.

"ООО Спутник" занимаемся производством умных домофонов различных классов, от доступных до премиум-сегмента, при этом накопив значительный опыт в работе с умными устройствами. Команда компании состоит из профессионалов в области IT, менеджмента, юридического сопровождения и технического обслуживания, каждый из которых вносит свой ценный вклад в создание готового уникального продукта.

Перед компанией стоят следующие задачи:

1. Улучшение качества наших продуктов: это включает в себя усовершенствование материалов корпуса, а также повышение качества комплектующих.
2. Развитие рекламной и маркетинговой стратегии: цель - привлечь к сотрудничеству крупные корпорации и индивидуальных потребителей.
3. Обеспечение качественного сервиса послепродажного обслуживания: компания предоставляет гарантийный ремонт, а также решает возникшие проблемы на расстоянии.

## Анализ существующих аналогов ПО

Среди существующих аналогов разрабатываемого приложения можно выделить такие платформы, как веб-сайт Битрикс-24, телеграмм-бот и мобильное приложение. Первые два варианта также позволяют изменять стадии сделки, однако это требует ввода номера панели вручную. В свою очередь, мобильная версия предлагает возможность сканирования NFC-метки панели с помощью смартфона, что значительно экономит время.

Для разработки приложения могут быть использованы различные платформы и технологии. В частности, можно использовать CRM-системы, которые предлагают широкий спектр инструментов для управления взаимоотношениями с клиентами. Также можно использовать технологии интеграции сервисов, такие как CSI (Crystal Service Integration).

Для создания коммерческого предложения могут быть выбраны различные ИТ-платформы и технологии. В частности, можно использовать платформы для создания диаграмм, такие как XMind, и технологии для моделирования процессов, такие как UML.

В процессе работы могут быть использованы различные технологии и методы. В частности, для управления проектами и задачами могут быть использованы методыуправления проектами, такие как Agile или Scrum.

В процессе моей работы я использую различные технологии и методы. Для управления проектами и задачами я могу использовать методы управления проектами, такие как Agile или Scrum. Также я могу использовать технологии для обработки данных, такие как Microsoft Office Excel. Кроме того, в моих средствах использования также присутствуют Visual Studio и Microsoft SQL Server для разработки и работы с базами данных.

Пример технических характеристик для приложения, разработанного для операционной системы Windows:

1. Операционная система: Приложение совместимо с Windows XP и более поздними версиями.
2. Процессор: процессор с частотой не менее 1 ГГц.
3. Оперативная память (RAM): Минимальное требование - 1 ГБ для 32-битной системы или 2 ГБ для 64-битной системы.
4. Жесткий диск: Минимальное свободное пространство на жестком диске - 4 ГБ
5. Видеокарта: DirectX 9 или более поздняя версия с драйвером WDDM 1.0.
6. Интернет-соединение: основные функции требуют подключение к интернету.

1.3 Определение требований к приложению.

Основная задача приложения - обеспечить возможность отслеживания и измени статуса ремонта домофонов. Пользователь вводит ID домофона в соответствующее поле и выбирает одну из четырех стадий: "Принята на склад", "Готово к отправке", "В ремонте", "В проверку". После этого отправляется WEB хук на удалённый сервер Битрикс-24 информация обновляется в системе, где заявка с указанным ID перемещается в соответствующую стадию или в случае когда заявка находится на стадии из которой нельзя перемещать возвращается сообщение с ошибкой. Все действия фиксируются в базе данных и логах.

Описание основного бизнес-процесса приложения:

1. Пользователь вводит ID домофона вручную.
2. Пользователь нажимает на одну из четырех кнопок со следующими названиями: "Принята на склад", "Готово к отправке", "В ремонте", "В проверку».
3. В систему Битрикс-24 отправляется WEB-хук для получения сделки на номер этой панели.
4. После запроса приходят данные в json формате и от туда извлекается номер последней сделки.
5. Формируется новый json, который отправляется в систему Битрикс-24, где заявка с указанным ID перемещается в соответствующую стадию.
6. При успешном выполнении команды пользователю выводится соответствующее сообщение.
7. При ошибке выполнения команды пользователю выводится сообщение об ошибке.
8. Все действия фиксируются в базе данных и логах.

Дополнительные функции приложения:

1. Просмотр истории изменений стадий
2. Фильтрация по дате и времени изменения
3. Фильтрация по сотруднику
4. Фильтрация по стадии

В будущем в приложении может быть добавлена возможность работать с заявкой: указывать выявленные дефекты и проведенные работы.

Определение пользователей и их функционал:

1. Оператор: Вводит ID домофона, выбирает стадию, просматривает сообщения об успешном выполнении команды или об ошибке управление логами.

Защита от несанкционированного доступа:

Доступ к приложению должен быть защищен с помощью аутентификации и авторизации. Пользователи должны вводить токен для доступа к функционалу приложения.

Диаграмма прецедентов:

Диаграмма прецедентов будет включать в себя следующие действия: ввод ID домофона, выбор стадии, просмотр сообщений, просмотр логов, управление доступом.

2 Проектирование приложения

2.1 Определение сущностей и их характеристик

2.2 Схема базы данных

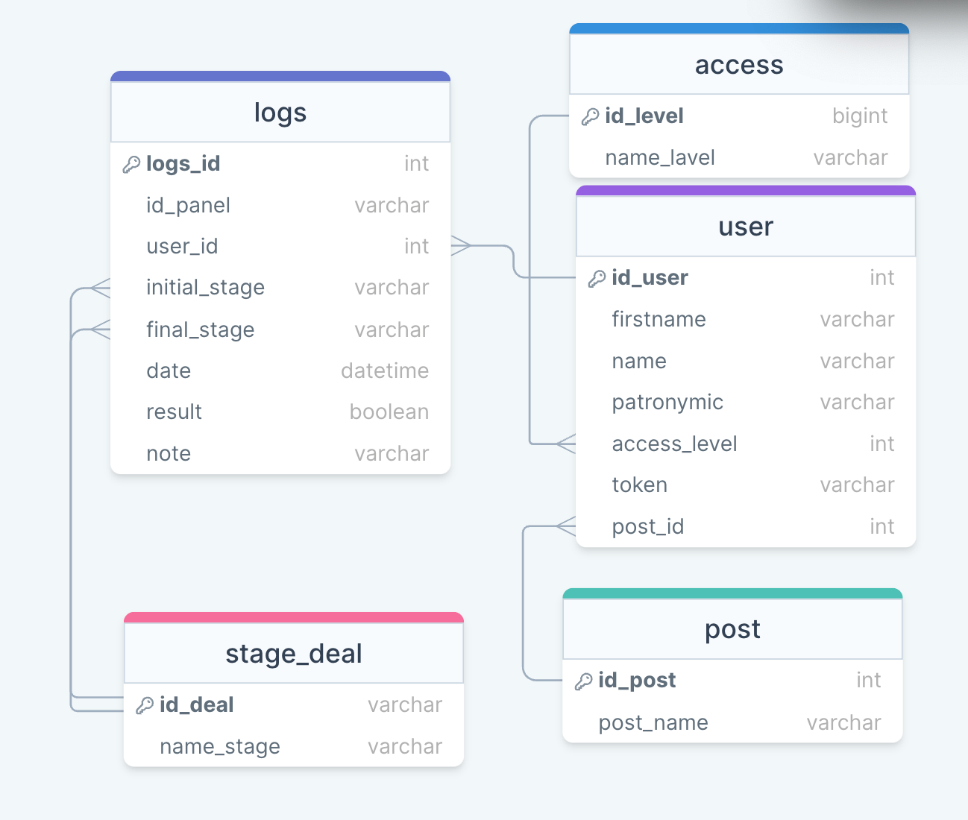


Рисунок 1 - Схема данных

## 2.3 Словарь данных

Таблица 1 - Словарь данных

|  |  |  |
| --- | --- | --- |
| logs | | |
| Поле | Тип данных | Примечание |
| logs\_id | int | PK |
| id\_panel | varchar(10) |  |
| user\_id | int | FK |
| initial\_stage | nvarchar(60) | FK |
| final\_stage | nvarchar(60) | FK |
| result | bit |  |
| note | nvarchar(MAX) |  |
| date | datetime |  |
| stage\_deal | | |
| Поле | Тип данных | Примечание |
| id\_deal | nvarchar(60) | FK |
| name\_stage | nvarchar(160) |  |
| access | | |
| Поле | Тип данных | Примечание |
| id\_level | int | PK |
| name\_lavel | nvarchar(60) |  |
| user | | |
| Поле | Тип данных | Примечание |
| id\_user | int | PK |
| firstname | nvarchar(100) |  |
| name | nvarchar(100) |  |
| patronymic | nvarchar(100) |  |
| access\_level | int | FK |
| token | nvarchar(200) |  |
| post\_id | int | FK |
| post | | |
| Поле | Тип данных | Примечание |
| id\_post | int | PK |
| post\_name | nvarchar(70) |  |

2.4 Сценарий работы приложения

1. Разработка алгоритма для изменения стадии сделки в системе Bitrix24 и ведения логов включает несколько важных шагов.
2. 1. Определение подсистем приложения: Приложение, которое разрабатывается для изменения стадии сделки и ведения логов, состоит из нескольких подсистем. Это подсистемы для отображения интерфейса пользователя, управления базой данных, управления WEB-хуками и ведения логов. Каждая из этих подсистем играет важную роль в работе приложения.
3. 2. Разработка алгоритма: Для успешной работы приложения необходимо разработать алгоритм, который будет определять последовательность действий для изменения стадии сделки и ведения логов. Этот алгоритм включает следующие шаги:
4. Получение последнего id сделки, которая заведена на введённый номер панели из системы Bitrix24.
5. Получение информации о текущей стадии сделки.
6. Изменение стадии сделки в соответствии с заданными правилами: переход на стадию «Принята на склад» если сделка находится в одном из следующих этапов: «Новая заявка» или «Проверка заявки» или «Заявка готова к отправке», в фазу «В ремонте» переходит если находится в стадии: «Принята на склад», «Отложено», «В ремонте», «Выходной контроль», «Готово к отправке», «Паркинг», в этап «Готов к отправке» можно перемещать если не находится в стадиях: «Заявка отменена», «Ошибка заведения», «Заявка завершена», в этап «В проверку» отправляется если на годится в фазе: «Принята на склад», «Отложено», «В ремонте», «Выходной контроль», «Готово к отправке», «Паркинг».
7. Запись действий в лог: Приложение ведёт логирование всех действий, выполняемых для изменения стадии сделки. Это помогает отслеживать историю изменений и обнаруживать возможные ошибки или проблемы.
8. Обновление информации о сделке в системе Bitrix24.

3. Разработка интерфейса: Важной частью разработки приложения является создание удобного и понятного интерфейса для пользователей. Интерфейс содержит такие элементы как: поле для ввода номера панели, выпадающие списки, всплывающий календарь, уточняющие пункты выбора для фильтрации истории взаимодействия.

2.5 Дизайн интерфейса приложения

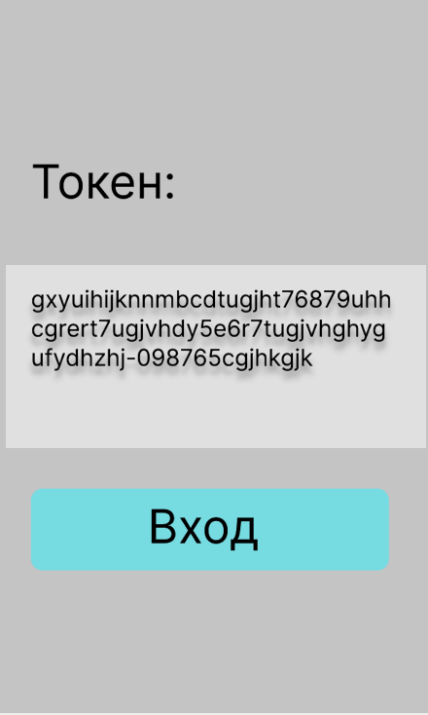


Рисунок 2 - Окно входа



Рисунок 3 - Главное окно

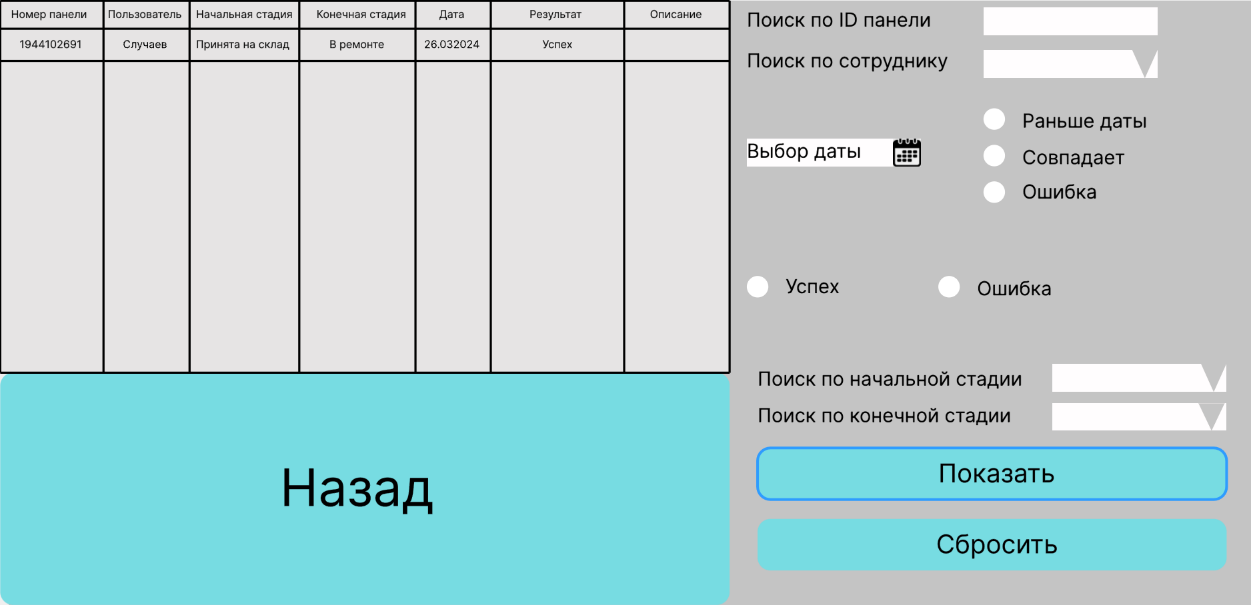


Рисунок 4 - Окно просмотра логов

## 2.6 Описание и формат входных данных. Надежность и достоверность данных

Для работы с моим приложением необходимо предоставить две составляющие входных данных.

Первая составляющая - это номер панели. Номер панели должен содержать 10 цифр без пробелов, букв и наличия специальных символов. Перед использованием номер панели проходит несколько шагов проверки. Сначала проверяется наличие символов в поле ввода, чтобы обеспечить гарантию того, что поле не осталось пустым. Затем проверяется длина номера панели, чтобы убедиться, что введено ровно 10 символов. После этого осуществляется проверка на наличие букв и пробелов в номере панели. Если в номере панели найдены буквы или пробелы, считается, что данные недостоверны и происходит вывод соответствующего сообщения.

Вторая составляющая - это файл в формате JSON, полученный от Битрикс-24 после отправки GET-запроса. Этот файл содержит дополнительные данные, необходимые приложению для выполнения задачи пользователя. Важно отметить, что достоверность и надежность данных, предоставленных пользователем после получения GET-запроса, уже были проверены и подтверждены системой Битрикс-24. Эта система считается автоматизированной и надежной, и поэтому шанс возникновения ошибки в предоставленных данных минимален.

## 2.7 Описание выходных данных.

Выходными данными работы приложения является json-файл отправляемый в систему Битрикс-24

# 3. Разработка приложения

## 3.1 Разработка БД

Открыл Microsoft SQL Server Management Studio (SSMS) и подключился к серверу баз данных, создал новую базу данных с именем courseproject. Затем, создал таблицу access с полями id\_level и name\_level, добавив столбцы id\_level (тип данных INT, PRIMARY KEY) и name\_level (тип данных NVARCHAR). После этого я создал таблицу logs с полями logs\_id, idpanel, userid, initialstage, finalstage, date, result, reason, добавив столбцы logs\_id (тип данных INT, PRIMARY KEY), id\_panel (тип данных INT, FOREIGN KEY связанный с id\_level в таблице access), user\_id (тип данных INT, FOREIGN KEY связанный с iduser в таблице user), initial\_stage (тип данных NVARCHAR), final\_stage (тип данных NVARCHAR), date (тип данных DATETIME), result (тип данных BIT), reason (тип данных NVARCHAR). Далее я создал таблицу stage\_deal с полями id\_deal и name\_stage, добавив столбцы id\_deal (тип данных NVARCHAR, PRIMARY KEY) и name\_stage (тип данных NVARCHAR). После этого я создал таблицу user с полями id\_user, firstname, name, patronymic, access\_level, token, post, добавив столбцы id\_user (тип данных INT, PRIMARY KEY), firstname (тип данных NVARCHAR), name (тип данных NVARCHAR), patronymic (тип данных NVARCHAR), access\_level (тип данных INT, FOREIGN KEY связанный с id\_level в таблице access), token (тип данных NVARCHAR), post (тип данных NVARCHAR). Затем я связал таблицы между собой, установив соответствующие ключи FOREIGN KEY и связи, создал необходимые индексы для оптимизации запросов, проверил работоспособность базы данных, выполнив несколько тестовых запросов и операций.

### 3.2.1. Разработка стиля приложения

Первое окно – это окно регистрации, состоит из текстового поля для ввода токена пользователя и кнопки для проверки соответствия введённых данных информации в БД.

Второе окно – это окно изменения стадии. Состоит из 6 кнопок и текстового поля ввода номера панели. Каждая кнопка имеет свою надпись, в которой заключён смысл её существования.

Третье окно – это окно просмотра логов, состоит из таблицы для отображения информации и различных элементов для поиска и фильтрации необходимых данных в БД.

В стиле приложения преобладают серые тона, на втором окне идёт секционная заливка цветом, уникальный цвет под каждый раздел. Во всех остальных случаях задний фон закрашен обычным серым, все кнопки имеют цвет морской волны, который гармонично вписывается в общий дизайн приложения.

### 3.2.2 Разработка подсистемы регистрации и авторизации

Окно регистрации в системе (Window2), представленное на рисунке 2, разработано в соответствии со стилем приложения. Для создания окна, необходимо нажать ПКМ по проекту, выбрать "Добавить", "Окно" (см. рисунок 5). В всплывающем окне снизу указать название окна, нажать "Добавить" В файл Window2.xaml, расположенный в обозревателе решений, необходимо скопировать xaml код из приложения Б, а в Window2.xaml.cs C# код.

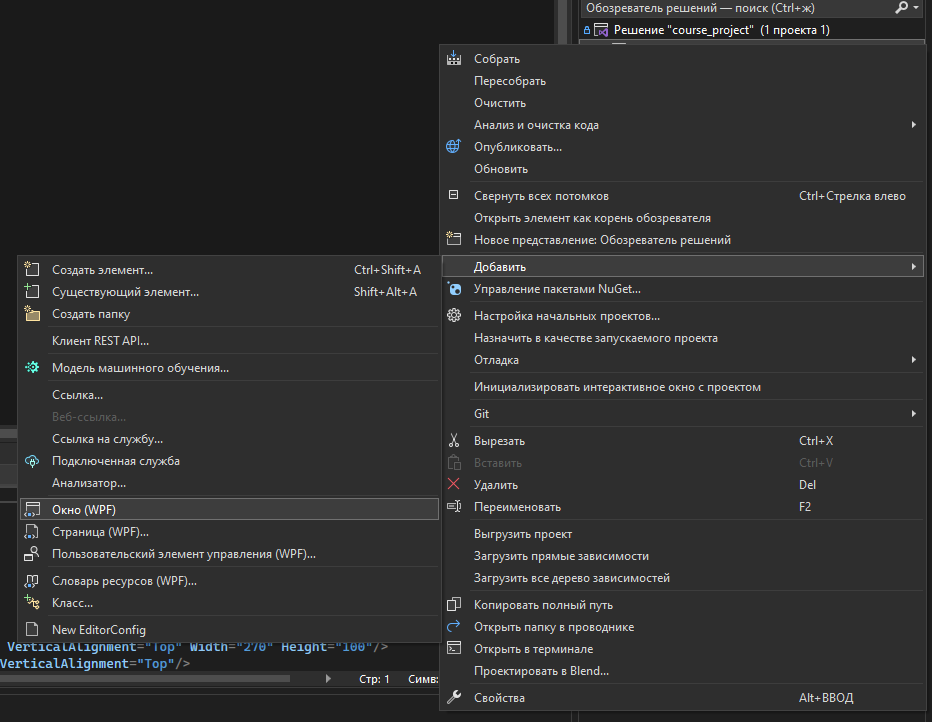


Рисунок 5 - Добавление окна регистрации

3.2.3. Разработка подсистем приложения

Пространство предназначенное для работы пользователя это главное окно и просмотра логов, были созданы согласно стилю приложения, содержат элементы, указанные в пункте 3.2.1. При разработке были добавлены таким же образом что и окно авторизации.

# 4. Тестирование