

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет

вычислительной техники

Кафедра

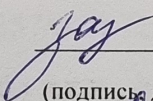
«Вычислительная техника»

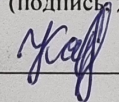
Направление подготовки: 09.03.01 «Информатика и вычислительная техника»
Профиль подготовки: «Программное обеспечение средств вычислительной
техники и автоматизированных систем»

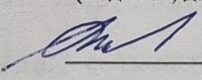
БАКАЛАВРСКАЯ РАБОТА

на тему

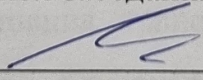
**АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА
СОСТОЯНИЯ ТРОЛЛЕЙБУСНОГО ПАРКА**

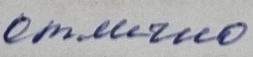
Студент  31.05.2024 Зацепилина Е.С.
(подпись, дата)

Руководитель  31.05.2024 Карамышева Н.С.
(подпись, дата)

Нормоконтролёр  31.05.2024 Бычков А.С.
(подпись, дата)

Работа допущена к защите (протокол заседания кафедры от 27.05.2024 г. №11)

Заведующий кафедрой  М.А. Митрохин
(подпись)

Работа защищена с отметкой  (протокол заседания ГЭК от 10.06.24 № 2)

Секретарь ГЭК  Е.И. Гурин
(подпись)

Пенза 2024

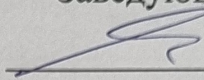
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»

Заведующий кафедрой

 М.А. Митрохин

«19» 04 2024г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

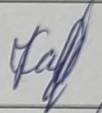
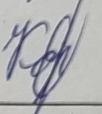
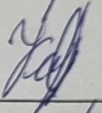
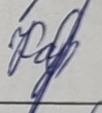
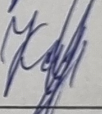
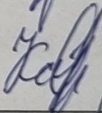
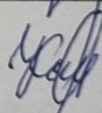
1. Студент Зацепилина Екатерина Сергеевна гр.20BBB1 факультета вычислительной техники.
2. Тема работы: «Автоматизированная система учета состояния троллейбусного парка»
Тема утверждена приказом ПГУ №357/о от "19" апреля 2024 г.
3. Руководитель бакалаврской работы к.т.н., доцент, каф. ВТ ПГУ Карамышева Н.С.
4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать приложение «Автоматизированная система учета состояния троллейбусного парка», позволяющее автоматизировать ручной процесс записи в журнал о сходе транспортного средства с линии, создания заявок на ремонт, создания рекламационных актов, отслеживания текущего состояния транспортных средств. Приложение должно обладать графическим и интуитивно понятным интерфейсом, для взаимодействия с пользователем. Система должна предоставлять возможность для взаимодействия с информацией о транспортных средствах, о заявках на ремонт, о видах технического обслуживания и о сотрудниках, работающих с данной автоматизированной системой. Разрабатываемое приложение также должно выполнять задачи по формированию документов в виде рекламационных актов.

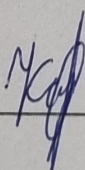
5. Перечень подлежащих разработке вопросов

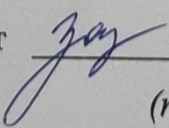
Требуется произвести анализ предметных областей: изучить существующие автоматизированные системы, выделить их достоинства и недостатки. Выбрать язык программирования, который подойдет для создания графического интерфейса. Определить архитектуру приложения и вспомогательные средства разработки. Изучить функционал библиотек для взаимодействия с файлами формата docx.

6. Календарный график выполнения работы

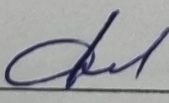
№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	26.04.2024 – 28.05.2024	
2	Анализ требований	32	29.04.2024 – 30.05.2024	
3	Создание макета графического интерфейса	30	1.04.2024 – 3.05.2024	
4	Выбор технологий, применяемых при проектировании и разработке	40	4.05.2024 – 8.05.2024	
5	Реализация графического интерфейса	66	9.05.2024 – 15.05.2024	
6	Разработка основной части программы	70	16.05.2024 – 23.05.2024	
7	Оформление пояснительной записки	54	24.05.2024 – 31.05.2024	

Дата выдачи задания "19" апреля 2024 г.

Руководитель бакалаврской работы  19.04.2024 Карамышева Н.С.
(подпись, дата)

Задание к исполнению принял студент  19.04.2024 Зацепилина Е.С.
(подпись, дата)

Работу к защите допустить

Декан факультета  07.06.2024 Фионова Л.Р.
(подпись, дата)

Бакалаврская работа содержит 223 лист, 18 рисунков 13 таблиц, 7 источников, 3 приложения

АВТОМАТИЗИРОВАННАЯ СИСТЕМА, ТРОЛЛЕЙБУСНЫЙ ПАРК, MVVM, C#, WPF, POSTGRESQL

Цель работы – разработать систему, позволяющую автоматизировать процесс учета троллейбусов в троллейбусном парке.

Объектом разработки является приложение, предназначенное для ведения базы данных для учета троллейбусов в троллейбусном парке, а также для формирования рекламационных актов, по которым будет осуществляется гарантийный ремонт и техническое обслуживание. Приложение должно обладать графическим пользовательским интерфейсом.

Разработка проводилась в среде программирования Visual Studio 2022. В качестве языка программирования был выбран C#. В качестве СУБД была выбрана PostgreSQL.

В результате проведенной работы разработано приложение, удовлетворяющее поставленной задаче, а именно: удобный пользовательский интерфейс, возможность взаимодействия с информацией троллейбусах в троллейбусном парке, автоматическое создание документов.

Созданное приложение внедрено в троллейбусный парк «Универсал Сервис» для автоматизации учета троллейбусов и планирования технического обслуживания и ремонта.

					ПГУ 1.090301.07.001 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата	Автоматизированная система учета состояния троллейбусного парка Пояснительная записка	Лит.	Лист	Листов
Разраб.		Зацепилина Е.С.		31.05.24				
Провер.		Карамышева Н.С.		31.05.24				
Н. Контр		Бычков А.С.		31.05.24			2	223
Утверд.		Митрохин М.А.		31.05.24		ФВТ, каф ВТ гр. 20BVB1		

Содержание

Перечень сокращений и обозначений.....	4
Введение.....	5
1 Автоматизированная система учета состояния троллейбусного парка..	6
1.1 Анализ предметной области.....	6
1.1.1 Анализ актуальности разработки.....	6
1.1.2 Обзор аналогов.....	7
1.1.3 Анализ требований к разработке.....	11
1.1.4 Постановка задачи.....	13
1.2 Выбор инструментов разработки.....	15
1.2.1 Архитектура приложения.....	15
1.2.2 Выбор языка программирования.....	15
1.2.3 Выбор СУБД.....	18
1.2.4 Выбор паттерна проектирования.....	23
1.2.5 Выбранные технологии.....	24
1.3 проектирование информационной системы.....	28
1.3.1 Проектирование базы данных.....	28
1.3.2 проектирование приложения	35
1.4 Реализация.....	37
1.4.1 Создание базы данных.....	37
1.4.2 Работа с базой данных и описание программы.....	39
1.5 Тестирование и отладка.....	43
2 Безопасность жизнедеятельности.....	46
2.1 Вред здоровью при работе с ПК.....	46
2.2 Снижение вреда для здоровья при работе с ПК.....	46
2.3 Техника безопасности при работе с компьютером.....	47
Заключение.....	48
Список использованных источников.....	49
Приложение А – UML-диаграммы.....	50
Приложение Б – Листинг.....	55
Приложение В – Презентация.....	211

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

АС – автоматизированная система

БД – база данных

ГОСТ – региональный стандарт, принятый Межгосударственным советом по стандартизации, метрологии и сертификации Содружества Независимых Государств

ПК – персональный компьютер

ПЭВМ – персональная электронно-вычислительная машина

СУБД – система управления базой данных

Фреймворк – программная платформа, которая упрощает разработку программного продукта, определяет структуру проекта

MVVM (Model View ViewModel) – шаблон проектирования архитектуры приложения

WPF (Windows Presentation Foundation) – система для построения клиентских приложений Windows с визуальными возможностями взаимодействия с пользователем

XAML – декларативный язык разметки

XLSX – формат файлов табличного типа

.NET – кроссплатформенная платформа для создания различных типов приложений

					ПГУ 1.090301.07.001 ПЗ	Лист
						4
Изм.	Лист	№ докум.	Подпись	Дата		

Введение

В настоящее время трудно представить нашу жизнь без информационных технологий. Вряд ли можно найти организацию, которая в той или иной степени не была бы оснащена средствами приёма, хранения, передачи, обработки информации.

Информация является одним из основных ресурсов роста производительности. Одним из видов программного обеспечения являются автоматизированные системы. Автоматизированная система состоит из компонентов, предназначенных для выполнения набора задач, которые были предварительно настроены. Функциональные и утомительные задачи становятся менее обременительными и делают рабочий процесс проще и легче.

Целью выпускной работы является разработка автоматизированной системы для учета троллейбусов в троллейбусном парке.

В ходе данной работы необходимо разработать приложение для автоматизации процесса учета троллейбусов в троллейбусном парке. Приложение должно иметь удобный графический интерфейс, для визуализации всей необходимой информации для учета троллейбусов. Входными данными являются информация о троллейбусах, а также об их состоянии в текущий момент, о сотрудниках предприятия, в том числе пользователей приложения, а также справочная информация о видах технического обслуживания. Результатом работы приложения является файл, содержащий в себе рекламационный акт о неисправности троллейбуса.

1 Автоматизированная система учета состояния троллейбусного парка

1.1 Анализ предметной области

1.1.1 Анализ актуальности разработки

Актуальность данной темы, прежде всего, обусловлена отсутствием собственной автоматизированной системы учета троллейбусов в троллейбусном парке «Универсал Сервис».

На данный момент алгоритм выполнения работ по ремонту транспортного средства выглядит так:

- при выходе находящегося на линии троллейбуса из строя, водитель сообщает об этом диспетчеру, при этом, в случае необходимости, сообщает свое местоположение и указывает необходимость вызова эвакуатора;

- диспетчер фиксирует в журнале время схода транспортного средства с линии, сообщает об этом мастеру, находящемуся на смене, который отвечает за доставку троллейбуса в парк;

- по прибытию транспортного средства в парк (своим ходом или с помощью эвакуатора) диспетчер сообщает об этом мастеру. Обязанности по выполнению работ с данной машиной переходят к мастеру;

- мастер проводит диагностику троллейбуса, фиксирует поломки и заносит все данные в свой журнал. Все эти данные должны перейти к сотруднику гарантийного отдела, который составляет рекламационный акт о необходимости выполнения работ по ремонту. Данный рекламационный акт отправляется на завод изготовитель;

- процессе отправки рекламационного акта или до отправки, мастер начинает выполнение ремонтных работ. Все этапы выполнения работ (принятие заявки на ремонт, начало выполнения ремонтных работ и конец выполнения) фиксируются в журнале. На основании данного журнала и записей мастера сотрудник гарантийного отдела проводит свою работу.

					ПГУ 1.090301.07.001 ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

Помимо информации о проведении ремонтных работ в бумажных журналах также фиксируется информация о проведении технического обслуживания, о пробеге транспортных средств, о статусе троллейбуса, таких как:

- на линии – машина работает на маршруте;
- сход – машина сошла с линии, но еще не прибыла в парк;
- в парке – машина прибыла в парк, но мастер еще не начал проводить ремонтные работы;
- ремонт – проводится ремонт троллейбуса;
- готово – ремонтные работы проведены, но троллейбус находится в состоянии ожидания;
- резерв – после проведения ремонтных работ троллейбус ушел в резерв;
- на линии.

Так как запись в журнал о техническом состоянии транспортного средства и оформление рекламационных актов производится вручную, присутствует большая трата времени на повторно выполняемую работу. Руководством предприятия было принято решение автоматизировать процесс данных работ, путем внедрения автоматизированной системы.

1.1.2 Обзор аналогов

Исходя из требований заказчика по работе программного обеспечения, были выбраны аналоги, которые позволяют вести учет транспортных средств, такие как:

- АС «ЗАВГАР.ОНЛАЙН»;
- АС «TIS-Online»;
- «1С:Предприятие 8. Управление автотранспортом Проф.»

Автоматизированная система «ЗАВГАР.ОНЛАЙН»[1]. Графический интерфейс приложения «ЗАВГАР.ОНЛАЙН» представлен на рисунке 1.1.

Согласно описанию автоматизированной системы, из основных особенностей данного аналога можно выделить удобный интерфейс и полную

сводную информацию о техническом обслуживании и ремонте транспортных средств.

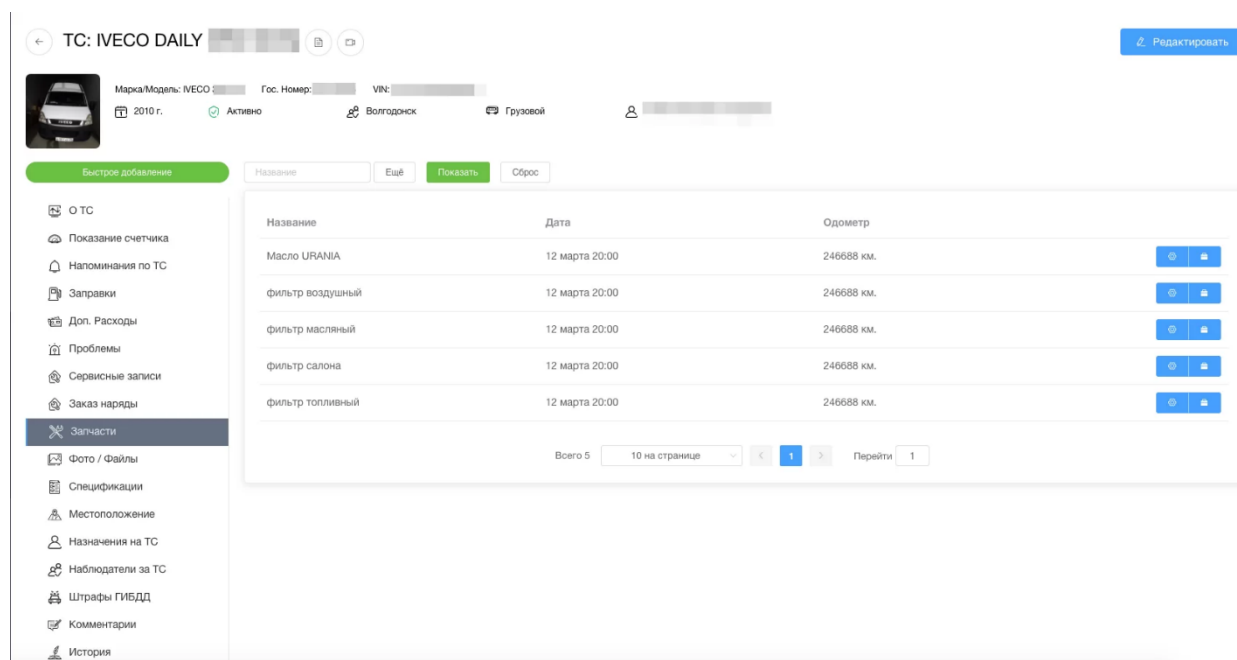


Рисунок 1.1 – Графический интерфейс приложения
«ЗАВГАР.ОНЛАЙН»

Из плюсов данного приложения можно выделить следующее:

- большой функционал;
- удобный и многофункциональный интерфейс;
- наличие большого функционала по созданию отчетов и графиков расходов.

Из минусов можно выделить следующее:

- неактуальность большинства функций данного приложения;
- отсутствие возможности создания документа по определенному шаблону;
- высокая стоимость автоматизированной системы (минимум 400 рублей в месяц за одно транспортное средство) и, соответственно, распространение на коммерческой основе;

Автоматизированная система «TIS-Online»[2]. Графический интерфейс приложения «TIS-Online» представлен на рисунке 1.2.

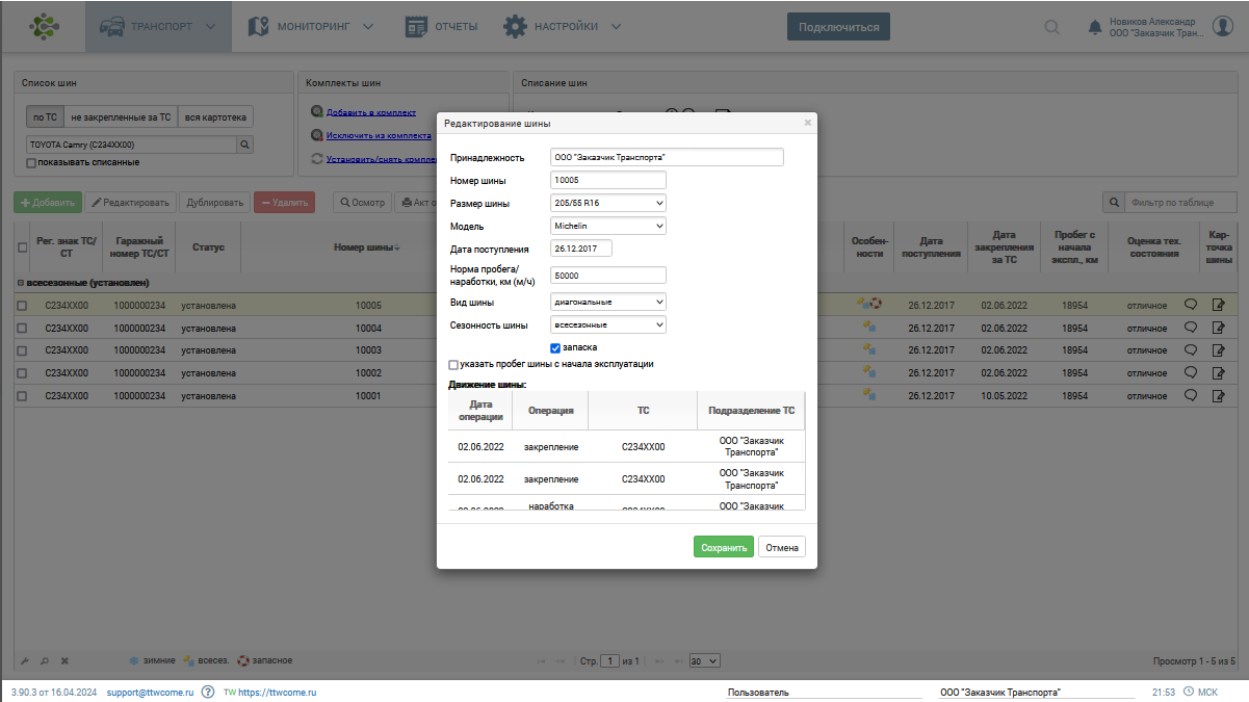


Рисунок 1.2 – Графический интерфейс приложения «TIS-Online»

TIS-Online – это система электронного документооборота, разработанная для управления и отслеживания грузовых перевозок. Она предоставляет возможность автоматизации процессов связанных с транспортировкой грузов, включая создание электронных накладных, отслеживание движения товаров, обмен информацией с поставщиками и партнерами.

С помощью TIS-Online компании могут значительно увеличить эффективность логистических операций, уменьшить риски потери грузов и оптимизировать расходы на перевозки. Система обеспечивает прозрачность и контроль над всем процессом доставки товаров, позволяя быстро реагировать на возникающие проблемы и уменьшать время доставки.

Минусы данной системы:

– работа с системой возможна только при наличии доступа в интернет;

- риск потери данных при взломе аккаунта злоумышленником;
- высокая стоимость системы (минимум 200 рублей в месяц за одно транспортное средство) и, соответственно, распространение на коммерческой основе.

Автоматизированная система «1С:Предприятие 8. Управление автотранспортом Проф.»[3] Графический интерфейс приложения «1С:Предприятие 8. Управление автотранспортом Проф.» представлен на рисунке 1.3.

The screenshot displays the '1С:Предприятие 8. Управление автотранспортом Проф.' application window. The title bar reads '★ Транспортное средство: к 745 ое 777 (Транспортные средства и оборудован...'. The interface includes a top navigation bar with tabs: 'Основное', 'Доп. сведения о ТС', 'История изменения статуса блокировки топливных карт', and 'Еще...'. Below the navigation bar are buttons: 'Записать и закрыть', 'Записать', 'Заполнить по модели', 'Настройки', and 'Еще - ?'. The main form contains fields for vehicle identification: 'Наименование: к 745 ое 777', 'Гос. номер: к 745 ое 777', 'Гаражный номер: 2', 'Марка: Газель ГАЗ-3302', and 'Тип ТС: Грузовые фургоны'. A 'Принадлежность' section has radio buttons for 'Собственное', 'Привлеченное', and 'Виртуальное'. On the left is a sidebar menu with options like 'Основное', 'Доп. параметры', 'Маршрутизация', 'Маршруты', 'Состав ТС', 'Документы', 'Экипаж', 'Агрегаты', 'Местонахождение', 'Топливные карты', 'Нормы расхода ГСМ', 'Нормы расхода ТЖ', 'Мониторинг', and 'Бух. учет'. The right pane shows 'Текущее местонахождение' with fields for 'Организация: Мегатранс ООО', 'Подразделение: Транспортный отдел', 'Колонна: Грузовые автомобили', 'Дата ввода: 28.02.2018', 'Дата выбытия:', and 'Начальный пробег, км: 1 000,0 ?'. Below this is a 'Лицензионная карточка' section with 'Серия' and '№' fields, and a 'Дополнительная информация' section at the bottom.

Рисунок 1.3 – Интерфейс программы «1С:Предприятие 8. Управление автотранспортом Проф.»

Система имеет удобную систему учета и распределения заказов на автомобили:

- заказы на автотранспорт могут учитываться от внутренних подразделений компании и от сторонних контрагентов;
- система автоматически изменяет статус и состояние заказов и грузов при выполнении различных операций с грузом;

- есть учет сопроводительных документов на груз, дополнительных услуг и операций;
- можно фиксировать плановые и фактические доходы и расходы по заказу.

Минусы системы:

- высокая стоимость лицензии (88500 рублей или 1860 рублей в месяц);
- платное внедрение дополнительного функционала;
- невозможность создания документов по определенному шаблону.

1.1.3 Анализ требований к разработке

Исходя из описания алгоритма выполнения работ по ремонту транспортного средства на предприятии «Универсал Сервис», который описан в пункте 1.1.1, а также из описания заказчика, алгоритм работы программного обеспечения должен выглядеть так:

- с помощью программы диспетчер вносит информацию о выходе транспортного средства из строя (указывается описания неисправности со слов водителя и необходимость вызова эвакуатора). Статус транспортного средства меняется на «Сход»;
- механику, у которого в данный момент запущено программное обеспечение, должно прийти уведомление, в котором указывается гаражный номер транспортного средства, дата и время схода с линии, необходимость вызова эвакуатора и неисправность со слов водителя. Механик подтверждает уведомление, в базе данных фиксируется дата и время подтверждения;
- в момент прибытия троллейбуса в парк, диспетчер фиксирует дату и время. Механику приходит уведомление о прибытии транспортного средства в парк. Статус транспортного средства меняется на «В парке»;
- механик проводит диагностику транспортного средства и фиксирует дату и время начала ремонтных работ. Статус транспортного средства меняется на «Ремонт»;

– после проведения ремонта механик фиксирует дату и время конца ремонтных работ. Статус транспортного средства меняется на «Готово»;

– сотрудник гарантийного отдела с помощью программного обеспечения отслеживает всю информацию, заносимую диспетчером и механиком. Ему приходят уведомления об изменении статуса транспортного средства. Он может указать на запрет эксплуатации троллейбуса, с помощью программы составить рекламационный акт, вручную изменить статус транспортного средства, а также вносить всю необходимую информацию. Помимо этого, сотрудник гарантийного отдела, как и механик, может внести информацию о поломке транспортного средства, если поломка была обнаружена до выхода на линию.

Автоматизированная система должна предполагать четыре режима работы. В зависимости от должности сотрудника, программного обеспечения должно предоставлять разный функционал:

– режим диспетчера – в данном режиме сотрудник может просматривать информацию о всех транспортных средствах, вручную просматривать состояние транспортных средств, создавать заявки о сходе транспортных средств, отменять эти заявки, фиксировать факт прибытия транспортного средства в парк;

– режим механика – в данном режиме сотрудник может изменять статус транспортного средства, фиксировать дату и время принятия заявки, начала и конца проведения ремонтных работ, вносить информацию о проведении технического обслуживания;

– режим руководителя – в данном режиме сотрудник может просматривать всю информацию, которая хранится в базе данных, но при этом не может изменять её или удалять;

– режим сотрудника гарантийного отдела – в данном режиме сотрудник может добавлять, удалять и изменять любую информацию. Также

сотрудник может составить рекламационный акт на основании данных, заполненных механиком.

1.1.4 Постановка задачи

Исходя требований к разработке, описанных в пункте 1.1.3, для проектируемой автоматизированной системы была сформирована диаграмма вариантов использования, которая изображена на рисунке 1.4.

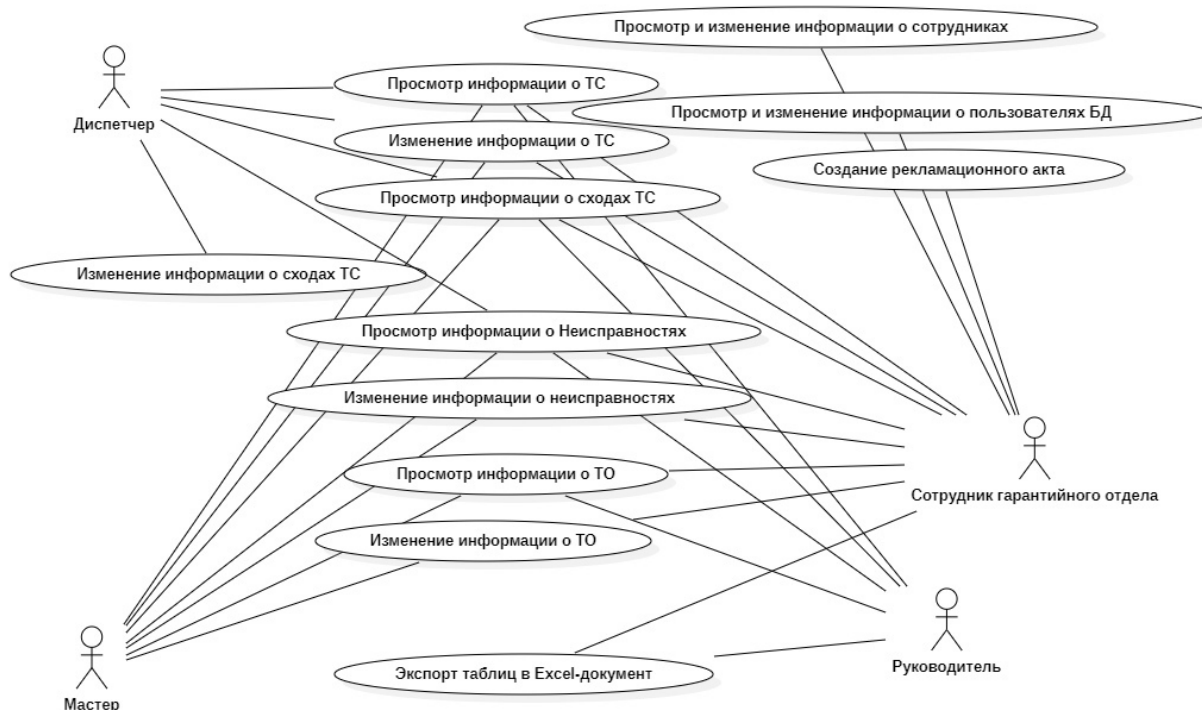


Рисунок 1.4 – UML диаграмма вариантов использования

Требуется разработать приложение, которое должно советовать следующим целям:

- автоматизирование процесса учета троллейбусов и их состояния на текущий момент времени;
- создание рекламационных актов в формате docx;
- удобный и интуитивно понятный графический интерфейс для работы пользователя с приложением;

– расположение графических элементов должно предоставлять пользователю быстрое ориентирование в интерфейсе;

– возможность масштабирования интерфейса в процессе работы.

В процессе разработки должны быть решены следующие задачи:

– отображение и хранение информации о троллейбусах (номер троллейбуса, государственный номер, VIN-номер, текущее состояние, дата ввода в эксплуатацию, год выпуска, марка, текущий статус, наличие автохода);

– отображение и хранение информации о сотрудниках предприятия (ФИО, должность), а также о пользователях автоматизированной системы (сотрудник и закрепленный за ним пароль);

– отображение и хранение информации о заявках диспетчера о сходе троллейбуса с линии (время схода, причина, необходимость эвакуатора, время отмены заявки и причина отмены (в случае отмены), время прибытия транспортного средства в парк, диспетчер);

– отображение и хранение информации о заявках мастера на ремонт (пробег на момент поломки транспортного средства, время подтверждения заявки диспетчера, время начала ремонта, время конца ремонта, информация о неисправности, мастер и слесари, выполняющие работу, комментарий к поломке, запрет на эксплуатацию транспортного средства);

– отображение и хранение информации о техническом обслуживании троллейбусов (вид ТО, пробег на момент ТО, дата начала ТО, слесари выполняющие ТО и мастер, вид следующего ТО и пробег для следующего ТО);

– отображение и хранение справочной информации о видах должностей, видах ТО, типах состояний троллейбусов;

– создание рекламационного акта по запросу пользователя на каждую неисправность транспортного средства. Шаблон рекламационного акта представлен на рисунке 1.5;

– уведомление пользователей об изменении состояния троллейбусов;

					ПГУ 1.090301.07.001 ПЗ	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

- разграничить доступ к добавлению, изменению, удалению и просмотру информации базы данных в зависимости от должности пользователя приложения;
- блокировать некорректный ввод данных пользователем.

**Общество с ограниченной ответственностью
«Универсал Сервис»**

440028, г. Пенза, проспект Победы, 65, офис 2
Тел: 8 (8412) 42-13-07
ИНН 5835137029
E-mail: universalservis58@mail.ru

	АО «Уфимский трамвайно-троллейбусный завод» РФ, Уфа, пр-т Октября, 143 servisuttz@mail.ru , uttz@mail.ru
	Директору ГКУ «Организатор перевозок Пензенской области» Л.В. Панковой 440066, г. Пенза, пр. Победы, 150 gku_oppo@mail.ru

РЕКЛАМАЦИОННЫЙ АКТ №
на троллейбус пассажирский мод. 6241.01 «Горожанин»

1.	Дата составления	
2.	Заводской № (VIN) №	
3.	Инвентарный (городской) №	
4.	Дата ввода в эксплуатацию	
5.	Дата обнаружения дефекта	
6.	Сведения об отстранении от эксплуатации	
7.	Общий пробег на момент обнаружения дефекта	
8.	Сведения о проведении ТО: дата и пробег	
9.	Описание и условия проявления дефекта	
10.	Дополнительная информация	--
11.	Приложения	Копия сервисной книжки (Приложение 1)
12.	Дилер, у которого приобретён троллейбус	АО «Уфимский трамвайно-троллейбусный завод»

-- «_____» 20__

М.П.

_____ А.Ю. Кузнецов

Рисунок 1.5 – Шаблон рекламационного акта

1.2 Выбор инструментов разработки

1.2.1 Архитектура приложения

Приложение должно быть построено по архитектуре «клиент-сервер». Данная архитектура была выбрана, поскольку данная автоматизированная система будет использоваться на нескольких ЭВМ, которые соединены в одну локальную сеть. На одном из компьютеров будет развернут сервер базы данных PostgreSQL, который будет выполнять функции серверной части

					ПГУ 1.090301.07.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

автоматизированной системы. Диаграмма развертывания автоматизированной системы представлена на рисунке 1.6.

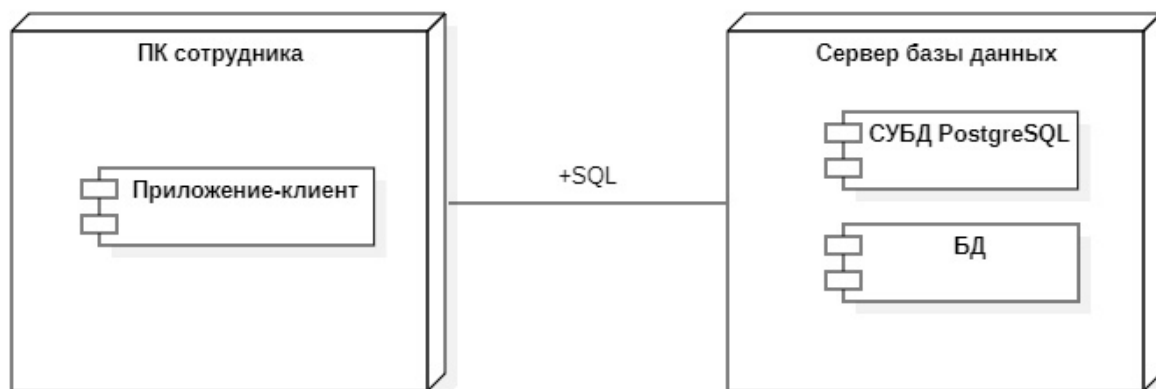


Рисунок 1.6 – Диаграмма развертывания автоматизированной системы

1.2.2 Выбор языка программирования

– Язык программирования C++ – объектно-ориентированный, компилируемый, строго типизированный язык программирования. C++ поддерживает различные способы программирования, такие как процедурный, объектно-ориентированный, функциональный и т. д. Это делает C++ мощным и гибким. Но C++ не является достаточно гибким для создания графического интерфейса.

– Язык программирования Java – кроссплатформенный строго типизированный, объектно-ориентированный язык программирования. Кроссплатформенность достигается использованием виртуальной машины `jvm – java virtual machine`. Также данный язык имеет библиотеки, которые подходят для работы с файлами формата `docx`. К недостаткам можно отнести то, что при работе с библиотеками для создания оконных приложений, процесс создание сложных, настраиваемых интерфейсов может стать довольно долгим.

– Язык программирования Python – объектно-ориентированный интерпретируемый язык программирования высокого уровня с динамической

типизацией. Он имеет очень простой синтаксис, который упрощает изучение этого языка, а также повышает скорость разработки.

Интерпретатор Python и стандартные библиотеки распространяются бесплатно, поскольку они доступны в исходном или двоичном виде, а отладка программ на Python очень проста благодаря их функциям.

Несмотря на явные достоинства Python, у него есть и недостатки, о которых следует помнить. Python. Это интерпретируемый скриптовый язык, поэтому он работает относительно медленнее своих скомпилированных аналогов. Также у существующих библиотек для графического интерфейса сложно модифицировать макет приложений.

– Язык программирования C# – это язык программирования, разработанный и запущенный Microsoft в 2001 году. C# – строго типизированный объектно-ориентированный язык программирования, который предоставляет современным разработчикам гибкость и лёгкость для создания программного обеспечения.[4]

Данный язык программирования имеет открытый исходный код в рамках .NET Foundation. Благодаря открытости, есть множество сообществ, активно занимающихся улучшением языка.

Поскольку C# и Windows разработаны Microsoft, разработчики могут легко создавать настольные приложения на основе Windows.

Также можно добавить к описанным выше преимуществам наличие библиотек для автоматизации работы в Excel и Word. Данные библиотеки позволяют выполнять такие действия, как импорт данных из таблиц, а также создание документа в формате docx по определённому шаблону. Все действия, которые можно выполнить вручную через пользовательский интерфейс в приложениях Microsoft Office, также можно выполнить программным путем через данные библиотеки.

По результатам анализа существующих аналогов было принято решение разработать автоматизированную систему, которая будет решать задачи по

					ПГУ 1.090301.07.001 ПЗ	Лист
						17
Изм.	Лист	№ докум.	Подпись	Дата		

хранению и отображению информации о транспортных средствах троллейбусного парка, а также предоставлять функционал по проведению манипуляций добавления, удаления и изменения этих данных. Выходными данными данного приложения будут являться рекламационные акты, составленные на основании данных, введённых пользователями автоматизированной системы.

Разработка будет производиться с помощью языка C#, так как C# позволяет создавать удобный и понятный для пользователя графический интерфейс.

1.2.3 Выбор СУБД

Неудивительно, что корпорация Oracle предлагает одноимённый продукт, с которого обычно начинается рассмотрение вариантов популярных СУБД. Первая версия Oracle была создана в конце 70-х годов. На данный момент у этого продукта блестящая репутация. Кроме того, существует несколько версий этого продукта для удовлетворения потребностей конкретной организации.

Актуальная версия Oracle предназначена для облачных сред и может быть размещена на одном или нескольких серверах, это позволяет управлять базами данных, которые содержат миллиарды записей. Некоторые из функций новейшей версии Oracle включают в себя grid framework и использования как физических, так и логических структур.

Это означает, что физическое управление данными не влияет на доступ к логическим структурам. Кроме того, безопасность в этой версии доведена до высочайшего уровня, потому что каждая транзакция изолирована от других.

Достоинства:

– Самые свежие инновации и впечатляющий функционал уже внедрены в этом продукте, поскольку компания Oracle стремится держать планку даже на фоне других разработчиков СУБД;

– Оракул является крайне надёжной, фактически это эталон надёжности среди подобных систем.

Недостатки:

– Стоимость Oracle может оказаться непомерно высокой, особенно для небольших организаций;

– Система может потребовать значительных ресурсов уже сразу после установки, поэтому возможно потребуется модернизировать оборудование для внедрения Oracle.

Идеально подходит для крупных организаций, которые работают с огромными базами данных и разнообразными функциями.

MySQL - одна из самых популярных СУБД для веб-приложений. Фактически, является стандартом для веб-серверов, которые работают под управлением операционной системы Linux. MySQL - это бесплатный пакет программ, однако новые версии выходят постоянно, расширяя функционал и улучшая безопасность. Существуют специальные платные версии, предназначенные для коммерческого использования. В бесплатной версии наибольший упор делается на скорость и надежность, а не на полноту функционала, который может стать и достоинством и недостатком - в зависимости от области внедрения.

Разработку и поддержку MySQL осуществляет корпорация Oracle, получившая права на торговую марку вместе с поглощённой Sun Microsystems, которая ранее приобрела шведскую компанию MySQL AB. Продукт распространяется как под GNU General Public License, так и под собственной коммерческой лицензией. Помимо этого, разработчики создают функциональность по заказу лицензионных пользователей. Именно благодаря такому заказу почти в самых ранних версиях появился механизм репликации.

Эта СУБД позволяет выбирать различные движки для системы хранения, которые позволяют менять функционал инструмента и выполнять обработку данных, хранящихся в различных типах таблиц. Гибкость СУБД

MySQL обеспечивается поддержкой большого количества типов таблиц: пользователи могут выбрать как таблицы типа MyISAM, поддерживающие полнотекстовый поиск, так и таблицы InnoDB, поддерживающие транзакции на уровне отдельных записей. Более того, СУБД MySQL поставляется со специальным типом таблиц EXAMPLE, демонстрирующим принципы создания новых типов таблиц. Благодаря открытой архитектуре и GPL-лицензированию, в СУБД MySQL постоянно появляются новые типы таблиц. Она также имеет простой в использовании интерфейс, и пакетные команды, которые позволяют удобно обрабатывать огромные объемы данных. Система невероятно надежна и не стремится подчинить себе все доступные аппаратные ресурсы.

Достоинства:

- Распространяется бесплатно
- Прекрасно документирована
- Предлагает много функций, даже в бесплатной версии
- Пакет MySQL включен в стандартные репозитории наиболее распространённых дистрибутивов операционной системы Linux, что позволяет устанавливать её элементарно просто
- Поддерживает набор пользовательских интерфейсов
- Может работать с другими базами данных, включая DB2 и Oracle.

Недостатки:

- Придётся потратить много времени и усилий, чтобы заставить MySQL выполнять несложные задачи, хотя другие системы делают это автоматически, например: создавать инкрементные резервные копии.
- Отсутствует встроенная поддержка XML или OLAP;
- Для бесплатной версии доступна только платная поддержка;

Идеально подходит для: организаций, которым требуется надежный инструмент управления базами данных, но бесплатный.

Ещё одной из популярных СУБД является программный продукт Microsoft SQL-сервер. Это система управления базами данных, движок которой работает на облачных серверах, а также локальных серверах, причем можно комбинировать типы применяемых серверов одновременно. Вскоре после выпуска Microsoft SQL Server 2016, Microsoft адаптировала продукт для операционной системы Linux, а на платформе Windows он работал изначально.

Одной из уникальных особенностей версии 2016 года является temporal data support (временная поддержка данных), которая позволяет отслеживать изменения данных с течением времени. Последняя версия Microsoft SQL-сервер поддерживает dynamic data masking (динамическую маскировку данных), которая гарантирует, что только авторизованные пользователи будут видеть конфиденциальные данные.

Достоинства:

- Продукт очень прост в использовании;
- Текущая версия работает быстро и стабильно;
- Движок предоставляет возможность регулировать и отслеживать уровни производительности, которые помогают снизить использование ресурсов;
- Вы сможете получить доступ к визуализации на мобильных устройствах;
- Он очень хорошо взаимодействует с другими продуктами Microsoft.

Недостатки:

- Цена для юридических лиц оказывается неприемлемой для большей части организаций.
- Даже при тщательной настройке производительности SQL Server способен задействовать все доступные ресурсы.
- Сообщается о проблемах с использованием службы интеграции для импорта файлов.

– Есть смысл покупать лицензию на этот продукт, если уже внедрена (читай "куплена") экосистема Microsoft.

Идеально подходит для: крупных организаций, которые уже используют ряд продуктов Microsoft.

PostgreSQL является одним из нескольких бесплатных популярных вариантов СУБД, часто используется для ведения баз данных веб-сайтов. Это весьма старая система, поэтому в настоящее время она хорошо развита, и позволяет пользователям управлять как структурированными, так и неструктурированными данными. Может быть использована на большинстве основных платформ, включая Linux (где особенно хорошо проявляется производительность). Прекрасно справляется с задачами импорта информации из других типов баз данных с помощью собственного инструментария.

Движок БД может быть размещен в ряде сред, в том числе виртуальных, физических и облачных. Самая свежая версия предлагает обработку больших объемов данных и увеличение числа одновременно работающих пользователей. Безопасность была улучшена благодаря поддержке DBMS_SESSION.

Достоинства:

- Является масштабируемым решением и позволяет обрабатывать терабайты данных;
- Поддерживает формат json;
- Существует множество predefined функций;
- Доступен ряд интерфейсов.

Недостатки:

- Документация туманна, поэтому, возможно, ответы на некоторые вопросы придется искать в интернете;
- Конфигурация может смутить неподготовленного пользователя;

– Скорость работы может падать во время проведения пакетных операций или выполнения запросов чтения.

Идеально подходит для организаций с ограниченным бюджетом, но требует привлечения квалифицированных специалистов, когда требуется возможность выбрать уникальный интерфейс и использовать json.

В качестве системы управления базой данных было решено выбрать СУБД PostgreSQL — одна из самых функциональных реляционных СУБД, направленная, прежде всего, на соответствие стандартам ANSI/ISO и расширяемость.

Главное отличие PostgreSQL от других СУБД — наличие объектно-ориентированных функциональных возможностей, среди которых поддержка концепции ACID (Atomicity, Consistency, Isolation, Durability — атомарность, согласованность, изолированность, долговечность).

PostgreSQL способна справляться с одновременной обработкой нескольких задач, поддерживает большое количество типов данных и вспомогательных инструментов для работы.

1.2.4 Выбор паттерна проектирования

При разработке приложения основным паттерном разработки был выбран MVVM. Паттерн MVVM (Model-View-ViewModel) позволяет отделить логику приложения от визуальной части (представления). Данный паттерн является архитектурным, то есть он задает общую архитектуру приложения. [5][6]

Данный паттерн был представлен Джоном Госсманом в 2005 году как модификация шаблона Presentation Model и был первоначально нацелен на разработку приложений в WPF. И хотя сейчас данный паттерн вышел за пределы WPF и применяется в самых различных технологиях, в том числе при разработке под Android, iOS, тем не менее WPF является довольно показательной технологией, которая раскрывает возможности данного паттерна.

Паттерн можно разделить на три отдельных компонента:

- представление (View), которое инкапсулирует пользовательский интерфейс и логику пользовательского интерфейса;
- модель представления (ViewModel), которая инкапсулирует логику представления и состояние;
- модель (Model), которая инкапсулирует различные данные приложения.

На рисунке 1.7 показаны три компонента, использующиеся в паттерне MVVM и отображено взаимодействие между ними.

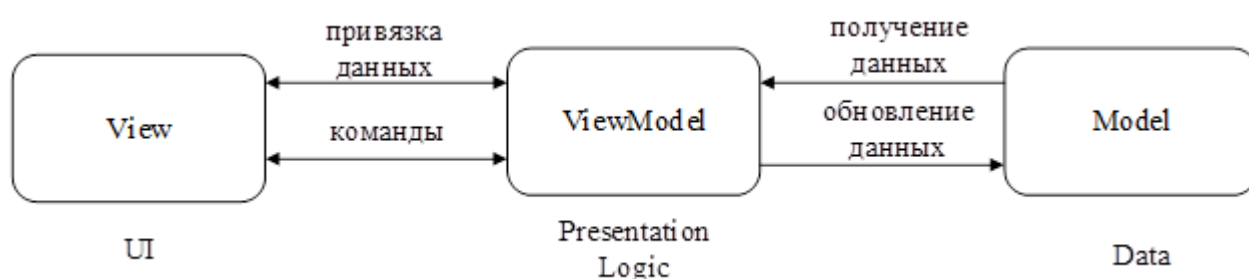


Рисунок 1.7 – Визуализация взаимодействия компонентов при использовании паттерна MVVM

1.2.5 Выбранные технологии

1.2.5.1 Выбор платформы для реализации графического интерфейса

Windows Forms — интерфейс программирования приложений (API), отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework. Данный интерфейс упрощает доступ к элементам интерфейса Microsoft Windows за счет создания обёртки для существующего Win32 API в управляемом коде. Причём управляемый код — классы, реализующие API для Windows Forms, не зависят от языка разработки. То есть программист одинаково может использовать Windows Forms как при написании ПО на C#, C++, так и на VB.Net, J# и др.

Преимущества WinForms:

- Технология более старая и, соответственно, лучше испытанная и протестированная.

– На данный момент существует огромное множество готовых элементов управления, которые можно купить либо использовать бесплатно.

– С точки зрения написания, дизайнер Visual Studio лучше приспособлен к WinForms, так как в WPF больше необходимо делать самому.

Windows Presentation Foundation – это платформа для создания приложений Windows, которая позволяет разрабатывать многофункциональные пользовательские интерфейсы с меньшей сложностью кода.

Основной особенностью данной платформы является наличие векторного движка рендеринга, который использует аппаратное ускорение современных графических видеокарт, что делает графический интерфейс более быстрым и легко масштабируемым.

WPF можно рассматривать как эволюцию Windows Forms. Платформа предлагает легкое разделение между пользовательским интерфейсом и функциональной логикой.

Все графические элементы определяются с помощью языка XAML. XAML – это язык декларативной разметки на основе XML для указания и установки характеристик классов. Синтаксический анализатор XAML будет автоматически анализировать, и создавать фактический объект во время визуализации приложения. С помощью данного языка определяется весь макет пользовательского интерфейса.

Также основными преимуществами WPF при создании графического интерфейса являются:

– стили WPF, которые позволяют устанавливать определённые свойства для графических элементов управления и повторно использовать эти конкретные настройки для нескольких элементов, для придания единообразного внешнего вида всему приложению;

– шаблоны элементов, позволяющие определять внешний вид элемента управления. Например, внешний вид кнопки по умолчанию можно изменить,

сделав кнопку круглой, с наличием картинки и определенного текста, изменится лишь только внешний вид, а поведение кнопки останется прежним;

— механизм привязки данных, который позволят выполнять переноса данных из кода на уровень пользовательского интерфейса или наоборот.

В качестве графической платформы приложения была выбрана платформа WPF.

Исходя из вышеописанных преимуществ и особенностей, платформа WPF удовлетворяет требованиям для создания графического интерфейса разрабатываемой автоматизированной системы.

1.2.5.2 Библиотека для взаимодействия с СУБД

В нашем случае, в качестве модели данных будут выступать классы, которые описывают сущности из базы данных. Для этого было принято решение применить библиотеку Entity Framework. Данная библиотека позволяет производить обратное проектирование, при котором база данных создается автоматически уже после того, как были описаны классы на языке программирования. В нашем же случае данный подход не имеет смысла, поскольку для автоматического обновления статусов транспортных средств необходимо применять триггерные функции, которые можно создать только с помощью СУБД. Но, помимо этого, Entity Framework предлагает автоматическое создание классов для каждой из таблиц, а также создание контекста для лёгкого подключения к базе данных. Данный подход позволит значительно сократить время разработки, поскольку не будет необходимости заново описывать на языке программирования то, что уже было описано в базе данных.

Более того, данный фреймворк позволяет полностью отказаться от ручного написания SQL-запросов, поскольку формат использования контекста предлагает в качестве возвращаемого значения нумеруемые списки. Это позволяет воспользоваться возможностями библиотеки LINQ, которые полностью покрывают все потребности при осуществлении доступа к данным

в разрабатываемой автоматизированной системе. Данная библиотека предлагает множество методов для сортировки и фильтрации нумерованных структур данных, при этом позволяя осуществить гораздо более сложные запросы с меньшими затратами по времени, чем с применением SQL-оператора SELECT.

1.2.5.3 Выбор библиотек для реализации интерфейса

Для реализации интерфейса было принято решение выбрать библиотеку Handy Control – это свободно распространяемая библиотека элементов управления WPF, которая содержит в себе до 80 дополнительных элементов управления, а также расширяет возможности использования нативных элементов управления. Более того, данная библиотека расширяет возможности использования команд – в WPF это механизм выполнения определенной задачи при выполнении события (пример: нажатие кнопки, наведение мыши на элемент управления). Команды позволяют полностью отказаться от обработчиков событий, что в свою очередь позволяет: во-первых, применять одни и те же команды повторно при выполнении разных событий с разными элементами управления, а во-вторых, позволяет описывать логику работы событий в части ModelView. Данный подход позволит в любой момент жизненного цикла программы полностью переписать интерфейс автоматизированной системы без необходимости создавать новые обработчики событий, поскольку они уже были описаны в части ModelView и к ним всего лишь нужно осуществить привязку.

Для улучшения интерфейса также было принято решение использовать свободно распространяемую библиотеку иконок MahApps.Metro.IconPacks. Данная библиотека содержит более 20 стилей и более 10000 видов иконок. Наличие визуального отображения назначения той или иной кнопки позволит сократить время обучения будущих пользователей автоматизированной системы, поскольку интерфейс станет более интуитивно понятным.

					ПГУ 1.090301.07.001 ПЗ	Лист
						27
Изм.	Лист	№ докум.	Подпись	Дата		

1.3 Проектирование информационной системы

1.3.1 Проектирование базы данных

ER-модель (от англ. Entity-Relationship model, модель «сущность — связь») — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование схемы, созданной на основе ER-модели, в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма «сущность-связь» (англ. Entity-Relationship diagram, ERD, ER-диаграмма).[7]

Диаграмма отношений сущностей UML-ER диаграмма базы данных представлена на рисунке 1.7.

1.3.1.1 Описание сущностей базы данных

tv_statuses – справочник статусов транспортного средства. Описание атрибутов данной сущности представлено в таблице 1.1.

Таблица 1.1 — Описание атрибутов сущности tv_statuses

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер статуса	+	-
status_name	Название статуса	-	-

Таблица 1.3 — Описание атрибутов сущности to_type

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер типа	+	-
name	Название типа	-	-

employee – сотрудники парка. Описание атрибутов данной сущности представлено в таблице 1.4.

Таблица 1.4 — Описание атрибутов сущности employee

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер сотрудника	+	-
name	Имя	-	-
lastname	Фамилия	-	-
otchestvo	Отчество	-	-
post	Должность	-	+
park	Парк	-	+

db_users – пользователи автоматизированной системы. Описание атрибутов данной сущности представлено в таблице 1.5.

Таблица 1.5 — Описание атрибутов сущности db_users

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер пользователя БД	+	-
password	Пароль для входа в приложение	-	-
employee	Сотрудник	-	+

to_tv – записи о проведении технического обслуживания. Описание атрибутов данной сущности представлено в таблице 1.6.

Таблица 1.6 — Описание атрибутов сущности to_tv

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер ТО	+	-
to_type	Тип ТО	-	+
probeg	Пробег, на котором выполнялось ТО	-	-
complete_date	Дата выполнения ТО	-	-
master	Мастер ответственный за проведение ТО	-	+
tv_number	Номер троллейбуса	-	+
next_to	Тип следующего ТО	-	+
next_to_probeg	Пробег для следующего ТО	-	-

to_tv_executor – сущность, предназначенная для реализации отношения «многие ко многим» между сущностями to_tv и employee (сотрудники, которые участвовали в проведении ТО). Описание атрибутов данной сущности представлено в таблице 1.7.

Таблица 1.7 — Описание атрибутов сущности to_executor

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер связи	+	-
employee	Сотрудник	-	+
to_tv	Проведение ТО	-	+

shod – записи о сходах транспортного средства с линии. Описание атрибутов данной сущности представлено в таблице 1.8.

Таблица 1.8 — Описание атрибутов сущности shod

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер схода	+	-
tv_number	Номер троллейбуса	-	+
departure_time	Время схода	-	-
malfunction	Неисправность	-	-
evacuator	Нужен ли эвакуатор	-	-
cancel_comment	Комментарий, в случае отмены	-	-
cancel_time	Время отмены	-	-
arrival	Время прибытия в парк	-	-
dispatcher	Диспетчер, отметивший сход	-	+

park – список троллейбусных парков. Описание атрибутов данной сущности представлено в таблице 1.9.

Таблица 1.9 — Описание атрибутов сущности park

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер парка	+	-
name	Название парка	-	-

transport_venichle – список транспортных средств. Описание атрибутов данной сущности представлено в таблице 1.10.

Таблица 1.10 — Описание атрибутов сущности transport_venichle

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
number	Уникальный номер троллейбуса	+	-

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
mark	Марка троллейбуса	-	-
gov_number	Гос. номер	-	-
vin	VIN-номер	-	-
commissioning_date	Дата ввода в эксплуатацию	-	-
status	Статус троллейбуса	-	+
autohod	Наличие автохода	-	-
probeg	Пробег	-	-
release_year	Год выпуска	-	-
park	Парк	-	+

malfunction – список неисправностей транспортных средств. Описание атрибутов данной сущности представлено в таблице 1.11.

Таблица 1.11 — Описание атрибутов сущности malfunction

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер неисправности	+	-
zav_disp	Сход	-	+
tv_number	Номер троллейбуса	-	+
probeg	Пробег	-	-
accept_time	Время приема заявки мастером	-	-
begin_time	Время начала ремонта	-	-
ready_time	Время окончания ремонта	-	-
master	Мастер	-	+
malfunction	Неисправность, описанная мастером	-	-
zapret_expluatation	Есть ли запрет на эксплуатацию с этой неисправностью	-	-

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
work_comment	Комментарий мастера по проведению работ	-	-
tech_parts	Запчасти	-	-
reklamation	Номер рекламационного акта	-	-
malfunction_recl	Неисправность, описанная работником гарантийного отдела	-	-
malfunction_recl_comment	Комментарий работника гарантийного отдела	-	-
signed	Подписан ли рекламационный акт	-	-
done	Готово ли	-	-

malfunction_executor – сущность, предназначенная для реализации отношения «многие ко многим» между сущностями malfunction и employee (сотрудники, которые участвовали в проведении ремонта неисправности транспортного средства). Описание атрибутов данной сущности представлено в таблице 1.12.

Таблица 1.12 — Описание атрибутов сущности malfunction_executor

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер связи	+	-
employee	Сотрудник	-	+
malfunction	Неисправность	-	+

notifications – таблица уведомлений. Описание атрибутов данной сущности представлено в таблице 1.13.

Таблица 1.13 — Описание атрибутов сущности notifications

Имя атрибута	Назначение	Первичный ключ	Внешний ключ
id	Уникальный номер уведомления	+	-
message	Сообщение	-	-
posts	Массив номеров должностей, которым должно прийти сообщение	-	-
parks	Массив номеров парков, сотрудникам которых должно придти сообщение	-	-

1.3.2 Проектирование приложения

Для промежуточного хранения информации из базы данных в процессе работы приложения используются классы моделей. Они представляют из себя описание сущностей из базы данных. При обращении к базе с помощью библиотеки Entity Framework Core, создаются объекты данных классов. Диаграмма классов для Model представлена на рисунке 1.8.

Графический интерфейс должен предоставлять возможности для взаимодействия с приложением в соответствии с UML диаграммой вариантов использования, приведенной на рисунке 1.1.

Графический интерфейс состоит из разных элементов - представлений (View), каждый из которых предназначен для определённого взаимодействия с пользователем:

- EditDbUser – представление позволяющее добавлять или изменять информацию о пользователе базы данных;
- EditEmployee – представление, позволяющее добавлять или изменять информацию о сотруднике;

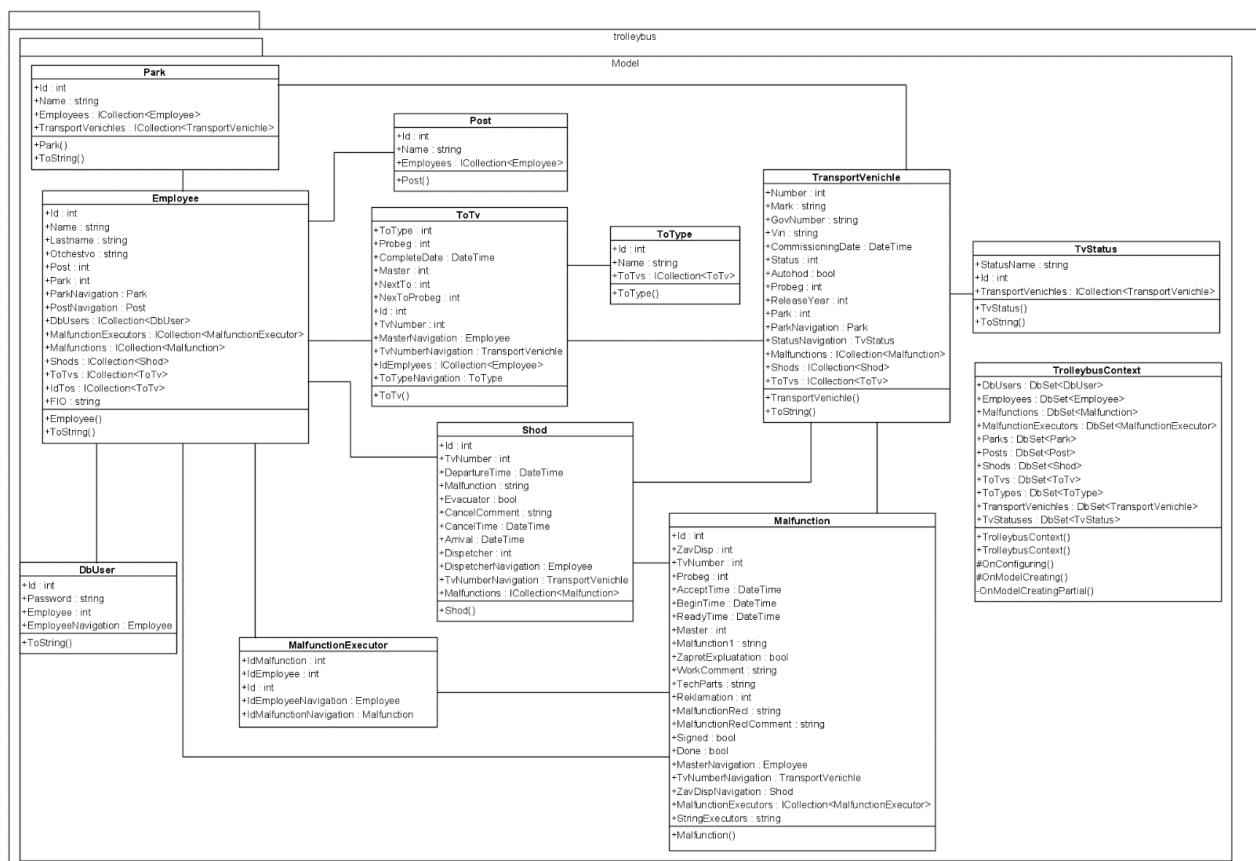


Рисунок 1.8 – Диаграмма классов для моделей

– EditMalfunction – представление позволяющее добавлять или изменять информацию о неисправности транспортного средства;

– EditPark – представление позволяющее добавлять или изменять информацию о троллейбусном парке;

– EditShod – представление позволяющее добавлять или изменять информацию о сходе транспортного средства с линии;

– EditTv – представление позволяющее добавлять или изменять информацию о транспортном средстве;

EditToTv – представление позволяющее добавлять или изменять информацию о техническом обслуживании;

– DispatcherWindow – представление главного окна приложения в режиме диспетчера;

– MasterWindow – представление главного окна приложения в режиме мастера;

- GuaranteeWindow – представление главного окна приложения в режиме сотрудника гарантийного отдела;
- RukovoditelWindow – представление главного окна приложения в режиме руководителя.

Для реализации связи между интерфейсом и моделями разработаны классы, которые предоставляет интерфейсу обновляемые свойства. Класс MainWindowViewModel будет являться контекстом данных для модулей интерфейса главной формы, а именно DispatcherWindow, MasterWindow, GuaranteeWindow и RukovoditelWindow.

Классы EditDbUserViewModel, EditEmployeeViewModel, EditMalfunctionViewModel, EditParkViewModel, EditShodViewModel, EditToTvViewModel и EditTvViewModel будут являться контекстом данных для модулей интерфейса добавления или изменения сущностей, а именно для модулей EditDbUser, EditEmployee, EditMalfunction, EditPark, EditShod, EditToTv, EditTv соответственно.

Разграничение доступа в данном случае реализовано через интерфейс. Так как для каждого режима описан разный интерфейс, в каждом из которых был убран к той информации или функциям, которая не должна отображаться в том или ином режиме. Утечка данных в данном случае исключена, поскольку модуль MainWindowViewModel, предоставляет данные только в том случае, если элементы управления интерфейса обращаются к ним.

1.4 Реализация

1.4.1 Создание базы данных

Для создания базы данных была использована утилита pgAdmin4, которая позволяет подключиться к серверу базы данных и создать базу данных без использования языка SQL. Интерфейс утилиты pgAdmin4 представлен на рисунке 1.9.

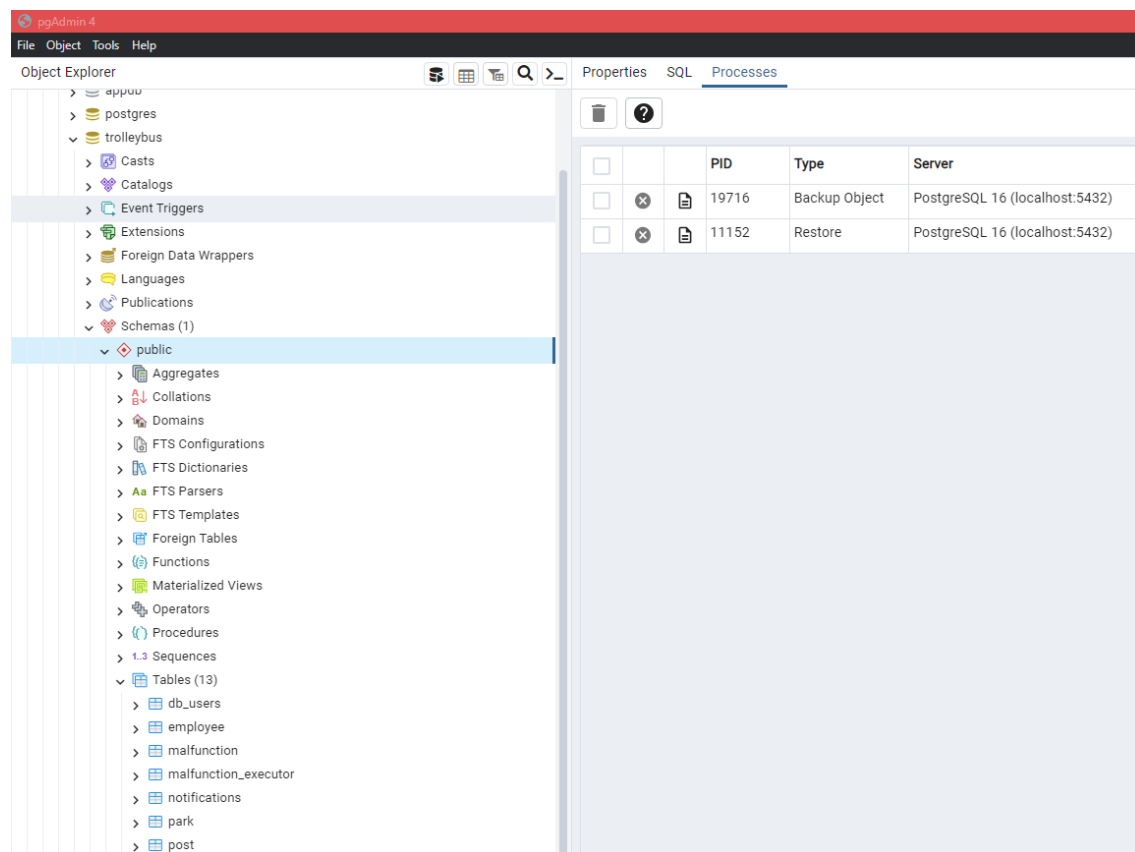


Рисунок 1.9 – Интерфейс приложения pgAdmin4

На рисунке 1.10 представлены таблицы, которые были созданы с помощью pgAdmin4.

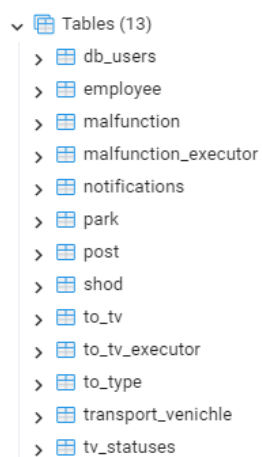


Рисунок 1.10 – Список таблиц

С помощью триггерных функций была реализована возможность изменения статуса троллейбуса в той или иной ситуации. Так изменение статуса происходит следующим образом:

- Статус меняется на «Сход» при добавлении записи в таблицу shod (add_shod()), а также автоматически добавляется новая запись в таблицу malfunction;
- Статус меняется на «В парке» или «На линии» в случае изменения записи в таблице shod, где в новой записи присутствует время отмены или прибытия соответственно (update_shod());
- Статус меняется на «Ремонт», когда в запись о неисправности автомобиля вносится информация о начале работ, а также статус меняется на «Готово» », когда в запись о неисправности автомобиля вносится информация о конце работ (update_malfunction());
- При изменении записи в таблице transport_venichles, проверяется статус в новой и старой записи. Если он изменился добавляется запись с сообщением в таблицу notifications (update_tv()).

Список триггерных функций представлен на рисунке 1.11

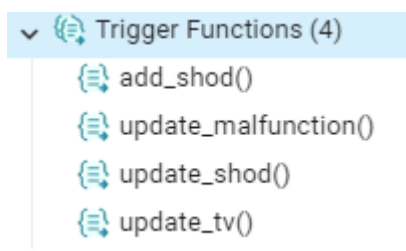


Рисунок 1.11 – Список триггерных функций базы данных

1.4.2 Работа с базой данных и описание программы

Разработка приложения осуществляется с использованием среды разработки Visual Studio 2022. Для создания контекста данных используется команда Scaffold-DbContext (рисунок 1.12). С помощью неё создается класс для подключения к базе данных, а также модель базы данных (рисунок 1.13).

Для того, чтобы при каждом новом обновлении модели не перезаписывался код, который добавляется в файлы классов моделей, данные классы объявляются частичными, что позволяет дописать код для модели в отдельных файлах и не беспокоиться, что он будет перезаписан при обновлении модели. Такие файлы описаны как <Имя класса модели>Partial (рисунок 1.13).

```
PM> Scaffold-DbContext "Host=localhost;Port=5432;Database=trolleybus;Username=postgres;Password=12345678" Npgsql.EntityFrameworkCore.PostgreSQL -OutputDir Model -F
Build started...
Build succeeded.
```

Рисунок 1.12 – Создание модели базы данных

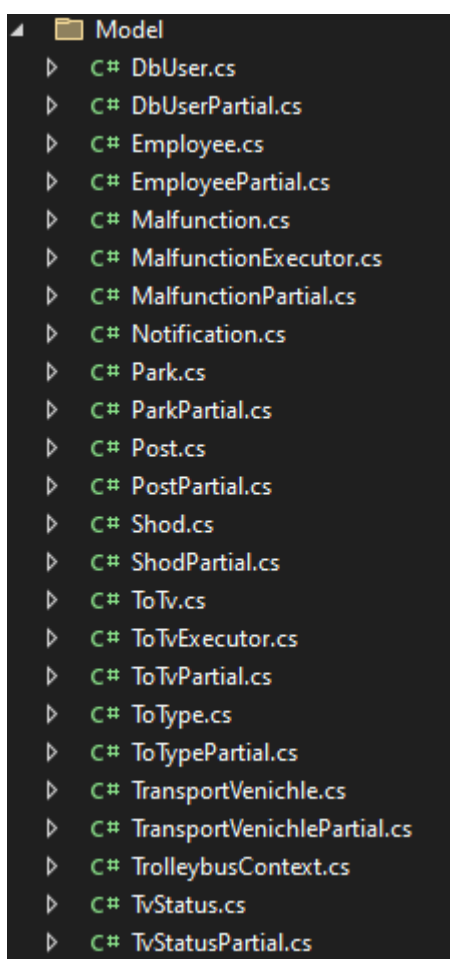


Рисунок 1.13 – Модель базы данных

Для обращения к сущностям базы используется класс TrolleybusContext. Пример использования данного класса и получения списка сотрудников реализован через свойство Employees. Описание свойства на языке C#:

```

public List<Employee> Employees
{
    get
    {
        var db = new TrolleybusContext();
        return db.Employees.Where(p => p.Id != 14).ToList();
    }
}

```

Пример добавления и изменения записи в базе данных:

```

using var db = new TrolleybusContext();
if (newEntity)
{
    db.Add(Entity);
}
else
{
    db.Update(Entity);
}
db.SaveChanges();

```

Пример удаления записи из базы данных:

```

using var db = new TrolleybusContext();
db.Remove(SelectedEmployee);
db.SaveChanges();

```

Для реализации вывода уведомлений был создан параллельный поток, который запускается в месте с созданием объекта класса MainWindowViewModel. При создании потока он получает начальное количество записей в таблице Notifications, и выводит новое уведомление каждый раз, когда количество записей увеличивается. Пример уведомления приведен на рисунке 1.14. Код для потока приведен ниже:

```

var updater = new Task
    (() =>
    {
        var db = new TrolleybusContext();
        int oldCount = db.Notifications.OrderByDescending(x
=>
            x.Id).Where(p
=>
p.Posts.Contains(CurrentUser.Employee.Post.Value))
            .Where(p
=>
p.Posts.Contains(CurrentUser.Employee.Park.Value)).Count();
        while (true)
        {
            if (ct.IsCancellationRequested) return;

```

```

        db = new TrolleybusContext();
        int newCount =
db.Notifications.OrderByDescending(x => x.Id).Where(p =>
p.Posts.Contains(CurrentUser.Employee.Post.Value))
        .Where(p =>
p.Posts.Contains(CurrentUser.Employee.Park.Value)).Count();
        if (newCount > oldCount)
        {
            var notif =
db.Notifications.OrderByDescending(x => x.Id).Where(p =>
p.Posts.Contains(CurrentUser.Employee.Post.Value))
        .Where(p =>
p.Posts.Contains(CurrentUser.Employee.Park.Value)).ToList();
            int toShowCount = newCount - oldCount;
            for (int i = 0; i < toShowCount; i++)
            {
                MessageBox.Show(notif[i].Message,
"Уведомление", MessageBoxButton.OK, MessageBoxImage.Information);
            }

            oldCount = newCount;
        }
    }, ct);
    updater.Start();

```

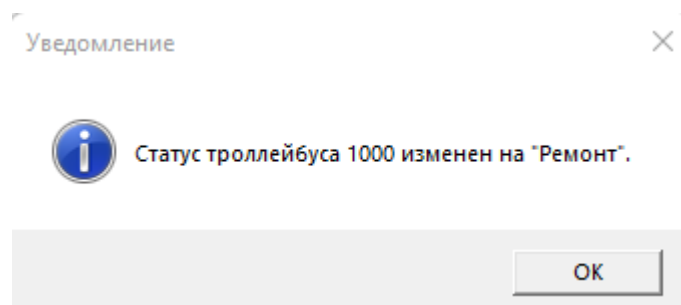


Рисунок 1.14 – Пример уведомления

Для создания рекламаций была использована библиотека TemplateEngine.Docx. В заранее созданном шаблоне были созданы закладки, по ссылкам на которые размещается необходимая информация в документ. Код реализации данной возможности приведен ниже:

```

File.Copy("шаблон_акт.docx", newAkt);

var valuesToFill = new Content
(
    new FieldContent("AktNumber", aktNum.Value.ToString()),

```

```

        new FieldContent("AktDate",
DateTime.Now.Date.ToString("dd.MM.yyyy")),
        new FieldContent("TvNumber",
selected.TvNumber.ToString()),
        new FieldContent("Vin",
selected.TvNumberNavigation.Vin),
        new FieldContent("DateExpluatation",
selected.TvNumberNavigation.CommissioningDate.Value.ToString("dd
.MM.yyyy")),
        new FieldContent("MalfunctionDate",
selected.AcceptTime.Value.ToString("dd.MM.yyyy")),
        new FieldContent("ZapretExpluatation",
selected.ZapretExpluatation.Value ? "Да" : "Нет"),
        new FieldContent("Probeg", selected.Probeg.ToString()),
        new FieldContent("TO", "Тут сведения о проведении ТО"),
        new FieldContent("ActMalfunction",
selected.MalfunctionRecl),
        new FieldContent("ActComments",
selected.MalfunctionReclComment)
    );

using (var outputDocument = new TemplateProcessor(newAkt))
{
    outputDocument.FillContent(valuesToFill);
    outputDocument.SaveChanges();
}

```

1.5 Тестирование и отладка

При запуске приложения открывается форма авторизации. Форма авторизации изображена на рисунке 1.15.

Рисунок 1.15 – Форма авторизации

На данной форме необходимо выбрать пользователя и ввести пароль. В случае правильности введенного пароля открывается главная форма программы (рисунок 1.16). В случае неправильности ввода пароля выводится сообщение об ошибке (рисунок 1.17).

Троллейбусное депо

Транспортные средства

Сход

Неисправности

Техническое обслуживание

Сотрудники

Пользователи БД

Парки

+

Маг

Гос. Ном

VIN-ном

Год выпу

Дата ввода в эксплуатац

Про

П:

Авто

Статус

0 записей найдено из 0

Сход

Неисправности

ТО

+

Номер ТС

Время схода

Неисправность

Эвакуатор

Время прибытия

Рисунок 1.16 – Главная форма программы

Ошибка

Не верный пароль

Подтвердить

Рисунок 1.17 – Сообщение об ошибке ввода пароля

Для функций создания, изменения, удаления и обновления записей в таблицах были созданы соответствующие кнопки над каждой из таблиц. При нажатии на кнопку создания и изменения записи открывается окно, в котором есть все необходимые поля. При нажатии на кнопку «сохранить» данные сохраняются в базе данных. Пример формы добавления для троллейбуса представлен на рисунке 1.18.

Троллейбус

Номер ТС

1000

Марка ТС

ВАЗ

Гос. номер

м456кр58

VIN

123456789

Пробег

10000

Год выпуска

2023

Дата начала эксплуатации

13.05.2024

☒ Автоход

Статус

Ремонт

Парк

Арбеково

Сохранить

Троллейбусное депо

Транспортные средства

Сход

Неисправности

Техническое обслуживание

Сотрудники

Пользователи БД

Парки

+

N	Маг	Гос. Ном	VIN-ном	Год выпу.	Дата ввода в эксплуатац	Про	Парк	Авто	Статус
1000	ВАЗ	м456кр58	123456789	2023	13.05.2024	10000	Арбеково	☒	Ремонт

Рисунок 1.18 – Пример добавления записи

2 Безопасность жизнедеятельности

При взаимодействии с компьютером, необходимо соблюдать меры безопасности, чтобы не было неприятных последствий для здоровья после использования. Если пользователь ПК будет пренебрегать техникой безопасности, то проблемы со здоровьем будут неизбежны.

2.1 Вред здоровью при работе с ПК

Проблемы со здоровьем в процессе взаимодействия с компьютером имеют свойство накапливаться, то есть здоровье ухудшается не сразу, а постепенно. Наиболее опасными факторами являются:

- нагрузка на суставы. При неправильной посадке за компьютером, на суставы будет идти большая нагрузка, а именно кистей рук, из-за чего может произойти даже онемение рук;
- нагрузка на спину. Опять же, если принять неправильно положение осанки при работе за компьютером, может появиться искривление позвоночника и даже грыжа межпозвоночного диска;
- нагрузка на органы зрения. При долгой работе с мониторами, будет ухудшаться здоровья глаз, впоследствии могут появиться синдромы «красного глаза» и «сухого глаза»;
- нагрузка на психику и нервную. При долгом взаимодействии с ПК, у пользователя могут возникать проблемы с памятью, а также может появиться депрессия и даже бессонница.

2.2 Снижение вреда для здоровья при работе с ПК

При соблюдении простых правил можно избежать серьезных проблем со здоровьем, для этого нужно:

- соблюдать необходимое расстояние до монитора. Правильным расстоянием до дисплея является расстояние, при котором пользователь может коснуться указательным пальцем до верхнего края монитора;

- соблюдать оптимальный угол экрана. Оптимальным углом является наклон монитора в 30 градусов;
- клавиатуру ставить в нужное положение. Клавиатура должна находиться на расстоянии 20-30 сантиметров от края стола;
- делать перерыв. Примерно каждый час необходимо делать разминку для органов зрения, а именно оторваться от монитора и делать гимнастику для глаз: рисовать различные фигуры глазами, фокусировать внимание на дальние объекты и переводить фокус с одних предметов на другие. Также примерно каждые полтора-два часа необходимо выполнять зарядку;
- выбрать правильный стул или кресло. При выборе данной мебели, необходимо приобрести такую, которая будет обеспечивать прямую осанку, при которой спина будет опираться в спинку.

2.3 Техника безопасности при работе с компьютером

При работе с компьютером необходимо соблюдать следующие правила, которые связаны с техникой безопасности:

- не оставлять компьютер без присмотра на длительное время;
- не использовать компьютер, который работает от питания без заземления;
- не работать за ПК, у которого есть повреждение в системном блоке или в его периферийных устройствах;
- не включать компьютер, если в помещении слишком высокая влажность;
- не класть на корпус компьютера или монитора посторонние предметы.

Заключение

В ходе выполнения данной выпускной квалификационной работы были получены навыки написания автоматизированной системы для учета состояния троллейбусов в троллейбусном парке, изучен стек технологий и библиотек платформы .NET. Результатом работы является разработанное клиент-серверное приложение «Троллейбус», которое называется TrolleyBus, а также отчёт, в котором представлены результаты тестирования разработанного программного обеспечения, описание разработки и подробный разбор его основных функциональных особенностей.

Разработанное приложение позволяет пользователю работать с базой данных, предназначенной для ведения учета состояния троллейбусов троллейбусном парке. После проведения тестирования было выяснено, что разработанное приложение соответствует всем поставленным в начале работы требованиям и работает корректно, не вызывая критических ошибок в системе, что является частой проблемой драйверов операционной системы Windows.

В дальнейшем приложение можно усовершенствовать путем добавления поддержки шифрования данных. Разработанный интерфейс пользователя можно улучшить путем усовершенствования дизайна для повышения комфорта использования.

Список использованных источников

1. zavgar.online: [Электронный ресурс]. URL: <https://zavgar.online/> (дата обращения: 26.04.2024)
2. tis-online.com: [Электронный ресурс]. URL: <https://tis-online.com/> (дата обращения: 27.04.2024)
3. solutions.1c.ru: [Электронный ресурс]. URL: <https://solutions.1c.ru/catalog/autotransport-prof> (дата обращения: 27.04.2024)
4. Зиборов, В. *Visual C# 2010 на примерах* / Виктор Зиборов. - М.: БХВ-Петербург, 2011. - 432 с.
5. Натан А. *WPF 4. Подробное руководство*. - Пер. с англ. - СПб.: Символ-Плюс, 2011. - 880 с.
6. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2018 — 1328 с.
7. Дж. Рамбо, М. Блаха *UML 2.0. Объектно-ориентированное моделирование и разработка*, 2-е издание. – СПб.: Питер, 2007 — 544 с.

Приложение А

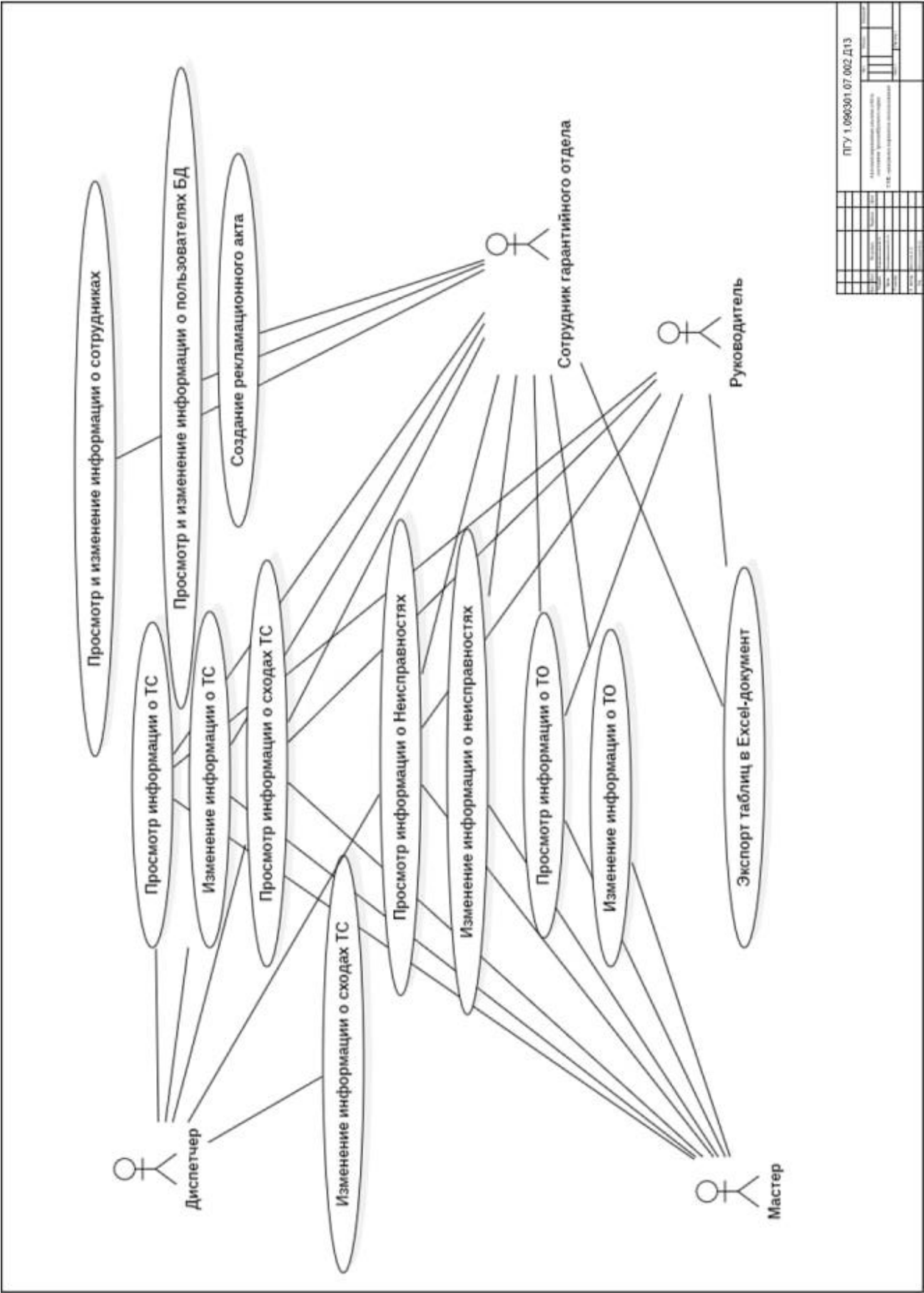
(Обязательное)

Автоматизированная система учета состояния троллейбусного парка

UML–диаграммы

					ПГУ 1.090301.07.001 ПЗ	Лист
						50
Изм.	Лист	№ докум.	Подпись	Дата		

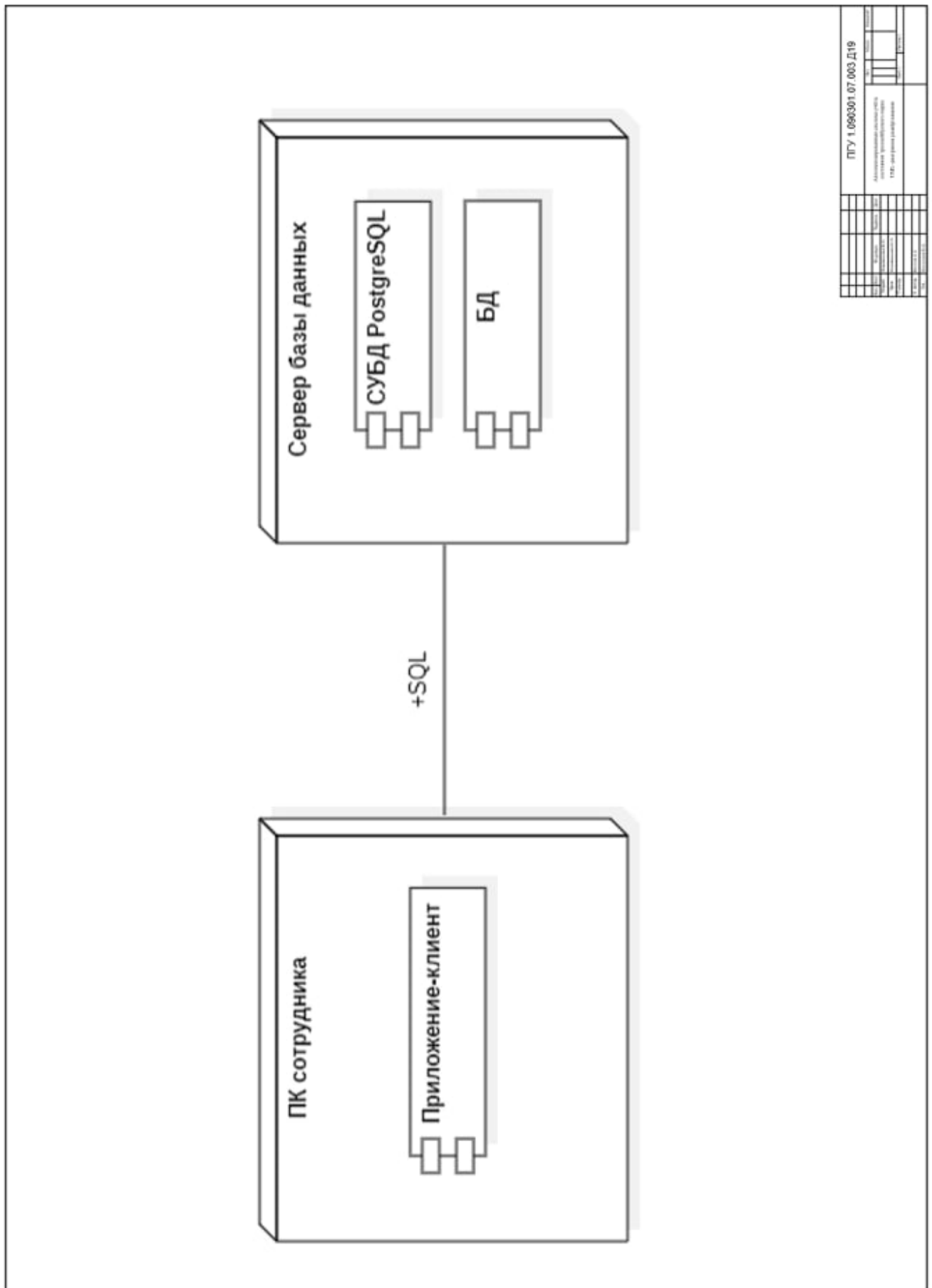
Приложение А.1 – UML – диаграмма вариантов использования



ПГУ 1.090301.07.002 Д13	
Исполнитель	Сотрудник гарантийного отдела
Проверенный	Руководитель
Дата	2013.07.01

Изм.	Лист	№ докум.	Подпись	Дата

Приложение А.2 – UML – диаграмма развертывания

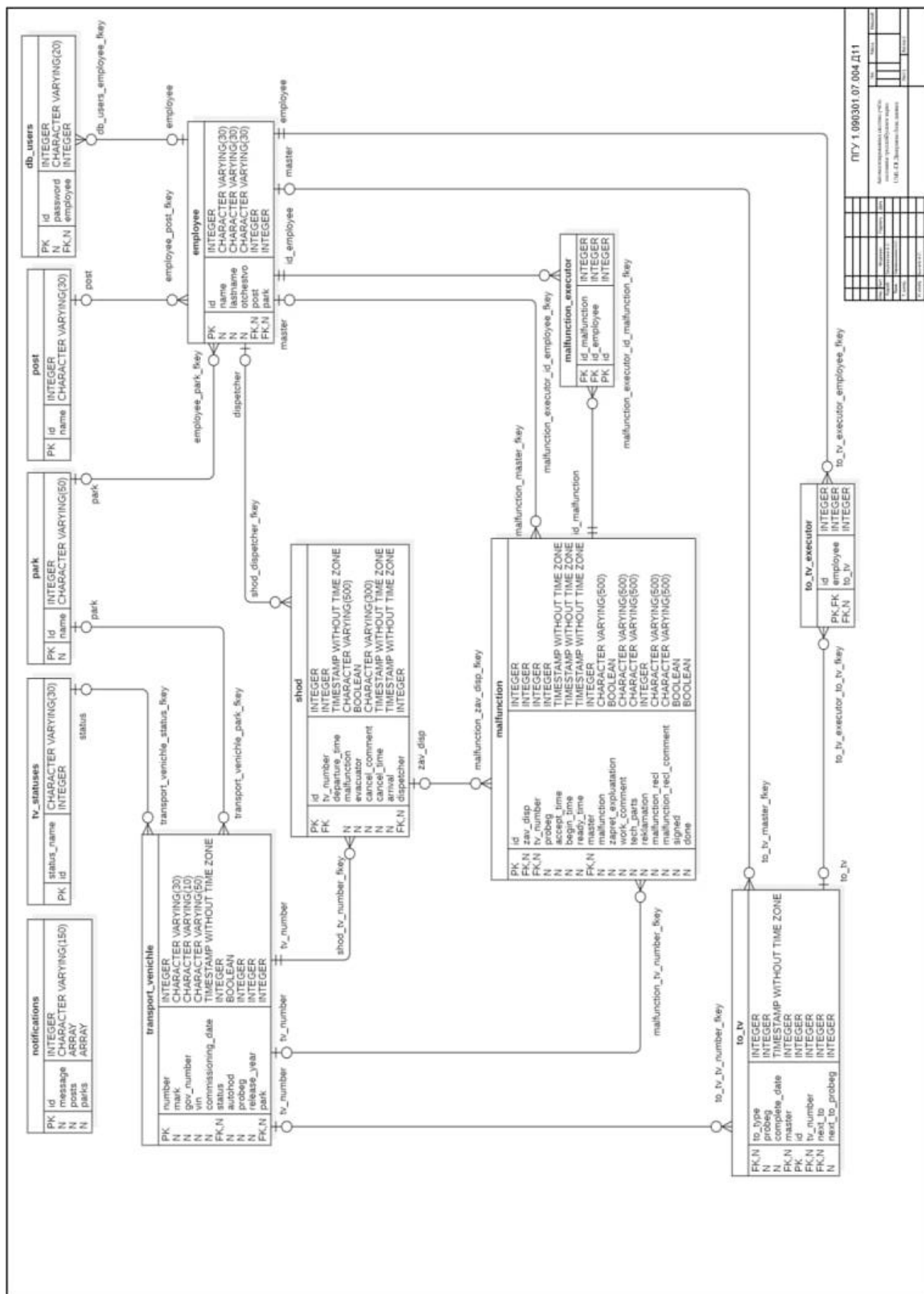


Изм.	Лист	№ докум.	Подпись	Дата

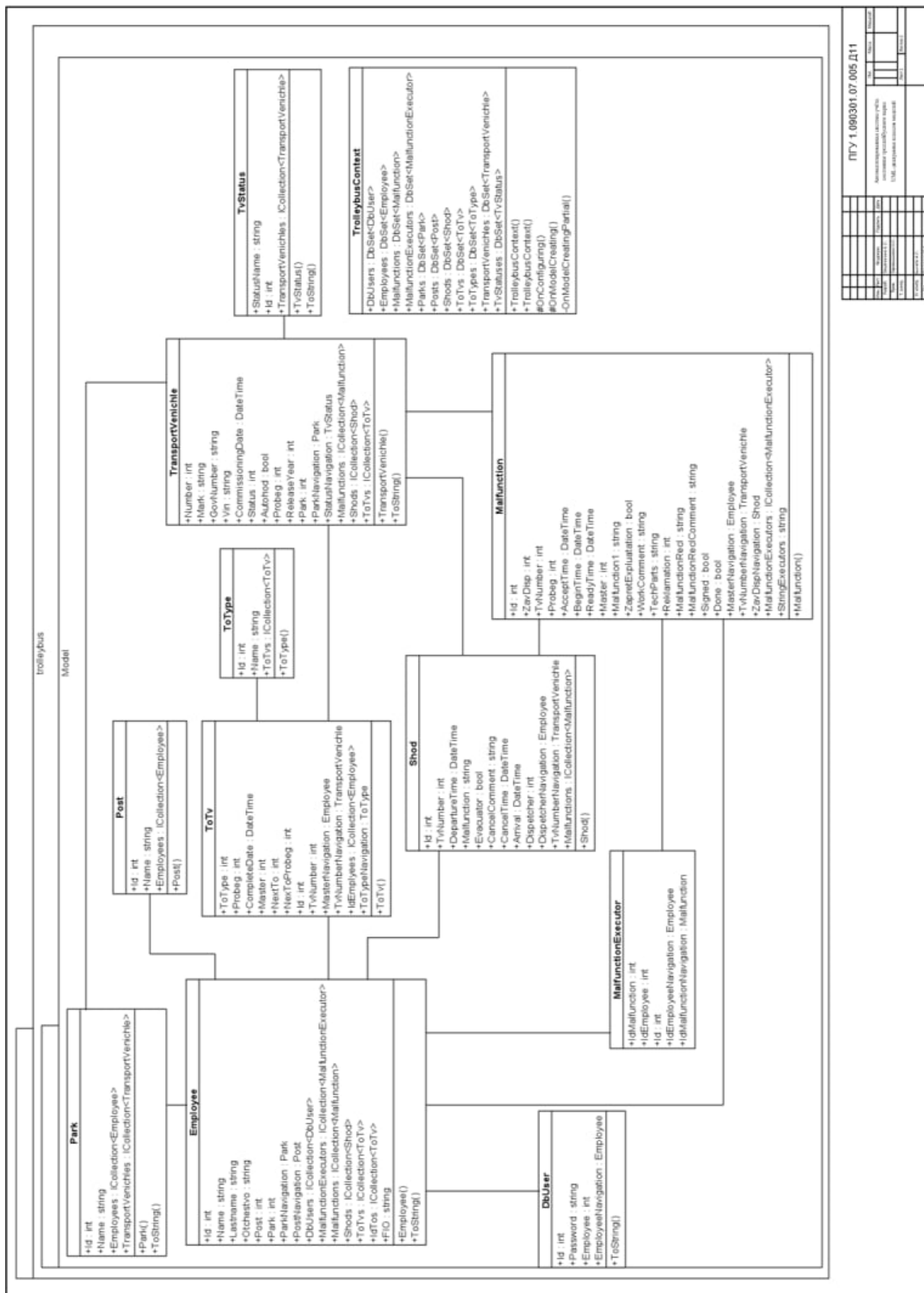
ПГУ 1.090301.07.001 ПЗ

Лист
52

Приложение А.3 – UML – диаграмма классов моделей



Приложение А.4 – UML-ER диаграмма базы данных



ПТУ 1.090301.07.005 Д11									
№	№	№	№	№	№	№	№	№	№
1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

Приложение Б

(Обязательное)

Автоматизированная система учета состояния троллейбусного парка

Листинг программы

					ПГУ 1.090301.07.001 ПЗ	Лист
						55
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «Dbuser.cs»

```
namespace trolleybus.Model;
public partial class DbUser
{
    public int Id { get; set; }
    public string? Password { get; set; }
    public int? Employee { get; set; }
    public virtual Employee? EmployeeNavigation { get; set; }
}
```

Файл «DbuserPartial.cs»

```
using System.Linq;
namespace trolleybus.Model
{
    public partial class DbUser
    {
        public override string ToString()
        {
            return EmployeeView.ToString();
        }
        public Employee EmployeeView
        {
            get
            {
                using var db = new TrolleybusContext();
                var list = db.Employees.Where(p => p.Id ==
Employee).ToList();
                if(list.Count != 0)
                {
                    return list.First();
                }
                else
                {
                    return null;
                }
            }
        }
    }
}
```


}

Файл «Employee.cs»

```
using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class Employee
{
    public int Id { get; set; }

    public string? Name { get; set; }

    public string? Lastname { get; set; }

    public string? Otchestvo { get; set; }

    public int? Post { get; set; }

    public int? Park { get; set; }

    public virtual ICollection<DbUser> DbUsers { get; set; } = new
List<DbUser>();

    public virtual ICollection<MalfunctionExecutor>
MalfunctionExecutors { get; set; } = new
List<MalfunctionExecutor>();

    public virtual ICollection<Malfunction> Malfunctions { get;
set; } = new List<Malfunction>();

    public virtual Park? ParkNavigation { get; set; }

    public virtual Post? PostNavigation { get; set; }
```

					ПГУ 1.090301.07.001 ПЗ	Лист
						57
Изм.	Лист	№ докум.	Подпись	Дата		

```

        public virtual ICollection<Shod> Shods { get; set; } = new
List<Shod>();

        public virtual ToTvExecutor? ToTvExecutor { get; set; }

        public virtual ICollection<ToTv> ToTvs { get; set; } = new
List<ToTv>();
    }

```

Файл «EmployeePartial.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace trolleybus.Model
{
    public partial class Employee
    {
        public string FIO
        {
            get
            {
                return ToString();
            }
        }

        public Park ParkView
        {
            get
            {
                using var db = new TrolleybusContext();
                var list = db.Parks.Where(p => p.Id ==
Park).ToList();
                if (list.Count != 0)
                {
                    return list.First();
                }
                else
                {
                    return null;
                }
            }
        }
    }

```

```

    }
}

public Post PostView
{
    get
    {
        using var db = new TrolleybusContext();
        var list = db.Posts.Where(p => p.Id ==
Post).ToList();
        if (list.Count != 0)
        {
            return list.First();
        }
        else
        {
            return null;
        }
    }
}

public override string ToString()
{
    if (Name != "" && Lastname != "")
    {
        if (Otchestvo != "")
        {
            var result = Lastname + " " + Name[0] + "." +
Otchestvo[0] + ".";
            return result;
        }
        else
        {
            var result = Lastname + " " + Name[0] + ".";
            return result;
        }
    }

    return "";
}
}
}

```

Файл «Malfunction.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class Malfunction
{
    public int Id { get; set; }
}

```

```

public int? ZavDisp { get; set; }

public int? TvNumber { get; set; }

public int? Probeg { get; set; }

public DateTime? AcceptTime { get; set; }

public DateTime? BeginTime { get; set; }

public DateTime? ReadyTime { get; set; }

public int? Master { get; set; }

public string? Malfunction1 { get; set; }

public bool? ZapretExpluatation { get; set; }

public string? WorkComment { get; set; }

public string? TechParts { get; set; }

public int? Reklamation { get; set; }

public string? MalfunctionRecl { get; set; }

public string? MalfunctionReclComment { get; set; }

public bool? Signed { get; set; }

public bool? Done { get; set; }

public virtual ICollection<MalfunctionExecutor>
MalfunctionExecutors { get; set; } = new
List<MalfunctionExecutor>();

public virtual Employee? MasterNavigation { get; set; }

public virtual TransportVenichle? TvNumberNavigation { get;
set; }

public virtual Shod? ZavDispNavigation { get; set; }
}

```

Файл «MalfunctionExecutor.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

```

```

public partial class MalfunctionExecutor
{
    public int IdMalfunction { get; set; }

    public int IdEmployee { get; set; }

    public int Id { get; set; }

    public virtual Employee IdEmployeeNavigation { get; set; } =
null!;

    public virtual Malfunction IdMalfunctionNavigation { get; set;
} = null!;
}

```

Файл «MalfunctionPartial.cs»

```

using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations.Schema;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;

namespace trolleybus.Model
{
    public partial class Malfunction
    {
        public Employee MasterView
        {
            get
            {
                using var db = new TrolleybusContext();
                var list = db.Employees.Where(p => p.Id ==
Master).ToList();
                if (list.Count != 0)
                {
                    return list.First();
                }
                else
                {
                    return null;
                }
            }
        }

        public string StringExecutors
        {
            get

```

```

        {
            string result = "Исполнители: ";

            using var db = new TrolleybusContext();
            var idExs = db.MalfunctionExecutors.Where(p =>
p.IdMalfunction == Id).Select(p => p.IdEmployee).ToList();

            if (idExs.Count == 0) return result + "нет";

            var exs = db.Employees.Where(p
idExs.Contains(p.Id)).ToList();
            for (int i = 0; i < exs.Count() - 1; i++)
            {
                result += exs[i].ToString() + ", ";
            }
            result += exs[exs.Count() - 1].ToString();

            return result;
        }
    }
}

```

Файл «Notification.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class Notification
{
    public int Id { get; set; }

    public string? Message { get; set; }

    public int[]? Posts { get; set; }

    public int[]? Parks { get; set; }
}

```

Файл «Park.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class Park
{

```

```

        public int Id { get; set; }

        public string? Name { get; set; }

        public virtual ICollection<Employee> Employees { get; set; }
= new List<Employee>();

        public          virtual          ICollection<TransportVenichle>
TransportVenichles { get; set; } = new List<TransportVenichle>();
    }

```

Файл «ParkPartial.cs»

```

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

namespace trolleybus.Model
{
    public partial class Park
    {
        public override string ToString()
        {
            return Name;
        }
    }
}

```

Файл «Post.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class Post
{
    public int Id { get; set; }

    public string Name { get; set; } = null!;

    public virtual ICollection<Employee> Employees { get; set; }
= new List<Employee>();
}

```

Файл «PostPartial.cs»

					ПГУ 1.090301.07.001 ПЗ	Лист
						63
Изм.	Лист	№ докум.	Подпись	Дата		

```

using DocumentFormat.OpenXml.Drawing.Charts;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

```

```

namespace trolleybus.Model
{
    public partial class Post
    {
        public override string ToString()
        {
            return Name;
        }
    }
}

```

Файл «Shod.cs»

```

using System;
using System.Collections.Generic;

```

```

namespace trolleybus.Model;

```

```

public partial class Shod
{
    public int Id { get; set; }

    public int TvNumber { get; set; }

    public DateTime DepartureTime { get; set; }

    public string? Malfunction { get; set; }

    public bool? Evacuator { get; set; }

    public string? CancelComment { get; set; }

    public DateTime? CancelTime { get; set; }

    public DateTime? Arrival { get; set; }

    public int? Dispetcher { get; set; }

    public virtual Employee? DispetcherNavigation { get; set; }

    public virtual ICollection<Malfunction> Malfunctions { get;
set; } = new List<Malfunction>();

    public virtual TransportVenichle TvNumberNavigation { get;
set; } = null!;
}

```

					ПГУ 1.090301.07.001 ПЗ	Лист
						64
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «ShodPartial.cs»

```

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

namespace trolleybus.Model
{
    public partial class Shod
    {
        public Employee DispetcherView
        {
            get
            {
                using var db = new TrolleybusContext();
                var list = db.Employees.Where(p => p.Id ==
Dispetcher).ToList();
                if (list.Count != 0)
                {
                    return list.First();
                }
                else
                {
                    return null;
                }
            }
        }
    }
}

```

Файл «ToTv.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class ToTv
{
    public int? ToType { get; set; }

    public int? Probeg { get; set; }

    public DateTime? CompleteDate { get; set; }

    public int? Master { get; set; }

    public int Id { get; set; }

    public int? TvNumber { get; set; }

    public int? NextTo { get; set; }
}

```

```

        public int? NextToProbeg { get; set; }

        public virtual Employee? MasterNavigation { get; set; }

        public virtual ToType? NextToNavigation { get; set; }

        public virtual ICollection<ToTvExecutor> ToTvExecutors { get;
set; } = new List<ToTvExecutor>();

        public virtual ToType? ToTypeNavigation { get; set; }

        public virtual TransportVenichle? TvNumberNavigation { get;
set; }
    }

```

Файл «ToTvExecutor.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class ToTvExecutor
{
    public int Id { get; set; }

    public int Employee { get; set; }

    public int? ToTv { get; set; }

    public virtual Employee EmployeeNavigation { get; set; } =
null!;

    public virtual ToTv? ToTvNavigation { get; set; }
}

```

Файл «ToTvPartial.cs»

```

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

namespace trolleybus.Model
{
    public partial class ToTv
    {
        public Employee MasterView
        {
            get
            {
                using var db = new TrolleybusContext();

```

```

        var list = db.Employees.Where(p => p.Id ==
Master).ToList();
        if (list.Count != 0)
        {
            return list.First();
        }
        else
        {
            return null;
        }
    }
}
public ToType ToTypeView
{
    get
    {
        using var db = new TrolleybusContext();
        var list = db.ToTypes.Where(p => p.Id ==
ToType).ToList();
        if (list.Count != 0)
        {
            return list.First();
        }
        else
        {
            return null;
        }
    }
}

public ToType NextToView
{
    get
    {
        using var db = new TrolleybusContext();
        var list = db.ToTypes.Where(p => p.Id ==
NextTo).ToList();
        if (list.Count != 0)
        {
            return list.First();
        }
        else
        {
            return null;
        }
    }
}
public string StringExecutors
{
    get
    {

```

```

        string result = "Исполнители: ";

        using var db = new TrolleybusContext();
        var idExs = db.ToTvExecutors.Where(p => p.ToTv ==
Id).Select(p => p.Employee).ToList();

        if (idExs.Count == 0) return result + "нет";

        var exs = db.Employees.Where(p
=> idExs.Contains(p.Id)).ToList();
        for (int i = 0; i < exs.Count() - 1; i++)
        {
            result += exs[i].ToString() + ", ";
        }
        result += exs[exs.Count() - 1].ToString();

        return result;
    }
}
}

```

Файл «ToType.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class ToType
{
    public int Id { get; set; }

    public string Name { get; set; } = null!;

    public virtual ICollection<ToTv> ToTvNextToNavigations { get;
set; } = new List<ToTv>();

    public virtual ICollection<ToTv> ToTvToTypeNavigations { get;
set; } = new List<ToTv>();
}

```

Файл «ToTypedPartial.cs»

```

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

namespace trolleybus.Model
{
    public partial class ToType
    {
        public override string ToString()

```

```

        {
            return Name;
        }
    }
}

```

Файл « TransportVenichle.cs»

```

using System;
using System.Collections.Generic;

namespace trolleybus.Model;

public partial class TransportVenichle
{
    public int Number { get; set; }

    public string? Mark { get; set; }

    public string? GovNumber { get; set; }

    public string? Vin { get; set; }

    public DateTime? CommissioningDate { get; set; }

    public int? Status { get; set; }

    public bool? Autohod { get; set; }

    public int? Probeg { get; set; }

    public int? ReleaseYear { get; set; }

    public int? Park { get; set; }

    public virtual ICollection<Malfunction> Malfunctions { get; set; } = new List<Malfunction>();

    public virtual Park? ParkNavigation { get; set; }

    public virtual ICollection<Shod> Shods { get; set; } = new List<Shod>();

    public virtual TvStatus? StatusNavigation { get; set; }

    public virtual ICollection<ToTv> ToTvs { get; set; } = new List<ToTv>();
}

```

Файл « TransportVenichlePartial.cs»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.ComponentModel.DataAnnotations.Schema;

```

```

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

namespace trolleybus.Model
{
    public partial class TransportVenichle
    {
        public Park ParkView
        {
            get
            {
                using var db = new TrolleybusContext();
                var list = db.Parks.Where(p => p.Id ==
Park).ToList();
                if (list.Count != 0)
                {
                    return list.First();
                }
                else
                {
                    return null;
                }
            }
        }

        public TvStatus StatusView
        {
            get
            {
                using var db = new TrolleybusContext();
                var list = db.TvStatuses.Where(p => p.Id ==
Status).ToList();
                if (list.Count != 0)
                {
                    return list.First();
                }
                else
                {
                    return null;
                }
            }
        }

        public override string ToString()
        {
            return Number.ToString();
        }
    }
}

```

Файл «TrolleybusContext.cs»

```

using System;
using System.Collections.Generic;
using Microsoft.EntityFrameworkCore;

namespace trolleybus.Model;

public partial class TrolleybusContext : DbContext
{
    public TrolleybusContext()
    {
    }

    public TrolleybusContext(DbContextOptions<TrolleybusContext>
options)
    : base(options)
    {
    }

    public virtual DbSet<DbUser> DbUsers { get; set; }

    public virtual DbSet<Employee> Employees { get; set; }

    public virtual DbSet<Malfunction> Malfunctions { get; set; }

    public virtual DbSet<MalfunctionExecutor>
MalfunctionExecutors { get; set; }

    public virtual DbSet<Notification> Notifications { get; set; }

    public virtual DbSet<Park> Parks { get; set; }

    public virtual DbSet<Post> Posts { get; set; }

    public virtual DbSet<Shod> Shods { get; set; }

    public virtual DbSet<ToTv> ToTvs { get; set; }

    public virtual DbSet<ToTvExecutor> ToTvExecutors { get; set; }

    public virtual DbSet<ToType> ToTypes { get; set; }

    public virtual DbSet<TransportVenichle> TransportVenichles {
get; set; }

    public virtual DbSet<TvStatus> TvStatuses { get; set; }

    protected override void OnConfiguring(DbContextOptionsBuilder
optionsBuilder)

```

#warning To protect potentially sensitive information in your connection string, you should move it out of source code. You can avoid scaffolding the connection string by using the Name= syntax to read it from configuration - see <https://go.microsoft.com/fwlink/?linkid=2131148>. For more guidance on storing connection strings, see <http://go.microsoft.com/fwlink/?LinkId=723263>.

=>

```
optionsBuilder.UseNpgsql("Host=localhost;Port=5432;Database=trolleybus;Username=postgres;Password=12345678");
```

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
```

```
{
```

```
modelBuilder.Entity<DbUser>(entity =>
```

```
{
```

```
entity.HasKey(e => e.Id).HasName("db_users_pkey");
```

```
entity.ToTable("db_users");
```

```
entity.Property(e => e.Id).HasColumnName("id");
```

```
entity.Property(e
```

=>

```
e.Employee).HasColumnName("employee");
```

```
entity.Property(e => e.Password)
```

```
.HasMaxLength(20)
```

```
.HasColumnName("password");
```

```
entity.HasOne(d => d.EmployeeNavigation).WithMany(p
```

```
=> p.DbUsers)
```

```
.HasForeignKey(d => d.Employee)
```

```
.OnDelete(DeleteBehavior.Cascade)
```

```
.HasConstraintName("db_users_employee_fkey");
```

```
});
```

```
modelBuilder.Entity<Employee>(entity =>
```

```
{
```

```
entity.HasKey(e => e.Id).HasName("employee_pkey");
```

```
entity.ToTable("employee");
```

```
entity.Property(e => e.Id).HasColumnName("id");
```

```
entity.Property(e => e.Lastname)
```

```
.HasMaxLength(30)
```

```
.HasColumnName("lastname");
```

```
entity.Property(e => e.Name)
```

```
.HasMaxLength(30)
```

```
.HasColumnName("name");
```

```
entity.Property(e => e.Otchestvo)
```

```
.HasMaxLength(30)
```

```
.HasColumnName("otchestvo");
```

```
entity.Property(e => e.Park).HasColumnName("park");
```

					ПГУ 1.090301.07.001 ПЗ	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		


```

        entity.Property(e => e.Post).HasColumnName("post");

        entity.HasOne(d => d.ParkNavigation).WithMany(p =>
p.Employees)
            .HasForeignKey(d => d.Park)
            .OnDelete(DeleteBehavior.SetNull)
            .HasConstraintName("employee_park_fkey");

        entity.HasOne(d => d.PostNavigation).WithMany(p =>
p.Employees)
            .HasForeignKey(d => d.Post)
            .OnDelete(DeleteBehavior.SetNull)
            .HasConstraintName("employee_post_fkey");
    });

    modelBuilder.Entity<Malfunction>(entity =>
    {
        entity.HasKey(e => e.Id).HasName("zav_master_pkey");

        entity.ToTable("malfunction");

        entity.Property(e => e.Id)

.HasDefaultValueSql("nextval('zav_master_id_seq'::regclass)")
            .HasColumnName("id");
        entity.Property(e => e.AcceptTime)
            .HasColumnType("timestamp without time zone")
            .HasColumnName("accept_time");
        entity.Property(e => e.BeginTime)
            .HasColumnType("timestamp without time zone")
            .HasColumnName("begin_time");
        entity.Property(e => e.Done)
            .HasDefaultValueSql("false")
            .HasColumnName("done");
        entity.Property(e => e.Malfunction1)
            .HasMaxLength(500)
            .HasColumnName("malfunction");
        entity.Property(e => e.MalfunctionRecl)
            .HasMaxLength(500)
            .HasColumnName("malfunction_recl");
        entity.Property(e => e.MalfunctionReclComment)
            .HasMaxLength(500)
            .HasColumnName("malfunction_recl_comment");
        entity.Property(e
e.Master).HasColumnName("master");
        entity.Property(e
e.Probeg).HasColumnName("probeg");
        entity.Property(e => e.ReadyTime)
            .HasColumnType("timestamp without time zone")
            .HasColumnName("ready_time");
    });

```

```

        entity.Property(e => e.Signed)
        .HasDefaultValueSql("false")
        .HasColumnName("signed");
        entity.Property(e => e.TechParts)
        .HasMaxLength(500)
        .HasColumnName("tech_parts");
        entity.Property(e => e.WorkComment)
        .HasMaxLength(500)
        .HasColumnName("work_comment");
        entity.Property(e => e.ZapretExpluatation)
        .HasColumnName("zapret_expluatation");
        entity.Property(e => e.ZavDisp)
        .HasColumnName("zav_disp");

        entity.HasOne(d => d.MasterNavigation).WithMany(p =>
p.Malfunctions)
        .HasForeignKey(d => d.Master)
        .OnDelete(DeleteBehavior.SetNull)
        .HasConstraintName("malfunction_master_fkey");

        entity.HasOne(d => d.TvNumberNavigation).WithMany(p
=> p.Malfunctions)
        .HasForeignKey(d => d.TvNumber)
        .OnDelete(DeleteBehavior.Cascade)

        .HasConstraintName("malfunction_tv_number_fkey");

        entity.HasOne(d => d.ZavDispNavigation).WithMany(p =>
p.Malfunctions)
        .HasForeignKey(d => d.ZavDisp)
        .OnDelete(DeleteBehavior.SetNull)
        .HasConstraintName("malfunction_zav_disp_fkey");
    });

    modelBuilder.Entity<MalfunctionExecutor>(entity =>
    {
        entity.HasKey(e
e.Id).HasName("malfunction_executor_pkey");

        entity.ToTable("malfunction_executor");

        entity.Property(e => e.Id).HasColumnName("id");
        entity.Property(e
e.IdEmployee).HasColumnName("id_employee");
        entity.Property(e
e.IdMalfunction).HasColumnName("id_malfunction");
    });

```

```

        entity.HasOne(d => d.IdEmployeeNavigation).WithMany(p
=> p.MalfunctionExecutors)
            .HasForeignKey(d => d.IdEmployee)

        .HasConstraintName("malfunction_executor_id_employee_fkey");

        entity.HasOne(d
=> d.IdMalfunctionNavigation).WithMany(p => p.MalfunctionExecutors)
            .HasForeignKey(d => d.IdMalfunction)

        .HasConstraintName("malfunction_executor_id_malfunction_fkey");
    });

    modelBuilder.Entity<Notification>(entity =>
    {
        entity.HasKey(e
=> e.Id).HasName("notifications_pkey");

        entity.ToTable("notifications");

        entity.Property(e => e.Id).HasColumnName("id");
        entity.Property(e => e.Message)
            .HasMaxLength(150)
            .HasColumnName("message");
        entity.Property(e => e.Parks).HasColumnName("parks");
        entity.Property(e => e.Posts).HasColumnName("posts");
    });

    modelBuilder.Entity<Park>(entity =>
    {
        entity.HasKey(e => e.Id).HasName("park_pkey");

        entity.ToTable("park");

        entity.Property(e => e.Name)
            .HasMaxLength(50)
            .HasColumnName("name");
    });

    modelBuilder.Entity<Post>(entity =>
    {
        entity.HasKey(e => e.Id).HasName("post_pkey");

        entity.ToTable("post");

        entity.Property(e => e.Id).HasColumnName("id");
        entity.Property(e => e.Name)
            .HasMaxLength(30)
            .HasColumnName("name");
    });

```

```

modelBuilder.Entity<Shod>(entity =>
{
    entity.HasKey(e => e.Id).HasName("zav_disp_pkey");

    entity.ToTable("shod");

    entity.Property(e => e.Id)

.HasDefaultValueSql("nextval('zav_disp_id_seq'::regclass)")
    .HasColumnName("id");
    entity.Property(e => e.Arrival)
        .HasColumnType("timestamp without time zone")
        .HasColumnName("arrival");
    entity.Property(e => e.CancelComment)
        .HasMaxLength(300)
        .HasColumnName("cancel_comment");
    entity.Property(e => e.CancelTime)
        .HasColumnType("timestamp without time zone")
        .HasColumnName("cancel_time");
    entity.Property(e => e.DepartureTime)
        .HasColumnType("timestamp without time zone")
        .HasColumnName("departure_time");
    entity.Property(e
=>
e.Dispatcher).HasColumnName("dispetcher");
    entity.Property(e => e.Evacuator)
        .HasDefaultValueSql("false")
        .HasColumnName("evacuator");
    entity.Property(e => e.Malfunction)
        .HasMaxLength(500)
        .HasColumnName("malfunction");
    entity.Property(e
=>
e.TvNumber).HasColumnName("tv_number");

    entity.HasOne(d => d.DispetcherNavigation).WithMany(p
=> p.Shods)
        .HasForeignKey(d => d.Dispetcher)
        .OnDelete(DeleteBehavior.SetNull)
        .HasConstraintName("shod_dispetcher_fkey");

    entity.HasOne(d => d.TvNumberNavigation).WithMany(p
=> p.Shods)
        .HasForeignKey(d => d.TvNumber)
        .OnDelete(DeleteBehavior.SetNull)
        .HasConstraintName("shod_tv_number_fkey");
});

modelBuilder.Entity<ToTv>(entity =>
{
    entity.HasKey(e => e.Id).HasName("to_tv_pkey");

    entity.ToTable("to_tv");

```

```

        entity.Property(e => e.Id).HasColumnName("id");
        entity.Property(e => e.CompleteDate)
            .HasColumnType("timestamp without time zone")
            .HasColumnName("complete_date");
        entity.Property(e => e.Master)
            .HasColumnName("master");
        entity.Property(e => e.NextTo)
            .HasColumnName("next_to");
        entity.Property(e => e.NextToProbeg)
            .HasColumnName("next_to_probeg");
        entity.Property(e => e.Probeg)
            .HasColumnName("probeg");
        entity.Property(e => e.ToType)
            .HasColumnName("to_type");
        entity.Property(e => e.TvNumber)
            .HasColumnName("tv_number");

        entity.HasOne(d => d.MasterNavigation).WithMany(p => p.ToTvs)
            .HasForeignKey(d => d.Master)
            .OnDelete(DeleteBehavior.SetNull)
            .HasConstraintName("to_tv_master_fkey");

        entity.HasOne(d => d.NextToNavigation).WithMany(p => p.ToTvNextToNavigations)
            .HasForeignKey(d => d.NextTo)
            .OnDelete(DeleteBehavior.SetNull)
            .HasConstraintName("to_tv_next_to_fkey");

        entity.HasOne(d => d.ToTypeNavigation).WithMany(p => p.ToTvToTypeNavigations)
            .HasForeignKey(d => d.ToType)
            .OnDelete(DeleteBehavior.SetNull)
            .HasConstraintName("to_tv_to_type_fkey");

        entity.HasOne(d => d.TvNumberNavigation).WithMany(p => p.ToTvs)
            .HasForeignKey(d => d.TvNumber)
            .OnDelete(DeleteBehavior.Cascade)
            .HasConstraintName("to_tv_tv_number_fkey");
    });

    modelBuilder.Entity<ToTvExecutor>(entity =>
    {
        entity.HasKey(e => e.Employee)
            .HasName("to_tv_executor_pkey");

        entity.ToTable("to_tv_executor");

        entity.Property(e => e.Employee)

```

```

        .ValueGeneratedNever()
        .HasColumnName("employee");
entity.Property(e => e.Id)
    .ValueGeneratedOnAdd()
    .HasColumnName("id");
entity.Property(e => e.ToTv).HasColumnName("to_tv");

entity.HasOne(d => d.EmployeeNavigation).WithOne(p =>
p.ToTvExecutor)
    .HasForeignKey<ToTvExecutor>(d => d.Employee)

.HasConstraintName("to_tv_executor_employee_fkey");

entity.HasOne(d => d.ToTvNavigation).WithMany(p =>
p.ToTvExecutors)
    .HasForeignKey(d => d.ToTv)
    .OnDelete(DeleteBehavior.Cascade)
    .HasConstraintName("to_tv_executor_to_tv_fkey");
});

modelBuilder.Entity<ToType>(entity =>
{
    entity.HasKey(e => e.Id).HasName("to_type_pkey");

    entity.ToTable("to_type");

    entity.Property(e => e.Id).HasColumnName("id");
    entity.Property(e => e.Name)
        .HasColumnType("character varying")
        .HasColumnName("name");
});

modelBuilder.Entity<TransportVenichle>(entity =>
{
    entity.HasKey(e
e.Number).HasName("transport_venichle_pkey");

    entity.ToTable("transport_venichle");

    entity.Property(e => e.Number)
        .ValueGeneratedNever()
        .HasColumnName("number");
    entity.Property(e => e.Autohod)
        .HasDefaultValueSql("false")
        .HasColumnName("autohod");
    entity.Property(e => e.CommissioningDate)
        .HasColumnType("timestamp without time zone")
        .HasColumnName("commissioning_date");
    entity.Property(e => e.GovNumber)
        .HasMaxLength(10)
        .HasColumnName("gov_number");
}
=>

```

```

        entity.Property(e => e.Mark)
            .HasMaxLength(30)
            .HasColumnName("mark");
        entity.Property(e => e.Park).HasColumnName("park");
        entity.Property(e
e.Probeg).HasColumnName("probeg");
        entity.Property(e
e.ReleaseYear).HasColumnName("release_year");
        entity.Property(e
e.Status).HasColumnName("status");
        entity.Property(e => e.Vin)
            .HasMaxLength(50)
            .HasColumnName("vin");

        entity.HasOne(d => d.ParkNavigation).WithMany(p =>
p.TransportVenichles)
            .HasForeignKey(d => d.Park)
            .OnDelete(DeleteBehavior.SetNull)

.HasConstraintName("transport_venichle_park_fkey");

        entity.HasOne(d => d.StatusNavigation).WithMany(p =>
p.TransportVenichles)
            .HasForeignKey(d => d.Status)
            .OnDelete(DeleteBehavior.SetNull)

.HasConstraintName("transport_venichle_status_fkey");
    });

    modelBuilder.Entity<TvStatus>(entity =>
    {
        entity.HasKey(e => e.Id).HasName("tv_statuses_pkey");

        entity.ToTable("tv_statuses");

        entity.Property(e => e.Id).HasColumnName("id");
        entity.Property(e => e.StatusName)
            .HasMaxLength(30)
            .HasColumnName("status_name");
    });

    OnModelCreatingPartial(modelBuilder);
}

partial void OnModelCreatingPartial(ModelBuilder
modelBuilder);
}

Файл «TvStatus.cs»

using System;
using System.Collections.Generic;

```

```

namespace trolleybus.Model;

public partial class TvStatus
{
    public string StatusName { get; set; } = null!;

    public int Id { get; set; }

    public virtual ICollection<TransportVenichle>
TransportVenichles { get; set; } = new List<TransportVenichle>();
    Файл «TvStatusPartial.cs»

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

namespace trolleybus.Model
{
    public partial class TvStatus
    {
        public override string ToString()
        {
            return StatusName;
        }
    }
}

    Файл «CheckComBoxHelper.cs»
using HandyControl.Controls;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;

namespace trolleybus.Other
{
    public class CheckComBoxHelper
    {
        static bool _isUpdating = false;

        public static readonly DependencyProperty
SelectedItemsProperty =
        DependencyProperty.RegisterAttached("SelectedItems",
        typeof(List<string>), typeof(CheckComBoxHelper),
        new FrameworkPropertyMetadata(null, new
PropertyChangingCallback(OnPropertyChanged)));

        public static List<string>
GetSelectedItems(DependencyObject d)

```



```

        {
            return d.GetValue(SelectedItemsProperty) as
List<string>;
        }

        public static void SetSelectedItems(DependencyObject d,
List<string> value)
        {
            d.SetValue(SelectedItemsProperty, value);
        }

        private static void OnPropertyChanged(DependencyObject d,
DependencyPropertyChangedEventArgs e)
        {
            CheckComboBox checkComboBox = d as CheckComboBox;

            if (!_isUpdating)
            {
                var result = e.NewValue as List<string>;
                checkComboBox.SelectedItems.Clear();
                if (result != null)
                {
                    foreach (var item in
(List<string>)e.NewValue)
                    {
                        checkComboBox.SelectedItems.Add(item);
                    }
                }
                else
                {
                    checkComboBox.SelectedItems.Add(e.NewValue);
                }
            }
        }

        public static readonly DependencyProperty AttachProperty
=
        DependencyProperty.RegisterAttached("Attach",
typeof(bool), typeof(CheckComBoxHelper),
        new FrameworkPropertyMetadata(default(bool), new
PropertyChangedCallback(OnAttached)));

        public static bool GetAttach(DependencyObject d)
        {
            return (bool)d.GetValue(AttachProperty);
        }

        public static void SetAttach(DependencyObject d, bool
value)
        {
            d.SetValue(AttachProperty, value);
        }

```

```

    }

    private static void OnAttached(DependencyObject d,
DependencyPropertyChangedEventArgs e)
    {
        CheckComboBox checkComboBox = d as CheckComboBox;
        checkComboBox.SelectionChanged +=
CheckComboBoxSelectionChanged;
    }

    private static void CheckComboBoxSelectionChanged(object
sender, RoutedEventArgs e)
    {
        CheckComboBox checkComboBox = sender as CheckComboBox;
        _isUpdating = true;
        List<string> temp = new List<string>();
        foreach (var item in checkComboBox.SelectedItems)
        {
            temp.Add(item.ToString());
        }
        SetSelectedItems(checkComboBox, temp);
        _isUpdating = false;
    }
}

```

Файл «CurrentUser.cs»

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using trolleybus.Model;

namespace trolleybus.Other
{
    public static class CurrentUser
    {
        public static Employee Employee { get; set; }

        public static Visibility IsVisible
        {
            get
            {
                if (Employee.Post == 4)
                {
                    return Visibility.Visible;
                }
                else
                {
                    return Visibility.Collapsed;
                }
            }
        }
    }
}

```

Файл « MessageBoxes.cs»

```
using HandyControl.Data;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using trolleybus.Model;

namespace trolleybus.Other
{
    public static class MessageBoxes
    {
        public static MessageBoxResult WrongPassword()
        {
            return HandyControl.Controls.MessageBox.Show(new
MessageBoxInfo
            {
                Message = "Не верный пароль",
                Caption = "Ошибка",
                Button = MessageBoxButton.OK,
            });
        }
    }
}
```

Файл « RelayCommand.cs»

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;

namespace trolleybus.Other
{
    public class RelayCommand: ICommand
    {
        private readonly Func<object, bool> _canExecute;
        private readonly Action<object> _onExecute;

        public event EventHandler CanExecuteChanged
        {
            add { CommandManager.RequerySuggested += value; }
            remove { CommandManager.RequerySuggested -= value; }
        }

        public RelayCommand(Action<object> execute, Func<object,
bool> canExecute = null)
        {
            _onExecute = execute;
        }
    }
}
```

```

        _canExecute = canExecute;
    }

    public bool CanExecute(object parameter) => _canExecute ==
    null ? true : _canExecute.Invoke(parameter);

    public void Execute(object parameter) =>
    _onExecute?.Invoke(parameter);
}
}

```

Файл «DispatcherWindow.xaml»

```

<hc:Window x:Class="trolleybus.View.DispatcherWindow"

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:hc="https://handyorg.github.io/handycontrol"

xmlns:iconPacks="http://metro.mahapps.com/winfx/xaml/iconpacks"
    xmlns:local="clr-namespace:trolleybus.View"

xmlns:filterdg="http://filterdatagrid.control.com/2021"
    mc:Ignorable="d"
    Title="Троллейбусное депо"
    Height="450" Width="800"
    FontSize="14"
    WindowStartupLocation="CenterScreen"
    WindowState="Maximized"
    Closing="Window_Closing"
>
    <Window.Resources>
        <Style x:Key="BaseDataGrid"
TargetType="local:ExcelFilteredDataGrid">
            <Setter Property="SelectedItem" Value="{Binding
Selected, Mode=TwoWay}" />
            <Setter Property="FontSize" Value="14" />
            <Setter Property="AutoGenerateColumns"
Value="False" />
            <Setter Property="HeadersVisibility" Value="Column" />
            <Setter Property="ColumnWidth" Value="Auto" />
            <Setter Property="IsReadOnly" Value="True" />
            <Setter Property="hc:DataGridAttach.ShowRowNumber"
Value="True" />
            <Setter Property="FilterLanguage" Value="Russian" />
            <Setter Property="ShowStatusBar" Value="True" />

```

```

        <Setter                                Property="FilterPopupBackground"
Value="White"/>
        <Setter                                Property="DateFormatString"
Value="dd.MM.yyyy"/>
    </Style>
</Window.Resources>
<Grid >
    <Grid.RowDefinitions>
        <RowDefinition Height="30"/>
        <RowDefinition Height="1*"/>
    </Grid.RowDefinitions>
    <TabControl Grid.Row="1">
        <hc:Interaction.Triggers>
            <hc:EventTrigger EventName="SelectionChanged">
                <hc:EventToCommand                Command="{Binding
SelectionCommand}" PassEventArgsToCommand="True"/>
            </hc:EventTrigger>
        </hc:Interaction.Triggers>
        <TabItem Header="Транспортные средства" Name="Tv">
            <Grid Background="White">
                <Grid.RowDefinitions>
                    <RowDefinition Height="50"/>
                    <RowDefinition                Height="1*"
MinHeight="100"/>
                    <RowDefinition Height="Auto"/>
                    <RowDefinition                Height="1*"
MinHeight="100"/>
                </Grid.RowDefinitions>
                <StackPanel Orientation="Horizontal"
                    Grid.Row="0">
                    <Button Height="50"
                        Width="50"
                        Command="{Binding EditTvCommand}"
                        IsEnabled="{Binding
ElementName=TvGrid, Path=SelectedItem, Converter={StaticResource
Object2BooleanConverter}}">
                        <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                    </Button>
                    <Button Height="50"
                        Width="50"
                        Command="{Binding
UpdateTvCommand}">
                        <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"

```

```

HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
    <local:ExcelFilteredDataGrid x:Name="TvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding TransportVenichles}"

SelectedItem="{Binding SelectedTv}"

                                Grid.Row="1">
        <filterdg:FilterDataGrid.Columns>
            <filterdg:DataGridTextColumn
Header="№"

Binding="{Binding Path=Number}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мапка"

Binding="{Binding Path=Mark}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Гос. номер"

Binding="{Binding Path=GovNumber}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="VIN-номер"

Binding="{Binding Path=Vin}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Год выпуска"

Binding="{Binding Path=ReleaseYear}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Дата ввода в эксплуатацию"

Binding="{Binding Path=CommissioningDate}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"

```

```

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Парк"

Binding="{Binding Path=ParkView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Автоход"

Binding="{Binding Path=Autohod}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Статус"

Binding="{Binding Path=StatusView}"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                <GridSplitter      Grid.Row="2"      Height="5"
HorizontalAlignment="Stretch"/>
                                <TabControl Grid.Row="3">
                                    <TabItem Header="Сход">
                                        <Grid Background="White">
                                            <Grid.RowDefinitions>
                                                <RowDefinition Height="50"/>
                                                <RowDefinition Height="1*" />
                                            </Grid.RowDefinitions>
                                            <StackPanel
Orientation="Horizontal"
Grid.Row="0">
                                                <Button Height="50"
Width="50"
Command="{Binding
AddShodTvCommand}"
IsEnabled="{Binding
ElementName=TvGrid, Path=SelectedItem, Converter={StaticResource
Object2BooleanConverter}}">
<iconPacks:PackIconMaterial Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

```

```

</Button>
<Button Height="50"
        Width="50"
        Command="{Binding
EditShodTvCommand}"
        IsEnabled="{Binding
ElementName=ShodTvGrid,      Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Pen"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
</Button>
</StackPanel>
<local:ExcelFilteredDataGrid
x:Name="ShodTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ShodsTv}"

SelectedItem="{Binding SelectedShodTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Homep TC"

Binding="{Binding Path=TvNumber}"

IsReadOnly="True"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время схода"

Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsReadOnly="True"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

Width="350"

CanUserResize="False"

```



```

Binding="{Binding Path=Malfunction}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Диспетчер"

Binding="{Binding Path=DispatcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

Width="*" />

```

```

</filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Неисправности">
                                    <Grid Background="White">
                                        <local:ExcelFilteredDataGrid
x:Name="MalfunctionTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding MalfunctionsTv}"

SelectedItem="{Binding SelectedMalfunctionTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Заявка принята"

Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

Binding="{Binding Path=Malfunction1}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

```

```

<filterdg:DataGridTextColumn Header="Время конца работ"
Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"
IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Запрет эксплуатации"
Binding="{Binding Path=ZapretExpluatation}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Мастер"
Binding="{Binding Path=MasterView}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий работника"
Binding="{Binding Path=WorkComment}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Запчасти"
Binding="{Binding Path=TechParts}"
IsColumnFiltered="True"
Width="*" />

</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
            <TextBlock
Text="{Binding Path=StringExecutors}" />
        </StackPanel>
    </DataTemplate>
</filterdg:FilterDataGrid.RowDetailsTemplate>
</local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
<TabItem Header="ТО">
    <Grid Background="White">

```

```

        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition
Width="Auto"/>
                <ColumnDefinition
Width="1*"/>
            </Grid.ColumnDefinitions>
            <StackPanel
Orientation="Horizontal"
Grid.Row="0">
                <Button Height="50"
Width="50"
Command="{Binding DeleteToTvTvCommand}"
IsEnabled="{Binding ElementName=ToTvTvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                    <iconPacks:PackIconMaterial Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                </Button>
            </StackPanel>
        </Grid>
        <local:ExcelFilteredDataGrid
x:Name="ToTvTvGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding ToTvsTv}"
SelectedItem="{Binding SelectedToTvTv}"
Grid.Row="1">
            <filterdg:FilterDataGrid.Columns>
                <filterdg:DataGridTextColumn Header="Номер TC"
Binding="{Binding Path=TvNumber}"
IsColumnFiltered="True"/>
                <filterdg:DataGridTextColumn Header="Дата проведения"

```

```

Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробер"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Тип ТО"

Binding="{Binding Path=ToTypeView}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Следующее ТО"

Binding="{Binding Path=NextToView}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег для следующего ТО"

Binding="{Binding Path=NextToProbeg}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>

</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
                                <DataTemplate>
                                    <StackPanel

Orientation="Horizontal">
                                <TextBlock

Text="{Binding Path=StringExecutors}" />
                                </StackPanel>
                                </DataTemplate>

</filterdg:FilterDataGrid.RowDetailsTemplate>
                                </local:ExcelFilteredDataGrid>
                                </Grid>

```

```

        </TabItem>
    </TabControl>
</Grid>
</TabItem>
<TabItem Header="Сход" Name="Shod">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal"
            Grid.Row="0">
            <Button Height="50"
                Width="50"
                Command="{Binding
AddShodCommand}">
                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                Width="50"
                Command="{Binding
EditShodCommand}"
                IsEnabled="{Binding
ElementName=ShodGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                Width="50"
                Command="{Binding
DeleteShodCommand}"
                IsEnabled="{Binding
ElementName=ShodGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
        </StackPanel>
    </Grid>
</TabItem>

```

```

        <Button Height="50"
                Width="50"
                Command="{Binding
UpdateShodCommand}">
                <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
</StackPanel>
<local:ExcelFilteredDataGrid
x:Name="ShodGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Shods}"
SelectedItem="{Binding SelectedShod}"
                                Grid.Row="1">
        <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn
Header="Homep TC"
Binding="{Binding Path=TvNumber}"
IsReadOnly="True"
IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Время схода"
Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"
IsReadOnly="True"
IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Неисправность"
Width="350"
CanUserResize="False"
Binding="{Binding Path=Malfunction}"
IsColumnFiltered="True"/>
        <filterdg:DataGridCheckBoxColumn
Header="Эвакуатор"

```

```

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Диспетчер"

Binding="{Binding Path=DispetcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Неисправности" Name="Malfunction">
                                    <Grid Background="White">

```



```

        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="1*"/>
            </Grid.ColumnDefinitions>
            <StackPanel Orientation="Horizontal"
                Grid.Row="0">
                <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateMalfunctionCommand}">
                    <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                </Button>
            </StackPanel>
        </Grid>
        <local:ExcelFilteredDataGrid
x:Name="MalfunctionGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Malfunctions}"
SelectedItem="{Binding SelectedMalfunction}"
                Grid.Row="1">
                <filterdg:FilterDataGrid.Columns>
                    <filterdg:DataGridTextColumn
Header="Номер ТС"
Binding="{Binding Path=TvNumber}"
IsColumnFiltered="True"/>
                    <filterdg:DataGridTextColumn
Header="Заявка принята"
Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"
IsColumnFiltered="True"/>
                    <filterdg:DataGridTextColumn
Header="Неисправность"
Binding="{Binding Path=Malfunction1}"

```

```

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"
Binding="{Binding Path=Probeg}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время начала работ"
Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время конца работ"
Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Запрет эксплуатации"
Binding="{Binding Path=ZapretExpluatation}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мастер"
Binding="{Binding Path=MasterView}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий работника"
Binding="{Binding Path=WorkComment}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Запчасти"
Binding="{Binding Path=TechParts}"
IsColumnFiltered="True"
Width="*" />
                                </filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>

```

```

Orientation="Horizontal">
                                <StackPanel
                                <TextBlock      Text="{Binding
Path=StringExecutors}" />
                                </StackPanel>
                                </DataTemplate>

</filterdg:FilterDataGrid.RowDetailsTemplate>
    </local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
<TabItem Header="Техническое обслуживание">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto"/>
                <ColumnDefinition Width="1*"/>
            </Grid.ColumnDefinitions>
            <StackPanel Orientation="Horizontal"
                Grid.Row="0">
                <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateToTvCommand}">
                    <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                </Button>
            </StackPanel>
        </Grid>
        <local:ExcelFilteredDataGrid
x:Name="ToTvGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding ToTvs}"
SelectedItem="{Binding SelectedToTv}"
                                Grid.Row="1">
                                <filterdg:FilterDataGrid.Columns>
                                    <filterdg:DataGridTextColumn
Header="Номер ТС"
Binding="{Binding Path=TvNumber}"

```

```

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Дата проведения"

Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Тип ТО"

Binding="{Binding Path=ToTypeView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Следующее ТО"

Binding="{Binding Path=NextToView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег для следующего ТО"

Binding="{Binding Path=NextToProbeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>
                                </filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
                                <DataTemplate>
                                    <StackPanel
Orientation="Horizontal">
                                                <TextBlock Text="{Binding
Path=StringExecutors}" />
                                                </StackPanel>
                                </DataTemplate>

```

```

</filterdg:FilterDataGrid.RowDetailsTemplate>
    </local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
</TabControl>
</Grid>
</hc:Window>

Файл «EditDbUser.xaml»
<Window x:Class="trolleybus.View.EditDbUser"

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:trolleybus.View"
    xmlns:hc="https://handyorg.github.io/handycontrol"
    mc:Ignorable="d"
    Title="Пользователь БД"
    SizeToContent="WidthAndHeight"
    WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">
    <TextBlock Text="Сотрудник"/>
    <hc:ComboBox Name="TvCB"
        SelectedValue = "{Binding Entity.Employee}"
        SelectedValuePath="Id"
        DisplayMemberPath="FIO"
        ItemsSource="{Binding Employees}"
        IsEnabled="{Binding NewDbUser}"
        Height="46"/>

    <TextBlock Text="Пароль"/>
    <TextBox Name="MalfunctionTextBox"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.Password,
UpdateSourceTrigger=PropertyChanged}"
        MaxLength="500"
        Grid.Row="1"
        Width="250"
        Height="46"/>

    <Button Name="Save"
        IsDefault="True"
        Content="Сохранить"
        Command="{Binding SaveCommand}"
        Click="Save_Click"

```

```

        Grid.Row="10"
        Margin="0,15,0,0"
        HorizontalAlignment="Stretch"/>
    </StackPanel>
</Window>
<Window x:Class="trolleybus.View.EditEmployee"

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:trolleybus.View"
    xmlns:hc="https://handyorg.github.io/handycontrol"
    mc:Ignorable="d"
    Title="Пользователь БД"
    SizeToContent="WidthAndHeight"
    WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">
    <TextBlock Text="Фамилия"/>
    <TextBox Name="LastNameTB"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.Lastname}"
        MaxLength="500"
        Grid.Row="1"
        Width="250"
        Height="46"/>

    <TextBlock Text="Имя"/>
    <TextBox Name="NameTb"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.Name}"
        MaxLength="500"
        Grid.Row="1"
        Width="250"
        Height="46"/>

    <TextBlock Text="Отчество"/>
    <TextBox Name="OtchestvoTb"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.Otchestvo}"
        MaxLength="500"
        Grid.Row="1"

```

```

        Width="250"
        Height="46"/>

<TextBlock Text="Должность"/>
<hc:ComboBox Name="PostCB"
    SelectedValue = "{Binding Entity.Post}"
    SelectedValuePath="Id"
    DisplayMemberPath="Name"
    ItemsSource="{Binding Posts}"
    Height="46"/>

<TextBlock Text="Парк"/>
<hc:ComboBox Name="ParkCB"
    SelectedValue = "{Binding Entity.Park}"
    SelectedValuePath="Id"
    DisplayMemberPath="Name"
    ItemsSource="{Binding Parks}"
    Height="46"/>

<Button Name="Save"
    IsDefault="True"
    Content="Сохранить"
    Command="{Binding SaveCommand}"
    Click="Save_Click"
    Grid.Row="10"
    Margin="0,15,0,0"
    HorizontalAlignment="Stretch"/>
</StackPanel>
</Window>

        Файл «EditMalfunction.xaml»
<Window x:Class="trolleybus.View.EditMalfunction"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:trolleybus.View"
    xmlns:other="clr-namespace:trolleybus.Other"
    xmlns:hc="https://handyorg.github.io/handycontrol"
    Title="Неисправность"
    SizeToContent="WidthAndHeight"
    WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">
    <TextBlock Text="Номер ТС"/>
    <hc:ComboBox Name="TvCB"
        SelectedValue = "{Binding Entity.TvNumber}"
        SelectedValuePath="Number"
        DisplayMemberPath="Number"

```

```

ItemsSource="{Binding TransportVenichles}"
IsEnabled="{Binding NewMalfunction}"
Height="46"/>

<TextBlock Text="Пробег"/>
<hc:NumericUpDown Name="ProbegTB"
    HorizontalContentAlignment="Left"
    VerticalContentAlignment="Center"
    Value="{Binding Entity.Probeg,
UpdateSourceTrigger=PropertyChanged}"
    Width="250"
    Height="46"/>

<TextBlock Text="Неисправность"/>
<TextBox Name="CancelTextBox"
    HorizontalContentAlignment="Left"
    VerticalContentAlignment="Top"
    Padding="5"
    Text="{Binding Entity.Malfunction1,
UpdateSourceTrigger=PropertyChanged}"
    MaxLength="500"
    TextWrapping="Wrap"
    AcceptsReturn="True"
    Grid.Row="4"
    Width="250"
    Height="46"/>

<CheckBox Content="Запрет эксплуатации"
    IsChecked="{Binding Entity.ZapretExpluatation}"
    Grid.Row="2"
    Height="46"/>

<TextBlock Text="Комментарий мастера"/>
<TextBox Name="WorkCommentTB"
    HorizontalContentAlignment="Left"
    VerticalContentAlignment="Top"
    Padding="5"
    Text="{Binding Entity.WorkComment,
UpdateSourceTrigger=PropertyChanged}"
    MaxLength="500"
    TextWrapping="Wrap"
    AcceptsReturn="True"
    Grid.Row="4"
    Width="250"
    Height="46"/>

<TextBlock Text="Запчасти"/>
<TextBox Name="TechPartsTB"
    HorizontalContentAlignment="Left"
    VerticalContentAlignment="Top"
    Padding="5"

```



```

        Text="{Binding Entity.TechParts,
UpdateSourceTrigger=PropertyChanged}"
        MaxLength="500"
        TextWrapping="Wrap"
        AcceptsReturn="True"
        Grid.Row="4"
        Width="250"
        Height="46"/>

<TextBlock Text="Исполнители"/>
<hc:CheckComboBox Grid.Column="1"
        TabIndex="19"
        Name="ExecutorCCBox"
        ItemsSource="{Binding Executors}"
        ShowSelectAllButton="True"
        MinHeight="35"
        SelectionMode="Multiple"
        other:CheckComBoxHelper.Attach="True"

other:CheckComBoxHelper.SelectedItems="{Binding
        SelectedExecutors, Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}"/>

<Button Name="Begin"
        IsDefault="True"
        Content="Начать работу"
        Command="{Binding BeginCommand}"
        Visibility="{Binding IsToBegin,
Converter={StaticResource Boolean2VisibilityConverter}}"
        Grid.Row="10"
        Margin="0,15,0,0"
        HorizontalAlignment="Stretch"/>

<Button Name="End"
        IsDefault="True"
        Content="Закончить работу"
        Command="{Binding EndCommand}"
        Visibility="{Binding IsToEnd,
Converter={StaticResource Boolean2VisibilityConverter}}"
        Grid.Row="10"
        Margin="0,15,0,0"
        HorizontalAlignment="Stretch"/>

<TextBlock Text="Неисправность для акта"
        Visibility="{Binding Vision}"/>
<TextBox Name="MalfAktTB"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.MalfunctionRecl,
UpdateSourceTrigger=PropertyChanged}"

```

```

        MaxLength="500"
        TextWrapping="Wrap"
        AcceptsReturn="True"
        Visibility="{Binding Vision}"
        Grid.Row="4"
        Width="250"
        Height="46"/>

<TextBlock Text="Комментарий для акта"
    Visibility="{Binding Vision}"/>
<TextBox Name="AktCommentTB"
    HorizontalContentAlignment="Left"
    VerticalContentAlignment="Top"
    Padding="5"
    Text="{Binding Entity.MalfunctionReclComment,
UpdateSourceTrigger=PropertyChanged}"
    MaxLength="500"
    TextWrapping="Wrap"
    AcceptsReturn="True"
    Visibility="{Binding Vision}"
    Grid.Row="4"
    Width="250"
    Height="46"/>

<CheckBox Content="Подписан"
    IsChecked="{Binding Entity.Signed}"
    Visibility="{Binding Vision}"
    Grid.Row="2"
    Height="46"/>

<CheckBox Content="Устранено"
    IsChecked="{Binding Entity.Done}"
    Visibility="{Binding Vision}"
    Grid.Row="2"
    Height="46"/>

<Button Name="Save"
    IsDefault="True"
    Content="Сохранить"
    Command="{Binding SaveCommand}"
    Grid.Row="10"
    Margin="0,15,0,0"
    HorizontalAlignment="Stretch"
    Click="Save_Click"/>
</StackPanel>
</Window>

        Файл « EditPark.xaml »
<Window x:Class="trolleybus.View.EditPark"

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:trolleybus.View"
xmlns:hc="https://handyorg.github.io/handycontrol"
mc:Ignorable="d"
Title="Пользователь БД"
SizeToContent="WidthAndHeight"
WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">
  <TextBlock Text="Название"/>
  <TextBox Name="MalfunctionTextBox"
    HorizontalContentAlignment="Left"
    VerticalContentAlignment="Top"
    Padding="5"
    Text="{Binding Entity.Name,
UpdateSourceTrigger=PropertyChanged}"
    MaxLength="500"
    Grid.Row="1"
    Width="250"
    Height="46"/>

  <Button Name="Save"
    IsDefault="True"
    Content="Сохранить"
    Command="{Binding SaveCommand}"
    Click="Save_Click"
    Grid.Row="10"
    Margin="0,15,0,0"
    HorizontalAlignment="Stretch"/>
</StackPanel>
</Window>

Файл « EditShod.xaml »
<Window x:Class="trolleybus.View.EditShod"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
xmlns:hc="https://handyorg.github.io/handycontrol"
Title="Сход"
SizeToContent="WidthAndHeight"
WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">
  <TextBlock Text="Номер ТС"/>
  <hc:ComboBox Name="TvCB"
    SelectedValue = "{Binding Entity.TvNumber}"
    SelectedValuePath="Number"
    DisplayMemberPath="Number"
    ItemsSource="{Binding TransportVenichles}"

```

```

        IsEnabled="{Binding NewShod}"
        Height="46"/>

<TextBlock Text="Неисправность"/>
<TextBox Name="MalfunctionTextBox"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.Malfunction,
UpdateSourceTrigger=PropertyChanged}"
        IsReadOnly="{Binding NewShod,
Converter={StaticResource Boolean2BooleanReConverter}}"
        MaxLength="500"
        TextWrapping="Wrap"
        AcceptsReturn="True"
        Grid.Row="1"
        Width="250"
        Height="46"/>

<CheckBox Content="Эвакуатор"
        IsChecked="{Binding Entity.Evacuator}"
        IsEnabled="{Binding NewShod}"
        Grid.Row="2"
        Height="46"/>

<CheckBox Name="Canec1Check"
        Content="Отмена"
        IsChecked="{Binding Cancel}"
        IsEnabled="{Binding NewShod,
Converter={StaticResource Boolean2BooleanReConverter}}"
        Grid.Row="3"
        Height="46"/>

<TextBox Name="CancelTextBox"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Top"
        Padding="5"
        Text="{Binding Entity.CancelComment,
UpdateSourceTrigger=PropertyChanged}"
        MaxLength="500"
        TextWrapping="Wrap"
        AcceptsReturn="True"
        Grid.Row="4"
        IsEnabled="{Binding ElementName=Canec1Check,
Path=IsChecked}"
        IsReadOnly="{Binding NewShod}"
        Width="250"
        Height="46"/>
<CheckBox Content="Прибыл"
        IsChecked="{Binding Arrived}"

```

```

        IsEnabled="{Binding NewShod,
Converter={StaticResource Boolean2BooleanReConverter}}"
        Grid.Row="5"
        Height="46"/>

        <Button Name="Save"
            IsDefault="True"
            Content="Сохранить"
            Command="{Binding SaveCommand}"
            Click="Save_Click"
            Grid.Row="10"
            Margin="0,15,0,0"
            HorizontalAlignment="Stretch"/>
    </StackPanel>
</Window>

    Файл « EditToTv.xaml »
<Window x:Class="trolleybus.View.EditToTv"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation
"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:hc="https://handyorg.github.io/handycontrol"
    Title="ТО"
    SizeToContent="WidthAndHeight"
    xmlns:other="clr-namespace:trolleybus.Other"
    WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">
    <TextBlock Text="Home TC"/>
    <hc:ComboBox Name="TvCB"
        SelectedValue = "{Binding Entity.TvNumber}"
        SelectedValuePath="Number"
        DisplayMemberPath="Number"
        ItemsSource="{Binding TransportVenichles}"
        Height="46"/>

    <TextBlock Text="Тип ТО"/>
    <hc:ComboBox Name="ToType"
        SelectedItem="{Binding
Entity.ToTypeNavigation}"
        SelectedValue = "{Binding Entity.ToType}"
        SelectedValuePath="Id"
        DisplayMemberPath="Name"
        ItemsSource="{Binding ToTypes}"
        Height="46"/>

    <TextBlock Text="Пробег"/>
    <hc:NumericUpDown Name="Probeg"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Center"
        Value="{Binding Entity.Probeg,
UpdateSourceTrigger=PropertyChanged}"

```

```

Width="250"
Height="46"/>

<TextBlock Text="Исполнители"/>
<hc:CheckComboBox Grid.Column="1"
    TabIndex="19"
    Name="ExecutorCCBox"
    ItemsSource="{Binding Executors}"
    ShowSelectAllButton="True"
    MinHeight="35"
    SelectionMode="Multiple"
    other:CheckComBoxHelper.Attach="True"

other:CheckComBoxHelper.SelectedItems="{Binding
    SelectedExecutors,                      Mode=TwoWay,
UpdateSourceTrigger=PropertyChanged}"/>

<Button Name="Save"
    IsDefault="True"
    Content="Сохранить"
    Command="{Binding SaveCommand}"
    Click="Save_Click"
    Grid.Row="10"
    Margin="0,15,0,0"
    HorizontalAlignment="Stretch"/>
</StackPanel>
</Window>

Файл « EditTv.xaml »
<Window x:Class="trolleybus.View.EditTv"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation
"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:hc="https://handyorg.github.io/handycontrol"
    Title="Троллейбус"
    SizeToContent="WidthAndHeight"
    WindowStartupLocation="CenterScreen">
<StackPanel Margin="15" Width="250">

    <TextBlock Text="Номер ТС"
        Visibility="{Binding Vision}"/>
    <hc:NumericUpDown Name="NumberTB"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Center"
        Visibility="{Binding Vision}"
        Value="{Binding                      Entity.Number,
UpdateSourceTrigger=PropertyChanged}"
        Width="250"
        Height="46"/>

```

					ПГУ 1.090301.07.001 ПЗ	Лист
						110
Изм.	Лист	№ докум.	Подпись	Дата		

```

        <TextBlock Text="Mapka TC"
                Visibility="{Binding Vision}"/>
        <TextBox Name="MarkTB"
                HorizontalContentAlignment="Left"
                VerticalContentAlignment="Center"
                Text="{Binding                                Entity.Mark,
UpdateSourceTrigger=PropertyChanged}"
                MaxLength="30"
                Visibility="{Binding Vision}"
                Width="250"
                Height="46"/>

        <TextBlock Text="Гос. номер"
                Visibility="{Binding Vision}"/>
        <TextBox Name="GosNomerTB"
                HorizontalContentAlignment="Left"
                VerticalContentAlignment="Center"
                Text="{Binding                                Entity.GovNumber,
UpdateSourceTrigger=PropertyChanged}"
                MaxLength="10"
                Visibility="{Binding Vision}"
                Width="250"
                Height="46"/>

        <TextBlock Text="VIN"
                Visibility="{Binding Vision}"/>
        <TextBox Name="VINTB"
                HorizontalContentAlignment="Left"
                VerticalContentAlignment="Center"
                Text="{Binding                                Entity.Vin,
UpdateSourceTrigger=PropertyChanged}"
                MaxLength="30"
                Visibility="{Binding Vision}"
                Width="250"
                Height="46"/>

        <TextBlock Text="Пробег"/>
        <hc:NumericUpDown Name="Probeg"
                HorizontalContentAlignment="Left"
                VerticalContentAlignment="Center"
                Value="{Binding                                Entity.Probeg,
UpdateSourceTrigger=PropertyChanged}"
                Width="250"
                Height="46"/>

        <TextBlock Text="Год выпуска"
                Visibility="{Binding Vision}"/>
        <hc:TextBox Name="ReleaseYearTB"
                Visibility="{Binding Vision}"
                HorizontalContentAlignment="Left"
                VerticalContentAlignment="Center"

```

```

        Text="{Binding Entity.ReleaseYear,
UpdateSourceTrigger=PropertyChanged}"
        MaxLength="4"
        Width="250"
        Height="46"/>

        <TextBlock Text="Дата начала эксплуатации"
        Visibility="{Binding Vision}"/>
        <hc:DatePicker Name="CommissioningDateDP"
        HorizontalContentAlignment="Left"
        VerticalContentAlignment="Center"
        SelectedDate="{Binding
Entity.CommissioningDate, UpdateSourceTrigger=PropertyChanged}"
        Visibility="{Binding Vision}"
        Width="250"
        Height="46"/>
        <CheckBox Content="Автоход"
        Name="AutohodCB"
        IsChecked="{Binding Entity.Autohod,
UpdateSourceTrigger=PropertyChanged}"
        Height="46"/>

        <TextBlock Text="Статус"/>
        <hc:ComboBox Name="StatusCB"
        SelectedValue = "{Binding Entity.Status}"
        SelectedValuePath="Id"
        DisplayMemberPath="StatusName"
        ItemsSource="{Binding Statuses}"
        Height="46"/>

        <TextBlock Text="Парк"
        Visibility="{Binding Vision}"/>
        <hc:ComboBox Name="ParkCB"
        SelectedValue = "{Binding Entity.Park}"
        SelectedValuePath="Id"
        DisplayMemberPath="Name"
        Visibility="{Binding Vision}"
        ItemsSource="{Binding Parks}"
        Height="46"/>

        <Button Name="Save"
        Margin="0,15,0,0"
        HorizontalAlignment="Stretch"
        Content="Сохранить"
        Command="{Binding SaveCommand}"
        Click="Save_Click"/>
    </StackPanel>
</Window>

```


Файл « ExcelFilteredDataGrid.xaml»

```

<filterdg:FilterDataGrid
x:Class="trolleybus.View.ExcelFilteredDataGrid"
    x:Name="efdg"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:local="clr-namespace:trolleybus.View"
    mc:Ignorable="d"
    d:DesignHeight="450" d:DesignWidth="800"

xmlns:filterdg="http://filterdatagrid.control.com/2021"

xmlns:hc="https://handyorg.github.io/handycontrol"
    FilterLanguage="Russian">
    <filterdg:FilterDataGrid.ContextMenu>
        <ContextMenu>
            <MenuItem Header="Экспорт" В эксель"
Click="Excel_Click"/>
        </ContextMenu>
    </filterdg:FilterDataGrid.ContextMenu>
</filterdg:FilterDataGrid>

```

Файл « GuaranteeWindow.xaml»

```

<hc:Window x:Class="trolleybus.View.GuaranteeWindow"

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:hc="https://handyorg.github.io/handycontrol"

xmlns:iconPacks="http://metro.mahapps.com/winfx/xaml/iconpacks"
    xmlns:local="clr-namespace:trolleybus.View"

xmlns:filterdg="http://filterdatagrid.control.com/2021"
    mc:Ignorable="d"
    Title="Троллейбусное депо"
    Height="450" Width="800"
    FontSize="14"

```

```

        WindowStartupLocation="CenterScreen"
        WindowState="Maximized"
        Closing="Window_Closing"
    >
    <Window.Resources>
        <Style
            x:Key="BaseDataGrid"
            TargetType="local:ExcelFilteredDataGrid">
            <Setter Property="SelectedItem" Value="{Binding
                Selected, Mode=TwoWay}" />
            <Setter Property="FontSize" Value="14" />
            <Setter
                Property="AutoGenerateColumns"
                Value="False" />
            <Setter Property="HeadersVisibility" Value="Column" />
            <Setter Property="ColumnWidth" Value="Auto" />
            <Setter Property="IsReadOnly" Value="True" />
            <Setter
                Property="hc:DataGridAttach.ShowRowNumber"
                Value="True" />
            <Setter Property="FilterLanguage" Value="Russian" />
            <Setter Property="ShowStatusBar" Value="True" />
            <Setter
                Property="FilterPopupBackground"
                Value="White" />
            <Setter
                Property="DateFormatString"
                Value="dd.MM.yyyy" />
        </Style>
    </Window.Resources>
    <Grid >
        <Grid.RowDefinitions>
            <RowDefinition Height="30" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <TabControl Grid.Row="1">
            <hc:Interaction.Triggers>
                <hc:EventTrigger EventName="SelectionChanged">
                    <hc:EventToCommand
                        Command="{Binding
                            SelectionCommand}"
                        PassEventArgsToCommand="True" />
                </hc:EventTrigger>
            </hc:Interaction.Triggers>
            <TabItem Header="Транспортные средства" Name="Tv">
                <Grid Background="White">
                    <Grid.RowDefinitions>
                        <RowDefinition Height="50" />
                        <RowDefinition
                            Height="1*"
                            MinHeight="100" />
                        <RowDefinition Height="Auto" />
                        <RowDefinition
                            Height="1*"
                            MinHeight="100" />
                    </Grid.RowDefinitions>
                    <StackPanel Orientation="Horizontal"
                        Grid.Row="0">
                        <Button Height="50"
                            Width="50"

```

```

        Command="{Binding AddTvCommand}">
        <iconPacks:PackIconMaterial

Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding EditTvCommand}"
                IsEnabled="{Binding
ElementName=GridView, Path=SelectedItem, Converter={StaticResource
Object2BooleanConverter}}">
        <iconPacks:PackIconMaterial

Kind="Pen"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
DeleteTvCommand}"
                IsEnabled="{Binding
ElementName=GridView, Path=SelectedItem, Converter={StaticResource
Object2BooleanConverter}}">
        <iconPacks:PackIconMaterial

Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
UpdateTvCommand}">
        <iconPacks:PackIconMaterial

Kind="Update"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
    <local:ExcelFilteredDataGrid x:Name="GridView"

Style="{StaticResource BaseDataGrid}"

```

```

ItemsSource="{Binding TransportVenichles}"

SelectedItem="{Binding SelectedTv}"
                                Grid.Row="1">
                                <filterdg:FilterDataGrid.Columns>
                                <filterdg:DataGridTextColumn
Header="№"

Binding="{Binding Path=Number}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мапка"

Binding="{Binding Path=Mark}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Гос. номер"

Binding="{Binding Path=GovNumber}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="VIN-номер"

Binding="{Binding Path=Vin}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Год выпуска"

Binding="{Binding Path=ReleaseYear}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Дата ввода в эксплуатацию"

Binding="{Binding Path=CommissioningDate}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Папк"

```

```

Binding="{Binding Path=ParkView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Автоход"

Binding="{Binding Path=Autohod}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Статус"

Binding="{Binding Path=StatusView}"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                <GridSplitter      Grid.Row="2"      Height="5"
HorizontalAlignment="Stretch"/>
                                <TabControl Grid.Row="3">
                                    <TabItem Header="Сход">
                                        <Grid Background="White">
                                            <Grid.RowDefinitions>
                                                <RowDefinition Height="50"/>
                                                <RowDefinition Height="1*" />
                                            </Grid.RowDefinitions>
                                            <StackPanel
Orientation="Horizontal"
Grid.Row="0">
                                                <Button Height="50"
Width="50"
Command="{Binding
AddShodTvCommand}"
IsEnabled="{Binding
ElementName=TVGrid, Path=SelectedItem, Converter={StaticResource
Object2BooleanConverter}}">
<iconPacks:PackIconMaterial Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
</Button>
<Button Height="50"
Width="50"
Command="{Binding
EditShodTvCommand}"

```

```

        IsEnabled="{Binding
ElementName=ShodTvGrid,           Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Pen"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

        </Button>
        <Button Height="50"
        Width="50"
        Command="{Binding
DeleteShodTvCommand}"
        IsEnabled="{Binding
ElementName=ShodTvGrid,           Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

        </Button>
    </StackPanel>
<local:ExcelFilteredDataGrid
x:Name="ShodTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ShodsTv}"

SelectedItem="{Binding SelectedShodTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsReadOnly="True"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время схода"

Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"

```

```

IsReadOnly="True"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Malfunction}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>

```

					ПГУ 1.090301.07.001 ПЗ	Лист
						119
Изм.	Лист	№ докум.	Подпись	Дата		

```

<filterdg:DataGridTextColumn Header="Диспетчер"
Binding="{Binding Path=DispatcherView}"
IsReadOnly="True"
IsColumnFiltered="True"
Width="*" />

</filterdg:FilterDataGrid.Columns>
    </local:ExcelFilteredDataGrid>
    </Grid>
</TabItem>
<TabItem Header="Неисправности">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition
Width="Auto" />
                <ColumnDefinition
Width="1*" />
            </Grid.ColumnDefinitions>
            <StackPanel
Orientation="Horizontal"
Grid.Row="0">
                <Button Height="50"
Width="50"
Command="{Binding AddMalfunctionTvCommand}"
IsEnabled="{Binding ElementName=TvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
<iconPacks:PackIconMaterial Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center" />
            </Button>
            <Button Height="50"
Width="50"
Command="{Binding EditMalfunctionTvCommand}"
IsEnabled="{Binding ElementName=MalfunctionTvGrid,

```



```

Path=SelectedItem,
Object2BooleanConverter}}">
Converter={StaticResource

<iconPacks:PackIconMaterial Kind="Pen"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

</Button>
<Button Height="50"
Width="50"

Command="{Binding DeleteMalfunctionTvCommand}"

IsEnabled="{Binding
Path=SelectedItem,
Object2BooleanConverter}}">
ElementName=MalfunctionTvGrid,
Converter={StaticResource

<iconPacks:PackIconMaterial Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

</Button>
<Button Height="50"
Width="50"

Command="{Binding ReclamationCommand}"

IsEnabled="{Binding
Path=SelectedItem,
Object2BooleanConverter}}">
ElementName=MalfunctionTvGrid,
Converter={StaticResource

CommandParameter="parent">

<iconPacks:PackIconMaterial Kind="FileDocument"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

</Button>
</StackPanel>
</Grid>
<local:ExcelFilteredDataGrid

x:Name="MalfunctionTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding MalfunctionsTv}"

SelectedItem="{Binding SelectedMalfunctionTv}"

```

```

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Заявка принята"

Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

Binding="{Binding Path=Malfunction1}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время конца работ"

Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Запрет эксплуатации"

Binding="{Binding Path=ZapretExpluatation}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Мастер"

Binding="{Binding Path=MasterView}"

```

```

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий работника"
Binding="{Binding Path=WorkComment}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Запчасти"
Binding="{Binding Path=TechParts}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Номер акта"
Binding="{Binding Path=Reklamation}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность для акта"
Binding="{Binding Path=MalfunctionRecl}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий для акта"
Binding="{Binding Path=MalfunctionReclComment}"
IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Подписан"
Binding="{Binding Path=Signed}"
IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Устранено"
Binding="{Binding Path=Done}"
IsColumnFiltered="True"
Width="*"/>

</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>

```

```

                                <DataTemplate>
                                    <StackPanel
Orientation="Horizontal">
                                <TextBlock
Text="{Binding Path=StringExecutors}" />
                                </StackPanel>
                                </DataTemplate>

</filterdg:FilterDataGrid.RowDetailsTemplate>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="ТО">
                                    <Grid Background="White">
                                        <Grid.RowDefinitions>
                                            <RowDefinition Height="50"/>
                                            <RowDefinition Height="1*"/>
                                        </Grid.RowDefinitions>
                                        <Grid Grid.Row="0">
                                            <Grid.ColumnDefinitions>
                                                <ColumnDefinition
Width="Auto"/>
                                                <ColumnDefinition
Width="1*"/>
                                            </Grid.ColumnDefinitions>
                                            <StackPanel
Orientation="Horizontal"
Grid.Row="0">
                                                <Button Height="50"
Width="50"

Command="{Binding AddToTvTvCommand}"

IsEnabled="{Binding ElementName=TvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
                                                </Button>
                                                <Button Height="50"
Width="50"

Command="{Binding EditToTvTvCommand}"

IsEnabled="{Binding ElementName=ToTvTvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Pen"

```

```

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
</Button>
<Button Height="50"
        Width="50"

Command="{Binding DeleteToTvTvCommand}"

IsEnabled="{Binding ElementName=ToTvTvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
</Button>
</StackPanel>
</Grid>
<local:ExcelFilteredDataGrid
x:Name="ToTvTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ToTvsTv}"

SelectedItem="{Binding SelectedToTvTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Дата проведения"

Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег"

Binding="{Binding Path=Probeg}"

```

```

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Тип ТО"
Binding="{Binding Path=ToTypeView}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Следующее ТО"
Binding="{Binding Path=NextToView}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег для следующего ТО"
Binding="{Binding Path=NextToProbeg}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Мастер"
Binding="{Binding Path=MasterView}"
IsColumnFiltered="True"/>

</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
            <TextBlock
Text="{Binding Path=StringExecutors}" />
        </StackPanel>
    </DataTemplate>
</filterdg:FilterDataGrid.RowDetailsTemplate>
</local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
</TabControl>
</Grid>
</TabItem>
<TabItem Header="Сход" Name="Shod">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>

```

```

        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal"
                    Grid.Row="0">
            <Button Height="50"
                    Width="50"
                    Command="{Binding
AddShodCommand}">
                                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                    Width="50"
                    Command="{Binding
EditShodCommand}"
                                <iconPacks:PackIconMaterial
IsEnabled="{Binding
ElementName=ShodGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                    Width="50"
                    Command="{Binding
DeleteShodCommand}"
                                <iconPacks:PackIconMaterial
IsEnabled="{Binding
ElementName=ShodGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                                <iconPacks:PackIconMaterial
Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateShodCommand}">
                                <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"

```

```

HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
    <local:ExcelFilteredDataGrid
x:Name="ShodGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding Shods}"

SelectedItem="{Binding SelectedShod}"

Grid.Row="1">
    <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn
Header="Homep TC"

Binding="{Binding Path=TvNumber}"

IsReadOnly="True"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Время схода"

Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsReadOnly="True"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Неисправность"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Malfunction}"

IsColumnFiltered="True"/>
        <filterdg:DataGridCheckBoxColumn
Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Время прибытия"

Width="350"

```

					ПГУ 1.090301.07.001 ПЗ	Лист
						128
Изм.	Лист	№ докум.	Подпись	Дата		


```

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Диспетчер"

Binding="{Binding Path=DispatcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Неисправности" Name="Malfunction">
                                    <Grid Background="White">
                                        <Grid.RowDefinitions>
                                            <RowDefinition Height="50"/>
                                            <RowDefinition Height="1*" />
                                        </Grid.RowDefinitions>
                                        <Grid Grid.Row="0">
                                            <Grid.ColumnDefinitions>
                                                <ColumnDefinition Width="Auto"/>
                                                <ColumnDefinition Width="1*" />

```

```

        </Grid.ColumnDefinitions>
        <StackPanel Orientation="Horizontal"
                    Grid.Row="0">
            <Button Height="50"
                    Width="50"
                    Command="{Binding
AddMalfunctionCommand}">
                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                    Width="50"
                    Command="{Binding
EditMalfunctionCommand}"
                    IsEnabled="{Binding
ElementName=MalfunctionGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                    Width="50"
                    Command="{Binding
DeleteMalfunctionCommand}"
                    IsEnabled="{Binding
ElementName=MalfunctionGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            </Button>
            <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateMalfunctionCommand}">
                <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"

```

```

HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
ReclamationCommand}"
                IsEnabled="{Binding
ElementName=MalfunctionGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                CommandParameter="no">
        <iconPacks:PackIconMaterial
Kind="FileDocument"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
</Grid>
<local:ExcelFilteredDataGrid
x:Name="MalfunctionGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Malfunctions}"
SelectedItem="{Binding SelectedMalfunction}"
                Grid.Row="1">
        <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn
Header="Номер ТС"
Binding="{Binding Path=TvNumber}"
IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Заявка принята"
Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"
IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Неисправность"
Binding="{Binding Path=Malfunction1}"
IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Пробег"

```

```

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время конца работ"

Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Запрет эксплуатации"

Binding="{Binding Path=ZapretExpluatation}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий работника"

Binding="{Binding Path=WorkComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Запчасти"

Binding="{Binding Path=TechParts}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Номер акта"

Binding="{Binding Path=Reklamation}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Неисправность для акта"

Binding="{Binding Path=MalfunctionRecl}"

```

```

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий для акта"

Binding="{Binding Path=MalfunctionReclComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Подписан"

Binding="{Binding Path=Signed}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Устранено"

Binding="{Binding Path=Done}"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
                                <DataTemplate>
                                    <StackPanel
Orientation="Horizontal">
                                        <TextBlock          Text="{Binding
Path=StringExecutors}" />
                                        </StackPanel>
                                    </DataTemplate>
                                </filterdg:FilterDataGrid.RowDetailsTemplate>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Техническое обслуживание">
                                    <Grid Background="White">
                                        <Grid.RowDefinitions>
                                            <RowDefinition Height="50"/>
                                            <RowDefinition Height="1*" />
                                        </Grid.RowDefinitions>
                                        <Grid Grid.Row="0">
                                            <Grid.ColumnDefinitions>
                                                <ColumnDefinition Width="Auto"/>
                                                <ColumnDefinition Width="1*" />
                                            </Grid.ColumnDefinitions>
                                            <StackPanel Orientation="Horizontal"
                                                Grid.Row="0">

```

```

        <Button Height="50"
                Width="50"
                Command="{Binding
AddToTvCommand}">
                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
EditToTvCommand}"
                IsEnabled="{Binding
ElementName=ToTvGrid,
Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
DeleteToTvCommand}"
                IsEnabled="{Binding
ElementName=ToTvGrid,
Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
UpdateToTvCommand}">
                <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
</StackPanel>

```

```

</Grid>
<local:ExcelFilteredDataGrid
x:Name="ToTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ToTvs}"

SelectedItem="{Binding SelectedToTv}"

Grid.Row="1">
    <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn
Header="Homep TC"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Дата проведения"

Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Тип ТО"

Binding="{Binding Path=ToTypeView}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Следующее ТО"

Binding="{Binding Path=NextToView}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Пробег для следующего ТО"

Binding="{Binding Path=NextToProbeg}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Мастер"

```

```

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>
</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
                                <TextBlock      Text="{Binding
Path=StringExecutors}" />
                                </StackPanel>
        </DataTemplate>
    </filterdg:FilterDataGrid.RowDetailsTemplate>
    </local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
<TabItem Header="Сотрудники" Name="Employee">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal"
Grid.Row="0">
            <Button Height="50"
Width="50"
Command="{Binding
AddEmployeeCommand}">
                                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
            <Button Height="50"
Width="50"
Command="{Binding
EditEmployeeCommand}"
                                IsEnabled="{Binding
ElementName=EmployeeGrid,      Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"

```



```

HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
DeleteEmployeeCommand}"
                IsEnabled="{Binding
ElementName=EmployeeGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
UpdateEmployeeCommand}">
                <iconPacks:PackIconMaterial
Kind="Update"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
    <local:ExcelFilteredDataGrid
x:Name="EmployeeGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding Employees}"

SelectedItem="{Binding SelectedEmployee}"
                Grid.Row="1">
        <filterdg:FilterDataGrid.Columns>
            <filterdg:DataGridTextColumn
Header="Фамилия"

Binding="{Binding Path=Lastname}"

IsColumnFiltered="True"/>
            <filterdg:DataGridTextColumn
Header="Имя"

Binding="{Binding Path=Name}"

IsColumnFiltered="True"/>

```

```

                                <filterdg:DataGridTextColumn
Header="Отчество"

Binding="{Binding Path=Otchestvo}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Должность"

Binding="{Binding Path=PostView.Name}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Парк"

Binding="{Binding Path=ParkView}"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Пользователи БД" Name="DbUser">
                                    <Grid Background="White">
                                        <Grid.RowDefinitions>
                                            <RowDefinition Height="50" />
                                            <RowDefinition Height="1*" />
                                        </Grid.RowDefinitions>
                                        <StackPanel Orientation="Horizontal"
                                            Grid.Row="0">
                                            <Button Height="50"
                                                Width="50"
                                                Command="{Binding
AddDbUserCommand}">
                                                <iconPacks:PackIconMaterial
Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center" />
                                            </Button>
                                            <Button Height="50"
                                                Width="50"
                                                Command="{Binding
EditDbUserCommand}"

IsEnabled="{Binding
ElementName=DbUsersGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

```

```

Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
</Button>
<Button Height="50"
Width="50"
Command="{Binding
DeleteDbUserCommand}"
IsEnabled="{Binding
ElementName=DbUsersGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
<iconPacks:PackIconMaterial
Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
</Button>
<Button Height="50"
Width="50"
Command="{Binding
UpdateDbUserCommand}">
<iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
</Button>
</StackPanel>
<local:ExcelFilteredDataGrid
x:Name="DbUsersGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding DbUsers}"
SelectedItem="{Binding SelectedDbUser}"
Grid.Row="1">
<filterdg:FilterDataGrid.Columns>
<filterdg:DataGridTextColumn
Header="Сотрудник"
Binding="{Binding Path=EmployeeView}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn
Header="Пароль"

```

```

Binding="{Binding Path=Password}"

IsColumnFiltered="True"

Width="*" />
        </filterdg:FilterDataGrid.Columns>
    </local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
<TabItem Header="Папки" Name="Park">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal"
            Grid.Row="0">
            <Button Height="50"
                Width="50"
                Command="{Binding
AddParkCommand}">
                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center" />
            <Button Height="50"
                Width="50"
                Command="{Binding
EditParkCommand}"
                IsEnabled="{Binding
ElementName=ParksGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center" />
            <Button Height="50"
                Width="50"
                Command="{Binding
DeleteParkCommand}"
                IsEnabled="{Binding
ElementName=ParksGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

```

```

Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
</Button>
<Button Height="50"
Width="50"
Command="{Binding
UpdateParkCommand}"
HorizontalAlignment="Right">
<iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
</Button>
</StackPanel>
<local:ExcelFilteredDataGrid
x:Name="ParksGrid"
Grid.Row="1"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Parks}"
SelectedItem="{Binding SelectedPark}">
<filterdg:FilterDataGrid.Columns>
<filterdg:DataGridTextColumn
Header="Название"
Binding="{Binding Path=Name}"
IsColumnFiltered="True"
Width="*" />
</filterdg:FilterDataGrid.Columns>
</local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
</TabControl>
</Grid>
</hc:Window>
Файл « MasterWindow.xaml »
<hc:Window x:Class="trolleybus.View.MasterWindow"
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:hc="https://handyorg.github.io/handycontrol"

xmlns:iconPacks="http://metro.mahapps.com/winfx/xaml/iconpacks"
      xmlns:local="clr-namespace:trolleybus.View"

xmlns:filterdg="http://filterdatagrid.control.com/2021"
      mc:Ignorable="d"
      Title="Троллейбусное депо"
      Height="450" Width="800"
      FontSize="14"
      WindowStartupLocation="CenterScreen"
      WindowState="Maximized"
      Closing="Window_Closing"
    >
    <Window.Resources>
      <Style
        TargetType="local:ExcelFilteredDataGrid"
        x:Key="BaseDataGrid">
        <Setter Property="SelectedItem" Value="{Binding
        Selected, Mode=TwoWay}"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter
          Value="False"/>
          Property="AutoGenerateColumns"
        <Setter Property="HeadersVisibility" Value="Column"/>
        <Setter Property="ColumnWidth" Value="Auto"/>
        <Setter Property="IsReadOnly" Value="True"/>
        <Setter
          Value="True"/>
          Property="hc:DataGridAttach.ShowRowNumber"
        <Setter Property="FilterLanguage" Value="Russian"/>
        <Setter Property="ShowStatusBar" Value="True"/>
        <Setter
          Value="White"/>
          Property="FilterPopupBackground"
        <Setter
          Property="DateFormatString"
          Value="dd.MM.yyyy"/>
        </Style>
      </Window.Resources>
    <Grid >
      <Grid.RowDefinitions>
        <RowDefinition Height="30"/>
        <RowDefinition Height="1*" />
      </Grid.RowDefinitions>
      <TabControl Grid.Row="1">
        <hc:Interaction.Triggers>
          <hc:EventTrigger EventName="SelectionChanged">
            <hc:EventToCommand
              SelectionCommand}" PassEventArgsToCommand="True"/>

```

```

        </hc:EventTrigger>
    </hc:Interaction.Triggers>
    <TabItem Header="Транспортные средства" Name="Tv">
        <Grid Background="White">
            <Grid.RowDefinitions>
                <RowDefinition Height="50"/>
                <RowDefinition Height="1*"
MinHeight="100"/>
                <RowDefinition Height="Auto"/>
                <RowDefinition Height="1*"
MinHeight="100"/>
            </Grid.RowDefinitions>
            <StackPanel Orientation="Horizontal"
                Grid.Row="0">
                <Button Height="50"
                    Width="50"
                    Command="{Binding EditTvCommand}"
                    IsEnabled="{Binding
ElementName=TvGrid, Path=SelectedItem, Converter={StaticResource
Object2BooleanConverter}}">
                    <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                </Button>
                <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateTvCommand}">
                    <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                </Button>
            </StackPanel>
            <local:ExcelFilteredDataGrid x:Name="TvGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding TransportVenichles}"
SelectedItem="{Binding SelectedTv}"
                Grid.Row="1">
                <filterdgc:FilterDataGrid.Columns>
                <filterdgc:DataGridTextColumn
Header="№"

```

```

Binding="{Binding Path=Number}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мапка"

Binding="{Binding Path=Mark}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Гос. номер"

Binding="{Binding Path=GovNumber}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="VIN-номер"

Binding="{Binding Path=Vin}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Год выпуска"

Binding="{Binding Path=ReleaseYear}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Дата ввода в эксплуатацию"

Binding="{Binding Path=CommissioningDate}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Парк"

Binding="{Binding Path=ParkView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Автоход"

Binding="{Binding Path=Autohod}"

```



```

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Статус"
Binding="{Binding Path=StatusView}"
IsColumnFiltered="True"
Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                <GridSplitter Grid.Row="2" Height="5"
HorizontalAlignment="Stretch" />
                                <TabControl Grid.Row="3">
                                    <TabItem Header="Сход">
                                        <Grid Background="White">
                                            <local:ExcelFilteredDataGrid
x:Name="ShodTvGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding ShodsTv}"
SelectedItem="{Binding SelectedShodTv}"
Grid.Row="1">
<filterdg:FilterDataGrid.Columns>
<filterdg:DataGridTextColumn Header="Номер ТС"
Binding="{Binding Path=TvNumber}"
IsReadOnly="True"
IsColumnFiltered="True" />
<filterdg:DataGridTextColumn Header="Время схода"
Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"
IsReadOnly="True"
IsColumnFiltered="True" />
<filterdg:DataGridTextColumn Header="Неисправность"
Width="350"

```

```

CanUserResize="False"

Binding="{Binding Path=Malfunction}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Диспетчер"

Binding="{Binding Path=DispetcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

```

```

Width="*" />

</filterdg:FilterDataGrid.Columns>
    </local:ExcelFilteredDataGrid>
    </Grid>
</TabItem>
<TabItem Header="Неисправности">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition
Width="Auto" />
                <ColumnDefinition
Width="1*" />
            </Grid.ColumnDefinitions>
            <StackPanel
Orientation="Horizontal"
Grid.Row="0">
                <Button Height="50"
Width="50"

Command="{Binding AddMalfunctionTvCommand}"

IsEnabled="{Binding ElementName=TvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
<iconPacks:PackIconMaterial Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center" />
                </Button>
                <Button Height="50"
Width="50"

Command="{Binding EditMalfunctionTvCommand}"

IsEnabled="{Binding ElementName=MalfunctionTvGrid,
Path=SelectedItem,
Converter={StaticResource
Object2BooleanConverter}}">
<iconPacks:PackIconMaterial Kind="Pen"

VerticalAlignment="Center"

HorizontalAlignment="Center" />

```

```

</Button>
<Button Height="50"
        Width="50"

Command="{Binding DeleteMalfunctionTvCommand}"

IsEnabled="{Binding                               ElementName=MalfunctionTvGrid,
Path=SelectedItem,                               Converter={StaticResource
Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>

</Button>
</StackPanel>
</Grid>
<local:ExcelFilteredDataGrid

x:Name="MalfunctionTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding MalfunctionsTv}"

SelectedItem="{Binding SelectedMalfunctionTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Заявка принята"

Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

Binding="{Binding Path=Malfunction1}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробер"

```

```

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время конца работ"

Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Запрет эксплуатации"

Binding="{Binding Path=ZapretExpluatation}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий работника"

Binding="{Binding Path=WorkComment}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Запчасти"

Binding="{Binding Path=TechParts}"

IsColumnFiltered="True"

Width="*" />

</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel

Orientation="Horizontal">

```

```

Text="{Binding Path=StringExecutors}" />
        <TextBlock
        </StackPanel>
    </DataTemplate>

</filterdg:FilterDataGrid.RowDetailsTemplate>
    </local:ExcelFilteredDataGrid>
    </Grid>
</TabItem>
<TabItem Header="T0">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*"/>
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition
                    Width="Auto"/>
                <ColumnDefinition
                    Width="1*"/>
            </Grid.ColumnDefinitions>
            <StackPanel
                Orientation="Horizontal"
                Grid.Row="0">
                <Button Height="50"
                    Width="50"

Command="{Binding AddToTvTvCommand}"

IsEnabled="{Binding ElementName=TvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Plus"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
                </Button>
                <Button Height="50"
                    Width="50"

Command="{Binding EditToTvTvCommand}"

IsEnabled="{Binding ElementName=ToTvTvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Pen"

VerticalAlignment="Center"

```

```

HorizontalAlignment="Center"/>
</Button>
<Button Height="50"
        Width="50"

Command="{Binding DeleteToTvTvCommand}"

IsEnabled="{Binding ElementName=ToTvTvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">

<iconPacks:PackIconMaterial Kind="Delete"

VerticalAlignment="Center"

HorizontalAlignment="Center"/>
</Button>
</StackPanel>
</Grid>
<local:ExcelFilteredDataGrid
x:Name="ToTvTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ToTvsTv}"

SelectedItem="{Binding SelectedToTvTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Дата проведения"

Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>

```

```

<filterdg:DataGridTextColumn Header="Тип ТО"
Binding="{Binding Path=ToTypeView}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Следующее ТО"
Binding="{Binding Path=NextToView}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Пробег для следующего ТО"
Binding="{Binding Path=NextToProbeg}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Мастер"
Binding="{Binding Path=MasterView}"
IsColumnFiltered="True"/>
</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
            <TextBlock
Text="{Binding Path=StringExecutors}" />
        </StackPanel>
    </DataTemplate>
</filterdg:FilterDataGrid.RowDetailsTemplate>
    </local:ExcelFilteredDataGrid>
    </Grid>
    </TabItem>
</TabControl>
</Grid>
</TabItem>
<TabItem Header="Сход" Name="Shod">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <StackPanel Orientation="Horizontal"

```



```

Grid.Row="0">
<Button Height="50"
Width="50"
Command="{Binding
UpdateShodCommand}">
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
</Button>
</StackPanel>
<local:ExcelFilteredDataGrid
x:Name="ShodGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Shods}"
SelectedItem="{Binding SelectedShod}"
Grid.Row="1">
<filterdg:FilterDataGrid.Columns>
<filterdg:DataGridTextColumn
Header="Homep TC"
Binding="{Binding Path=TvNumber}"
IsReadOnly="True"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn
Header="Время схода"
Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"
IsReadOnly="True"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn
Header="Неисправность"
Width="350"
CanUserResize="False"
Binding="{Binding Path=Malfunction}"
IsColumnFiltered="True"/>

```

```

                                <filterdg:DataGridCheckBoxColumn
Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Диспетчер"

Binding="{Binding Path=DispatcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

Width="*/>
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>

```

```

        <TabItem Header="Неисправности" Name="Malfunction">
            <Grid Background="White">
                <Grid.RowDefinitions>
                    <RowDefinition Height="50"/>
                    <RowDefinition Height="1*"/>
                </Grid.RowDefinitions>
                <Grid Grid.Row="0">
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="Auto"/>
                        <ColumnDefinition Width="1*"/>
                    </Grid.ColumnDefinitions>
                    <StackPanel Orientation="Horizontal"
                        Grid.Row="0">
                        <Button Height="50"
                            Width="50"
                            Command="{Binding
AddMalfunctionCommand}">
                            <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                        </Button>
                        <Button Height="50"
                            Width="50"
                            Command="{Binding
EditMalfunctionCommand}"
                            IsEnabled="{Binding
ElementName=MalfunctionGrid,
Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                            <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                        </Button>
                        <Button Height="50"
                            Width="50"
                            Command="{Binding
UpdateMalfunctionCommand}">
                            <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                        </Button>
                    </StackPanel>
                </Grid>
            </Grid>
        </TabItem>
    </TabControl>

```

```

        <local:ExcelFilteredDataGrid
x:Name="MalfunctionGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding Malfunctions}"

SelectedItem="{Binding SelectedMalfunction}"

                                Grid.Row="1">
        <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn
Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Заявка принята"

Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Неисправность"

Binding="{Binding Path=Malfunction1}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Время конца работ"

Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
        <filterdg:DataGridCheckBoxColumn
Header="Запрет эксплуатации"

Binding="{Binding Path=ZapretExpluatation}"

```

```

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мастер"
Binding="{Binding Path=MasterView}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий работника"
Binding="{Binding Path=WorkComment}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Запчасти"
Binding="{Binding Path=TechParts}"
IsColumnFiltered="True"
Width="*" />
                                </filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
                                <DataTemplate>
                                    <StackPanel
Orientation="Horizontal">
                                        <TextBlock      Text="{Binding
Path=StringExecutors}" />
                                        </StackPanel>
                                    </DataTemplate>
                                </filterdg:FilterDataGrid.RowDetailsTemplate>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Техническое обслуживание">
                                    <Grid Background="White">
                                        <Grid.RowDefinitions>
                                            <RowDefinition Height="50"/>
                                            <RowDefinition Height="1*" />
                                        </Grid.RowDefinitions>
                                        <Grid Grid.Row="0">
                                            <Grid.ColumnDefinitions>
                                                <ColumnDefinition Width="Auto"/>
                                                <ColumnDefinition Width="1*" />
                                            </Grid.ColumnDefinitions>
                                            <StackPanel Orientation="Horizontal"
                                                Grid.Row="0">

```

```

        <Button Height="50"
                Width="50"
                Command="{Binding
AddToTvCommand}">
                <iconPacks:PackIconMaterial
Kind="Plus"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
EditToTvCommand}"
                IsEnabled="{Binding
ElementName=ToTvGrid,
Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Pen"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
DeleteToTvCommand}"
                IsEnabled="{Binding
ElementName=ToTvGrid,
Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
                <iconPacks:PackIconMaterial
Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
        <Button Height="50"
                Width="50"
                Command="{Binding
UpdateToTvCommand}">
                <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>

```

```

</Grid>
<local:ExcelFilteredDataGrid
x:Name="ToTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ToTvs}"

SelectedItem="{Binding SelectedToTv}"
Grid.Row="1">
    <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn
Header="Homep TC"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Дата проведения"

Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Тип ТО"

Binding="{Binding Path=ToTypeView}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Следующее ТО"

Binding="{Binding Path=NextToView}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Пробег для следующего ТО"

Binding="{Binding Path=NextToProbeg}"

IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn
Header="Мастер"

```

```

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>
</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
                                <TextBlock      Text="{Binding
Path=StringExecutors}" />
                                </StackPanel>
        </DataTemplate>
    </filterdg:FilterDataGrid.RowDetailsTemplate>
</local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
</TabControl>
</Grid>
</hc:Window>

```

Файл « RukovoditelWindow.xaml »

```

<hc:Window x:Class="trolleybus.View.RukovoditelWindow"

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:hc="https://handyorg.github.io/handycontrol"

xmlns:iconPacks="http://metro.mahapps.com/winfx/xaml/iconpacks"
    xmlns:local="clr-namespace:trolleybus.View"

xmlns:filterdg="http://filterdatagrid.control.com/2021"
    mc:Ignorable="d"
    Title="Троллейбусное депо"
    Height="450" Width="800"
    FontSize="14"
    WindowStartupLocation="CenterScreen"
    WindowState="Maximized"
    Closing="Window_Closing"
    >
    <Window.Resources>
        <Style                                x:Key="BaseDataGrid"
TargetType="local:ExcelFilteredDataGrid">

```

					ПГУ 1.090301.07.001 ПЗ	Лист
						160
Изм.	Лист	№ докум.	Подпись	Дата		


```

        <Setter      Property="SelectedItem"      Value="{Binding
Selected, Mode=TwoWay}"/>
        <Setter Property="FontSize" Value="14"/>
        <Setter      Property="AutoGenerateColumns"
Value="False"/>
        <Setter Property="HeadersVisibility" Value="Column"/>
        <Setter Property="ColumnWidth" Value="Auto"/>
        <Setter Property="IsReadOnly" Value="True"/>
        <Setter      Property="hc:DataGridAttach.ShowRowNumber"
Value="True"/>
        <Setter Property="FilterLanguage" Value="Russian"/>
        <Setter Property="ShowStatusBar" Value="True"/>
        <Setter      Property="FilterPopUpBackground"
Value="White"/>
        <Setter      Property="DateFormatString"
Value="dd.MM.yyyy"/>
    </Style>
</Window.Resources>
<Grid >
    <Grid.RowDefinitions>
        <RowDefinition Height="30"/>
        <RowDefinition Height="1*" />
    </Grid.RowDefinitions>
    <TabControl Grid.Row="1">
        <hc:Interaction.Triggers>
            <hc:EventTrigger EventName="SelectionChanged">
                <hc:EventToCommand      Command="{Binding
SelectionCommand}" PassEventArgsToCommand="True"/>
            </hc:EventTrigger>
        </hc:Interaction.Triggers>
        <TabItem Header="Транспортные средства" Name="Tv">
            <Grid Background="White">
                <Grid.RowDefinitions>
                    <RowDefinition Height="50"/>
                    <RowDefinition      Height="1*"
MinHeight="100"/>
                    <RowDefinition Height="Auto"/>
                    <RowDefinition      Height="1*"
MinHeight="100"/>
                </Grid.RowDefinitions>
                <StackPanel Orientation="Horizontal"
Grid.Row="0">
                    <Button Height="50"
Width="50"
Command="{Binding
UpdateTvCommand}">
                        <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"

```

```

HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
    <local:ExcelFilteredDataGrid x:Name="TvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding TransportVenichles}"

SelectedItem="{Binding SelectedTv}"

                                Grid.Row="1">
        <filterdg:FilterDataGrid.Columns>
            <filterdg:DataGridTextColumn
Header="№"

Binding="{Binding Path=Number}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мапка"

Binding="{Binding Path=Mark}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Гос. номер"

Binding="{Binding Path=GovNumber}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="VIN-номер"

Binding="{Binding Path=Vin}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Год выпуска"

Binding="{Binding Path=ReleaseYear}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Дата ввода в эксплуатацию"

Binding="{Binding Path=CommissioningDate}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"

```

```

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Парк"

Binding="{Binding Path=ParkView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Автомобиль"

Binding="{Binding Path=Autohod}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Статус"

Binding="{Binding Path=StatusView}"

IsColumnFiltered="True"

Width="*" />
                                </filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                <GridSplitter Grid.Row="2" Height="5"
HorizontalAlignment="Stretch"/>
                                <TabControl Grid.Row="3">
                                    <TabItem Header="Сход">
                                        <Grid Background="White">
                                            <local:ExcelFilteredDataGrid
x:Name="ShodTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding ShodsTv}"

SelectedItem="{Binding SelectedShodTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Homep TC"

Binding="{Binding Path=TvNumber}"

IsReadOnly="True"

IsColumnFiltered="True"/>

```

```

<filterdg:DataGridTextColumn Header="Время схода"

Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsReadOnly="True"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Malfunction}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий к отмене"

Width="350"

```

					ПГУ 1.090301.07.001 ПЗ	Лист
						164
Изм.	Лист	№ докум.	Подпись	Дата		

```

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Диспетчер"

Binding="{Binding Path=DispatcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

Width="*" />

</filterdg:FilterDataGrid.Columns>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Неисправности">
                                    <Grid Background="White">
                                        <local:ExcelFilteredDataGrid
x:Name="MalfunctionTvGrid"

Style="{StaticResource BaseDataGrid}"

ItemsSource="{Binding MalfunctionsTv}"

SelectedItem="{Binding SelectedMalfunctionTv}"

Grid.Row="1">

<filterdg:FilterDataGrid.Columns>

<filterdg:DataGridTextColumn Header="Номер ТС"

Binding="{Binding Path=TvNumber}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Заявка принята"

Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность"

```

```

Binding="{Binding Path=Malfunction1}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Время конца работ"

Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Запрет эксплуатации"

Binding="{Binding Path=ZapretExpluatation}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий работника"

Binding="{Binding Path=WorkComment}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Запчасти"

Binding="{Binding Path=TechParts}"

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Номер акта"

Binding="{Binding Path=Reklamation}"

```

```

IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Неисправность для акта"
Binding="{Binding Path=MalfunctionRecl}"
IsColumnFiltered="True"/>

<filterdg:DataGridTextColumn Header="Комментарий для акта"
Binding="{Binding Path=MalfunctionReclComment}"
IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Подписан"
Binding="{Binding Path=Signed}"
IsColumnFiltered="True"/>

<filterdg:DataGridCheckBoxColumn Header="Устранено"
Binding="{Binding Path=Done}"
IsColumnFiltered="True"
Width="*" />

</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
            <TextBlock
Text="{Binding Path=StringExecutors}" />
        </StackPanel>
    </DataTemplate>
</filterdg:FilterDataGrid.RowDetailsTemplate>
</local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
<TabItem Header="ТО">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50"/>
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>

```

```

<Grid Grid.Row="0">
    <Grid.ColumnDefinitions>
        <ColumnDefinition
Width="Auto"/>
        <ColumnDefinition
Width="1*"/>
    </Grid.ColumnDefinitions>
    <StackPanel
Orientation="Horizontal"
Grid.Row="0">
        <Button Height="50"
Width="50"
Command="{Binding DeleteToTvTvCommand}"
IsEnabled="{Binding ElementName=ToTvTvGrid, Path=SelectedItem,
Converter={StaticResource Object2BooleanConverter}}">
            <iconPacks:PackIconMaterial Kind="Delete"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
        </Button>
    </StackPanel>
</Grid>
<local:ExcelFilteredDataGrid
x:Name="ToTvTvGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding ToTvsTv}"
SelectedItem="{Binding SelectedToTvTv}"
Grid.Row="1">
    <filterdg:FilterDataGrid.Columns>
        <filterdg:DataGridTextColumn Header="Homep TC"
Binding="{Binding Path=TvNumber}"
IsColumnFiltered="True"/>
        <filterdg:DataGridTextColumn Header="Дата проведения"
Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"
IsColumnFiltered="True"/>

```



```

<filterdg:DataGridTextColumn Header="Пробег"
Binding="{Binding Path=Probeg}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Тип ТО"
Binding="{Binding Path=ToTypeView}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Следующее ТО"
Binding="{Binding Path=NextToView}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Пробег для следующего ТО"
Binding="{Binding Path=NextToProbeg}"
IsColumnFiltered="True"/>
<filterdg:DataGridTextColumn Header="Мастер"
Binding="{Binding Path=MasterView}"
IsColumnFiltered="True"/>
</filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
    <DataTemplate>
        <StackPanel
Orientation="Horizontal">
            <TextBlock
Text="{Binding Path=StringExecutors}" />
        </StackPanel>
    </DataTemplate>
</filterdg:FilterDataGrid.RowDetailsTemplate>
</local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
</TabControl>
</Grid>
</TabItem>
<TabItem Header="Сход" Name="Shod">

```

```

        <Grid Background="White">
            <Grid.RowDefinitions>
                <RowDefinition Height="50"/>
                <RowDefinition Height="1*"/>
            </Grid.RowDefinitions>
            <StackPanel Orientation="Horizontal"
                Grid.Row="0">
                <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateShodCommand}">
                    <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                </Button>
            </StackPanel>
            <local:ExcelFilteredDataGrid
x:Name="ShodGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Shods}"
SelectedItem="{Binding SelectedShod}"
                Grid.Row="1">
                <filterdg:FilterDataGrid.Columns>
                    <filterdg:DataGridTextColumn
Header="Homep TC"
Binding="{Binding Path=TvNumber}"
IsReadOnly="True"
IsColumnFiltered="True"/>
                    <filterdg:DataGridTextColumn
Header="Время схода"
Binding="{Binding Path=DepartureTime, StringFormat=dd.MM.yyyy
HH:mm}"
IsReadOnly="True"
IsColumnFiltered="True"/>
                    <filterdg:DataGridTextColumn
Header="Неисправность"
Width="350"

```

```

CanUserResize="False"

Binding="{Binding Path=Malfunction}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Эвакуатор"

Binding="{Binding Path=Evacuator}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время прибытия"

Width="350"

CanUserResize="False"

Binding="{Binding Path=Arrival, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время отмены"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий к отмене"

Width="350"

CanUserResize="False"

Binding="{Binding Path=CancelComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Диспетчер"

Binding="{Binding Path=DispetcherView}"

IsReadOnly="True"

IsColumnFiltered="True"

```

```

Width="*" />
        </filterdg:FilterDataGrid.Columns>
    </local:ExcelFilteredDataGrid>
</Grid>
</TabItem>
<TabItem Header="Неисправности" Name="Malfunction">
    <Grid Background="White">
        <Grid.RowDefinitions>
            <RowDefinition Height="50" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <Grid Grid.Row="0">
            <Grid.ColumnDefinitions>
                <ColumnDefinition Width="Auto" />
                <ColumnDefinition Width="1*" />
            </Grid.ColumnDefinitions>
            <StackPanel Orientation="Horizontal"
                Grid.Row="0">
                <Button Height="50"
                    Width="50"
                    Command="{Binding
UpdateMalfunctionCommand}">
                    <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center" />
                </Button>
            </StackPanel>
        </Grid>
        <local:ExcelFilteredDataGrid
x:Name="MalfunctionGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding Malfunctions}"
SelectedItem="{Binding SelectedMalfunction}"
                Grid.Row="1">
                <filterdg:FilterDataGrid.Columns>
                    <filterdg:DataGridTextColumn
Header="Номер ТС"
Binding="{Binding Path=TvNumber}"
IsColumnFiltered="True" />
                    <filterdg:DataGridTextColumn
Header="Заявка принята"

```

```

Binding="{Binding Path=AcceptTime, StringFormat=dd.MM.yyyy
HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Неисправность"

Binding="{Binding Path=Malfunction1}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег"

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время начала работ"

Binding="{Binding Path=BeginTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Время конца работ"

Binding="{Binding Path=ReadyTime, StringFormat=dd.MM.yyyy HH:mm}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Запрет эксплуатации"

Binding="{Binding Path=ZapretExpluatation}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий работника"

Binding="{Binding Path=WorkComment}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Запчасти"

Binding="{Binding Path=TechParts}"

```

```

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Номер акта"
Binding="{Binding Path=Reklamation}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Неисправность для акта"
Binding="{Binding Path=MalfunctionRecl}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Комментарий для акта"
Binding="{Binding Path=MalfunctionReclComment}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Подписан"
Binding="{Binding Path=Signed}"
IsColumnFiltered="True"/>
                                <filterdg:DataGridCheckBoxColumn
Header="Устранено"
Binding="{Binding Path=Done}"
IsColumnFiltered="True"
Width="*" />
                                </filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
                                <DataTemplate>
                                    <StackPanel
Orientation="Horizontal">
                                        <TextBlock          Text="{Binding
Path=StringExecutors}" />
                                    </StackPanel>
                                </DataTemplate>

</filterdg:FilterDataGrid.RowDetailsTemplate>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                <TabItem Header="Техническое обслуживание">

```

```

        <Grid Background="White">
            <Grid.RowDefinitions>
                <RowDefinition Height="50"/>
                <RowDefinition Height="1*"/>
            </Grid.RowDefinitions>
            <Grid Grid.Row="0">
                <Grid.ColumnDefinitions>
                    <ColumnDefinition Width="Auto"/>
                    <ColumnDefinition Width="1*"/>
                </Grid.ColumnDefinitions>
                <StackPanel Orientation="Horizontal"
                    Grid.Row="0">
                    <Button Height="50"
                        Width="50"
                        Command="{Binding
UpdateToTvCommand}">
                        <iconPacks:PackIconMaterial
Kind="Update"
VerticalAlignment="Center"
HorizontalAlignment="Center"/>
                    </Button>
                </StackPanel>
            </Grid>
            <local:ExcelFilteredDataGrid
x:Name="ToTvGrid"
Style="{StaticResource BaseDataGrid}"
ItemsSource="{Binding ToTvs}"
SelectedItem="{Binding SelectedToTv}"
                    Grid.Row="1">
                <filterdg:FilterDataGrid.Columns>
                    <filterdg:DataGridTextColumn
Header="Homep TC"
Binding="{Binding Path=TvNumber}"
IsColumnFiltered="True"/>
                    <filterdg:DataGridTextColumn
Header="Дата проведения"
Binding="{Binding Path=CompleteDate, StringFormat=dd.MM.yyyy
HH:mm}"
IsColumnFiltered="True"/>
                    <filterdg:DataGridTextColumn
Header="Пробер"

```

```

Binding="{Binding Path=Probeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Тип ТО"

Binding="{Binding Path=ToTypeView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Следующее ТО"

Binding="{Binding Path=NextToView}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Пробег для следующего ТО"

Binding="{Binding Path=NextToProbeg}"

IsColumnFiltered="True"/>
                                <filterdg:DataGridTextColumn
Header="Мастер"

Binding="{Binding Path=MasterView}"

IsColumnFiltered="True"/>
                                </filterdg:FilterDataGrid.Columns>

<filterdg:FilterDataGrid.RowDetailsTemplate>
                                <DataTemplate>
                                    <StackPanel
Orientation="Horizontal">
                                                <TextBlock      Text="{Binding
Path=StringExecutors}" />
                                    </StackPanel>
                                </DataTemplate>

</filterdg:FilterDataGrid.RowDetailsTemplate>
                                </local:ExcelFilteredDataGrid>
                                </Grid>
                                </TabItem>
                                </TabControl>
                                </Grid>
</hc:Window>
using Microsoft.EntityFrameworkCore;
using System;
using System.Collections.Generic;
using System.Linq;

```



```

using System.Text;
using System.Threading.Tasks;
using trolleybus.Model;
        Файл «EditDbUserViewModel.cs»
namespace trolleybus.ViewModel
{
    public class EditDbUserViewModel :
    EditEntityViewModel<DbUser>
    {
        public bool NewDbUser { get; set; }

        private List<Employee> employees;
        public List<Employee> Employees
        {
            get
            {
                return employees;
            }

            set
            {
                employees = value;
                OnPropertyChanged();
            }
        }

        public EditDbUserViewModel(DbUser dbUser)
        {
            using var db = new TrolleybusContext();
            if (dbUser == null)
            {
                var employeesNotDbUsers = db.Employees.Include(p
=> p.DbUsers).Where(p => p.DbUsers.Count == 0).Select(p =>
p.Id).ToList();
                Employees = db.Employees.Where(p => p.Post != 2 &&
employeesNotDbUsers.Contains(p.Id)).ToList();
                if (Employees.Count == 0) throw new Exception("Нет
сотрудников, у которых нет профиля для подключения к БД.");
                Entity = new DbUser
                {
                    Employee = Employees.First().Id,
                    Password = ""
                };
                NewDbUser = true;
                newEntity = true;
            }
            else
            {
                Employees = db.Employees.Where(p => p.Post !=
2).ToList();
                Entity = dbUser;
            }
        }
    }
}

```

```

        NewDbUser = false;
    }
}

}

Файл « EditEmployeeViewModel.CS»
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using trolleybus.Model;

namespace trolleybus.ViewModel
{
    public class EditEmployeeViewModel :
    EditEntityViewModel<Employee>
    {
        public List<Post> Posts
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.Posts.ToList();
            }
        }

        public List<Park> Parks
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.Parks.ToList();
            }
        }

        public EditEmployeeViewModel(Employee employee)
        {
            if (employee == null)
            {
                Entity = new Employee
                {
                    Name = "",
                    Lastname = "",
                    Otchestvo = "",
                    Post = Posts.First().Id,
                    Park = Parks.First().Id
                };

                newEntity = true;
            }
        }
    }
}

```

```

        else
        {
            Entity = employee;
        }
    }
}

```

Файл « EditEntityViewModel.CS»

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;
using trolleybus.Model;
using trolleybus.Other;

namespace trolleybus.ViewModel
{
    public abstract class EditEntityViewModel<T> :
    INotifyPropertyChanged where T : class
    {
        private T entity;
        protected bool newEntity = false;

        public T Entity
        {
            get
            {
                return entity;
            }

            set
            {
                entity = value;
                OnPropertyChanged();
            }
        }

        ICommand saveCommand;
        public ICommand SaveCommand => saveCommand ?? (saveCommand
= new RelayCommand(OnSave));

        public virtual void OnSave(object parameter)
        {
            using var db = new TrolleybusContext();
            if(newEntity)
            {
                db.Add(Entity);
            }
        }
    }
}

```

```

        }
        else
        {
            db.Update(Entity);
        }
        db.SaveChanges();
    }

    public          event          PropertyChangedEventHandler?
PropertyChanged;

    public void OnPropertyChanged([CallerMemberName] string
prop = "")
    {
        if (PropertyChanged != null)
            PropertyChanged(this,          new
PropertyChangedEventArgs(prop));
    }
}

```

Файл « EditMalfunctionViewModel.CS»

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Input;
using trolleybus.Model;
using trolleybus.Other;

namespace trolleybus.ViewModel
{
    public          class          EditMalfunctionViewModel          :
EditEntityViewModel<Malfunction>
    {
        public Visibility Vision
        {
            get
            {
                return CurrentUser.IsVisible;
            }
        }

        ICommand beginCommand;
        public ICommand BeginCommand => beginCommand ??
(beginCommand = new RelayCommand(OnBegin));

        ICommand endCommand;
        public ICommand EndCommand => endCommand ?? (endCommand =
new RelayCommand(OnEnd));
    }
}

```

```

public bool NewMalfunction { get; set; }

private bool isToBegin;
public bool IsToBegin
{
    get
    {
        return isToBegin;
    }

    set
    {
        isToBegin = value;
        OnPropertyChanged();
    }
}

private bool isToEnd;
public bool IsToEnd
{
    get
    {
        return isToEnd;
    }

    set
    {
        isToEnd = value;
        OnPropertyChanged();
    }
}

public List<TransportVenichle> TransportVenichles
{
    get
    {
        using var db = new TrolleybusContext();
        return db.TransportVenichles.OrderBy(p =>
p.Number).ToList();
    }
}

public List<string> Executors
{
    get
    {
        using var db = new TrolleybusContext();
        var result = db.Employees.Where(p => p.Post ==
2).ToList();

        List<string> list = new List<string>();

```

```

        foreach ( var p in result )
        {
            list.Add(p.ToString());
        }
        return list;
    }
}

private List<string> selectedExecutors;

public List<string> SelectedExecutors
{
    get
    {
        return selectedExecutors;
    }

    set
    {
        selectedExecutors = value;
        OnPropertyChanged();
    }
}

public EditMalfunctionViewModel(Malfunction malfunction,
TransportVenichle tv)
{
    if (malfunction == null)
    {
        NewMalfunction = true;
        Entity = new Malfunction();
        Entity.Master = CurrentUser.Employee.Id;
        Entity.Probeg = 0;
        Entity.ZapretExpluatation = false;
        Entity.Done = false;
        Entity.Signed = false;
        newEntity = true;
    }
    else
    {
        NewMalfunction = false;
        Entity = malfunction;
    }

    if (tv != null)
    {
        Entity.TvNumber = tv.Number;
    }
    else
    {

```

```

        Entity.TvNumber
TransportVenichles.First().Number;
    }

    if(!Entity.BeginTime.HasValue)
    {
        IsToBegin = true;
        IsToEnd = false;
    }
    else if (!Entity.ReadyTime.HasValue)
    {
        IsToBegin = false;
        IsToEnd = true;
    }
    else
    {
        IsToBegin = false;
        IsToEnd = false;
    }

    using var db = new TrolleybusContext();
    var mes = Entity.MalfunctionExecutors.Select(p =>
p.IdEmployee).ToList();
    var exs = db.Employees.Where(e => e.Post == 2 &&
mes.Contains(e.Id)).ToList();
    SelectedExecutors = exs.Select(e =>
e.ToString()).ToList();
    }

    public override void OnSave(object parameter)
    {
        if (!Entity.AcceptTime.HasValue)
        {
            Entity.AcceptTime = DateTime.Now;
        }

        using var db = new TrolleybusContext();

        if (newEntity)
        {
            db.Add(Entity);
        }
        else
        {
            db.Update(Entity);
        }
        db.SaveChanges();

        var exs = db.Employees.Where(p => p.Post ==
2).ToList();
        List<int> idExs = new List<int>();

```

```

        foreach (var e in exs)
        {
            if (SelectedExecutors.Contains(e.ToString()))
            {
                idExs.Add(e.Id);
            }
        }

        foreach (var idEx in idExs)
        {
            if (db.MalfunctionExecutors.Where(p =>
p.IdEmployee == idEx && p.IdMalfunction == Entity.Id).Count() ==
0)
            {
                db.MalfunctionExecutors.Add(new
MalfunctionExecutor { IdMalfunction = Entity.Id, IdEmployee = idEx
});
            }
        }

        foreach (var me in db.MalfunctionExecutors.Where(p =>
p.IdMalfunction == Entity.Id))
        {
            if (!idExs.Contains(me.IdEmployee))
            {
                db.MalfunctionExecutors.Remove(me);
            }
        }

        db.SaveChanges();
    }

    public void OnBegin(object parameter)
    {
        Entity.BeginTime = DateTime.Now;
        IsToBegin = false;
    }

    public void OnEnd(object parameter)
    {
        Entity.ReadyTime = DateTime.Now;
        IsToEnd = false;
    }
}
}

```

Файл « EditParkViewModel.cs »
 using DocumentFormat.OpenXml.Vml.Office;
 using System;


```

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

namespace trolleybus.ViewModel
{
    public class EditParkViewModel : EditEntityViewModel<Park>
    {
        public EditParkViewModel(Park park)
        {
            if (park == null)
            {
                Entity = new Park();
                newEntity = true;
            }
            else
            {
                Entity = park;
            }
        }
    }
}

```

Файл « EditShodViewModel.CS»

```

using DocumentFormat.OpenXml.Vml.Office;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Input;
using trolleybus.Model;
using trolleybus.Other;

namespace trolleybus.ViewModel
{
    public class EditShodViewModel : EditEntityViewModel<Shod>
    {
        public bool NewShod { get; set; }
        public bool Cancel { get; set; }
        public bool Arrived { get; set; }

        public List<TransportVenichle> TransportVenichles
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.TransportVenichles.OrderBy(p =>
p.Number).ToList();
            }
        }
    }
}

```

```

    }

    public EditShodViewModel(Shod shod, TransportVenichle tv)
    {
        if (shod == null)
        {
            NewShod = true;
            Entity = new Shod();
            Entity.Evacuator = false;
            Entity.Dispatcher = CurrentUser.Employee.Id;
            newEntity = true;
        }
        else
        {
            NewShod = false;
            Entity = shod;
        }

        if (tv != null)
        {
            Entity.TvNumber = tv.Number;
        }
        else
        {
            Entity.TvNumber =
TransportVenichles.First().Number;
        }
    }

    public override void OnSave(object parameter)
    {
        if (NewShod)
        {
            Entity.CancelComment = "";
            Entity.DepartureTime = DateTime.Now;
        }
        else if (Cancel)
        {
            Entity.CancelTime = DateTime.Now;
        }
        else if (Arrived)
        {
            Entity.CancelComment = "";
            Entity.Arrival = DateTime.Now;
        }

        base.OnSave(parameter);
    }
}

```

Файл « EditToTvViewModel.CS»

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using trolleybus.Model;
using trolleybus.Other;

namespace trolleybus.ViewModel
{
    public class EditToTvViewModel : EditEntityViewModel<ToTv>
    {
        public Visibility Vision
        {
            get
            {
                return CurrentUser.IsVisible;
            }
        }

        public List<TransportVenichle> TransportVenichles
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.TransportVenichles.OrderBy(p =>
p.Number).ToList();
            }
        }

        public List<ToType> ToTypes
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.ToTypes.OrderBy(p => p.Id).ToList();
            }
        }

        public List<string> Executors
        {
            get
            {
                using var db = new TrolleybusContext();
                var result = db.Employees.Where(p => p.Post ==
2).ToList();

                List<string> list = new List<string>();
                foreach (var p in result)
                {
                    list.Add(p.ToString());
                }
            }
        }
    }
}

```

```

        }
        return list;
    }
}

private List<string> selectedExecutors;

public List<string> SelectedExecutors
{
    get
    {
        return selectedExecutors;
    }

    set
    {
        selectedExecutors = value;
        OnPropertyChanged();
    }
}

public EditToTvViewModel(ToTv toTv, TransportVenichle tv)
{
    if (toTv == null)
    {
        Entity = new ToTv();
        Entity.Master = CurrentUser.Employee.Id;
        Entity.Probeg = 0;
        newEntity = true;
    }
    else
    {
        Entity = toTv;
    }

    if (tv != null)
    {
        Entity.TvNumber = tv.Number;
    }
    else
    {
        Entity.TvNumber =
TransportVenichles.First().Number;
    }

    using var db = new TrolleybusContext();
    var mes = Entity.ToTvExecutors.Select(p =>
p.Employee).ToList();
    var exs = db.Employees.Where(e => e.Post == 2 &&
mes.Contains(e.Id)).ToList();

```

```

        SelectedExecutors = exs.Select(e
e.ToString()).ToList();
    }

    public override void OnSave(object parameter)
    {
        if (!Entity.CompleteDate.HasValue)
        {
            Entity.CompleteDate = DateTime.Now;
        }

        using var db = new TrolleybusContext();

        if (newEntity)
        {
            db.Add(Entity);
        }
        else
        {
            db.Update(Entity);
        }
        db.SaveChanges();

        var exs = db.Employees.Where(p => p.Post ==
2).ToList();
        List<int> idExs = new List<int>();
        foreach (var e in exs)
        {
            if (SelectedExecutors.Contains(e.ToString()))
            {
                idExs.Add(e.Id);
            }
        }

        foreach (var idEx in idExs)
        {
            if (db.ToTvExecutors.Where(p => p.Employee == idEx
&& p.ToTv == Entity.Id).Count() == 0)
            {
                db.ToTvExecutors.Add(new ToTvExecutor { ToTv
= Entity.Id, Employee = idEx });
            }
        }

        foreach (var me in db.ToTvExecutors.Where(p => p.ToTv
== Entity.Id))
        {
            if (!idExs.Contains(me.Employee))
            {
                db.ToTvExecutors.Remove(me);
            }
        }
    }
}

```

```

    }
}

db.SaveChanges();
}
}

Файл « EditTvViewModel.CS»
using DocumentFormat.OpenXml.Vml.Office;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using trolleybus.Model;
using trolleybus.Other;

namespace trolleybus.ViewModel
{
    public class EditTvViewModel :
    EditEntityViewModel<TransportVenichle>
    {
        public Visibility Vision
        {
            get
            {
                return CurrentUser.IsVisible;
            }
        }

        public List<TvStatus> Statuses
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.TvStatuses.ToList();
            }
        }

        public List<Park> Parks
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.Parks.ToList();
            }
        }

        public EditTvViewModel(TransportVenichle tv)
        {

```

```

        if (tv == null)
        {
            Entity = new TransportVenichle();
            Entity.Autohod = false;
            using var db = new TrolleybusContext();
            if (db.TransportVenichles.Count() != 0)
            {
                Entity.Number = db.TransportVenichles.Max(p
=> p.Number + 1);
            }
            else
            {
                Entity.Number = 1000;
            }
            Entity.Probeg = 0;
            newEntity = true;
        }
        else
        {
            Entity = tv;
        }
    }
}

```

Файл « MainWindowViewModel.CS»

```

using DocumentFormat.OpenXml.Drawing.Charts;
using HandyControl.Data;
using Microsoft.EntityFrameworkCore;
using Microsoft.EntityFrameworkCore.Metadata;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.IO;
using System.Linq;
using System.Runtime.CompilerServices;
using System.Text;
using System.Threading;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using TemplateEngine.Docx;
using trolleybus.Model;
using trolleybus.Other;
using trolleybus.View;

namespace trolleybus.ViewModel
{
    public class MainWindowViewModel : INotifyPropertyChanged
    {
        public List<TransportVenichle> TransportVenichles

```

```

        {
            get
            {
                using var db = new TrolleybusContext();
                return db.TransportVenichles.Include(p
=>
p.Shods).Include(p
=>
p.Malfunctions).Include(p
=>
p.ToTvs).OrderBy(p => p.Number).ToList();
            }
        }

private TransportVenichle selectedTv;
public TransportVenichle SelectedTv
{
    get
    {
        return selectedTv;
    }

    set
    {
        selectedTv = value;
        OnPropertyChanged();
        OnPropertyChanged("ShodsTv");
        OnPropertyChanged("MalfunctionsTv");
    }
}

public ICommand DeleteTvCommand => new
RelayCommand(OnDeleteTv);
public ICommand AddTvCommand => new RelayCommand(OnAddTv);
public ICommand EditTvCommand => new
RelayCommand(OnEditTv);
public ICommand UpdateTvCommand => new
RelayCommand(OnUpdateTv);

private void OnAddTv(object param)
{
    var dialog = new EditTv();
    dialog.ShowDialog();
    OnPropertyChanged("TransportVenichles");
}

private void OnEditTv(object param)
{
    var dialog = new EditTv(SelectedTv);
    dialog.ShowDialog();
    OnPropertyChanged("TransportVenichles");
}

private void OnDeleteTv(object param)
{

```



```

        using var db = new TrolleybusContext();
        db.Remove(SelectedTv);
        db.SaveChanges();
        OnPropertyChanged("TransportVenichles");
    }

    private void OnUpdateTv(object param)
    {
        OnPropertyChanged("TransportVenichles");
    }

    public List<Shod> ShodsTv
    {
        get
        {
            if (SelectedTv != null)
            {
                using var db = new TrolleybusContext();
                return db.Shods.OrderByDescending(s
s.DepartureTime).Where(p
SelectedTv.Number).ToList();
            }
            else
            {
                return new List<Shod>();
            }
        }
    }

    private Shod selectedShodTv;
    public Shod SelectedShodTv
    {
        get
        {
            return selectedShodTv;
        }
        set
        {
            selectedShodTv = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteShodTvCommand => new
RelayCommand(OnDeleteShodTv);
    public ICommand AddShodTvCommand => new
RelayCommand(OnAddShodTv);
    public ICommand EditShodTvCommand => new
RelayCommand(OnEditShodTv);

```

```

private void OnAddShodTv(object param)
{
    var dialog = new EditShod(SelectedTv);
    dialog.ShowDialog();
    OnPropertyChanged("ShodsTv");
}

private void OnEditShodTv(object param)
{
    var dialog = new EditShod(SelectedTv, SelectedShodTv);
    dialog.ShowDialog();
    OnPropertyChanged("ShodsTv");
}

private void OnDeleteShodTv(object param)
{
    using var db = new TrolleybusContext();
    db.Remove(SelectedShodTv);
    db.SaveChanges();
    OnPropertyChanged("ShodsTv");
}

public List<Malfunction> MalfunctionsTv
{
    get
    {
        if (SelectedTv != null)
        {
            using var db = new TrolleybusContext();
            return db.Malfunctions.OrderByDescending(s =>
s.AcceptTime).Include(p => p.MalfunctionExecutors).Where(p =>
p.TvNumber == SelectedTv.Number).ToList();
        }
        else
        {
            return new List<Malfunction>();
        }
    }
}

private Malfunction selectedMalfunctionTv;
public Malfunction SelectedMalfunctionTv
{
    get
    {
        return selectedMalfunctionTv;
    }

    set
    {
        selectedMalfunctionTv = value;
    }
}

```

```

        OnPropertyChanged();
    }
}

public ICommand DeleteMalfunctionTvCommand => new
 RelayCommand(OnDeleteMalfunctionTv);
public ICommand AddMalfunctionTvCommand => new
 RelayCommand(OnAddMalfunctionTv);
public ICommand EditMalfunctionTvCommand => new
 RelayCommand(OnEditMalfunctionTv);

private void OnAddMalfunctionTv(object param)
{
    var dialog = new EditMalfunction(SelectedTv);
    dialog.ShowDialog();
    OnPropertyChanged("MalfunctionsTv");
}

private void OnEditMalfunctionTv(object param)
{
    var dialog = new EditMalfunction(SelectedTv,
SelectedMalfunctionTv);
    dialog.ShowDialog();
    OnPropertyChanged("MalfunctionsTv");
}

private void OnDeleteMalfunctionTv(object param)
{
    using var db = new TrolleybusContext();
    db.Remove(SelectedMalfunctionTv);
    db.SaveChanges();
    OnPropertyChanged("MalfunctionsTv");
}

public List<ToTv> ToTvsTv
{
    get
    {
        if (SelectedTv != null)
        {
            using var db = new TrolleybusContext();
            return db.ToTvs.OrderByDescending(s =>
s.CompleteDate).Include(p => p.ToTvExecutors).Where(p =>
p.TvNumber == SelectedTv.Number)
                .ToList();
        }
        else
        {
            return new List<ToTv>();
        }
    }
}

```

```

    }

    private ToTv selectedToTvTv;
    public ToTv SelectedToTvTv
    {
        get
        {
            return selectedToTvTv;
        }

        set
        {
            selectedToTvTv = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteToTvTvCommand => new
RelayCommand(OnDeleteToTvTv);
    public ICommand AddToTvTvCommand => new
RelayCommand(OnAddToTvTv);
    public ICommand EditToTvTvCommand => new
RelayCommand(OnEditToTvTv);

    private void OnAddToTvTv(object param)
    {
        var dialog = new EditToTv(SelectedTv);
        dialog.ShowDialog();
        OnPropertyChanged("ToTvsTv");
    }

    private void OnEditToTvTv(object param)
    {
        var dialog = new EditToTv(SelectedTv, SelectedToTvTv);
        dialog.ShowDialog();
        OnPropertyChanged("ToTvsTv");
    }

    private void OnDeleteToTvTv(object param)
    {
        using var db = new TrolleybusContext();
        db.Remove(SelectedToTvTv);
        db.SaveChanges();
        OnPropertyChanged("ToTvsTv");
    }

    public List<Shod> Shods
    {
        get
        {
            using var db = new TrolleybusContext();

```

```

        return db.Shods.OrderByDescending(s
s.DepartureTime).ToList();
    }

    private Shod selectedShod;
    public Shod SelectedShod
    {
        get
        {
            return selectedShod;
        }

        set
        {
            selectedShod = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteShodCommand => new
RelayCommand(OnDeleteShod);
    public ICommand AddShodCommand => new
RelayCommand(OnAddShod);
    public ICommand EditShodCommand => new
RelayCommand(OnEditShod);
    public ICommand UpdateShodCommand => new
RelayCommand(OnUpdateShod);

    private void OnAddShod(object param)
    {
        var dialog = new EditShod(null);
        dialog.ShowDialog();
        OnPropertyChanged("Shods");
    }

    private void OnEditShod(object param)
    {
        var dialog = new EditShod(null, SelectedShod);
        dialog.ShowDialog();
        OnPropertyChanged("Shods");
    }

    private void OnDeleteShod(object param)
    {
        using var db = new TrolleybusContext();
        db.Remove(SelectedShod);
        db.SaveChanges();
        OnPropertyChanged("Shods");
    }

```

```

private void OnUpdateShod(object param)
{
    OnPropertyChanged("Shods");
}

public List<Malfunction> Malfunctions
{
    get
    {
        using var db = new TrolleybusContext();
        return db.Malfunctions.OrderByDescending(s =>
s.AcceptTime).Include(p => p.MalfunctionExecutors).ToList();
    }
}

private Malfunction selectedMalfunction;
public Malfunction SelectedMalfunction
{
    get
    {
        return selectedMalfunction;
    }

    set
    {
        selectedMalfunction = value;
        OnPropertyChanged();
    }
}

public ICommand DeleteMalfunctionCommand => new
RelayCommand(OnDeleteMalfunction);
public ICommand AddMalfunctionCommand => new
RelayCommand(OnAddMalfunction);
public ICommand EditMalfunctionCommand => new
RelayCommand(OnEditMalfunction);
public ICommand UpdateMalfunctionCommand => new
RelayCommand(OnUpdateMalfunction);
public ICommand ReclamationCommand => new
RelayCommand(OnReclamation);

private void OnAddMalfunction(object param)
{
    var dialog = new EditMalfunction(null);
    dialog.ShowDialog();
    OnPropertyChanged("Malfunctions");
}

private void OnEditMalfunction(object param)
{

```

```

        var dialog = new EditMalfunction(null,
SelectedMalfunction);
        dialog.ShowDialog();
        OnPropertyChanged("Malfunctions");
    }

    private void OnDeleteMalfunction(object param)
    {
        using var db = new TrolleybusContext();
        db.Remove(SelectedMalfunction);
        db.SaveChanges();
        OnPropertyChanged("Malfunctions");
    }

    private void OnUpdateMalfunction(object param)
    {
        OnPropertyChanged("Malfunctions");
    }

    private void OnReclamation(object param)
    {
        Malfunction selected;
        if (param.ToString() == "parent")
        {
            selected = SelectedMalfunctionTv;
        }
        else
        {
            selected = SelectedMalfunction;
        }

        if (!Directory.Exists(Directory.GetCurrentDirectory()
+ "\\АКТЫ"))
        {
            Directory.CreateDirectory(Directory.GetCurrentDirectory() +
"\\АКТЫ");
        }
        using var db = new TrolleybusContext();
        int? aktNum = db.Malfunctions.Select(p =>
p.Reclamation).Max();
        if (!aktNum.HasValue)
        {
            aktNum = 1;
        }
        else
        {
            aktNum = aktNum.Value;
        }
    }

```

```

        string newAkt = Directory.GetCurrentDirectory() +
        "\\Акты" + $"{aktNum}.docx";

        if (File.Exists(newAkt))
        {
            if (HandyControl.Controls.MessageBox.Show(new
        MessageBoxInfo
        {
            Message = $"Файл акта для этой неисправности
с номером {aktNum} уже существует. Продолжить?",
            Caption = "Подтверждение",
            Button = MessageBoxButton.YesNo,
        }) == MessageBoxResult.No)
        {
            return;
        }
        }

        File.Copy("шаблон_акт.docx", newAkt);

        var valuesToFill = new Content
        (
            new FieldContent("AktNumber",
        aktNum.Value.ToString()),
            new FieldContent("AktDate",
        DateTime.Now.Date.ToString("dd.MM.yyyy")),
            new FieldContent("TvNumber",
        selected.TvNumber.ToString()),
            new FieldContent("Vin",
        selected.TvNumberNavigation.Vin),
            new FieldContent("DateExluatation",
        selected.TvNumberNavigation.CommissioningDate.Value.ToString("dd
        .MM.yyyy")),
            new FieldContent("MalfunctionDate",
        selected.AcceptTime.Value.ToString("dd.MM.yyyy")),
            new FieldContent("ZapretExluatation",
        selected.ZapretExluatation.Value ? "Да" : "Нет"),
            new FieldContent("Probeg",
        selected.Probeg.ToString()),
            new FieldContent("ТО", "Тут сведения о проведении
        ТО"),
            new FieldContent("ActMalfunction",
        selected.MalfunctionRecl),
            new FieldContent("ActComments",
        selected.MalfunctionReclComment)
        );

        using (var outputDocument = new
        TemplateProcessor(newAkt))
        {
            outputDocument.FillContent(valuesToFill);
        }
    
```



```

        outputDocument.SaveChanges();
    }

    selected.Reklamation = aktNum;
    db.Update(selected);
    db.SaveChanges();

    OnPropertyChanged("Data");
    System.Diagnostics.Process.Start("explorer.exe",
newAkt);
    }

    public List<ToTv> ToTvs
    {
        get
        {
            using var db = new TrolleybusContext();
            return db.ToTvs.OrderByDescending(s
s.CompleteDate).Include(p => p.ToTvExecutors).ToList();
        }
    }

    private ToTv selectedToTv;
    public ToTv SelectedToTv
    {
        get
        {
            return selectedToTv;
        }

        set
        {
            selectedToTv = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteToTvCommand => new
RelayCommand(OnDeleteToTv);
    public ICommand AddToTvCommand => new
RelayCommand(OnAddToTv);
    public ICommand EditToTvCommand => new
RelayCommand(OnEditToTv);
    public ICommand UpdateToTvCommand => new
RelayCommand(OnUpdateToTv);

    private void OnAddToTv(object param)
    {
        var dialog = new EditToTv(null);
        dialog.ShowDialog();
        OnPropertyChanged("ToTvs");
    }

```

```

    }

    private void OnEditToTv(object param)
    {
        var dialog = new EditToTv(null, SelectedToTv);
        dialog.ShowDialog();
        OnPropertyChanged("ToTvs");
    }

    private void OnDeleteToTv(object param)
    {
        using var db = new TrolleybusContext();
        db.Remove(SelectedToTv);
        db.SaveChanges();
        OnPropertyChanged("ToTvs");
    }

    private void OnUpdateToTv(object param)
    {
        OnPropertyChanged("ToTvs");
    }

    public List<Employee> Employees
    {
        get
        {
            var db = new TrolleybusContext();
            return db.Employees.Where(p => p.Id !=
14).ToList();
        }
    }

    private Employee selectedEmployee;
    public Employee SelectedEmployee
    {
        get
        {
            return selectedEmployee;
        }

        set
        {
            selectedEmployee = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteEmployeeCommand => new
RelayCommand(OnDeleteEmployee);
    public ICommand AddEmployeeCommand => new
RelayCommand(OnAddEmployee);

```

```

        public ICommand EditEmployeeCommand => new
RelayCommand(OnEditEmployee);
        public ICommand UpdateEmployeeCommand => new
RelayCommand(OnUpdateEmployee);

        private void OnAddEmployee(object param)
        {
            var dialog = new EditEmployee();
            dialog.ShowDialog();
            OnPropertyChanged("Employees");
        }

        private void OnEditEmployee(object param)
        {
            var dialog = new EditEmployee(SelectedEmployee);
            dialog.ShowDialog();
            OnPropertyChanged("Employees");
        }

        private void OnDeleteEmployee(object param)
        {
            using var db = new TrolleybusContext();
            db.Remove(SelectedEmployee);
            db.SaveChanges();
            OnPropertyChanged("Employees");
        }

        private void OnUpdateEmployee(object param)
        {
            OnPropertyChanged("Employees");
        }

        public List<DbUser> DbUsers
        {
            get
            {
                using var db = new TrolleybusContext();
                return db.DbUsers.Where(p => p.Employee !=
14).ToList();
            }
        }

        private DbUser selectedDbUser;
        public DbUser SelectedDbUser
        {
            get
            {
                return selectedDbUser;
            }

            set

```

```

        {
            selectedDbUser = value;
            OnPropertyChanged();
        }
    }

    public ICommand DeleteDbUserCommand => new
    RelayCommand(OnDeleteDbUser);
    public ICommand AddDbUserCommand => new
    RelayCommand(OnAddDbUser);
    public ICommand EditDbUserCommand => new
    RelayCommand(OnEditDbUser);
    public ICommand UpdateDbUserCommand => new
    RelayCommand(OnUpdateDbUser);

    private void OnAddDbUser(object param)
    {
        try
        {
            var dialog = new EditDbUser();
            dialog.ShowDialog();
        }
        catch (Exception ex)
        {
            HandyControl.Controls.MessageBox.Show(new
    MessageBoxInfo
            {
                Message = ex.Message,
                Caption = "Ошибка",
                Button = MessageBoxButton.OK,
            });
        }
        OnPropertyChanged("DbUsers");
    }

    private void OnEditDbUser(object param)
    {
        var dialog = new EditDbUser(SelectedDbUser);
        dialog.ShowDialog();
        OnPropertyChanged("DbUsers");
    }

    private void OnDeleteDbUser(object param)
    {
        using var db = new TrolleybusContext();
        db.Remove(SelectedDbUser);
        db.SaveChanges();
        OnPropertyChanged("DbUsers");
    }

    private void OnUpdateDbUser(object param)

```

```

        {
            OnPropertyChanged("DbUsers");
        }

public List<Park> Parks
{
    get
    {
        using var db = new TrolleybusContext();
        return db.Parks.ToList();
    }
}

private Park selectedPark;
public Park SelectedPark
{
    get
    {
        return selectedPark;
    }

    set
    {
        selectedPark = value;
        OnPropertyChanged();
    }
}

public ICommand DeleteParkCommand => new
RelayCommand(OnDeletePark);
public ICommand AddParkCommand => new
RelayCommand(OnAddPark);
public ICommand EditParkCommand => new
RelayCommand(OnEditPark);
public ICommand UpdateParkCommand => new
RelayCommand(OnUpdatePark);

private void OnAddPark(object param)
{
    var dialog = new EditPark();
    dialog.ShowDialog();
    OnPropertyChanged("Parks");
}

private void OnEditPark(object param)
{
    var dialog = new EditPark(SelectedPark);
    dialog.ShowDialog();
    OnPropertyChanged("Parks");
}

```

```

private void OnDeletePark(object param)
{
    using var db = new TrolleybusContext();
    db.Remove(SelectedPark);
    db.SaveChanges();
    OnPropertyChanged("Parks");
}

private void OnUpdatePark(object param)
{
    OnPropertyChanged("Parks");
}

CancellationTokenSource CancellationTokenSource;

public MainWindowViewModel(Employee user)
{
    CurrentUser.Employee = user;
    CancellationTokenSource = new
CancellationTokenSource();
    var ct = CancellationTokenSource.Token;
    var updater = new Task
        (() =>
        {
            var db = new TrolleybusContext();
            int oldCount =
db.Notifications.OrderByDescending(x => x.Id).Where(p =>
p.Posts.Contains(CurrentUser.Employee.Post.Value))
                .Where(p =>
p.Posts.Contains(CurrentUser.Employee.Park.Value)).Count();
            while (true)
            {
                if (ct.IsCancellationRequested) return;
                db = new TrolleybusContext();
                int newCount =
db.Notifications.OrderByDescending(x => x.Id).Where(p =>
p.Posts.Contains(CurrentUser.Employee.Post.Value))
                .Where(p =>
p.Posts.Contains(CurrentUser.Employee.Park.Value)).Count();
                if (newCount > oldCount)
                {
                    var notif =
db.Notifications.OrderByDescending(x => x.Id).Where(p =>
p.Posts.Contains(CurrentUser.Employee.Post.Value))
                .Where(p =>
p.Posts.Contains(CurrentUser.Employee.Park.Value)).ToList();
                    int toShowCount = newCount - oldCount;
                    for (int i = 0; i < toShowCount; i++)
                    {

```

```

        MessageBox.Show(notif[i].Message, "Уведомление",
        MessageBoxButton.OK, MessageBoxImage.Information);
    }

    oldCount = newCount;
}
}
}, ct);
updater.Start();
}

public void BeforeClose()
{
    if(CancellationTokenSource != null)
    {
        CancellationTokenSource.Cancel();
    }
}

public ICommand SelectionCommand => new
RelayCommand(OnSelectionChanged);

private void OnSelectionChanged(object param)
{
    var source = (param as
SelectionChangedEventArgs).Source;
    if (source != null)
    {
        if(source is TabControl)
        {
            var index = (source as
TabControl).SelectedIndex;

            switch (index)
            {
                case 0:

OnPropertyChanged("TransportVenichles");
                return;
                case 1:
                    OnPropertyChanged("Shods");
                    return;
                case 2:
                    OnPropertyChanged("Malfunctions");
                    return;
                case 3:
                    OnPropertyChanged("ToTvs");
                    return;
                case 4:
                    OnPropertyChanged("Employees");
            }
        }
    }
}

```

```

        return;
    case 5:
        OnPropertyChanged("DbUsers");
        return;
    case 6:
        OnPropertyChanged("Parks");
        return;
    }
}

}

}

public event PropertyChangedEventHandler?
PropertyChanged;

public void OnPropertyChanged([CallerMemberName] string
prop = "")
{
    if (PropertyChanged != null)
        PropertyChanged(this, new
PropertyChangedEventArgs(prop));
}
}

```

Файл « App.xaml»

<Application x:Class="trolleybus.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:trolleybus"

xmlns:view="clr-namespace:trolleybus.View"

StartupUri="LoginWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<ResourceDictionary

Source="pack://application:,,,/HandyControl;component/Themes/SkinDefault.xaml"/>

<ResourceDictionary

Source="pack://application:,,,/HandyControl;component/Themes/Theme.xaml"/>

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

<!--Scaffold-DbContext

"Host=localhost;Port=5432;Database=trolleybus;Username=postgres;Password=12345678" Npgsql.EntityFrameworkCore.PostgreSQL -

OutputDir Model -F-->

</Application>

					ПГУ 1.090301.07.001 ПЗ	Лист
						208
Изм.	Лист	№ докум.	Подпись	Дата		

Файл « LoginWindow.xaml»

```
<hc:Window x:Class="trolleybus.LoginWindow"

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:trolleybus.View"
    mc:Ignorable="d"
    xmlns:hc="https://handyorg.github.io/handycontrol"
    Title="Авторизация"
    SizeToContent="WidthAndHeight"
    WindowStartupLocation="CenterScreen"
    ShowNonClientArea="False">
    <Grid HorizontalAlignment="Stretch" Margin="20,10,20,10">
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="1*" />
        </Grid.ColumnDefinitions>
        <Grid.RowDefinitions>
            <RowDefinition Height="60" />
            <RowDefinition Height="60" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="60" />
            <RowDefinition Height="Auto" />
            <RowDefinition Height="1*" />
        </Grid.RowDefinitions>
        <TextBlock Text="Авторизация"
            FontSize="35" HorizontalAlignment="Center"
VerticalAlignment="Center" />
        <TextBlock Text="Сотрудник"
            Grid.Row="1"
            FontSize="25" HorizontalAlignment="Center"
VerticalAlignment="Center" />
        <hc:ComboBox ItemsSource="{Binding DbUsers}"
            SelectedItem="{Binding SelectedDbUser,
Mode=TwoWay}"
            HorizontalContentAlignment="Center"
            Grid.Row="2"
            FontSize="25"
            Height="60" />
        <TextBlock Text="Пароль"
            Grid.Row="3"
            FontSize="25" HorizontalAlignment="Center"
VerticalAlignment="Center" />
        <hc:PasswordBox
            ShowEyeButton="True"
            IsSafeEnabled="False"
```

```

        UnsafePassword="{Binding Password}"
        Grid.Row="4" Height="60" FontSize="25"
        HorizontalContentAlignment="Center"/>
<Grid Grid.Row="5">
    <Grid.ColumnDefinitions>
        <ColumnDefinition/>
        <ColumnDefinition/>
    </Grid.ColumnDefinitions>
    <Button Content ="Авторизоваться"
        Name="Login"
        Click="Login_Click"
        IsEnabled="{Binding Ready}"
        Width="194"
        Height="54"
        Grid.Column="0"
        FontSize="20"/>
    <Button Content ="Выход"
        Name="Exit"
        Click="Exit_Click"
        Width="194"
        Height="54"
        Grid.Column="1"
        FontSize="20"/>

</Grid>
</Grid>
</hc:Window>

```

Приложение В

(Обязательное)

Автоматизированная система учета состояния троллейбусного парка

Презентация

					ПГУ 1.090301.07.001 ПЗ	Лист
						211
Изм.	Лист	№ докум.	Подпись	Дата		



Пензенский
государственный
университет



Кафедра
вычислительной
техники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА

АВТОМАТИЗИРОВАННАЯ СИСТЕМА УЧЕТА ТРОЛЛЕЙБУСНОГО ДЕПО

АВТОР РАБОТЫ: СТУДЕНТ ГР. 20ВВВ1 ЗАЦЕПИЛИНА Е.С.
НАУЧНЫЙ РУКОВОДИТЕЛЬ: ДОЦЕНТ, К.Т.Н. КАРАМЫШЕВА Н.С.



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.07.001 ПЗ

Лист
212

Актуальность

Актуальность данной темы, прежде всего, обусловлена отсутствием собственной автоматизированной системы учета троллейбусов в троллейбусном парке «Универсал Сервис».

Компания приняла решение внедрить автоматизированную систему в свой рабочий процесс для того чтобы увеличить производительность, облегчить и упростить рабочий процесс.



Изм.	Лист	№ докум.	Подпись	Дата

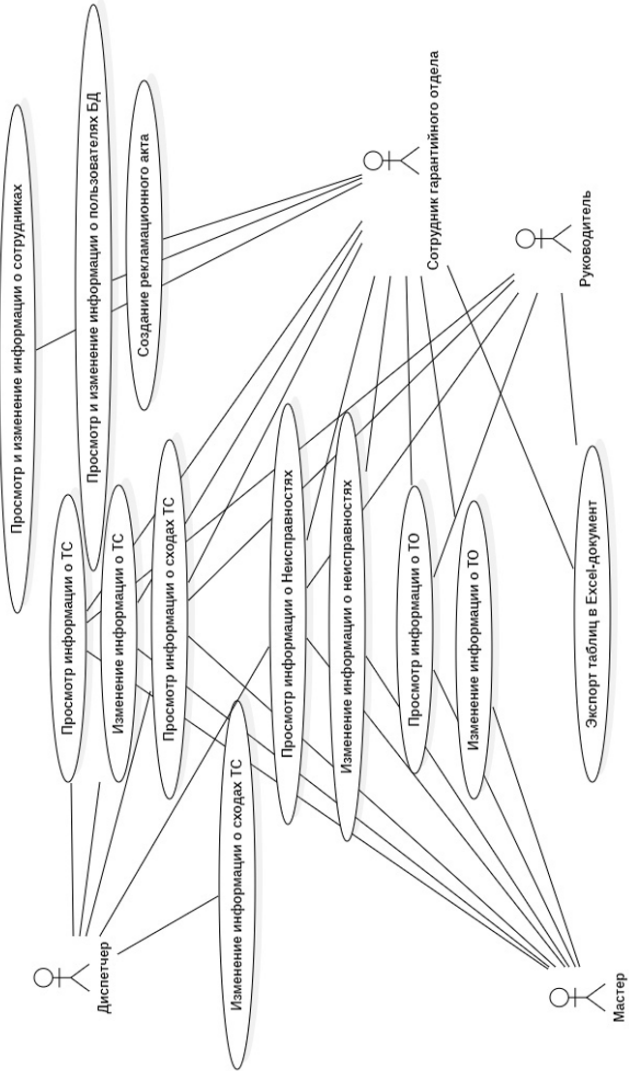
ПГУ 1.090301.07.001 ПЗ

Цели и задачи

Целью выпускной работы является разработка автоматизированной системы для учета состояния троллейбусов в троллейбусном парке.

В ходе данной работы необходимо разработать приложение для автоматизации процесса учета троллейбусов в троллейбусном парке. Приложение должно иметь удобный графический интерфейс, для визуализации всей необходимой информации для учета троллейбусов. Входными данными являются информация о троллейбусах, а также об их состоянии в текущий момент, о сотрудниках предприятия, в том числе пользователей приложения, а также справочная информация о видах технического обслуживания. Результатом работы приложения является файл, содержащий в себе рекламационный акт о неисправности троллейбуса.

Постановка задачи



Общество с ограниченной ответственностью
«Универсал Сервис»

440028, г. Пенза, проспект Победы 65, офис 2
ИНН 840313/07
ПИН 840313/029
E-mail: info@univservis@mail.ru

АО «Уфимский трамвайно-троллейбусный завод» РФ, Уфа, пр-т Октября, 143 zavt@uiz@mail.ru , info@uiz@mail.ru
Директору ГКУ «Организатор перевозок Пензенской области» Л.В. Панковой 440066, г. Пенза, пр. Победы, 150 zku_urolo@mail.ru

РЕКЛАМАЦИОННЫЙ АКТ №

на троллейбус пассажирский мод. 624101 «Горюшкин»

1. Дата составления	
2. Заводской № (VIN) №	
3. Инвентарный (серийный) №	
4. Дата ввоза в эксплуатацию	
5. Дата обнаружения дефекта	
6. Сведения об устранении от эксплуатации	
7. Обший пробег на момент обнаружения дефекта	
8. Сведения о проведении ТО, дата и пробег	
9. Описание и условия проявления дефекта	
10. Дополнительная информация	--
11. Приложение	Копия сервисной книжки (Приложение 1)
12. Дилер, у которого приобретен троллейбус	АО «Уфимский трамвайно-троллейбусный завод»

«__» _____ 20__ г.
МД

А.Ю. Кузнецов

Выбор решения

Язык разработки:

- C#



Среды разработки:

- Visual Studio 2022



Целевая платформа:

- Windows 10

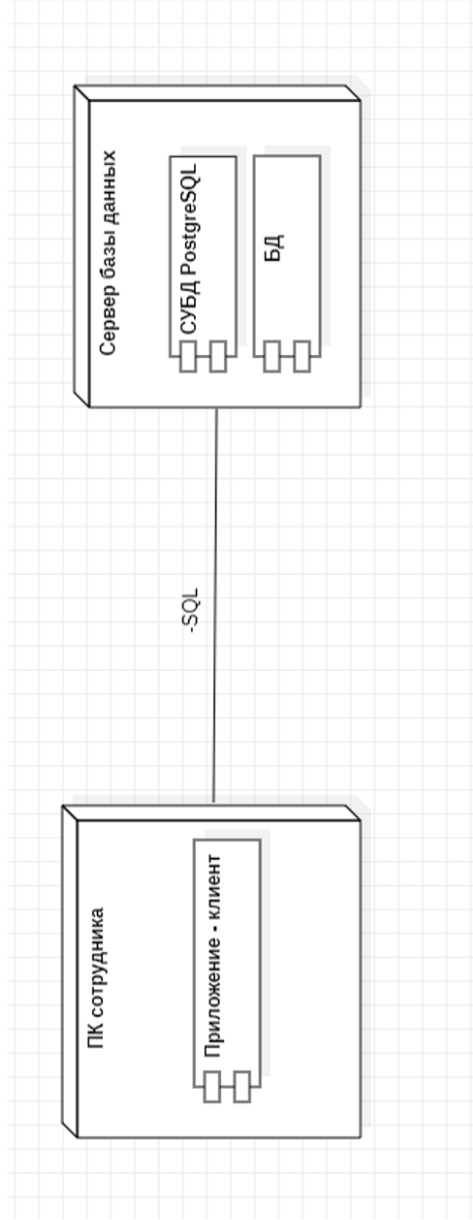
Другие ресурсы разработки: ПК, pgAdmin4



Изм.	Лист	№ докум.	Подпись	Дата

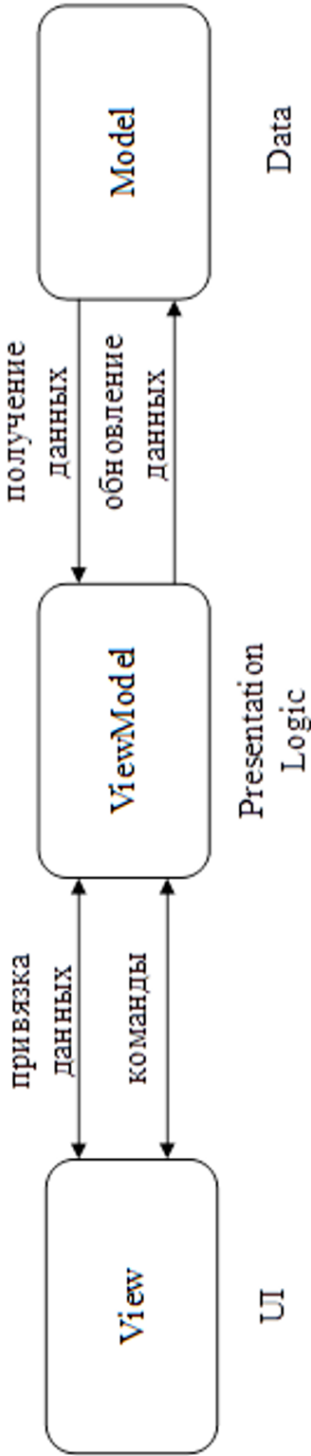
Процесс разработки

Диаграмма развертывания автоматизированной системы



Процесс разработки

Визуализация взаимодействия компонентов при использовании паттерна MVVM



Изм.	Лист	№ докум.	Подпись	Дата

Лист
219

[illegible]

Диаграмма классов для модели



Троллейбусное депо

+

🔍

📄

🔄

Транспортные средства

Сход

Неисправности

Техническое обслуживание

Сотрудники

Пользователи БД

Парси

▼

Май

▼

Гос. Нов.

▼

VIN-ноч.

▼

Год вылу

▼

Дата ввода в эксплуатац

▼

Про

▼

Пл

▼

Авто

▼

Статус

0 записей найдено из 0

Сход

Неисправности

ТО

+

🔍

📄

Номер ТС

▼

Время схода

▼

Неисправность

▼

Экзекатор

▼

Время прибытия

Авторизация

Сотрудник

Администратор А.А.

Пароль

Авторизоваться

Выход

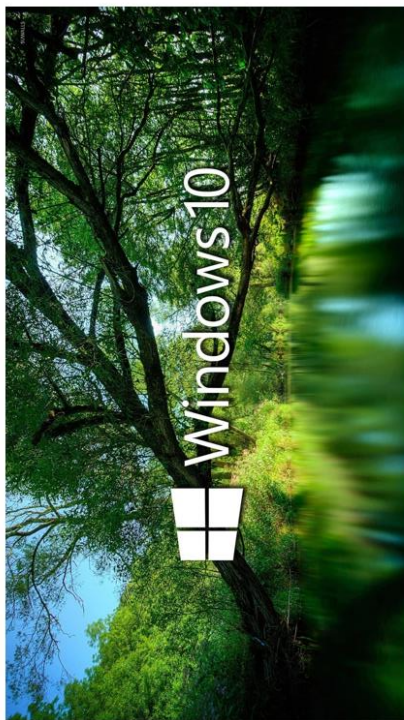
Заключение

В рамках разработки проекта были выявлены недостатки существующих аналогов и предложен свой вариант решения в виде реализации текущего проекта.

Итоговый вариант работоспособен и пригоден для внедрения на производство. В дальнейшем проект может быть расширен путем создания сервера для уведомлений.



Спасибо за внимание!



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ 1.090301.07.001 ПЗ