

Государственное автономное профессиональное образовательное учреждение
«Казанский авиационно-технический колледж имени П.В. Дементьева»

КУРСОВОЙ ПРОЕКТ
КП 09.02.07.21И1.17.000

(обозначение документа)

МДК.01.01 Разработка программных модулей

Тема: Разработка программного модуля: «Отдел гарантии IT компании»

Разработал

Случаев М.К.

(личная
подпись)

(инициалы, фамилия)

(дата)

Руководитель

Трифонова Э.М.

(личная
подпись)

(инициалы, фамилия)

(дата)

Заведующий
отделением

Цыбина Е.А.

(личная
подпись)

(инициалы, фамилия)

(дата)

2024

Государственное автономное профессиональное образовательное учреждение
«Казанский авиационно-технический колледж имени П.В. Дементьева»

Заместитель директора по учебной работе

_____.Э.Р.Соколова

_____2024

ЗАДАНИЕ

на курсовой проект

КП 09.02.07.21И1.17.000 3

(обозначение документа)

МДК.01.01 Разработка программных модулей

студенту

Случаеву Максиму Константиновичу

(фамилия, имя, отчество)

группы

21И1

(индекс группы)

Специальность:

«Информационные системы и программирование

базовый уровень

(наименование специальности, уровень СПО)

Тема:

Разработка программного модуля: «Отдел гарантии IT компании»

ОДОБРЕНО

Цикловой комиссией

Информационных технологий

Дата выдачи задания

01.02.2024

(наименование комиссии)

Дата окончания работы

13.05.2024

Протокол № 6 от 31.01 2024 года

Председатель

Сафиулина Л.М.

(личная подпись)

(инициалы, фамилия)

(дата)

Руководитель

Трифопова Э.М..

(личная подпись)

(инициалы, фамилия)

(дата)

2024

Исходные данные: список пользователей, номер панели и следующая стадия

1 Графические приложения

1.1 Графические приложения, перечень которых приведен ниже, должны быть выполнены в соответствии с требованиями стандартов Единой системы конструкторской документации (ЕСКД).

1 Схема данных

Текстовые документы

2.1 Комплект текстовых документов должен представлять:

- титульный лист;
- задание на курсовой проект;
- ведомость на курсовой проект;
- пояснительная записка к курсовому проекту.

2.2 Пояснительная записка должна быть оформлена по

ГОСТ 2.105-95 «ЕСКД. Общие требования к текстовым документам».

					КП 09.02.07.21И1.17.000 3			
Изм.	Лист	№ докум.	Подпись	Дат				
Разраб.		Случаев М. К.			Задание к разработке КП	Лит.	Лист	Листов
Провер.		Трифорова Э. М.				у		2 4
Н. контр.								
Утв.						КАТК зр. 21И1		

2.3 Пояснительная записка включает:

Содержание

Введение.....	3
1 Анализ предметной области	6
1.1 Описание области деятельности организации	6
1.2 Анализ существующих аналогов ПО	6
1.3 Определение требований к приложению.	8
2 Проектирование приложения.....	10
2.1. Определение сущностей и их характеристик	10
2.2 Схема базы данных.....	11
2.3 Словарь данных	12
2.4 Сценарий работы приложения	13
2.5 Дизайн интерфейса приложения.....	14
2.6 Описание и формат входных данных. Надежность и достоверность данных.....	15
2.7 Описание выходных данных.	16
3. Разработка приложения	10
3.1 Разработка БД	10
3.2.1. Разработка стиля приложения.....	11
3.2.2 Разработка подсистемы регистрации и авторизации.....	11
3.2.3. Разработка подсистем приложения	12
Тестирование	19
4.1 Unit - тестирование	19
4.2 Тестирование основных подсистем приложения.....	19
5. Инструкции по работе с приложением	22
5.1 Руководство пользователя	22
5.2. Руководство администратора БД.....	24
Заключение	25
Список литературы	26
Приложения	27

Приложение А.....	27
Приложение Б	30
Приложение В.....	33
Приложение Г	36
Приложение Д.....	40
Приложение Ж.....	42
Приложение З.....	45
Приложение И.....	45
Приложение К.....	47

					КП 09.02.07.21И1.17.000 З	Лист
Изм.	Лист	№. докум.	Подпись	Дата		4

№	Форма	Обозначение	Наименование	Кол. листов	№ экз.	Примечание
1			<u>Графическая документация</u>			
2	A3	КП 09.02.04.21И1.18.000	Схема данных	1	1	
3				1	1	
4						
5			<u>Текстовая документация</u>			
6	A4	КП 09.02.04.21И1.17.000 З	Задание	4	1	
7	A4	КП 09.02.04.21И1.17.000 ВД	Ведомость	1	1	
8	A4	КП 09.02.04.21И1.17.000 ПЗ	Пояснительная записка	3	1	
9			<u>Электронные носители</u>			
10		КП 09.02.04.21И1.17.000	Программный продукт на CD		1	
11						
12						
13						
14						
15						
16						
17						
18						
19						
20						
21						
22						
23						
24						
25						
26						
27						

					КП 09.02.04.21И1.17.000 ВД		
Изм.	Лист	№ документа.	Подпись	Дата			
Разраб.		Случаев М. К.			Ведомость КП		
Провер.		Трифонов Э. М.					
Н. Контр.							
Утверд.							
					Литера	Лист	Листов
						1	
					КАТК гр.21И1		

**Государственное автономное профессиональное образовательное учреждение
«Казанский авиационно-технический колледж имени П.В. Дементьева»**

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к курсовому проекту

КП 09.02.07.21И1.17.000 ПЗ

(обозначение документа)

МДК.01.01 Разработка программных модулей

Тема: _____

2024 год

Содержание

Введение	3
1 Анализ предметной области	6
1.1 Описание области деятельности организации	6
1.2 Анализ существующих аналогов ПО	6
1.3 Определение требований к приложению.	8
2 Проектирование приложения	10
2.1. Определение сущностей и их характеристик.....	10
2.2 Схема базы данных	11
2.3 Словарь данных.....	12
2.4 Сценарий работы приложения.....	13
2.5 Дизайн интерфейса приложения	14
2.6 Описание и формат входных данных. Надежность и достоверность данных	15
2.7 Описание выходных данных.....	16
3. Разработка приложения	10
3.1 Разработка БД.....	10
3.2.1. Разработка стиля приложения	11
3.2.2 Разработка подсистемы регистрации и авторизации	11
3.2.3. Разработка подсистем приложения.....	12
Тестирование.....	19
4.1 Unit - тестирование	19
4.2 Тестирование основных подсистем приложения.	19
5. Инструкции по работе с приложением.....	22
5.1 Руководство пользователя.....	22
5.2. Руководство администратора БД	24
Заключение.....	25

					КП 09.02.07.21И1.17.000 ПЗ								
Изм.	Лист	№ докум.	Подпись	Дата	ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КП				Лит.	Лист	Листов		
Разраб.		Случаев М. К..							✓		2	47	
Провер.		Трифопова Э.М.							КАТК гр.21И1				
Н. контр.													
Утв.													

Список литературы.....	26
Приложения.....	27
Приложение А	27
Приложение Б.....	30
Приложение В	33
Приложение Г	36
Приложение Д	40
Приложение Ж.....	42
Приложение З	45
Приложение И	45
Приложение К	47

					КП 09.02.07.21И1.17.000 ПЗ				
Изм.	Лист	№ докум.	Подпись	Дата	ПОЯСНИТЕЛЬНАЯ ЗАПИСКА КП	Лит.	Лист	Листов	
Разраб.		Случаев М. К..				✓		2	47
Провер.		Трифопова Э.М.							
Н. контр.									
Утв.						КАТК зр.21И1			

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-компаний» (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:

- Анализ предметной области - детальное изучение и понимание специфики ремонта умных домофонов и особенностей гарантийного обслуживания.
- Проектирование и разработка базы данных - создание структурированной и эффективной базы данных, являющейся основой для автоматизированной информационной системы.

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		3

- Разработка настольного приложения - создание удобного и функционального приложения, которое будет интегрировано с базой данных и обеспечит удобный интерфейс для работы с информацией.
- Тестирование
- 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-компаний.

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.

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		4

1 Анализ предметной области

1.1 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" занимается 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:

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

1.2 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.

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		6

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 методы управления 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:

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

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		7

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:

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

Дополнительные функции приложения:

- Просмотр истории изменений стадий

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		8

- Фильтрация по дате и времени изменения
- Фильтрация по сотруднику
- 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:

- Мастер: вводит ID домофона, выбирает стадию, просматривает сообщения об успешном выполнении команды или об ошибке управление логами.
- Администратор: 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: для мастера ввод ID домофона, смена стадии сделки, просмотр логов, для администратора просмотр логов и удаление записей.



Рисунок 1 - Диаграмма прецедентов

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

2 Проектирование приложения

2.1. Определение сущностей и их характеристик

Были определены 2 сущности и также построена ER диаграмма

Сущности:

1. Сотрудник

Сотрудник будет иметь следующие поля:

- Идентификатор
- ФИО
- Уровень доступа
- Токен
- Должность

2. Логи

Таблица Логи будет содержать следующие поля:

- Идентификатор
- Номер панели
- ID сотрудника
- Начальную стадию
- Конечную стадию
- Дату создания
- Результат
- Описание

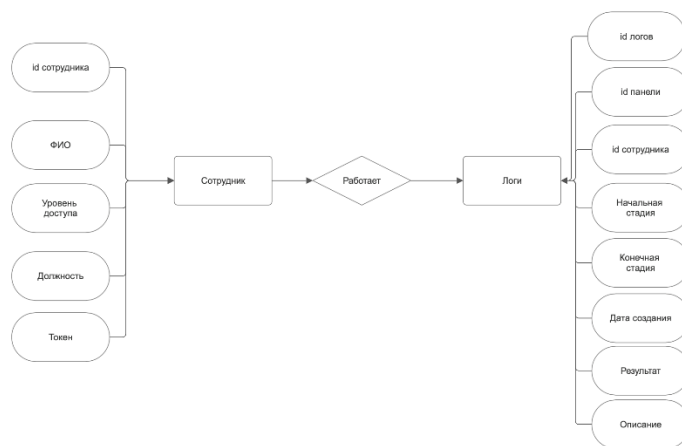


Рисунок 2 - ER модель

2.2 Схема базы данных

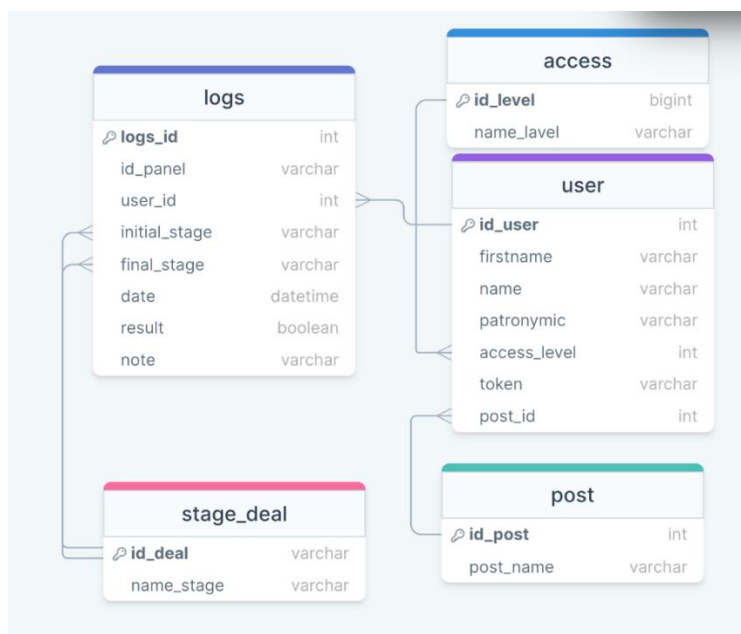


Рисунок 3 - Схема данных

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:

- Получение последнего id сделки, которая заведена на введённый номер панели из системы Bitrix24.
- Получение информации о текущей стадии сделки.
- 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 находится в фазе: «Принята на склад», «Отложено», «В ремонте», «Выходной контроль», «Готово к отправке», «Паркинг».
- 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.
- Обновление информации о сделке в системе Bitrix24.

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

13

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

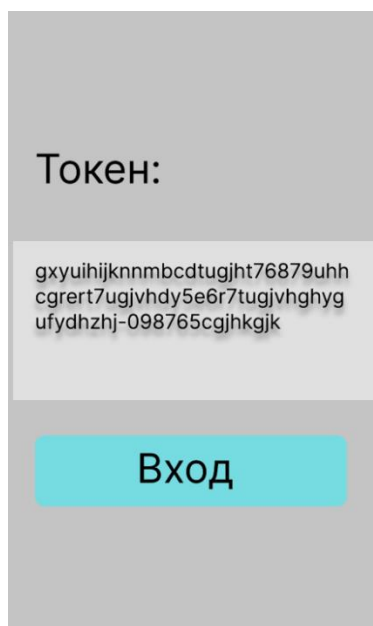


Рисунок 4 - Окно входа

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

14

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, который отправляется в систему Битрикс-24. Этот файл содержит информацию, которую об изменяемом поле и новом значении, которое будет записано в соответствующее место.

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		16

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 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, id_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), id_post (тип данных INT, FOREIGN KEY). После этого я создал таблицу post с полями id_post(тип данных INT, PRIMARY KEY), post_name(тип данных NVARCHAR) Затем я связал таблицы между собой, установив соответствующие ключи FOREIGN KEY и связи, проверил работоспособность базы данных, выполнив несколько тестовых запросов и операций.

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		10

3.2.1. Разработка стиля приложения

Первое окно – это окно регистрации, состоит из текстового поля для ввода токена пользователя и кнопки для проверки соответствия введенных 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 4, разработано в соответствии со стилем приложения приложение В. 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.

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

11

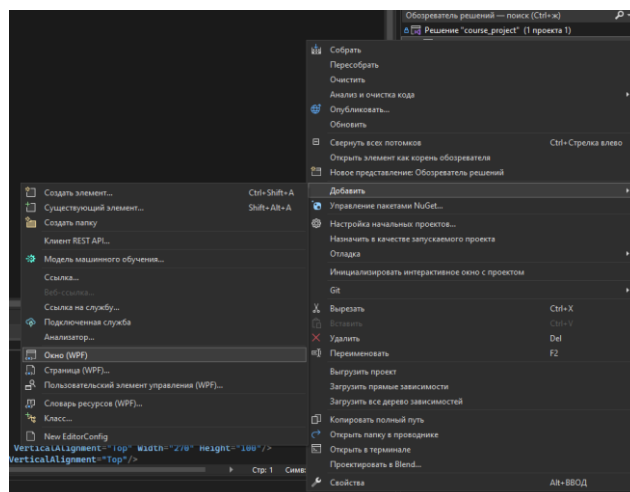


Рисунок 7 - Добавление окна регистрации

3.2.3. Разработка подсистем приложения

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

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

12

4.Тестирование

4.1 Unit - тестирование

Для оценки корректности работы написанных методов были созданы следующие Unit – тесты для методов проверки длины введённого номера панели, наличия в строке недопустимых символов, правильно введённого номера панели, для метода поиска последней созданной сделки на номер домофона и 4 функций смены стадии сделки.

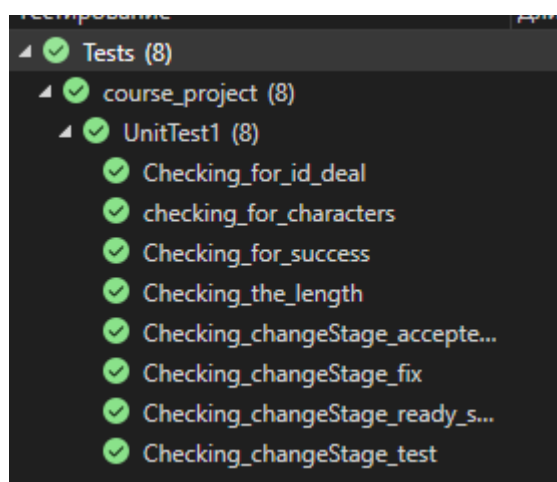


Рисунок 8 – Результат прохождения тестов

4.2 Тестирование основных подсистем приложения.

Система входа:

При вводе верного токена сотрудника и достаточного уровня доступа пользователя будет осуществляться переход на главную форму. Если сотрудник введёт не верный токен, то выйдет ошибка с сообщением, что токен является не верным(рисунок 9) .

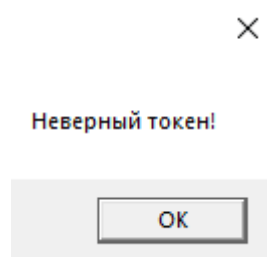


Рисунок 9 - Сообщение о неверном токене

При верном токене, не достаточном количестве прав у пользователя также будет ошибка, сообщение которой сообщает о недостаточном количестве прав(рисунок 10).

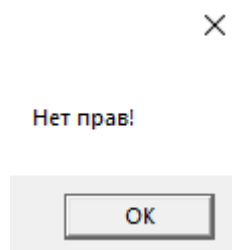


Рисунок 10 - Сообщение об отсутствии прав

Система изменения стадии сделки:

При попытке ввода некорректного номера панели выйдут сообщения с информацией, что не так с вводимой информацией(Рисунки 11, 12)

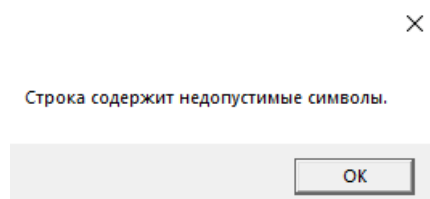


Рисунок 11 - Сообщение о недопустимых символах

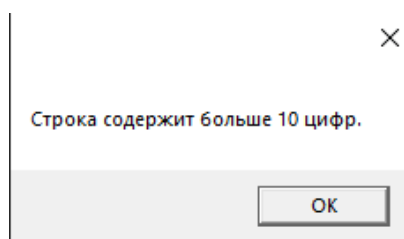


Рисунок 12 - Сообщение о несоответствии количества символов

При попытке изменить стадию сделки на недопустимую в данный момент также будет ошибка (Рисунок 13)

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

20

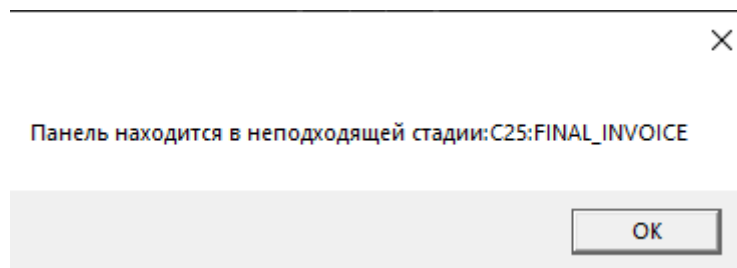


Рисунок 13 - Сообщение о неподходящей стадии

Система просмотров логов:

В данном окне возможно возникновение только одной ошибки с сообщением, что невозможно подключиться к БД(Рисунок 14).

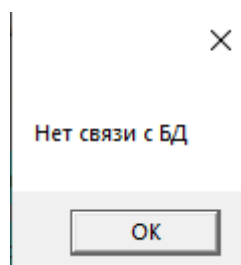


Рисунок 14 - Сообщение об отсутствии связи с БД

5. Инструкции по работе с приложением

5.1 Руководство пользователя

Данная инструкция предназначена для сотрудников компании Спутник, а именно участников отдела гарантии, к работе с созданной ИС допускаются люди, которые прошли инструктаж.

При запуске приложения пользователя встречает окно авторизации (рисунок 15), в соответствующее поле необходимо ввести свой уникальный токен сотрудника, при его корректности будет переход на следующее окно приложения, в противном случае выйдет сообщение с описание ошибки. После входа приложение запомнит пользователя и дальнейший запуск не потребует авторизации.



Рисунок 15 - Инструкция к окну входа

После авторизации пользователя встретит главное окно приложения (рисунок 16) в нём необходимо ввести ID панели, чей статус, в системе Битрикс-24, пользователь хочет изменить, каждая кнопка имеет своё название и перемещает сделку в соответствующую стадию, важно помнить порядок прохождения панелей этапов гарантийного обслуживания, при попытке переместить сделку в несоответствующую стадию, выйдет соответствующее,

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

22

при прочих ошибках необходимо проверить наличие интернета и перезагрузить приложение. Кнопка «Log Out» предназначена для выхода из учётной записи.

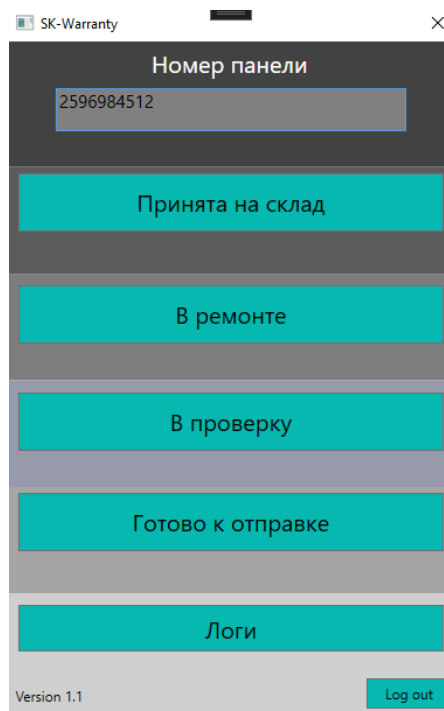


Рисунок 16 – Инструкция к главному окну

Окно просмотра логов содержит заметно больше элементов (рисунок 17), левый верхний угол занимает таблица с историей работы приложения, правее располагаются средства фильтрации данных, работа всех элементов интуитивно понятна, но фильтрация по дате может вызвать проблемы, сперва происходит выбор даты и дальше выбор идёт выбор трёх возможных вариантов: «Раньше даты», «Совпадает», «Позднее» выбор одного из 3 пунктов будет определять какие изменения будут отображены после применения фильтров.

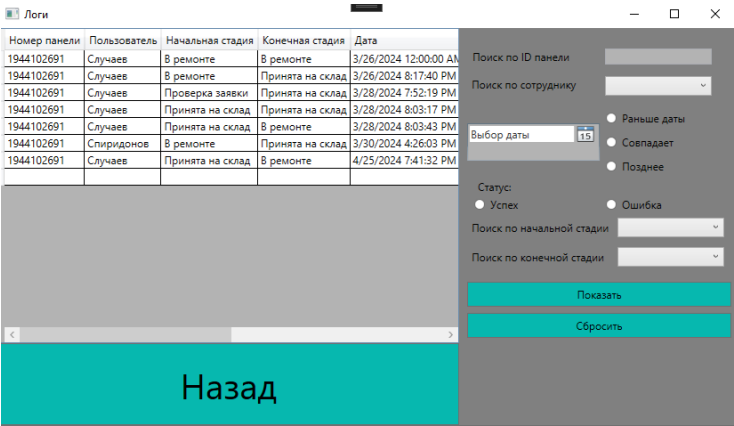


Рисунок 17 – Инструкция к окну логов

5.2. Руководство администратора БД

Данное руководство предназначено для пользователя с уровнем доступа администратор БД. При вводе токена, такого пользователя встретит окно с возможностью управления логами в БД включая возможность их удаления.

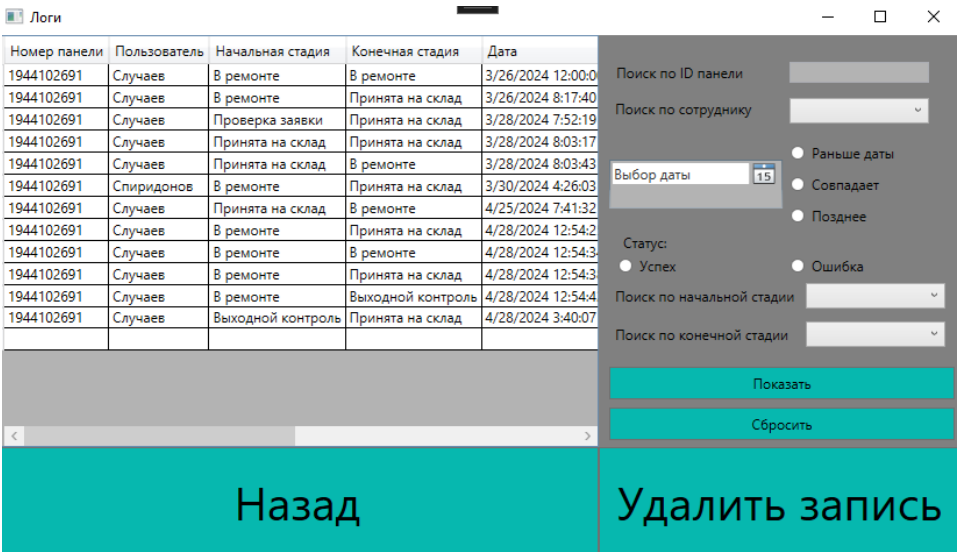


Рисунок 18 - Инструкция для администратора БД

Заключение

Результатом данного курсового проекта является АИС, предназначенная для управления отделом гарантии в IT компании. Во время работы был проведен анализ предметной области, проектирование и разработка базы данных, разработка приложения и руководства пользователя.

Разработанная система предоставляет удобное управление сделками на ремонт домофонов и просмотр истории их изменения, а гарантирует безопасность данных.

АИС имеет ряд преимуществ, среди которых:

- Простота добавления новых элементов
- Интуитивно понятный интерфейс
- Безопасность данных
- Возможность автоматизированной работы без связи с БД

В перспективе можно добавить возможность работать с заявкой: указывать выявленные дефекты и проведенные работы.

Следовательно, данная работа и в действительности обладает значительной ценностью, программный продукт может быть внедрен в деятельность организации и использоваться для решения вышеупомянутых задач.

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		25

Список литературы

1. Билл Гейтс. Искусство программирования на C#. - Москва: Вильямс, 2021. - 800 с. (Дата обращения: 22.03.2024)
2. Эндрю Хант, Дэвид Томас. Разработка ПО с использованием WPF. - Санкт-Петербург: Питер, 2022. - 896 с. (Дата обращения: 21.04.2024)
3. Стив Макконнелл. Чистый код и эффективное программирование. - Москва: ДМК Пресс, 2020. - 650 с. (Дата обращения: 10.04.2024)
4. Максим Шаров. Паттерны проектирования в .NET. - Санкт-Петербург: Питер, 2022. - 320 с. (Дата обращения: 04.04.2024)
5. Coursera: [Электронный ресурс] Курс "Базы данных и SQL" на Coursera. URL: <https://www.coursera.org/learn/sql-for-web-development> (Дата обращения: 01.04.2024)
6. Основы работы с базами данных на платформе GeekBrains. - URL: <https://geekbrains.ru/courses/68> (Дата обращения: 20.04.2024)
7. MozhnoEge: [Электронный ресурс] Информационные системы и базы данных. URL: <https://mozhnoege.ru/informatika/informatsionnye-sistemy-i-bazy-dannyh> (Дата обращения: 29.02.2024)
8. GeekBrains: [Электронный ресурс] Основы работы с базами данных. URL: <https://geekbrains.ru/courses/68> (Дата обращения: 29.01.2024)
9. W3Schools: [Электронный ресурс] Руководство по SQL. URL: <https://www.w3schools.com/sql/> (Дата обращения: 02.03.2024)
10. Docs.Microsoft: [Электронный ресурс] Документация по Windows Presentation Foundation (WPF). URL: <https://docs.microsoft.com/en-us/dotnet/desktop/wpf/?view=netdesktop-7.0> (Дата обращения: 15.02.2024)

Приложения

Приложение А

Скрипт создания БД

```
CREATE DATABASE [course_project]
GO
```

```
USE [course_project]
```

```
GO
```

```
/****** Object: Table [dbo].[access]   Script Date: 16.04.2024 20:45:35 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[access](
```

```
[id_level] [int] IDENTITY(1,1) NOT NULL,
```

```
[name_level] [nvarchar](25) NOT NULL,
```

```
CONSTRAINT [PK_access] PRIMARY KEY CLUSTERED
```

```
(
```

```
[id_level] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Table [dbo].[logs]   Script Date: 16.04.2024 20:45:35 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[logs](
```

```
[logs_id] [int] IDENTITY(1,1) NOT NULL,
```

```
[id_panel] [nvarchar](10) NOT NULL,
```

```
[user_id] [int] NOT NULL,
```

```
[initial_stage] [nvarchar](50) NOT NULL,
```

```
[final_stage] [nvarchar](50) NOT NULL,
```

```
[date] [datetime] NOT NULL,
```

```
[result] [bit] NOT NULL,
```

```
[reason] [nchar](100) NULL,
```

```
CONSTRAINT [PK_logs] PRIMARY KEY CLUSTERED
```

```
(
```

```
[logs_id] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
```

```
) ON [PRIMARY]
```

```
GO
```

```
/****** Object: Table [dbo].[stage_deal]   Script Date: 16.04.2024 20:45:35 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
CREATE TABLE [dbo].[stage_deal](
```

```
[id_deal] [nvarchar](50) NOT NULL,
```

```
[name_stage] [nvarchar](50) NOT NULL,
```

```
CONSTRAINT [PK_stage_deal] PRIMARY KEY CLUSTERED
```

```
(
```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

27

```

[id_deal] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[user]  Script Date: 16.04.2024 20:45:35 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[user](
[id_user] [int] IDENTITY(1,1) NOT NULL,
[firstname] [nvarchar](100) NOT NULL,
[name] [nvarchar](75) NOT NULL,
[patronymic] [nvarchar](100) NOT NULL,
[access_level] [int] NOT NULL,
[token] [nvarchar](100) NOT NULL,
[post] [nvarchar](50) NOT NULL,
CONSTRAINT [PK_user] PRIMARY KEY CLUSTERED
(
[id_user] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET IDENTITY_INSERT [dbo].[access] ON

INSERT [dbo].[access] ([id_level], [name_level]) VALUES (1, N'Низкий')
INSERT [dbo].[access] ([id_level], [name_level]) VALUES (2, N'Средний')
INSERT [dbo].[access] ([id_level], [name_level]) VALUES (3, N'')
SET IDENTITY_INSERT [dbo].[access] OFF
GO
SET IDENTITY_INSERT [dbo].[logs] ON

INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (1, N'1944102691', 1, N'C25:2', N'C25:2', CAST(N'2024-03-
26T00:00:00.000' AS DateTime), 1, NULL)
INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (2, N'1944102691', 1, N'C25:2', N'C25:EXECUTING',
CAST(N'2024-03-26T20:17:40.697' AS DateTime), 0, NULL)
INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (3, N'1944102691', 1, N'C25:PREPARATION',
N'C25:EXECUTING', CAST(N'2024-03-28T19:52:19.137' AS DateTime), 1, NULL)
INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (5, N'1944102691', 1, N'C25:EXECUTING', N'C25:EXECUTING',
CAST(N'2024-03-28T20:03:17.773' AS DateTime), 0, N'Панель находится в неподходящей
стадии
')
INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (6, N'1944102691', 1, N'C25:EXECUTING', N'C25:2',
CAST(N'2024-03-28T20:03:43.683' AS DateTime), 1, N'
')
INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (7, N'1111111111', 1, N'C25:APOLOGY', N'C25:EXECUTING',
CAST(N'2024-03-28T20:05:03.180' AS DateTime), 0, N'Панель находится в неподходящей
стадии
')
INSERT [dbo].[logs] ([logs_id], [id_panel], [user_id], [initial_stage], [final_stage], [date],
[result], [reason]) VALUES (8, N'1944102691', 2, N'C25:2', N'C25:EXECUTING',

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		28

```

CAST(N'2024-03-30T16:26:03.980' AS DateTime), 0, N'Панель находится в неподходящей
стадии
')
SET IDENTITY_INSERT [dbo].[logs] OFF
GO
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:1', N'Отложено')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:2', N'В ремонте')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:7', N'Готово к
отправке')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:APOLOGY', N'Ошибка
заведения')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:EXECUTING',
N'Принята на склад')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:FINAL_INVOICE',
N'Выходной контроль')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:LOSE', N'Заявка
отменена')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:NEW', N'Новая заявка')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:PREPARATION',
N'Проверка заявки')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES
(N'C25:PREPAYMENT_INVOIC', N'Заявка готова к отправке')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:UC_DMZGI5',
N'Паркинг')
INSERT [dbo].[stage_deal] ([id_deal], [name_stage]) VALUES (N'C25:WON', N'Заявка
завершена')
GO
SET IDENTITY_INSERT [dbo].[user] ON

INSERT [dbo].[user] ([id_user], [firstname], [name], [patronymic], [access_level], [token],
[post]) VALUES (1, N'Случаев', N'Максим', N'Константинович', 3,
N'ылдфроапоырфвпароывраплрфывроапрофыпвафыроваппрофыц', N'Мастер')
INSERT [dbo].[user] ([id_user], [firstname], [name], [patronymic], [access_level], [token],
[post]) VALUES (2, N'Спиридонов', N'Евгений', N'Андреевич', 3,
N'выаплоыврапошлщрываолдпролд', N'Мастер')
SET IDENTITY_INSERT [dbo].[user] OFF
GO
ALTER TABLE [dbo].[logs] WITH CHECK ADD CONSTRAINT [FK_logs_stage_deal]
FOREIGN KEY([initial_stage])
REFERENCES [dbo].[stage_deal] ([id_deal])
GO
ALTER TABLE [dbo].[logs] CHECK CONSTRAINT [FK_logs_stage_deal]
GO
ALTER TABLE [dbo].[logs] WITH CHECK ADD CONSTRAINT [FK_logs_stage_deal1]
FOREIGN KEY([final_stage])
REFERENCES [dbo].[stage_deal] ([id_deal])
GO
ALTER TABLE [dbo].[logs] CHECK CONSTRAINT [FK_logs_stage_deal1]
GO
ALTER TABLE [dbo].[logs] WITH CHECK ADD CONSTRAINT [FK_logs_user] FOREIGN
KEY([user_id])
REFERENCES [dbo].[user] ([id_user])
GO
ALTER TABLE [dbo].[logs] CHECK CONSTRAINT [FK_logs_user]
GO
ALTER TABLE [dbo].[user] WITH CHECK ADD CONSTRAINT [FK_user_access]
FOREIGN KEY([access_level])
REFERENCES [dbo].[access] ([id_level])

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		29

```

ON UPDATE CASCADE
GO
ALTER TABLE [dbo].[user] CHECK CONSTRAINT [FK_user_access]
GO
USE [master]
GO
ALTER DATABASE course_project SET READ_WRITE
GO

```

Приложение Б

Код главного окна

XAML:

```

<Window x:Class="course_project.MainWindow"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:course_project;assembly=course_project"
    mc:Ignorable="d"
    Title="SK-Warranty" Height="620" Width="400" ResizeMode="NoResize"
    Topmost="True">
    <Grid>
        <Grid.RowDefinitions>
            <RowDefinition Height="100*" />
            <RowDefinition Height="85*" />
            <RowDefinition Height="85*" />
            <RowDefinition Height="85*" />
            <RowDefinition Height="85*" />
            <RowDefinition Height="100*" />
        </Grid.RowDefinitions>
        <StackPanel Grid.Row="0" Background="#424242">
            <Label Content="Номер панели" FontSize="20" Height="40" Width="146"
                Foreground="White" />
            <TextBox x:Name="text_panel" FontSize="15" Width="300" Height="37"
                Background="Gray" />
        </StackPanel>
        <StackPanel Grid.Row="1" Background="#5B5B5B">
            <Button x:Name="warehouse" Background="#06b8af" Content="Принята на склад"
                FontSize="20" Margin="10, 5, 10, 0" Height="50" Click="warehouse_Click" />
        </StackPanel>
        <StackPanel Grid.Row="2" Background="#7D7D7D">
            <Button FontSize="20" x:Name="fix" Background="#06b8af" Content="В ремонте"
                Margin="10, 10, 10, 0" Height="50" Click="fix_Click" />
        </StackPanel>
        <StackPanel Grid.Row="3" Background="#979AAA">
            <Button x:Name="check" FontSize="20" Background="#06b8af" Content="В проверку"
                Margin="10, 10, 10, 0" Height="50" Click="check_Click" />
        </StackPanel>
        <StackPanel Grid.Row="4" Background="#A5A5A5">
            <Button x:Name="sending" FontSize="20" Background="#06b8af" Content="Готово к
                отправке" Margin="10, 5, 10, 0" Height="50" Click="sending_Click" />
        </StackPanel>
        <StackPanel Grid.Row="5" Background="#CFCFCF">

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		30

```

        <Button x:Name="log" FontSize="20" Content="Логн" Background="#06b8af" Margin="10, 10, 10, 0" Height="40" Click="log_Click"/>
        <Label Content="Version 1.1" Margin="0, 25, 255, 0" Width="124"/>
        <Button x:Name="out" Content="Log out" Margin="305, -30, 0, 0" Width="74" Height="26" Background="#06b8af" Click="out_Click"/>
    </StackPanel>
</Grid>
</Window>

```

C#:

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace course_project
{
    /// <summary>
    /// Логика взаимодействия для MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        public static string id_panel;
        public MainWindow()
        {
            InitializeComponent();
            if (Window2.flag == true)
            {
                log.IsEnabled = false;
            }
        }
        private void fix_Click(object sender, RoutedEventArgs e)
        {
            try
            {
                string result1 = ID_verification.CheckString(text_panel.Text);
                if (result1 != "Успех")
                {
                    MessageBox.Show(result1);
                }
            }
            else
            {
                id_panel = text_panel.Text;
                string id_deal = SearchDeal.SearchDealMethod(text_panel.Text);
                string result = DealManager.changeStage_fix(id_deal);
            }
        }
    }
}

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		31

```

        MessageBox.Show(result);
    }

}
catch
{
    MessageBox.Show("Произошла ошибка");
}
}
private void warehouse_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string result1 = ID_verification.CheckString(text_panel.Text);
        if (result1 != "Успех")
        {
            MessageBox.Show(result1);
        }
        else
        {
            id_panel = text_panel.Text;
            string id_deal = SearchDeal.SearchDealMethod(text_panel.Text);
            string result = DealManager.changeStage_accepted_warehouse(id_deal);
            MessageBox.Show(result);
        }
    }
    catch
    {
        MessageBox.Show("Произошла ошибка");
    }
}
private void check_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string result1 = ID_verification.CheckString(text_panel.Text);
        if (result1 != "Успех")
        {
            MessageBox.Show(result1);
        }
        else
        {
            id_panel = text_panel.Text;
            string id_deal = SearchDeal.SearchDealMethod(text_panel.Text);
            string result = DealManager.changeStage_test(id_deal);
            MessageBox.Show(result);
        }
    }
    catch
    {
        MessageBox.Show("Произошла ошибка");
    }
}
private void sending_Click(object sender, RoutedEventArgs e)
{
    try
    {

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

32

```

        string result1 = ID_verification.CheckString(text_panel.Text);
        if (result1 != "Успех")
        {
            MessageBox.Show(result1);
        }
        else
        {
            id_panel = text_panel.Text;
            string id_deal = SearchDeal.SearchDealMethod(text_panel.Text);
            string result = DealManager.changeStage_ready_ship(id_deal);
            MessageBox.Show(result);
        }
    }
    catch
    {
        MessageBox.Show("Произошла ошибка");
    }
}

private void log_Click(object sender, RoutedEventArgs e)
{
    Window1 window = new Window1();
    window.Show();
    this.Close();
}

private void out_Click(object sender, RoutedEventArgs e)
{
    file_write_and_read.file_write(" ");
    Window2 window = new Window2();
    window.Show();
    this.Close();
}
}
}

```

Приложение В

Код окна входа

XAML:

```

<Window x:Class="course_project.Window2"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:course_project"
    mc:Ignorable="d"
    Title="Вход" Height="343" Width="290" ResizeMode="NoResize">
    <Grid Background="Gray">
        <TextBox x:Name="tokenT" HorizontalAlignment="Center" Background="#b3b3b3"
            FontSize="20" Margin="0,64,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="270"
            Height="100"/>
        <TextBlock HorizontalAlignment="Left" Margin="101,26,0,0" TextWrapping="Wrap"
            Text="Токен" FontSize="25" VerticalAlignment="Top"/>
        <Button Content="Вход" HorizontalAlignment="Center" Background="#06b8af" FontSize="35"
            VerticalAlignment="Top" Height="97" Width="270" Margin="0,169,0,0" Click="Button_Click"/>
    </Grid>
</Window>

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

33

C#:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace course_project
{
    /// <summary>
    /// Логика взаимодействия для Window2.xaml
    /// </summary>
    public partial class Window2 : Window
    {
        public static int id_user = 0;
        public static int id_level;
        public static bool flag = false;
        course_projectEntities4 db = new course_projectEntities4();
        public Window2()
        {
            if (access_verification() == true)
            {
                MainWindow window = new MainWindow();
                window.Show();
                this.Close();
            }
            InitializeComponent();
        }
        private void Button_Click(object sender, RoutedEventArgs e)
        {
            if (tokenT.Text == "viva_laswegas")
            {
                flag = true;
                MessageBox.Show("Данные в БД не фиксируются");
                MainWindow window = new MainWindow();
                window.Show();
                this.Close();
            }
            else
            {
                var list = db.user.Where(x => x.token == tokenT.Text);
                int? level = 0;
                if (list != null)
                {
                    foreach (var item in list)
                    {
                        level = item.access_level;
                        id_user = item.id_user;
                        id_level = (int)level;
                    }
                    if (level == 4)
                    {
                        Window1 window = new Window1();
                        window.Show();
                        this.Close();
                    }
                }
            }
        }
    }
}
```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

34

```

        else
        {
            if (level > 1)
            {
file_write_and_read.file_write(AESEncryption.Encrypt(tokenT.Text, "sAL4-AVlWoCGELM-
"));
                MainWindow window = new MainWindow();
                window.Show();
                this.Close();
            }
            else if (level == 1)
            {
                MessageBox.Show($"Нет прав!");
            }
            else
            {
                MessageBox.Show($"Неверный токен!");
            }
        }
    }
}

public bool access_verification()
{
    try
    {
        string tokenAESE = file_write_and_read.file_read();
        string token = AESEncryption.Decrypt(tokenAESE, "sAL4-AVlWoCGELM-");
        var list = db.user.Where(x => x.token == token);
        int? level = 1;
        if (list != null)
        {
            foreach (var item in list)
            {
                level = item.access_level;

                id_user = item.id_user;
            }

            if (level > 1)
            {
                return true;
            }
            else
            {
                return false;
            }
        }
        else
        {
            return false;
        }
    }
    catch
    {
        return false;
    }
}
}
}

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		35

Приложение Г

Код окна логов

XAML:

```
<Window x:Class="course_project.Window1"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    xmlns:local="clr-namespace:course_project"
    mc:Ignorable="d"
    Title="Логги" Height="485" Width="845" MinHeight="485" MinWidth="845"
    Topmost="True">
    <Grid Background="Gray">
        <Grid.RowDefinitions>
            <RowDefinition Height="385*" />
            <RowDefinition Height="100*" />
        </Grid.RowDefinitions>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="250*" />
            <ColumnDefinition Width="150*" />
        </Grid.ColumnDefinitions>
        <DataGrid Grid.Row="0" x:Name="logsG" AutoGenerateColumns="False"
            Background="#b3b3b3">
            <DataGrid.Columns>
                <DataGridTextColumn Header="Номер панели" Binding="{Binding id_panel}" />
                <DataGridTextColumn Header="Пользователь" Binding="{Binding user.firstname}" />
                <DataGridTextColumn Header="Начальная стадия" Binding="{Binding
stage_deal.name_stage}" />
                <DataGridTextColumn Header="Конечная стадия" Binding="{Binding
stage_deal1.name_stage}" />
                <DataGridTextColumn Header="Дата" Binding="{Binding date}" />
                <DataGridTextColumn Header="Результат" Binding="{Binding result}" />
                <DataGridTextColumn Header="Причина" Binding="{Binding reason}" />
            </DataGrid.Columns>
        </DataGrid>

        <Button x:Name="comeback" FontSize="40" Content="Назад" Background="#06b8af"
            HorizontalAlignment="Stretch" VerticalAlignment="Stretch" Grid.Row="1" Click="comeback_Click"
            />
        <Label Grid.Column="1" Content="Поиск по сотруднику" HorizontalAlignment="Left"
            Margin="9,50,0,0" VerticalAlignment="Top" />
        <ComboBox Grid.Column="1" Background="#b3b3b3" x:Name="find_name"
            HorizontalAlignment="Left" Margin="164,54,0,0" VerticalAlignment="Top" Width="120" />
        <RadioButton Grid.Column="1" x:Name="earlier" GroupName="first" Content="Раньше даты"
            HorizontalAlignment="Left" Margin="164,94,0,0" VerticalAlignment="Top" Height="20"
            Width="120" />
        <RadioButton Grid.Column="1" x:Name="matches" GroupName="first" Content="Совпадает"
            HorizontalAlignment="Left" Margin="164,121,0,0" VerticalAlignment="Top" Height="20"
            Width="120" />
        <RadioButton Grid.Column="1" x:Name="later" GroupName="first" Content="Позднее"
            HorizontalAlignment="Left" Margin="164,148,0,0" VerticalAlignment="Top" Height="20"
            Width="120" />
    </Grid>
</Window>
```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

36

```

<DatePicker Background="#b3b3b3" Grid.Column="1"
PreviewKeyDown="DatePicker_PreviewKeyDown" x:Name="date" HorizontalAlignment="Left"
Margin="9,107,0,0" VerticalAlignment="Top" Height="42" Width="149"/>
<Label Grid.Column="1" Content="Поиск по начальной стадии" HorizontalAlignment="Left"
Margin="9,211,0,0" VerticalAlignment="Top"/>
<ComboBox Grid.Column="1" x:Name="stage" HorizontalAlignment="Left"
Margin="178,213,0,0" VerticalAlignment="Top" Width="120" />
<Button Grid.Column="1" Background="#06b8af" x:Name="show" Content="Показать"
HorizontalAlignment="Left" Margin="9,285,0,0" VerticalAlignment="Top" Width="297" Height="28"
Click="show_Click"/>
<Button Grid.Column="1" Background="#06b8af" x:Name="rest" Content="Сбросить"
HorizontalAlignment="Left" Margin="9,320,0,0" VerticalAlignment="Top" Width="297" Height="28"
Click="rest_Click"/>
<Label Grid.Column="1" Content="Поиск по конечной стадии" HorizontalAlignment="Left"
Margin="9,244,0,0" VerticalAlignment="Top"/>
<ComboBox Grid.Column="1" x:Name="stage_fin" HorizontalAlignment="Left"
Margin="178,246,0,0" VerticalAlignment="Top" Width="120" />
<RadioButton Grid.Column="1" x:Name="error" GroupName="second" Content="Ошибка"
HorizontalAlignment="Left" Margin="164,191,0,0" VerticalAlignment="Top" Height="20"
Width="120"/>
<RadioButton Grid.Column="1" x:Name="successful" GroupName="second" Content="Успех"
HorizontalAlignment="Left" Margin="16,191,0,0" VerticalAlignment="Top" Height="20"
Width="120"/>
<Label Grid.Column="1" Content="Поиск по ID панели" HorizontalAlignment="Left"
Margin="10,19,0,0" VerticalAlignment="Top"/>
<TextBox Background="#b3b3b3" Grid.Column="1" x:Name="id" HorizontalAlignment="Left"
Margin="164,23,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="120"/>
<Label Grid.Column="1" Content="Статус:" HorizontalAlignment="Left" Margin="16,165,0,0"
VerticalAlignment="Top"/>
<Button Grid.Column="1" Content="Удалить запись" x:Name="del" FontSize="40"
Visibility="Hidden" Background="#06b8af" Grid.Row="1" HorizontalAlignment="Stretch"
VerticalAlignment="Stretch" Click="del_Click"/>

```

```

</Grid>
</Window>

```

C#

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Reflection.Emit;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
namespace course_project
{
    /// <summary>
    /// Логика взаимодействия для Window1.xaml
    /// </summary>

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		37

```

public partial class Window1 : Window
{
    private course_projectEntities4 db = new course_projectEntities4();
    public static List<logs> list;
    public Window1()
    {
        InitializeComponent();
        if (Window2.id_level == 4)
        {
            del.Visibility = Visibility.Visible;
        }
        list = db.logs.ToList();
        logsG.ItemsSource = db.logs.ToList();
        find_name.ItemsSource = db.user.Select(x => x.firstname).ToList();
        stage.ItemsSource = db.stage_deal.Select(x => x.name_stage).ToList();
        stage_fin.ItemsSource = db.stage_deal.Select(x => x.name_stage).ToList();
    }
    private void comeback_Click(object sender, RoutedEventArgs e)
    {
        if(Window2.id_level == 4)
        {
            Window2 window = new Window2();
            window.Show();
            this.Close();
        }
        else
        {
            MainWindow window = new MainWindow();
            window.Show();
            this.Close();
        }
    }
    private void show_Click(object sender, RoutedEventArgs e)
    {
        try
        {
            var list2 = db.logs.ToList();
            string value = stage.Text;
            var id_arr = db.stage_deal.FirstOrDefault(x => x.name_stage == value);
            if (id.Text.Trim() != null)
            {
                var f = list2.Where(x => x.id_panel.Contains(id.Text)).ToList();
                list2 = f;
            }
            if (id_arr != null)
            {
                var f = list2.Where(x => x.initial_stage == id_arr.id_deal).ToList();
                list2 = f;
            }
            string value1 = stage_fin.Text;
            var id_arr1 = db.stage_deal.FirstOrDefault(x => x.name_stage == value1);
            if (id_arr1 != null)
            {
                var f = list2.Where(x => x.final_stage == id_arr1.id_deal).ToList();
                list2 = f;
            }
        }
    }
}

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		38

```

string value2 = find_name.Text;
var id_arr2 = db.user.FirstOrDefault(x => x.firstname == value2);
if (id_arr2 != null)
{
    var f = list2.Where(x => x.user_id == id_arr2.id_user).ToList();
    list2 = f;
}
if (error.IsChecked == true)
{
    var f = list2.Where(x => x.result == false).ToList();
    list2 = f;
}
if (successful.IsChecked == true)
{
    var f = list2.Where(x => x.result == true).ToList();
    list2 = f;
}
if (earlier.IsChecked == true)
{
    var f = list2.Where(x => x.date.Date > Convert.ToDateTime(date.Text).Date).ToList();
    list2 = f;
}
if (matches.IsChecked == true)
{
    var f = list2.Where(x => x.date.Date == Convert.ToDateTime(date.Text).Date).ToList();
    list2 = f;
}
if (later.IsChecked == true)
{
    var f = list2.Where(x => x.date.Date < Convert.ToDateTime(date.Text).Date).ToList();
    list2 = f;
}
list = list2;
logsG.ItemsSource = list2;

}
catch
{
    MessageBox.Show("Нет связи с БД");
}
}
private void rest_Click(object sender, RoutedEventArgs e)
{
    find_name.Text = "";
    stage_fin.Text = "";
    stage.Text = "";

    logsG.ItemsSource = list;
    error.IsChecked = false;
    successful.IsChecked = false;
    later.IsChecked = false;
    matches.IsChecked = false;
    earlier.IsChecked = false;
    date.Text = "";
    id.Text = "";
}
private void DatePicker_PreviewKeyDown(object sender, KeyEventArgs e)

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

39

```

    {
        e.Handled = true;
    }
    private void del_Click(object sender, RoutedEventArgs e)
    {
        try
        {
            var del = logsG.SelectedItem as logs;
            if (del != null)
            {
                db.logs.Remove(del);
                MessageBox.Show("Успех");
                logsG.ItemsSource = list;
            }
            else
            {
                MessageBox.Show("Выберите строку в таблице");
            }
        }
        catch
        {
            MessageBox.Show("Нет связи с БД");
        }
    }
}

```

Приложение Д

Класс для поиска сделки

```

using System;
using System.IO;
using System.Net;
using System.Xml.Linq;
using Newtonsoft.Json.Linq;
public class SearchDeal
{
    public static string SearchDealMethod(string idPanel)
    {
        string idDeal = "";
        try
        {
            // URL for getting JSON
            string urlString =
            // Creating URL object
            Uri url = new Uri(urlString);
            // Creating HttpWebRequest object
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
            request.Method = "GET";
            // Getting response from server
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            if (response.StatusCode == HttpStatusCode.OK)
            {
                // Reading data from input stream
                StreamReader reader = new StreamReader(response.GetResponseStream());
            }
        }
    }
}

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

40

```

        string responseContent = reader.ReadToEnd();
        // Processing received JSON
        JObject jsonObject = JObject.Parse(responseContent);
        JArray dealsArray = (JArray)jsonObject["result"];
        JObject lastDeal = (JObject)dealsArray[dealsArray.Count - 1];
        idDeal = lastDeal["ID"].ToString();
    }
    else
    {
        idDeal = "Error executing request. Response code: " + response.StatusCode;
    }
}
catch
{
    idDeal = "Error executing";
}
return idDeal;
}
}

```

Приложение Е

Класс для проверки стадии

```

using System.Net;
using Newtonsoft.Json.Linq;
using System.Text;
using System.IO;
using System;

public class CheckStage
{
    public static string Check_Stage(string id_deal)
    {
        string status = "";
        try
        {
            // URL for getting JSON
            string urlString =
            // Create a HttpWebRequest object
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(urlString);
            request.Method = "GET";
            // Get the response from the server
            HttpWebResponse response = (HttpWebResponse)request.GetResponse();
            if (response.StatusCode == HttpStatusCode.OK)
            {
                // Read data from the input stream
                using (StreamReader reader = new StreamReader(response.GetResponseStream(),
Encoding.UTF8))
                {
                    string inputLine;
                    StringBuilder jsonResponse = new StringBuilder();
                    while ((inputLine = reader.ReadLine()) != null)
                    {
                        jsonResponse.Append(inputLine);
                    }
                }
                reader.Close();
            }
        }
        catch { }
    }
}

```

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		41


```

        // Processing the received JSON
        JObject jsonObject = JObject.Parse(jsonResponse.ToString());
        JObject dealsArray = (JObject)jsonObject["result"];
        status = (string)dealsArray["STAGE_ID"];
    }
}
else
{
    status = "Error executing request. Response code: " + response.StatusCode;
}
response.Close();
}
catch (Exception e)
{
    status = "Error executing: " + e.Message;
}
return status;
}
}

```

Приложение Ж

Код для изменения стадии сделки

```

using course_project;
using System;
using System.Net.Http;
using System.Xml.XPath;

public static class DealManager
{
    private static string ChangeStage_universal(string dealId, string stageId, string stage)
    {
        string result = "";
        HttpClient client = new HttpClient();
        try
        {
            // URL и JSON-запрос
            // Создание запроса POST
            StringContent content = new StringContent(json);
            content.Headers.ContentType = new
System.Net.Http.Headers.MediaTypeHeaderValue("application/json");
            HttpResponseMessage response = client.PostAsync(url, content).Result;
            if (response.IsSuccessStatusCode)
            {
                // Преобразование ответа в строку
                string responseBody = response.Content.ReadAsStringAsync().Result;
                result = "Успех";
                if(Window2.flag == false)
                {
                    logs_save.save(Window2.id_user, MainWindow.id_panel, stage, stageId, true, "");
                }
            }
        }
        else
        {

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

42

```

        result = "Запрос отправлен, но сообщения об успехе нет";
        if (Window2.flag == false)
        {
            logs_save.save(Window2.id_user, MainWindow.id_panel, stage, stageId, false, result);
        }
    }
}
catch
{
    result = "Ошибка запроса";
    if (Window2.flag == false)
    {
        logs_save.save(Window2.id_user, MainWindow.id_panel, stage, stageId, false, result);
    }
}

return result;
}

public static string changeStage_accepted_warehouse(string dealId)
{
    string result = "";
    string stage = CheckStage.Check_Stage(dealId);
    if (stage.Length <= 30)
    {
        if (stage == "C25:NEW" || stage == "C25:PREPARATION" || stage ==
"C25:PREPAYMENT_INVOIC")
        {
            result = ChangeStage_universal(dealId, "C25:EXECUTING", stage);
        }
        else
        {
            result = $"Панель находится в неподходящей стадии: {stage}";
            if (Window2.flag == false)
            {
                logs_save.save(Window2.id_user, MainWindow.id_panel, stage, "C25:EXECUTING", false,
"Панель находится в неподходящей стадии");
            }
        }
    }
    else
    {
        result = stage;
    }
    return result;
}

public static string changeStage_fix(string dealId)
{
    string result = "";
    string stage = CheckStage.Check_Stage(dealId);
    if (stage.Length <= 30)
    {

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

43

```

        if (stage == "C25:1" || stage == "C25:2" || stage == "C25:PREPAYMENT_INVOIC" || stage ==
"C25:7" || stage == "C25:UC_DMZGI5" || stage == "C25:EXECUTING")
        {
            result = ChangeStage_universal(dealId, "C25:2", stage);
        }
        else
        {
            result = $"Панель находится в неподходящей стадии:{stage}";
            if (Window2.flag == false)
            {
                logs_save.save(Window2.id_user, MainWindow.id_panel, stage, "C25:2", false, "Панель
находится в неподходящей стадии");
            }
        }
    }
    else
    {
        result = stage;
    }
    return result;
}

public static string changeStage_ready_ship(string dealId)
{
    string result = "";
    string stage = CheckStage.Check_Stage(dealId);
    if (stage.Length <= 30)
    {
        if (stage != "C25:LOSE" && stage != "C25:APOLOGY" && stage != "C25:WON")
        {
            result = ChangeStage_universal(dealId, "C25:7", stage);
        }
        else
        {
            result = $"Панель находится в неподходящей стадии:{stage}";
            if (Window2.flag == false)
            {
                logs_save.save(Window2.id_user, MainWindow.id_panel, stage, "C25:7", false, "Панель
находится в неподходящей стадии");
            }
        }
    }
    else
    {
        result = stage;
    }
    return result;
}

public static string changeStage_test(string dealId)
{
    string result = "";
    string stage = CheckStage.Check_Stage(dealId);
    if (stage.Length <= 30)
    {

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

44

```

        result = ChangeStage_universal(dealId, "C25:FINAL_INVOICE", stage);
    }
    else
    {
        result = $"Панель находится в неподходящей стадии: {stage}";
        if (Window2.flag == false)
        {
            logs_save.save(Window2.id_user, MainWindow.id_panel, stage, "C25:FINAL_INVOICE",
false, "Панель находится в неподходящей стадии");
        }

    }
    return result;
}
}
}

```

Приложение 3

Код для сохранения логов

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace course_project
{
    public class logs_save
    {
        public static void save(int user_id, string id_panel, string first_stage, string final_stage, bool result,
string reason)
        {
            course_projectEntities4 db = new course_projectEntities4();

            logs logs = new logs();
            logs.user_id = user_id;
            logs.id_panel = id_panel;
            logs.result = result;
            logs.initial_stage = first_stage;
            logs.final_stage = final_stage;
            logs.date = DateTime.Now;
            logs.reason = reason;

            db.logs.Add(logs);
            db.SaveChanges();
        }
    }
}

```

Приложение И

Код для шифрования токена

					КП 09.02.07.21И1.17.000 ПЗ	Лист
Изм.	Лист	№. докум.	Подпись	Дата		45

```

using System;
using System.IO;
using System.Security.Cryptography;
using System.Text;

namespace course_project
{
    public class AESEncryption
    {
        public static string Encrypt(string plainText, string key)
        {
            using (Aes aesAlg = Aes.Create())
            {
                aesAlg.Key = Encoding.UTF8.GetBytes(key);
                aesAlg.IV = new byte[16]; // Используем IV длиной 16 байт для AES

                ICryptoTransform encryptor = aesAlg.CreateEncryptor(aesAlg.Key, aesAlg.IV);
                using (MemoryStream msEncrypt = new MemoryStream())
                {
                    using (CryptoStream csEncrypt = new CryptoStream(msEncrypt, encryptor,
CryptoStreamMode.Write))
                    {
                        using (StreamWriter swEncrypt = new StreamWriter(csEncrypt))
                        {
                            swEncrypt.Write(plainText);
                        }
                    }
                    return Convert.ToBase64String(msEncrypt.ToArray());
                }
            }
        }

        public static string Decrypt(string cipherText, string key)
        {
            using (Aes aesAlg = Aes.Create())
            {
                aesAlg.Key = Encoding.UTF8.GetBytes(key);
                aesAlg.IV = new byte[16]; // Используем IV длиной 16 байт для AES
                ICryptoTransform decryptor = aesAlg.CreateDecryptor(aesAlg.Key, aesAlg.IV);
                using (MemoryStream msDecrypt = new
MemoryStream(Convert.FromBase64String(cipherText)))
                {
                    using (CryptoStream csDecrypt = new CryptoStream(msDecrypt, decryptor,
CryptoStreamMode.Read))
                    {
                        using (StreamReader srDecrypt = new StreamReader(csDecrypt))
                        {
                            return srDecrypt.ReadToEnd();
                        }
                    }
                }
            }
        }
    }
}

```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

46

Приложение К

Код для записи зашифрованного токена в TXT файл

```
using System;
using System.IO;

namespace course_project
{
    public class file_write_and_read
    {
        public static void file_write(string text)
        {
            try
            {
                string filePath = "user.txt";

                File.WriteAllText(filePath, text);

            }
            catch (Exception e)
            {
            }
        }

        public static string file_read()
        {
            try
            {
                string filePath = "user.txt";
                string data = File.ReadAllText(filePath);
                return data;
            }
            catch (Exception e)
            {
                return "Ошибка при чтении данных из файла: " + e.Message;
            }
        }
    }
}
```

Изм.	Лист	№. докум.	Подпись	Дата

КП 09.02.07.21И1.17.000 ПЗ

Лист

47