СОДЕРЖАНИЕ

ВВЕДЕНИЕ 3

1 ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ 4

1.1 Основные требования по технике безопасности во время прохождения учебной практики 4

1.2 Нормативно-правовые документы 5

2 ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ 7

2.1 Описание предметной области 7

2.2 Группы пользователей информационной системы 8

2.3 Основные требования, предъявляемые к информационной системе 8

3 ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «РЕМОНТ ПАССАЖИРСКИХ ВАГОНОВ» 10

3.1 Моделирование некоторых динамических аспектов системы 10

3.1.1 Создание диаграммы вариантов использования 10

3.1.2 Создание диаграммы деятельности 10

3.2 Разработка функциональной структуры информационной системы. Карта навигации по информационной системе 11

3.3 Разработка макета информационной системы «Ремонт пассажирских вагонов» 11

4 РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «РЕМОНТ ПАССАЖИРСКИХ ВАГОНОВ» 12

4.1 Моделирование и разработка базы данных 12

4.2 Разработка программных модулей информационной системы «Ремонт пассажирских вагонов» 15

5 РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ 37

5.1 Назначение эксперимента. Выбор и обоснование методики проведения тестирования 37

5.2 Технология тестирования 37

5.3 Результаты проведения тестирования 39

ЗАКЛЮЧЕНИЕ 41

БИБЛИОГРАФИЧЕСКИЙ СПИСОК 42

ПРИЛОЖЕНИЯ А 44

ПРИЛОЖЕНИЯ Б 63

# ВВЕДЕНИЕ

Пассажирские вагоны играют ключевую роль в сфере железнодорожного транспорта, обеспечивая комфорт и безопасность путешествий. С течением времени эта отрасль претерпела значительное развитие, внедряя новые технологии и улучшая условия для пассажиров.

Начиная с классических вагонов с деревянными отделениями, современные пассажирские поезда предлагают разнообразные варианты, начиная от купейных до открытых салонов. Инновации в дизайне и строительстве включают использование легких и прочных материалов, улучшенные системы вентиляции и климат контроля.

С развитием высокоскоростных поездов, появлением беспроводного интернета и интеллектуальных систем управления, пассажирские вагоны становятся не только местом транспортировки, но и платформой для передовых технологий. Интеграция умных решений, таких как системы безопасности и развлечений, сделала поездки более современными и удобными.

В целом, развитие пассажирских вагонов продолжает стремительно продвигаться вперед, объединяя комфорт, технологии и ответственность перед окружающей средой.

# ЦЕЛИ И ЗАДАЧИ УЧЕБНОЙ ПРАКТИКИ

Разработка программных модулей для системы ремонта пассажирских вагонов направлена на повышение эффективности, надежности и прозрачности процессов технического обслуживания. Целью является разработка и проектирование информационной системы, способствующей оптимизации работы и обеспечивающей высокий уровень безопасности и качества обслуживания пассажирского транспорта.

Задачи:

1. Автоматизация учета технического состояния: разработка модулей для автоматизации мониторинга и учета технического состояния пассажирских вагонов.
2. Управление ремонтными работами: реализация функционала по управлению ремонтными работами, включая планирование, выделение ресурсов, контроль сроков и мониторинг качества выполненных работ.
3. Обеспечение безопасности данных: разработка модулей для защиты конфиденциальности данных, связанных с техническим состоянием вагонов и ремонтными работами, чтобы предотвратить несанкционированный доступ.

Цель и задачи в совокупности направлены на повышение эффективности обслуживания и обеспечение безопасности и надежности пассажирского транспорта через современные информационные технологии.

## Основные требования по технике безопасности во время прохождения учебной практики

К работе в компьютерном классе допускаются лица, ознакомленные с данной инструкцией по технике безопасности и правилам поведения. Необходимо неукоснительно соблюдать правила по технике безопасности, т.к. нарушение этих правил может привести к поражению электрическим током, вызвать возгорание и навредить вашему здоровью.

* Работа обучающегося в компьютерном классе разрешается только в присутствии преподавателя (лаборанта).
* Во время занятий посторонние лица могут находиться в классе только с разрешения преподавателя.
* Во время перемен между уроками проводится обязательное проветривание компьютерного кабинета с обязательным выходом обучающегося из класса.

Каждый обучающийся в ответе за состояние своего рабочего места и сохранность размещенного на нем оборудования.

Обучающегося, допустившего нарушение по технике безопасности и правилам поведения в компьютерном классе, может быть привлечен к дисциплинарной ответственности. Если нарушение техники безопасности связано с причинением имущественного ущерба, он несет и материальную ответственность в установленном законом порядке.

## Нормативно-правовые документы

Оформление и содержание технического задания должно соответствовать требованиям стандарта «ГОСТ 19.201–78. ЕСПД. Общие положения.

Техническое задание оформляют в соответствии с ГОСТ 19.106–78 на листах формата 11 и 12 по ГОСТ 2.301–68, как правило, без заполнения полей листа. Номера листов (страниц) проставляются в верхней части листа над текстом.

Лист утверждения и титульный лист оформляют в соответствии с ГОСТ 19.104–78.

Информационную часть (аннотацию и содержание), лист регистрации изменений допускается в документ не включать.

Для внесения изменений или дополнений в техническое задание на последующих стадиях разработки про1раммы или программного изделия выпускают дополнение к нему.

Согласование и утверждение дополнения к техническому заданию проводят в том же порядке, который установлен для технического задания.

Техническое задание должно содержать следующие разделы:

* Введение;
* Основания для разработки;
* Назначение разработки;
* Требования к программе или программному изделию;
* Требования к программной документации;
* Технико-экономические показатели;
* Стадии и этапы разработки;
* Порядок контроля и приемки;
* В техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

Вывод по главе: были описаны, техника безопасности во время прохождения учебной практики; нормативно-правовые документы, которые отвечают за стандарты разработки; цели и задачи учебной практики, включая вводную часть, где описана тематика отрасли и её тенденции развития.

# ПРЕДМЕТНАЯ ОБЛАСТЬ. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ. ПОСТАНОВКА ЗАДАЧИ

Предметная область информационной системы — это материальная система или система, характеризующая элементы материального мира, информация о которой хранится и обрабатывается. Предметная область рассматривается как некоторая совокупность реальных объектов и связей между ними.

## Описание предметной области

Необходимо разработать и спроектировать информационную систему для эффективного управления процессами технического обслуживания и ремонта пассажирских вагонов, направленной на повышение безопасности, надежности и оптимизации затрат.

Депо выполняет несколько видов ремонта. Деповской ремонт – после пробега вагоном 450 тыс. км, или два года эксплуатации (что наступит раньше). TO–2 – подготовка вагона к зимним или летним условиям эксплуатации. ТО–3 – текущее обслуживание – после пробега 150 тыс. км, или один год эксплуатации. Текущий ремонт – круглосуточно, при котором ремонтируются вагоны всех дорог России. Основные причины поступления вагона в текущий ремонт: неисправность колесной пары, неисправность буксового узла и т.д. Каждый вагон имеет уникальный номер. Тип вагона также имеет значение при ремонте: купейный, СВ., плацкартный, почтовый, багажный. Каждый вагон приписан к дирекции по обслуживанию пассажиров (ДОП–1, ДОП–2, ДОП–3 и т.д.). Текущий ремонт выполняют ремонтные бригады в четыре смены. Для выполнения остальных ремонтов привлекается, как правило, одна бригада. За высокое качество ремонта члены бригады получают премию.

Информационная система "Ремонт Пассажирских Вагонов" призвана содействовать повышению эффективности и безопасности в процессах обслуживания, а также уменьшению затрат за счет оптимизации ресурсов и раннего выявления потенциальных проблем.

## Группы пользователей информационной системы

Сотрудники, в зависимости от своих функциональных обязанностей, могут выполнять действия, направленные на мониторинг, управление, анализ данных и обеспечение безопасности в рамках проектируемой информационной системы "Ремонт Пассажирских Вагонов".

Роли и их обязанности:

* + Гость: просмотр данных.
  + Администратор: управление системами аутентификации, авторизации и шифрования; оптимизация работы системы для обеспечения высокой производительности; полный доступ ко всей системе.
  + Сотрудник: получение информации о текущем состоянии вагонов, диагностических данных и рекомендациях по обслуживанию; участие в разработке графика технического обслуживания и ремонтных работ.
  + Бригадир: получение информации о текущем состоянии вагонов, диагностических данных и рекомендациях по обслуживанию; участие в разработке графика технического обслуживания и ремонтных работ.

## Основные требования, предъявляемые к информационной системе

* + Гибкость: система должна обладать высокой степенью редактирования, чтобы пользователи могли адаптировать её под различные потребности в рамках своих обязанностей; возможность легкого масштабирования системы при увеличении объема данных или числа пользователей.
  + Надежность: обеспечение бесперебойной работы системы с минимальным количеством сбоев и перерывов; регулярное создание резервных копий данных для предотвращения потери информации в случае сбоев.
  + Эффективность: обеспечение высокой производительности при обработке данных, особенно в условиях одновременного использования множества пользователей; эффективное использование вычислительных ресурсов, чтобы система функционировала быстро и эффективно.
  + Безопасность: реализация надежной системы аутентификации для проверки личности пользователей и установление строгих прав доступа; и использование шифрования для защиты передаваемой и хранимой чувствительной информации.

Вывод по главе: было описано, что такое предметная область и её описание; группы пользователей, которые будут пользоваться информационной системой и основные системные требования к ней.

# ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ «РЕМОНТ ПАССАЖИРСКИХ ВАГОНОВ»

Проектированием информационных систем называется многоступенчатый процесс их создания и/или модернизации путём применения упорядоченной совокупности методологий и инструментария. Проектирование (в отличие от моделирования) предполагает работу с пока несуществующим объектом и направлено на создание информационной системы.

## Моделирование некоторых динамических аспектов системы

Диаграммы деятельности применяются для моделирования динамических аспектов поведения системы. Эти динамические аспекты могут включать деятельность на любом уровне абстракции в любом представлении системной архитектуры, включая классы, интерфейсы, компоненты и узлы.

При разработке программного обеспечения заблаговременное планирование и моделирование значительно упрощают программирование. На этапе создания концептуальной модели для описания функционального назначения используется диаграмма вариантов использования, а для описания логики процедур и бизнес-процессов используется диаграмма деятельности.

### Создание диаграммы вариантов использования

В приложении “А” на рисунке 1 представлен пример диаграммы вариантов использования информационной системы «Ремонт пассажирских вагонов».

### Создание диаграммы деятельности

В приложении “А” на рисунках 2 – 5 представлена диаграмма деятельности (по ролям) информационной системы «Ремонт пассажирских вагонов».

## Разработка функциональной структуры информационной системы. Карта навигации по информационной системе

Карта переходов по информационной системе «Ремонт пассажирских вагонов» представлена в приложении “А” на рисунке 6.

## Разработка макета информационной системы «Ремонт пассажирских вагонов»

В приложении “А” на рисунке 7 представлено окно авторизации.

На этапе авторизации пользователь может ввести логин и пароль, чтобы продолжить работу в информационной системе.

В случае если пользователь еще не зарегистрирован в системе, он может сделать это, нажав на кнопку «Зарегистрироваться». В этом случае откроется окно регистрации, представленное в приложении “А” на рисунке 8.

На этапе регистрации пользователю необходимо придумать свой логин и пароль.

После того, как пользователь успешно прошел авторизацию, он может продолжить работу с таблицами на главной форме. В приложении “А” на рисунках 9 – 12 представлена главная форма, которая будет отображена, в зависимости от прав пользователя в системе. Например, для гостя будут скрыты кнопки редактирования, добавления и удаления записи, так как данной группе пользователей доступен только просмотр данных.

Дизайн выполнен в простом стиле для того, чтобы пользователь с любым уровнем владения компьютером мог без препятствий работать с программным обеспечением.

Вывод по главе: была спроектирована диаграмма вариантов использования, диаграмма деятельности и макеты будущей информационной системы.

# РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ «РЕМОНТ ПАССАЖИРСКИХ ВАГОНОВ»

Прототип — это один из этапов разработки, который заключается в продумывании содержания и расположения важных элементов интерфейса. Прототип — это моделирование конечного продукта. Это интерактивный макет, который может иметь любую степень точности. Основная цель создания прототипов – проверить, насколько последователен путь пользователя, и выявить препятствия, которые могут возникнуть в процессе его взаимодействия с продуктом.

## Моделирование и разработка базы данных

Исходя из анализа предметной области, можно выделить восемь сущностей: «Железные Дороги», «Вагоны», «Ремонты», «Ремонтные Работы», «Бригады», «Бригадиры», «Сотрудники», «Пользователи».

В приложении “А” на рисунке 13 представлена логическая модель данных базы данных «Ремонт пассажирских вагонов».

Создание базы данных происходит в MS SQL, в примере представлено создание базы данных в MS SQL Server.

С помощью раздела «Создание» и конструктора таблиц MS SQL созданы все таблицы и поля, представленные на логической модели данных.

Названия полей и их типы данных представлены в таблицах 1-8.

Таблица 1 – Поля таблицы «Железные Дороги»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Дороги | Числовой |
| Внешняя Местная ЖД | Логический |
| Банк Внешней ЖД | Текстовый |
| ИНН Внешней ЖД | Числовой |
| Юридический Адрес | Текстовый |

Таблица 2 – Поля таблицы «Вагоны»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Вагона | Числовой |
| Код Дороги | Числовой |
| Номер Вагона | Числовой |
| Приписка Вагона К Дирекции | Текстовый |
| Приписка Вагона К Дороге | Текстовый |
| Тип Вагона | Текстовый |
| Год Выпуска | Дата |
| Фотография | OLE |

Таблица 3 – Поля таблицы «Ремонты»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Ремонта | Числовой |
| Тип Ремонта | Текстовый |
| Стоимость Ремонта | Денежный |
| Причина Поступления В Ремонт | Текстовый |

Таблица 4 – Поля таблицы «Ремонтные Работы»

| Название поля | Тип данных |
| --- | --- |
| Код Работы | Числовой |
| Код Вагона | Числовой |
| Код Ремонта | Числовой |
| Код Бригады | Числовой |
| Вид Работы | Текстовый |
| Качество Ремонта | Логический |
| Начало Ремонта | Дата |
| Окончание Ремонта | Дата |
| Премия Общая | Числовой |

Таблица 5 – Поля таблицы «Бригады»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Бригады | Числовой |
| Код Бригадира | Числовой |
| Код Сотрудника | Числовой |

Таблица 6 – Поля таблицы «Бригадиры»

| Название поля | Тип данных |
| --- | --- |
| Код Бригадира | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Образование | Текстовый |
| Дата Трудоустройства | Дата |
| Специальность | Текстовый |
| Номер Банковской Карты | Текстовый |
| Премия В Рублях | Денежный |
| Примечания. За Премию | Memo |

Таблица 7 – Поля таблицы «Сотрудники»

|  |  |
| --- | --- |
| Название поля | Тип данных |
| Код Сотрудника | Числовой |
| Фамилия | Текстовый |
| Имя | Текстовый |
| Отчество | Текстовый |
| Образование | Текстовый |
| Дата Трудоустройства | Дата |
| Специальность | Текстовый |
| Номер Банковской Карты | Текстовый |
| Премия В Рублях | Денежный |
| Примечания. За Премию | Memo |

Таблица 8 – Поля таблицы «Пользователи»

| Название поля | Тип данных |
| --- | --- |
| Код Пользователя | Числовой |
| Почта | Текстовый |
| Логин | Текстовый |
| Пароль | Числовой |

В приложении “А” на рисунке 14 представлены все созданные таблицы в обозревателе объектов базы данных.

После того, как таблицы созданы, необходимо создать между ними связи.

В приложении “А” на рисунке 15 представлена физическая модель данных, отражающая все таблицы и их поля, а также связи между таблицами.

## Разработка программных модулей информационной системы «Ремонт пассажирских вагонов»

Разработанные в MS Visual Studio формы продемонстрированы в приложении “А” на рисунках 16-21.

Свойства элементов формы Авторизация представлены в таблице 9.

Таблица 9 – Свойства элементов формы Авторизация

| Объект | Свойство | Значение |
| --- | --- | --- |
| Авторизация | FormBorderStyle | None |
| Text | Ремонт пассажирских вагонов |
| StartPosition | CenterScreen |
| Size | 224; 291 |
| ShowIcon | False |

Событийно – управляемые процедуры в форме Авторизация представлены в таблице 10.

Таблица 10 – Событийно-управляемые процедуры в форме «Авторизация»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Авторизация | public partial class Авторизация : Form  {  public Авторизация()  {  InitializeComponent();  }  } |
| Событие Логин\_KeyPress | private void Логин\_KeyPress(object sender, KeyPressEventArgs e)  {  Логин.MaxLength = 30;  if (char.IsDigit(e.KeyChar))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать цифры!", "Ошибка");  }  } |
| Событие Пароль\_KeyPress | private void Пароль\_KeyPress(object sender, KeyPressEventArgs e)  {  Пароль.MaxLength = 15;  if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать буквы!", "Ошибка");  }} |
| Кнопка Войти | private void Войти\_Click(object sender, EventArgs e)  {  try  {  if ((Логин.Text == "") || (Пароль.Text == ""))  {  MessageBox.Show("Поля не должны быть пустыми.", "Ошибка");  }  else  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  int flag = -1;  for (int i = 0; i < dt.Rows.Count; i++)  {  if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Админ") && (dt.Rows[i]["Логин"].ToString() == Логин.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль.Text))  flag = 0;  if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Сотрудник") && (dt.Rows[i]["Логин"].ToString() == Логин.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль.Text))  {  flag = 1;  }  if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Бригадир") && (dt.Rows[i]["Логин"].ToString() == Логин.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль.Text))  {  flag = 2;  }  }  if ((flag == 0) || (flag == 1) || (flag == 2))  {  if (flag == 0)  {  Hide();  Администратор UserForm1 = new Администратор();  UserForm1.Show();  }  if (flag == 1)  {  Hide();  Сотрудник UserForm2 = new Сотрудник();  UserForm2.Show();  }  if (flag == 2)  {  Hide();  Бригадир UserForm3 = new Бригадир();  UserForm3.Show();  }  }  else  MessageBox.Show("Пересоздайте пользователя! Такого пользователя не существует.", "Ошибка");  }  }  catch { }  } |
| Кнопка Регистрация | private void Регистрация\_Click(object sender, EventArgs e)  {  Регистрация UserForm4 = new Регистрация();  UserForm4.Show();  } |
| Кнопка Войти как гость | private void Войти\_Как\_Гость\_Click(object sender, EventArgs e)  {  Hide();  Гость UserForm = new Гость();  UserForm.Show();} |
| Кнопка Показать пароль | private void Показать\_Пароль\_Click(object sender, EventArgs e)  {  if (Показать\_Пароль.Text == "Показать пароль")  {  Пароль.PasswordChar = '\0';  Показать\_Пароль.Text = "Скрыть пароль";  }  else if (Показать\_Пароль.Text == "Скрыть пароль")  {  Пароль.PasswordChar = '\*';  Показать\_Пароль.Text = "Показать пароль";  }  } |
| Кнопка Выйти | private void Выйти\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие Авторизация\_MouseMove | Point Mouse;  private void Авторизация\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Событие Авторизация\_MouseDown | private void Авторизация\_MouseDown(object sender, MouseEventArgs e)  {  Mouse = new Point(e.X, e.Y);  } |

Свойства элементов формы Регистрация представлены в таблице 11.

Таблица 11 – Свойства элементов формы Регистрация

| Объект | Свойство | Значение |
| --- | --- | --- |
| Регистрация | FormBorderStyle | None |
| Text | Ремонт пассажирских вагонов |
| StartPosition | CenterScreen |
| Size | 224; 291 |
| ShowIcon | False |

Событийно – управляемые процедуры в форме Регистрация представлены в таблице 12.

Таблица 12 – Событийно-управляемые процедуры в форме «Регистрация»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Регистрация | public partial class Регистрация : Form  {  public Регистрация()  {  InitializeComponent();  }  } |
| Событие Почта\_KeyPress | private void Почта\_KeyPress(object sender, KeyPressEventArgs e)  {  Почта.MaxLength = 30;  if (char.IsDigit(e.KeyChar))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать цифры!", "Ошибка");  }} |
| Событие Логин\_KeyPress | private void Логин\_KeyPress(object sender, KeyPressEventArgs e)  {  Логин.MaxLength = 30;  if (char.IsDigit(e.KeyChar))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать цифры!", "Ошибка");  }  } |
| Событие Пароль\_KeyPress | private void Пароль\_KeyPress(object sender, KeyPressEventArgs e)  {  Пароль.MaxLength = 15;  if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))  {  e.Handled = true;  MessageBox.Show("Поле не может содержать буквы!", "Ошибка");  }  } |
| Кнопка Подтвердить | private void Подтвердить\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  if ((Почта.Text == "") || (Логин.Text == "") || (Пароль.Text == ""))  {  MessageBox.Show("Поля не должны быть пустыми.", "Ошибка");  }  else  {  SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Почта, Логин, Пароль) VALUES ('{Почта.Text}', '{Логин.Text}', '{Пароль.Text}');", sqlConnect);  DataTable dt = new DataTable();  da.Fill(dt);  MessageBox.Show("Вы зарегистрированы, как сотрудник.", "Уведомление");  Hide();  Сотрудник UserForm = new Сотрудник();  UserForm.Close();  }  } |
| Кнопка Выйти | private void Выйти\_Click(object sender, EventArgs e)  {  Hide();  Регистрация UserForm = new Регистрация();  UserForm.Close();} |
| Событие Регистрация\_MouseMove | Point Mouse;  private void Регистрация\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Событие Регистрация\_MouseDown | private void Регистрация\_MouseDown(object sender, MouseEventArgs e)  {  Mouse = new Point(e.X, e.Y);  } |

Свойства элементов формы Администратор представлены в таблице 13.

Таблица 13 – Свойства элементов формы Администратор

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Администратор | FormBorderStyle | None |
| Text | Ремонт пассажирских вагонов |
| StartPosition | CenterScreen |
| Size | 1064; 511 |
| ShowIcon | False |

Событийно – управляемые процедуры в форме Администратор представлены в таблице 14.

Таблица 14 – Событийно-управляемые процедуры в форме «Администратор»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Администратор | public partial class Администратор : Form  {  int s;  public Администратор()  {  InitializeComponent();  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  }  } |
| Событие Просмотр | public void Просмотр(int i)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  switch (i)  {  case 0: Таблица.DataSource = вагоныBindingSource; break;  case 1: Таблица.DataSource = ремонтныеРаботыBindingSource; break;  case 2: Таблица.DataSource = железныеДорогиBindingSource; break;  case 3: Таблица.DataSource = ремонтыBindingSource; break;  case 4: Таблица.DataSource = бригадыBindingSource; break;  case 5: Таблица.DataSource = сотрудникиBindingSource; break;  case 6: Таблица.DataSource = бригадирыBindingSource; break;  case 7: Таблица.DataSource = пользователиBindingSource; break;  }  s = i;  } |
| Кнопка Внести изменения | private void Внести\_Изменения\_Click(object sender, EventArgs e)  {  switch (s)  {  case 0:  вагоныTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  вагоныTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  break;  case 1:  ремонтные\_РаботыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  ремонтные\_РаботыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  break;  case 2:  железные\_ДорогиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  железные\_ДорогиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  break;  case 3:  ремонтыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  ремонтыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  break;  case 4:  бригадыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  бригадыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  break;  case 5:  сотрудникиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  сотрудникиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  break;  case 6:  бригадирыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры);  бригадирыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры);  break;  case 7:  пользователиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи);  пользователиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи);  break;  }  MessageBox.Show("Изменения внесены.", "Уводмление");  } |
| Кнопка Вагоны | private void Вагоны\_Click(object sender, EventArgs e)  {  Просмотр(0);  Фото\_Вагона.Visible = true;  Таблица.Size = new Size(826, 438);  } |
| Кнопка Ремонтные работы | private void Ремонтные\_Работы\_Click(object sender, EventArgs e)  {  Просмотр(1);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Железные дороги | private void Железные\_Дороги\_Click(object sender, EventArgs e)  {  Просмотр(2);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Ремонты | private void Ремонты\_Click(object sender, EventArgs e)  {  Просмотр(3);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);} |
| Кнопка Бригады | private void Бригады\_Click(object sender, EventArgs e)  {  Просмотр(4);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Сотрудники | private void Сотрудники\_Click(object sender, EventArgs e)  {  Просмотр(5);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Бригадиры | private void Бригадиры\_Click(object sender, EventArgs e)  {  Просмотр(6);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Пользователи | private void Пользователи\_Click(object sender, EventArgs e)  {  Просмотр(7);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Выйти | private void Выйти\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка Вернуться | private void Вернуться\_Click(object sender, EventArgs e)  {  Hide();  Авторизация UserForm = new Авторизация();  UserForm.Show();  } |
| Событие Администратор\_Load | private void Администратор\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.  this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.  this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи". При необходимости она может быть перемещена или удалена.  this.пользователиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.  this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.  this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады". При необходимости она может быть перемещена или удалена.  this.бригадыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры". При необходимости она может быть перемещена или удалена.  this.бригадирыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры);  } |
| Событие Поиск\_TextChanged | private void Поиск\_TextChanged(object sender, EventArgs e)  {  for (int i = 0; i < Таблица.RowCount; i++)  {  Таблица.Rows[i].Selected = false;  for (int j = 0; j < Таблица.ColumnCount; j++)  if (Таблица.Rows[i].Cells[j].Value != null)  if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Поиск.Text))  {  Таблица.Rows[i].Selected = true;  break;  }  }  } |
| Событие Администратор\_MouseMove | Point Mouse;  private void Администратор\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Событие Администратор\_MouseDown | private void Администратор\_MouseDown(object sender, MouseEventArgs e)  {  Mouse = new Point(e.X, e.Y);  } |
| Кнопка Выбрать фотографию | private void Выбрать\_Фотографию\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  Фото\_Вагона.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"UPDATE Вагоны SET Фотография = '{load.FileName}' WHERE Код\_Вагона = '{Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  }  } |
| Событие Таблица\_CellClick | string picture;  private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)  {  try  {  picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();  Фото\_Вагона.Image = Image.FromFile(picture);  }  catch { }  } |

Свойства элементов формы Бригадир представлены в таблице 15.

Таблица 15 – Свойства элементов формы Бригадир

| Объект | Свойство | Значение |
| --- | --- | --- |
| Бригадир | FormBorderStyle | None |
| Text | Ремонт пассажирских вагонов |
| StartPosition | CenterScreen |
| Size | 1064; 511 |
| ShowIcon | False |

Событийно – управляемые процедуры в форме Бригадир представлены в таблице 16.

Таблица 16 – Событийно-управляемые процедуры в форме «Бригадир»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Бригадир | public partial class Бригадир : Form  {  int s;  public Бригадир()  {  InitializeComponent();  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  }  } |
| Событие Просмотр | public void Просмотр(int i)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  switch (i)  {  case 0: Таблица.DataSource = вагоныBindingSource; break;  case 1: Таблица.DataSource = ремонтныеРаботыBindingSource; break;  case 2: Таблица.DataSource = железныеДорогиBindingSource; break;  case 3: Таблица.DataSource = ремонтыBindingSource; break;  case 4: Таблица.DataSource = бригадыBindingSource; break;  case 5: Таблица.DataSource = сотрудникиBindingSource; break;  }  s = i;  } |
| Кнопка Внести изменения | private void Внести\_Изменения\_Click(object sender, EventArgs e)  {  switch (s)  {  case 0:  вагоныTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  вагоныTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  break;  case 1:  ремонтные\_РаботыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  ремонтные\_РаботыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  break;  case 2:  железные\_ДорогиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  железные\_ДорогиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  break;  case 3:  ремонтыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  ремонтыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  break;  case 4:  бригадыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  бригадыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  break;  case 5:  сотрудникиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  сотрудникиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  break;  }  MessageBox.Show("Изменения внесены.", "Уводмление");  } |
| Кнопка Вагоны | private void Вагоны\_Click(object sender, EventArgs e)  {  Просмотр(0);  Фото\_Вагона.Visible = true;  Таблица.Size = new Size(826, 438);  } |
| Кнопка Ремонтные работы | private void Ремонтные\_Работы\_Click(object sender, EventArgs e)  {  Просмотр(1);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);} |
| Кнопка Железные дороги | private void Железные\_Дороги\_Click(object sender, EventArgs e)  {  Просмотр(2);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Ремонты | private void Ремонты\_Click(object sender, EventArgs e)  {  Просмотр(3);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Бригады | private void Бригады\_Click(object sender, EventArgs e)  {  Просмотр(4);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Сотрудники | private void Сотрудники\_Click(object sender, EventArgs e)  {  Просмотр(5);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Выйти | private void Выйти\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка Вернуться | private void Вернуться\_Click(object sender, EventArgs e)  {  Hide();  Авторизация UserForm = new Авторизация();  UserForm.Show();  } |
| Событие Бригадир\_Load | private void Бригадир\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады". При необходимости она может быть перемещена или удалена.  this.бригадыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.  this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.  this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.  this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.  this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.  this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады". При необходимости она может быть перемещена или удалена.  this.бригадыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.  this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.  this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.  this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.  this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  } |
| Событие Поиск\_TextChanged | private void Поиск\_TextChanged(object sender, EventArgs e)  {  for (int i = 0; i < Таблица.RowCount; i++)  {  Таблица.Rows[i].Selected = false;  for (int j = 0; j < Таблица.ColumnCount; j++)  if (Таблица.Rows[i].Cells[j].Value != null)  if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Таблица.Text))  {  Таблица.Rows[i].Selected = true;  break;  }  }} |
| Событие Бригадир\_MouseMove | Point Mouse;  private void Бригадир\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Событие Бригадир\_MouseDown | private void Бригадир\_MouseDown(object sender, MouseEventArgs e)  {  Mouse = new Point(e.X, e.Y);  } |
| Кнопка Выбрать фотографию | private void Выбрать\_Фотографию\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  Фото\_Вагона.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"UPDATE Вагоны SET Фотография = '{load.FileName}' WHERE Код\_Вагона = '{Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  }  } |
| Событие Таблица\_CellClick | string picture;  private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)  {  try  {  picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();  Фото\_Вагона.Image = Image.FromFile(picture);  }  catch { }  } |

Свойства элементов формы Сотрудник представлены в таблице 17.

Таблица 17 – Свойства элементов формы Сотрудник

|  |  |  |
| --- | --- | --- |
| Объект | Свойство | Значение |
| Сотрудник | FormBorderStyle | None |
| Text | Ремонт пассажирских вагонов |
| StartPosition | CenterScreen |
| Size | 1064; 511 |
| ShowIcon | False |

Событийно – управляемые процедуры в форме Сотрудник представлены в таблице 18.

Таблица 18 – Событийно-управляемые процедуры в форме «Сотрудник»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Сотрудник | public partial class Сотрудник : Form  {  int s;  public Сотрудник()  {  InitializeComponent();  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  }  } |
| Событие Просмотр | public void Просмотр(int i)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  switch (i)  {  case 0: Таблица.DataSource = вагоныBindingSource; break;  case 1: Таблица.DataSource = ремонтныеРаботыBindingSource; break;  case 2: Таблица.DataSource = железныеДорогиBindingSource; break;  case 3: Таблица.DataSource = ремонтыBindingSource; break;  }  s = i;} |
| Кнопка Внести изменения | private void Внести\_Изменения\_Click(object sender, EventArgs e)  {  switch (s)  {  case 0:  вагоныTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  вагоныTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  break;  case 1:  ремонтные\_РаботыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  ремонтные\_РаботыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  break;  case 2:  железные\_ДорогиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  железные\_ДорогиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  break;  case 3:  ремонтыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  ремонтыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  break;  }  MessageBox.Show("Изменения внесены.", "Уводмление");  } |
| Кнопка Вернуться | private void Вернуться\_Click(object sender, EventArgs e)  {  Hide();  Авторизация UserForm = new Авторизация();  UserForm.Show();  } |
| Кнопка Вагоны | private void Вагоны\_Click(object sender, EventArgs e)  {  Просмотр(0);  Фото\_Вагона.Visible = true;  Таблица.Size = new Size(826, 438);  } |
| Кнопка Ремонтные работы | private void Ремонтные\_Работы\_Click(object sender, EventArgs e)  {  Просмотр(1);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Железные дороги | private void Железные\_Дороги\_Click(object sender, EventArgs e)  {  Просмотр(2);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Ремонты | private void Ремонты\_Click(object sender, EventArgs e)  {  Просмотр(3);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Выйти | private void Выйти\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Событие Сотрудник\_Load | private void Сотрудник\_Load(object sender, EventArgs e)  {  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.  this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.  this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.  this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);  // TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.  this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);  } |
| Событие Поиск\_TextChanged | private void Поиск\_TextChanged(object sender, EventArgs e)  {  for (int i = 0; i < Таблица.RowCount; i++)  {  Таблица.Rows[i].Selected = false;  for (int j = 0; j < Таблица.ColumnCount; j++)  if (Таблица.Rows[i].Cells[j].Value != null)  if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Поиск.Text))  {  Таблица.Rows[i].Selected = true;  break;  }  }  } |
| Событие Сотрудник\_MouseMove | Point Mouse;  private void Сотрудник\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Событие Сотрудник\_MouseDown | private void Сотрудник\_MouseDown(object sender, MouseEventArgs e)  {  Mouse = new Point(e.X, e.Y);  } |
| Кнопка Выбрать фотографию | private void Выбрать\_Фотографию\_Click(object sender, EventArgs e)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  OpenFileDialog load = new OpenFileDialog();  if (load.ShowDialog() == DialogResult.OK)  {  Фото\_Вагона.ImageLocation = load.FileName;  SqlDataAdapter a = new SqlDataAdapter($"UPDATE Вагоны SET Фотография = '{load.FileName}' WHERE Код\_Вагона = '{Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);  DataSet dt = new DataSet();  a.Fill(dt);  }  } |
| Событие Таблица\_CellClick | string picture;  private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)  {  try  {  picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();  Фото\_Вагона.Image = Image.FromFile(picture);  }  catch { }  } |

Свойства элементов формы Гость представлены в таблице 19.

Таблица 19 – Свойства элементов формы Гость

| Объект | Свойство | Значение |
| --- | --- | --- |
| Гость | FormBorderStyle | None |
| Text | Ремонт пассажирских вагонов |
| StartPosition | CenterScreen |
| Size | 1064; 511 |
| ShowIcon | False |

Событийно – управляемые процедуры в форме Гость представлены в таблице 20.

Таблица 20 – Событийно-управляемые процедуры в форме «Гость»

| Объект | Событийно-управляемые процедуры |
| --- | --- |
| Класс Гость | public partial class Гость : Form  {  public Гость()  {  InitializeComponent();  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  }  } |
| Событие Просмотр | public void Просмотр(int i)  {  SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");  sqlConnect.Open();  SqlDataAdapter[] sa = new SqlDataAdapter[3];  sa[0] = new SqlDataAdapter("SELECT \* FROM Вагоны", sqlConnect);  sa[1] = new SqlDataAdapter("SELECT \* FROM Ремонтные\_Работы", sqlConnect);  sa[2] = new SqlDataAdapter("SELECT \* FROM Бригады", sqlConnect);  DataSet ds = new DataSet();  sa[i].Fill(ds);  Таблица.DataSource = ds.Tables[0];  } |
| Кнопка Выйти | private void Выйти\_Click(object sender, EventArgs e)  {  Application.Exit();  } |
| Кнопка Вагоны | private void Вагоны\_Click(object sender, EventArgs e)  {  Просмотр(0);  Фото\_Вагона.Visible = true;  Таблица.Size = new Size(826, 438);  } |
| Кнопка Ремонтные работы | private void Ремонтные\_Работы\_Click(object sender, EventArgs e)  {  Просмотр(1);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Бригады | private void Бригады\_Click(object sender, EventArgs e)  {  Просмотр(2);  Фото\_Вагона.Visible = false;  Таблица.Size = new Size(1041, 438);  } |
| Кнопка Вернуться | private void Вернуться\_Click(object sender, EventArgs e)  {  Hide();  Авторизация UserForm = new Авторизация();  UserForm.Show();} |
| Событие Поиск\_TextChanged | private void Поиск\_TextChanged(object sender, EventArgs e)  {  for (int i = 0; i < Таблица.RowCount; i++)  {  Таблица.Rows[i].Selected = false;  for (int j = 0; j < Таблица.ColumnCount; j++)  if (Таблица.Rows[i].Cells[j].Value != null)  if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Поиск.Text))  {  Таблица.Rows[i].Selected = true;  break;  }}} |
| Событие Гость\_MouseMove | Point Mouse;  private void Гость\_MouseMove(object sender, MouseEventArgs e)  {  if (e.Button == MouseButtons.Left)  {  Left += e.X - Mouse.X;  Top += e.Y - Mouse.Y;  }  } |
| Событие Гость\_MouseDown | private void Гость\_MouseDown(object sender, MouseEventArgs e)  {  Mouse = new Point(e.X, e.Y);  } |
| Событие Таблица\_CellClick | string picture;  private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)  {  try  {  picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();  Фото\_Вагона.Image = Image.FromFile(picture);  }  catch { }  } |

Вывод по главе: была полностью спроектирована и разработана система базы данных, были полностью спроектированы и разработаны все программные модули информационной системы “Ремонт пассажирских вагонов”.

# РАЗРАБОТКА ТЕСТОВЫХ НАБОРОВ И ТЕСТОВЫХ СЦЕНАРИЕВ

Тестирование программного обеспечения – процесс исследования, испытания программного продукта, имеющий своей целью проверку соответствия между реальным поведением программы и её ожидаемым поведением на конечном наборе тестов, выбранных определённым образом.

Тестовый сценарий – это неавтоматизированный или автоматизированный сценарий, содержащий инструкции по реализации тестового набора. Тестовый сценарий может быть написан вручную (для выполнения человеком) либо полностью или частично автоматизирован.

## Назначение эксперимента. Выбор и обоснование методики проведения тестирования

В данном разделе проведены два вида тестирования. Провести анализ тестирования и выбрать, как будет проходить тестирование ИС.

Модульное тестирование. Цель: проверить, что код работает именно так, как должен (при заданных входных параметрах выдает предполагаемый результат).

Анализ покрытия кода тестами. Цель: проверить, что весь наш код отрабатывает при модульном тестировании, что нет, не участвующих в тестировании участков кода.

Анализ стилистики кода. Цель: проверка кода на соблюдение стандартов разработки (отраслевых, корпоративных, принятых на проекте).

Анализ производительности. Цель: проанализировать производительность ключевых операций, потребление памяти, утечку памяти.

## Технология тестирования

Рассмотрим, какие технологии можно применить для модульного тестирования и покрытия кода тестами.

Как в ходе выполнения проекта, так и при развитии системы, изменения в ее алгоритмы вносятся в соответствии с техническими заданиями, хотя данный документ может иметь различные названия. В технических заданиях, как правило, описываются критерии приемки: как мы и заказчик убедимся в том, что заказанный функционал работает верно.

В качестве примера можно привести такой упрощенный критерий:

* Исходные данные: в систему внесен вагон, которому, например, нужно изменить покраску.
* Действия пользователя: пользователь ищет в таблице, какой это тип ремонта, сколько он будет стоить и т п.
* В результате в результате отображена сумма реализации, например в 5000 руб.
* Вручную такое тестирование, как правило, выполняется единожды, поскольку выполнять каждый раз полный комплект тестов очень трудоемко.
* Нет гарантии того, что тесты, описанные в техническом задании, покрывают функциональность полностью. Т.е. даже если тесты описаны и проводятся регулярно, вероятно будет существовать наличие не протестированного функционала.

Автоматизированное тестирование лишено данных недостатков. Конечно, возникает отдельная задача кодирования данного теста на основании критериев приемки, но это скорее дисциплинирующий фактор…

Прежде чем начинать юзабилити-тестирование необходимо собрать группу пользователей, которые будут тестировать данную информационную систему. Количество привлеченных пользователей должно быть не менее пяти человек. Пользователям будет представлен список некоторых вопросов, относящихся к графическому интерфейсу.

В таблице 21 представлены сведения о пользователях.

Таблица 21 – Информация о пользователях

|  |  |  |  |
| --- | --- | --- | --- |
| ФИО | Пол | Возраст | Образование |
| Аверьянов Николай Владимирович | Мужской | 24 | Среднее специальное |
| Туркина Надежда Михайловна | Женский | 34 | Среднее специальное |
| Окладников Антон Александрович | Мужской | 30 | Среднее специальное |
| Петрова Наталья Владимировна | Женский | 35 | Среднее специальное |
| Фурман Ксения Михайловна | Женский | 24 | Среднее специальное |

## Результаты проведения тестирования

В ходе тестирования был применен метод нагрузочного тестирования и юзабилити – тестирования, которые проверяют работу при нагрузке системы и сделать простым и удобным в пользовании информационной системы.

Суть тестирования заключается в проверке работы всех обязательных функций.

Результаты проведения нагрузочного тестирования

Для проведения нагрузочного тестирования составлена таблица 22, в которой указано количество запущенных копий программного продукта, нагрузка на процесс и оперативную память.

Таблица 22 – Результаты нагрузочного тестирования

|  |  |  |
| --- | --- | --- |
| Число копий | Нагрузка на процессор (%) | Нагрузка на память (%) |
| 1 | 0,1 | 0,2 |
| 5 | 0,2 | 0,10 |
| 10 | 0,3 | 0,20 |
| 15 | 0,4 | 0,30 |
| 20 | 0,5 | 0,40 |

Исходя из результатов нагрузочного тестирования, можно прийти к выводу, что нагрузка, оказываемая на центральный процессор и память, не имеет критичных значений. Это значит, что показатель производительности информационной системы на приемлемом уровне, то есть сбои и длительная загрузка не должны помешать пользователю при работе с данным программным продуктом.

Таблица 23 – Результаты тестирования

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Вопрос | Аверьянов Николай Владимирович | Туркина Надежда Михайловна | Окладников Антон Александрович | Петрова Наталья Владимировна | Фурман Ксения Михайловна |
| Присутствуют ли дизайн элементы, которые неприятны для зрения? | нет | нет | нет | нет | нет |
| Присутствует ли отвлекающая информация? | нет | нет | нет | нет | нет |
| Понятна ли структура приложения на интуитивном уровне? | да | да | да | да | да |
| Смогли ли вы найти интересующую вас информацию без затруднений? | да | да | да | нет | да |
| Понятен ли смысл приложения? | да | да | да | да | да |
| Общая оценка | 7/10 | 9/10 | 8/10 | 7/10 | 7/10 |

Исходя из средней оценки информационной системы, рассчитанной на основе общих оценок всех пользователей с учетом ответов на все поставленные вопросы, можно сделать вывод о том, что внешние характеристики графического интерфейса находятся на довольно высоком уровне.

Вывод по главе: были проведены тесты путём проведения модульного тестирования, анализа покрытия кода тестами, анализа стилистики кода и анализом производительности. Также было проведено нагрузочное тестирование, в результате чего нагрузка на процессор и память не имела критичных значений.

# ЗАКЛЮЧЕНИЕ

В результате учебной практики был проведен анализ предметной области, спроектирована и разработана информационная система «Ремонт пассажирских вагонов», серверная часть которой была реализована в MS SQL, а клиентская часть в MS Visual Studio.

В MS SQL была создана база данных для хранения информации о вагонах, персонале, должностях, видов ремонта и о самом ремонте вагонов.

Клиентская часть, реализованная в MS Visual Studio, отражает все хранящиеся в базе данных таблицы. Также разработаны некоторые запросы, позволяющие редактировать и удалять существующие данные, добавлять новые и осуществлять поиск по таблицам.

Цели учебной практики были достигнуты путем проведения анализа предметной области, проектирования информационной системы и разработки программных модулей.

Это позволяет сделать вывод, что поставленные цели и задачи выполнены в полном объеме.

В дальнейшем данная информационная система может быть модернизирована путем добавления формы личного кабинета, создания версии для слабовидящих людей или улучшения и изменения дизайна.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения [Электронный доступ]. — Режим доступа: [ГОСТ 20886-85. Организация данных в системах обработки данных. Термины и определения (rostest.info)](https://rostest.info/gost/001.001.040.035/gost-20886-85/#:~:text=%D0%93%D0%9E%D0%A1%D0%A2%2020886%2D85.%20%D0%9E%D1%80%D0%B3%D0%B0%D0%BD%D0%B8%D0%B7%D0%B0%D1%86%D0%B8%D1%8F%20%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85%20%D0%B2,%D0%BD%D0%B0%D1%83%D1%87%D0%BD%D0%BE%2D%D1%82%D0%B5%D1%85%D0%BD%D0%B8%D1%87%D0%B5%D1%81%D0%) (Дата обращения: 22.05.2013 г.).
2. ГОСТ Р. 59793–2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания [Электронный доступ]. — Режим доступа: [ГОСТ Р. 59793-2021. Информационные технологии. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания (internet-law.ru)](https://internet-law.ru/gosts/gost/77858/?ysclid=lifap277mg519765664) (Дата обращения: 30.04.2022 г.).
3. ГОСТ 19.201-78. ЕСПД. Общие положения. [Электронный доступ]. — Режим доступа: [ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению (swrit.ru)](https://www.swrit.ru/doc/espd/19.201-78.pdf?ysclid=lpi5u6naom706376064) (Дата обращения: 01.10.1981 г.).
4. Роберт Э. Уолтерс, Майкл Коулс. SQL Server 2008: ускоренный курс для профессионалов = Accelerated SQL Server 2008. — М: [«Вильямс»](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2008. — С. 768.
5. Microsoft SQL Server [Электронный доступ]. — Режим доступа: [Microsoft SQL Server — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_SQL_Server) (Дата обращения: 10.09.2023 г.).
6. Дебби Валковски. Microsoft Office Visio 2003 для «чайников» = Visio 2003 For Dummies. — М: [«Диалектика»](https://ru.wikipedia.org/w/index.php?title=%D0%94%D0%B8%D0%B0%D0%BB%D0%B5%D0%BA%D1%82%D0%B8%D0%BA%D0%B0_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2006. — С. 336.
7. Microsoft Visio [Электронный доступ]. — Режим доступа: [Microsoft Visio — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/Microsoft_Visio) (Дата обращения: 07.08.2023 г.).
8. Майо Д. Самоучитель Microsoft Visual Studio 2010 = Microsoft Visual Studio 2010: A Beginner's Guide (A Beginners Guide). — C: [«БХВ-Петербург»](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2010. — С. 464.
9. Microsoft Visual Studio [Электронный доступ]. — Режим доступа: [Руководство по разработке модулей расширений на C# — Павел Еремеев (pvs-studio.ru)](https://pvs-studio.ru/ru/blog/posts/csharp/a0082/) (Дата обращения: 29.08.2013 г.).
10. Microsoft Visual Studio [Электронный доступ]. — Режим доступа: [Visual Studio — Kalirgum (habr.com)](https://habr.com/ru/hubs/vs/articles/) (Дата обращения: 28.10.2023 г.).
11. Пауэрс Л., Снелл М. Microsoft Visual Studio 2008 = Microsoft Visual Studio 2008 Unleashed by Lars Powers and Mike Snell. — C: [«БХВ-Петербург»](https://ru.wikipedia.org/w/index.php?title=%D0%91%D0%A5%D0%92-%D0%9F%D0%B5%D1%82%D0%B5%D1%80%D0%B1%D1%83%D1%80%D0%B3_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2008. — С. 1200.
12. Крис Фиайли. SQL: Руководство по изучению языка. — М: Peachpit Press, 2003. — C. 456.
13. [К. Дж. Дейт](https://ru.wikipedia.org/wiki/%D0%94%D0%B5%D0%B9%D1%82,_%D0%9A%D1%80%D0%B8%D1%81%D1%82%D0%BE%D1%84%D0%B5%D1%80). [Введение в системы баз данных](https://ru.wikipedia.org/wiki/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5_%D0%B2_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D1%8B_%D0%B1%D0%B0%D0%B7_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) / Пер. с англ. — 8-е изд. — М: [Вильямс](https://ru.wikipedia.org/w/index.php?title=%D0%92%D0%B8%D0%BB%D1%8C%D1%8F%D0%BC%D1%81_(%D0%B8%D0%B7%D0%B4%D0%B0%D1%82%D0%B5%D0%BB%D1%8C%D1%81%D1%82%D0%B2%D0%BE)&action=edit&redlink=1), 2005. — C. 1328.
14. Microsoft SQL Server [Электронный доступ]. — Режим доступа: [Как устроен язык SQL и почему он так востребован — Анализ данных (practicum.yandex.ru)](https://practicum.yandex.ru/blog/chto-takoe-sql/) (Дата обращения: 27.07.2022 г.).
15. Microsoft Visual Studio [Электронный доступ]. — Режим доступа: [Руководство. Создание приложения — Microsoft (microsoft.com)](https://learn.microsoft.com/ru-ru/visualstudio/ide/walkthrough-building-an-application?view=vs-2022) (Дата обращения: 02.11.2023 г.).

# ПРИЛОЖЕНИЯ А

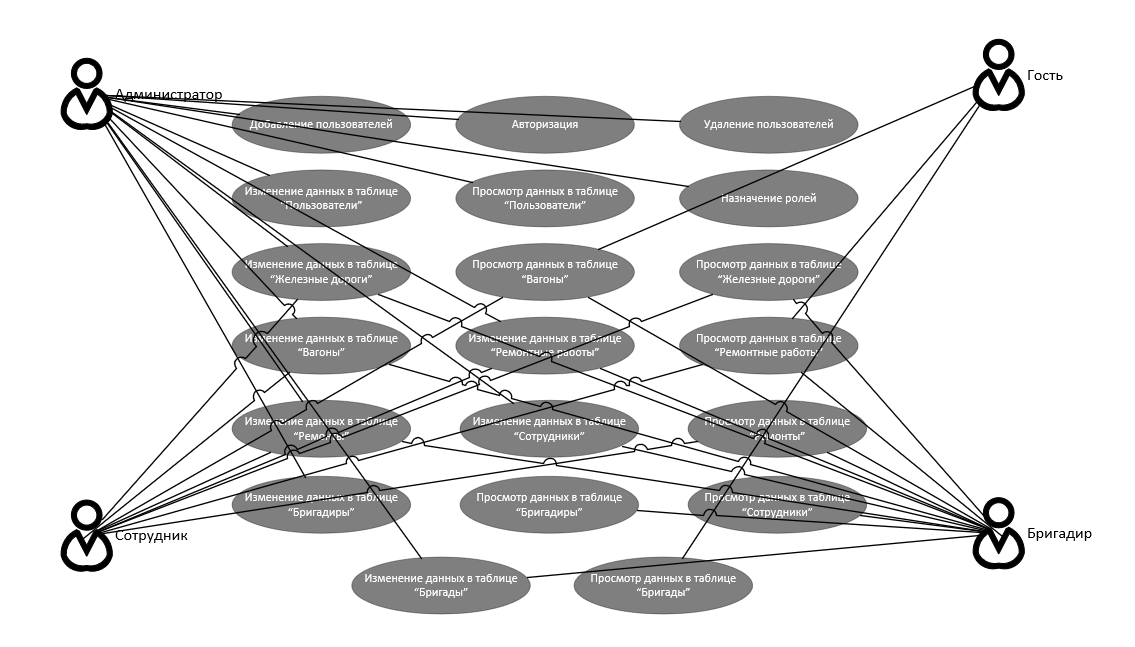


Рисунок 1 – Диаграмма вариантов использования



Рисунок 2 – Диаграмма деятельности (Администратор)

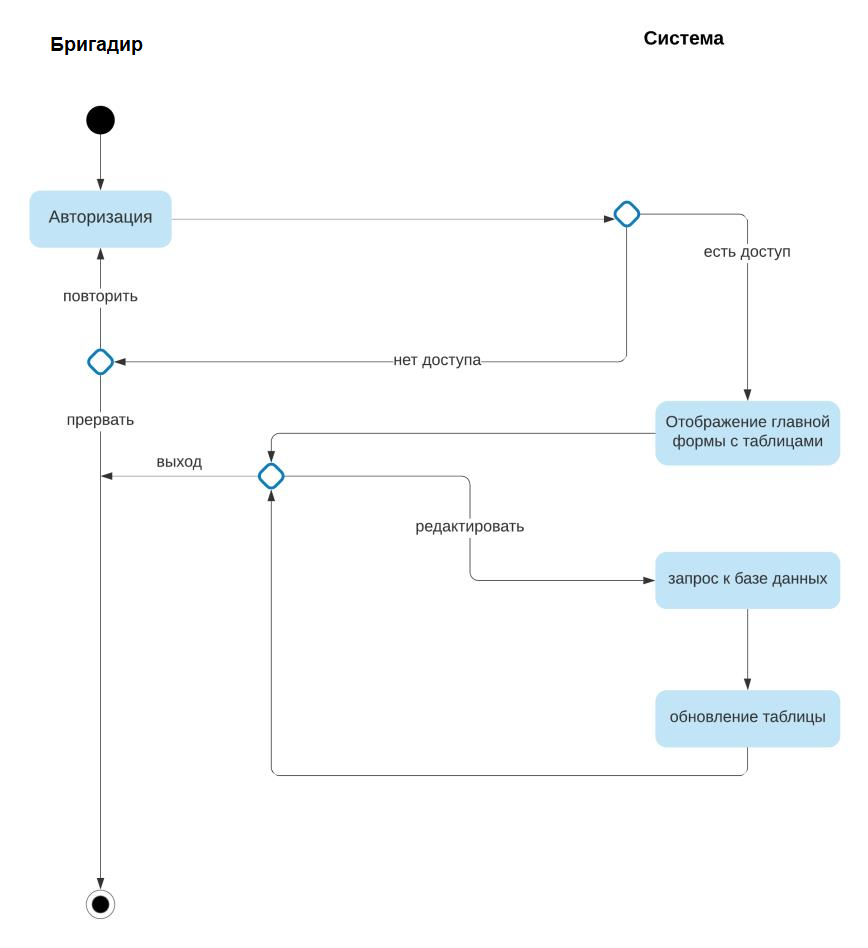


Рисунок 3 – Диаграмма деятельности (Бригадир)

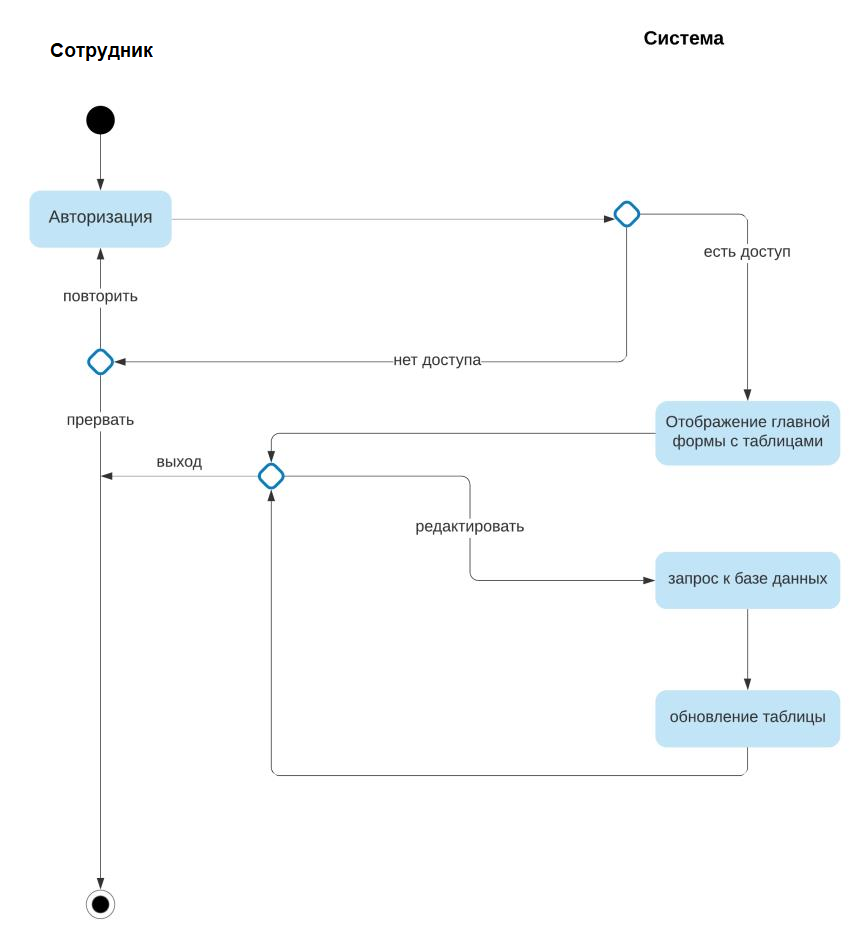


Рисунок 4 – Диаграмма деятельности (Сотрудник)



Рисунок 5 – Диаграмма деятельности (Гость)

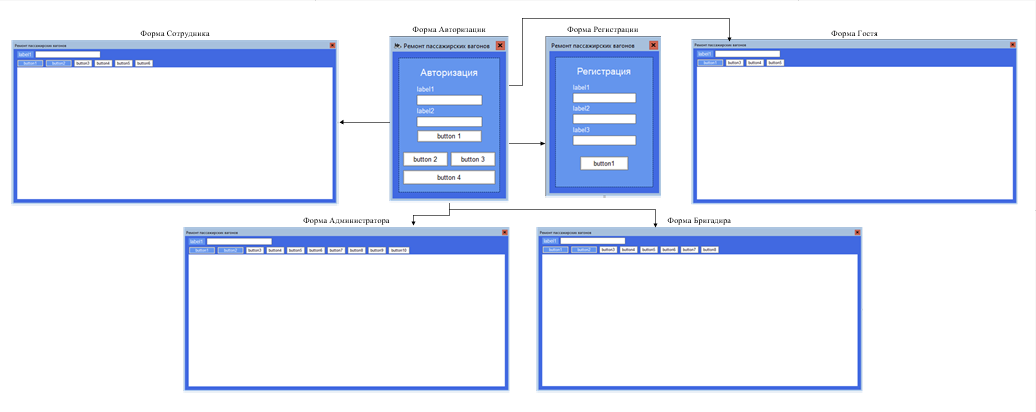


Рисунок 6 – Карта переходов

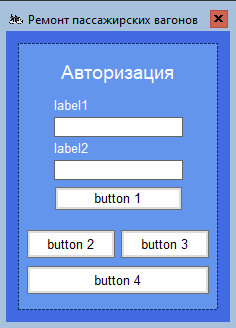


Рисунок 7 – Макет Авторизации

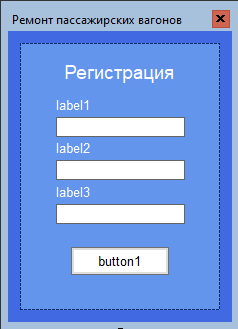


Рисунок 8 – Макет Регистрации



Рисунок 9 – Макет Администратора



Рисунок 10 – Макет Бригадира

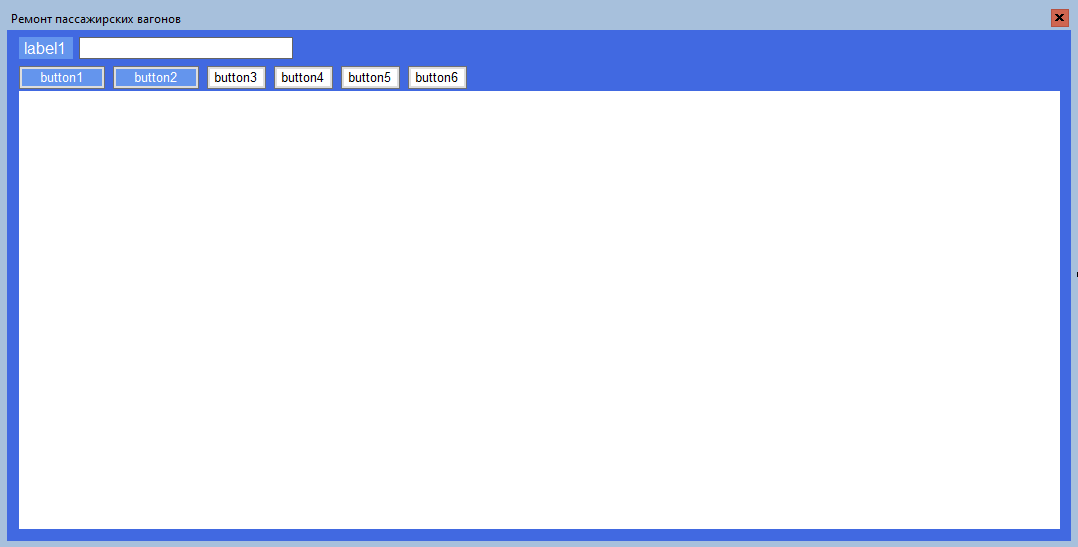


Рисунок 11 – Макет Сотрудника

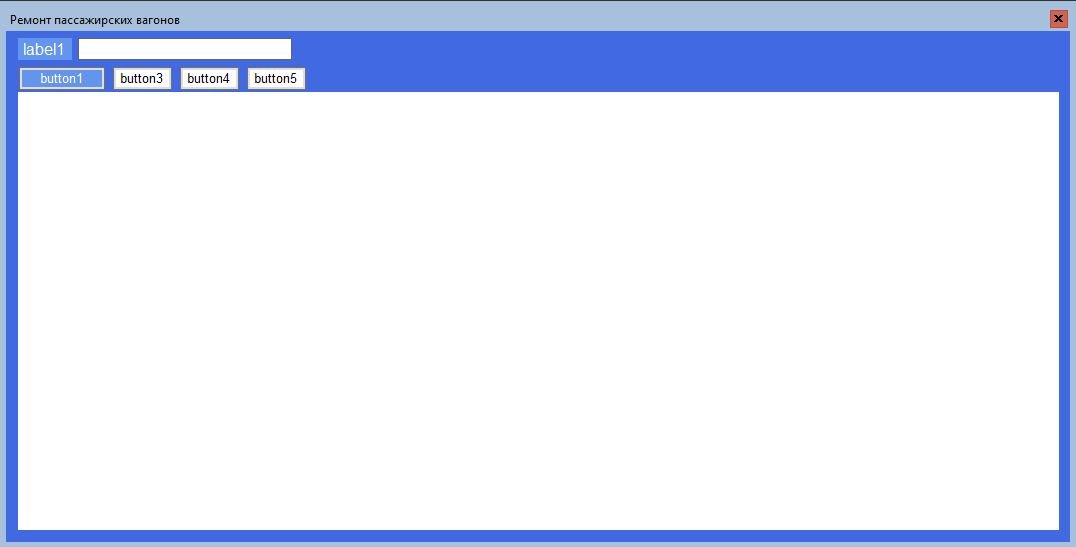


Рисунок 12 – Макет Гостя

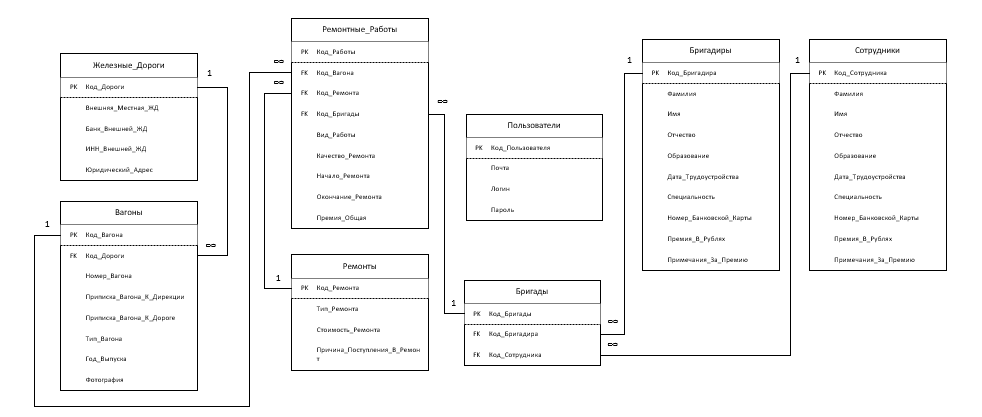


Рисунок 13 – Логическая модель данных

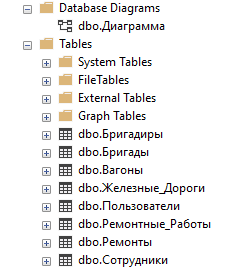


Рисунок 14 – Созданные таблицы

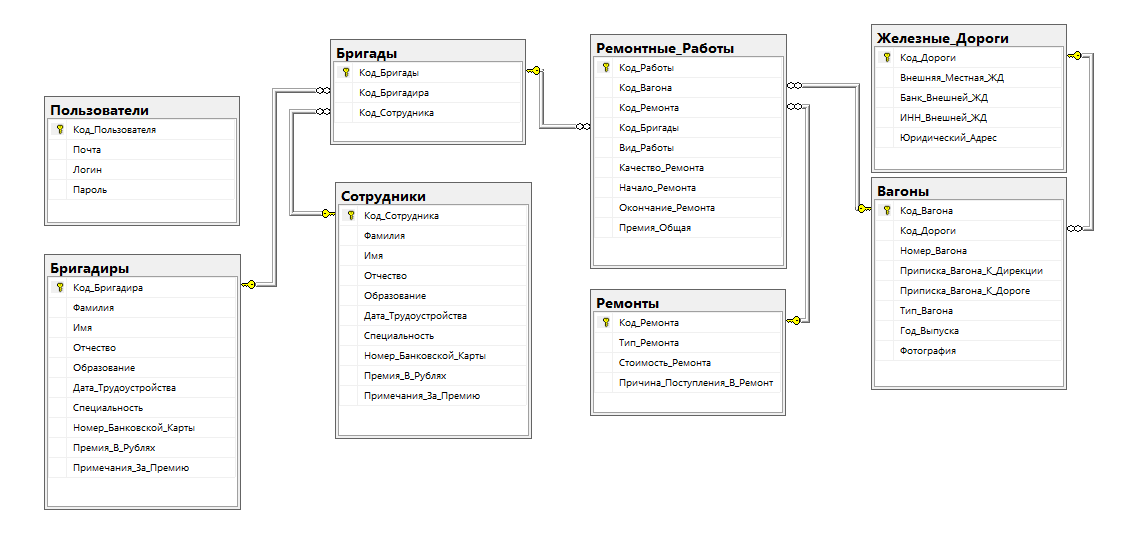


Рисунок 15 – Физическая модель базы данных

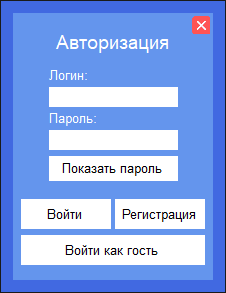


Рисунок 16 – Форма “Авторизация”

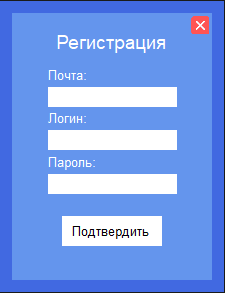


Рисунок 17 – Форма “Регистрация”

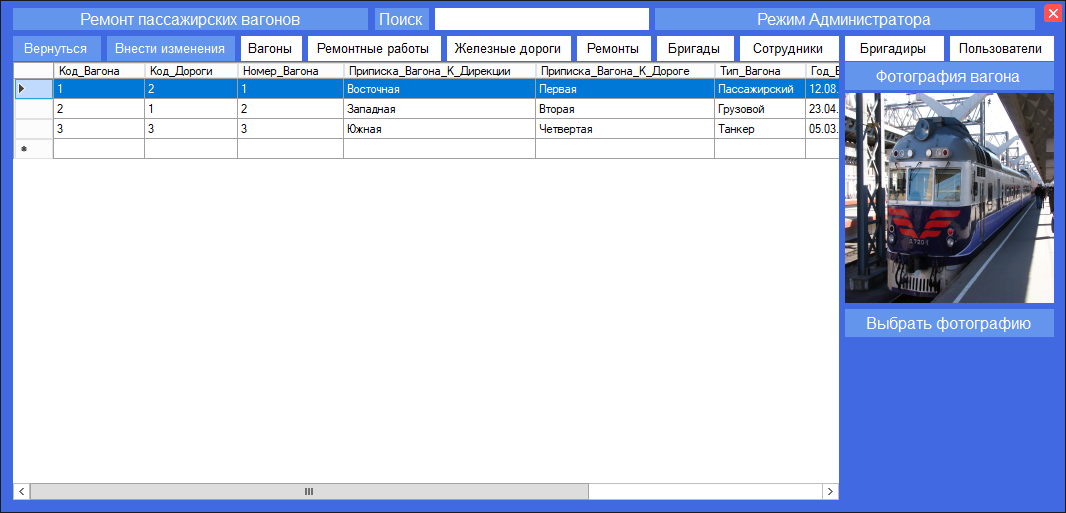


Рисунок 18 – Форма “Администратора”

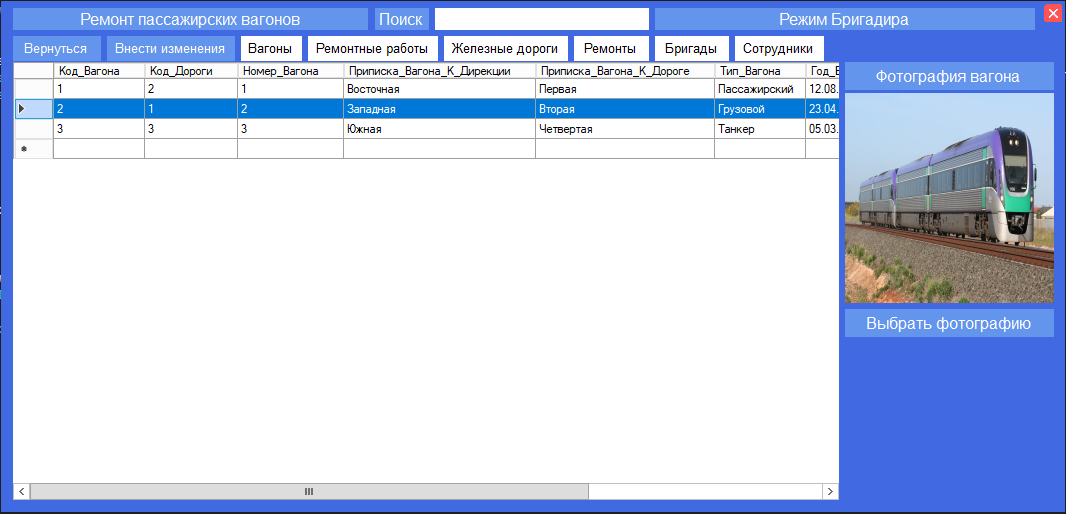
****

Рисунок 19 – Форма “Бригадира”

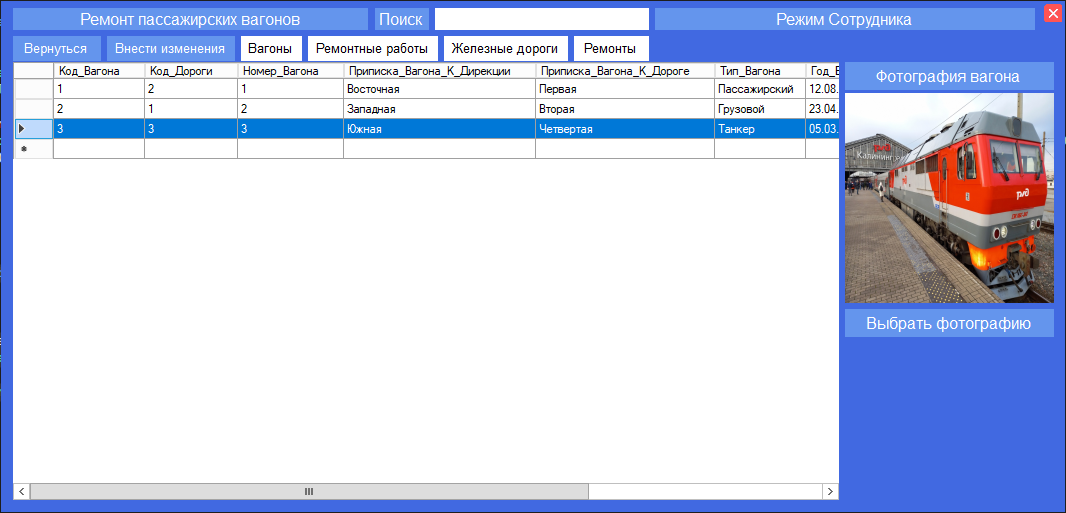
****

Рисунок 20 – Форма “Сотрудника”

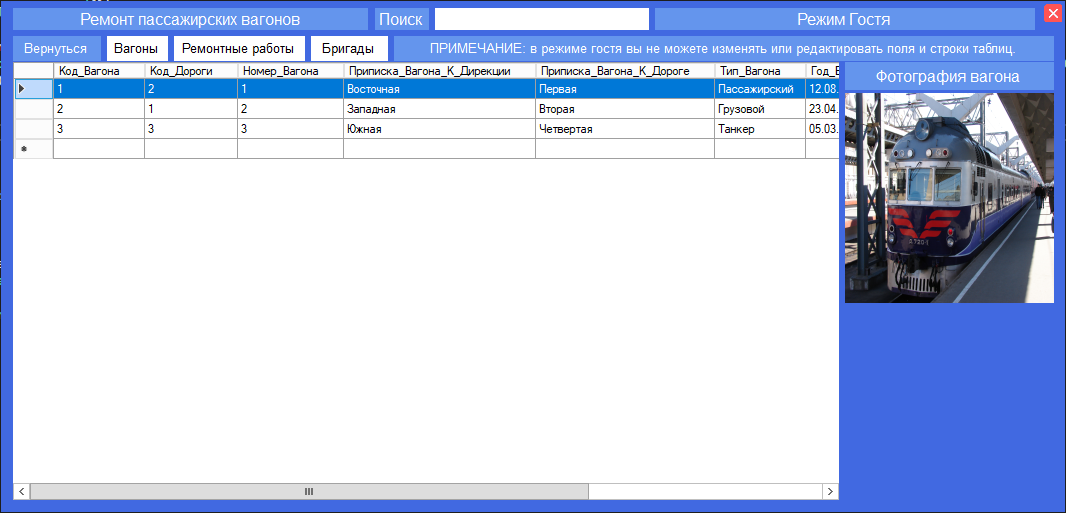
****

Рисунок 21 – Форма “Гостя”

# ПРИЛОЖЕНИЯ Б

Листинг кода формы “Авторизация”:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace Ремонт\_пассажирских\_вагонов

{

public partial class Авторизация : Form

{

public Авторизация()

{

InitializeComponent();

}

private void Логин\_KeyPress(object sender, KeyPressEventArgs e)

{

Логин.MaxLength = 30;

if (char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать цифры!", "Ошибка");

}

}

private void Пароль\_KeyPress(object sender, KeyPressEventArgs e)

{

Пароль.MaxLength = 15;

if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать буквы!", "Ошибка");

}

}

private void Войти\_Click(object sender, EventArgs e)

{

try

{

if ((Логин.Text == "") || (Пароль.Text == ""))

{

MessageBox.Show("Поля не должны быть пустыми.", "Ошибка");

}

else

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter da = new SqlDataAdapter("select \* from Пользователи", sqlConnect);

DataTable dt = new DataTable();

da.Fill(dt);

int flag = -1;

for (int i = 0; i < dt.Rows.Count; i++)

{

if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Админ") && (dt.Rows[i]["Логин"].ToString() == Логин.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль.Text))

flag = 0;

if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Сотрудник") && (dt.Rows[i]["Логин"].ToString() == Логин.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль.Text))

{

flag = 1;

}

if ((dt.Rows[i]["Права\_Доступа"].ToString() == "Бригадир") && (dt.Rows[i]["Логин"].ToString() == Логин.Text) && (dt.Rows[i]["Пароль"].ToString() == Пароль.Text))

{

flag = 2;

}

}

if ((flag == 0) || (flag == 1) || (flag == 2))

{

if (flag == 0)

{

Hide();

Администратор UserForm1 = new Администратор();

UserForm1.Show();

}

if (flag == 1)

{

Hide();

Сотрудник UserForm2 = new Сотрудник();

UserForm2.Show();

}

if (flag == 2)

{

Hide();

Бригадир UserForm3 = new Бригадир();

UserForm3.Show();

}

}

else

MessageBox.Show("Пересоздайте пользователя! Такого пользователя не существует.", "Ошибка");

}

}

catch { }

}

private void Регистрация\_Click(object sender, EventArgs e)

{

Регистрация UserForm4 = new Регистрация();

UserForm4.Show();

}

private void Войти\_Как\_Гость\_Click(object sender, EventArgs e)

{

Hide();

Гость UserForm = new Гость();

UserForm.Show();

}

private void Показать\_Пароль\_Click(object sender, EventArgs e)

{

if (Показать\_Пароль.Text == "Показать пароль")

{

Пароль.PasswordChar = '\0';

Показать\_Пароль.Text = "Скрыть пароль";

}

else if (Показать\_Пароль.Text == "Скрыть пароль")

{

Пароль.PasswordChar = '\*';

Показать\_Пароль.Text = "Показать пароль";

}

}

private void Выйти\_Click(object sender, EventArgs e)

{

Application.Exit();

}

Point Mouse;

private void Авторизация\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

Left += e.X - Mouse.X;

Top += e.Y - Mouse.Y;

}

}

private void Авторизация\_MouseDown(object sender, MouseEventArgs e)

{

Mouse = new Point(e.X, e.Y);

}

}

}

Листинг кода формы “Регистрация”:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace Ремонт\_пассажирских\_вагонов

{

public partial class Регистрация : Form

{

public Регистрация()

{

InitializeComponent();

}

private void Почта\_KeyPress(object sender, KeyPressEventArgs e)

{

Почта.MaxLength = 30;

if (char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать цифры!", "Ошибка");

}

}

private void Логин\_KeyPress(object sender, KeyPressEventArgs e)

{

Логин.MaxLength = 30;

if (char.IsDigit(e.KeyChar))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать цифры!", "Ошибка");

}

}

private void Пароль\_KeyPress(object sender, KeyPressEventArgs e)

{

Пароль.MaxLength = 15;

if (!char.IsDigit(e.KeyChar) && (e.KeyChar != (char)Keys.Back))

{

e.Handled = true;

MessageBox.Show("Поле не может содержать буквы!", "Ошибка");

}

}

private void Подтвердить\_Click(object sender, EventArgs e)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

if ((Почта.Text == "") || (Логин.Text == "") || (Пароль.Text == ""))

{

MessageBox.Show("Поля не должны быть пустыми.", "Ошибка");

}

else

{

SqlDataAdapter da = new SqlDataAdapter($"INSERT INTO Пользователи (Почта, Логин, Пароль) VALUES ('{Почта.Text}', '{Логин.Text}', '{Пароль.Text}');", sqlConnect);

DataTable dt = new DataTable();

da.Fill(dt);

MessageBox.Show("Вы зарегистрированы, как сотрудник.", "Уведомление");

Hide();

Сотрудник UserForm = new Сотрудник();

UserForm.Close();

}

}

private void Выйти\_Click(object sender, EventArgs e)

{

Hide();

Регистрация UserForm = new Регистрация();

UserForm.Close();

}

Point Mouse;

private void Регистрация\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

Left += e.X - Mouse.X;

Top += e.Y - Mouse.Y;

}

}

private void Регистрация\_MouseDown(object sender, MouseEventArgs e)

{

Mouse = new Point(e.X, e.Y);

}

}

}

Листинг кода формы “Администратор”:

using System;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

using System.Data;

namespace Ремонт\_пассажирских\_вагонов

{

public partial class Администратор : Form

{

int s;

public Администратор()

{

InitializeComponent();

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

public void Просмотр(int i)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

switch (i)

{

case 0: Таблица.DataSource = вагоныBindingSource; break;

case 1: Таблица.DataSource = ремонтныеРаботыBindingSource; break;

case 2: Таблица.DataSource = железныеДорогиBindingSource; break;

case 3: Таблица.DataSource = ремонтыBindingSource; break;

case 4: Таблица.DataSource = бригадыBindingSource; break;

case 5: Таблица.DataSource = сотрудникиBindingSource; break;

case 6: Таблица.DataSource = бригадирыBindingSource; break;

case 7: Таблица.DataSource = пользователиBindingSource; break;

}

s = i;

}

private void Внести\_Изменения\_Click(object sender, EventArgs e)

{

switch (s)

{

case 0:

вагоныTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

вагоныTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

break;

case 1:

ремонтные\_РаботыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

ремонтные\_РаботыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

break;

case 2:

железные\_ДорогиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

железные\_ДорогиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

break;

case 3:

ремонтыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

ремонтыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

break;

case 4:

бригадыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

бригадыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

break;

case 5:

сотрудникиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

сотрудникиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

break;

case 6:

бригадирыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры);

бригадирыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры);

break;

case 7:

пользователиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи);

пользователиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи);

break;

}

MessageBox.Show("Данные внесены.", "Уводмление");

}

private void Вагоны\_Click(object sender, EventArgs e)

{

Просмотр(0);

Фото\_Вагона.Visible = true;

Таблица.Size = new Size(826, 438);

}

private void Ремонтные\_Работы\_Click(object sender, EventArgs e)

{

Просмотр(1);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Железные\_Дороги\_Click(object sender, EventArgs e)

{

Просмотр(2);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Ремонты\_Click(object sender, EventArgs e)

{

Просмотр(3);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Бригады\_Click(object sender, EventArgs e)

{

Просмотр(4);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Сотрудники\_Click(object sender, EventArgs e)

{

Просмотр(5);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Бригадиры\_Click(object sender, EventArgs e)

{

Просмотр(6);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Пользователи\_Click(object sender, EventArgs e)

{

Просмотр(7);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Выйти\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void Вернуться\_Click(object sender, EventArgs e)

{

Hide();

Авторизация UserForm = new Авторизация();

UserForm.Show();

}

private void Поиск\_TextChanged(object sender, EventArgs e)

{

for (int i = 0; i < Таблица.RowCount; i++)

{

Таблица.Rows[i].Selected = false;

for (int j = 0; j < Таблица.ColumnCount; j++)

if (Таблица.Rows[i].Cells[j].Value != null)

if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Поиск.Text))

{

Таблица.Rows[i].Selected = true;

break;

}

}

}

Point Mouse;

private void Администратор\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

Left += e.X - Mouse.X;

Top += e.Y - Mouse.Y;

}

}

private void Администратор\_MouseDown(object sender, MouseEventArgs e)

{

Mouse = new Point(e.X, e.Y);

}

private void Администратор\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.

this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.

this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.

this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи". При необходимости она может быть перемещена или удалена.

this.пользователиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Пользователи);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.

this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.

this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады". При необходимости она может быть перемещена или удалена.

this.бригадыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры". При необходимости она может быть перемещена или удалена.

this.бригадирыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригадиры);

}

private void Выбрать\_Фотографию\_Click(object sender, EventArgs e)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

Фото\_Вагона.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"UPDATE Вагоны SET Фотография = '{load.FileName}' WHERE Код\_Вагона = '{Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

}

}

string picture;

private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();

Фото\_Вагона.Image = Image.FromFile(picture);

}

catch { }

}

}

}

Листинг кода формы “Бригадир”:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace Ремонт\_пассажирских\_вагонов

{

public partial class Бригадир : Form

{

int s;

public Бригадир()

{

InitializeComponent();

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

public void Просмотр(int i)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

switch (i)

{

case 0: Таблица.DataSource = вагоныBindingSource; break;

case 1: Таблица.DataSource = ремонтныеРаботыBindingSource; break;

case 2: Таблица.DataSource = железныеДорогиBindingSource; break;

case 3: Таблица.DataSource = ремонтыBindingSource; break;

case 4: Таблица.DataSource = бригадыBindingSource; break;

case 5: Таблица.DataSource = сотрудникиBindingSource; break;

}

s = i;

}

private void Внести\_Изменения\_Click(object sender, EventArgs e)

{

switch (s)

{

case 0:

вагоныTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

вагоныTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

break;

case 1:

ремонтные\_РаботыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

ремонтные\_РаботыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

break;

case 2:

железные\_ДорогиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

железные\_ДорогиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

break;

case 3:

ремонтыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

ремонтыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

break;

case 4:

бригадыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

бригадыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

break;

case 5:

сотрудникиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

сотрудникиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

break;

}

MessageBox.Show("Данные внесены.", "Уводмление");

}

private void Вагоны\_Click(object sender, EventArgs e)

{

Просмотр(0);

Фото\_Вагона.Visible = true;

Таблица.Size = new Size(826, 438);

}

private void Ремонтные\_Работы\_Click(object sender, EventArgs e)

{

Просмотр(1);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Железные\_Дороги\_Click(object sender, EventArgs e)

{

Просмотр(2);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Ремонты\_Click(object sender, EventArgs e)

{

Просмотр(3);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Бригады\_Click(object sender, EventArgs e)

{

Просмотр(4);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Сотрудники\_Click(object sender, EventArgs e)

{

Просмотр(5);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Выйти\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void Вернуться\_Click(object sender, EventArgs e)

{

Hide();

Авторизация UserForm = new Авторизация();

UserForm.Show();

}

private void Бригадир\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.

this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады". При необходимости она может быть перемещена или удалена.

this.бригадыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.

this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.

this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.

this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.

this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники". При необходимости она может быть перемещена или удалена.

this.сотрудникиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Сотрудники);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады". При необходимости она может быть перемещена или удалена.

this.бригадыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Бригады);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.

this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.

this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.

this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.

this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

}

private void Поиск\_TextChanged(object sender, EventArgs e)

{

for (int i = 0; i < Таблица.RowCount; i++)

{

Таблица.Rows[i].Selected = false;

for (int j = 0; j < Таблица.ColumnCount; j++)

if (Таблица.Rows[i].Cells[j].Value != null)

if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Таблица.Text))

{

Таблица.Rows[i].Selected = true;

break;

}

}

}

Point Mouse;

private void Бригадир\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

Left += e.X - Mouse.X;

Top += e.Y - Mouse.Y;

}

}

private void Бригадир\_MouseDown(object sender, MouseEventArgs e)

{

Mouse = new Point(e.X, e.Y);

}

private void Выбрать\_Фотографию\_Click(object sender, EventArgs e)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

Фото\_Вагона.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"UPDATE Вагоны SET Фотография = '{load.FileName}' WHERE Код\_Вагона = '{Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

}

}

string picture;

private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();

Фото\_Вагона.Image = Image.FromFile(picture);

}

catch { }

}

}

}

Листинг кода формы “Сотрудник”:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace Ремонт\_пассажирских\_вагонов

{

public partial class Сотрудник : Form

{

int s;

public Сотрудник()

{

InitializeComponent();

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

public void Просмотр(int i)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

switch (i)

{

case 0: Таблица.DataSource = вагоныBindingSource; break;

case 1: Таблица.DataSource = ремонтныеРаботыBindingSource; break;

case 2: Таблица.DataSource = железныеДорогиBindingSource; break;

case 3: Таблица.DataSource = ремонтыBindingSource; break;

}

s = i;

}

private void Внести\_Изменения\_Click(object sender, EventArgs e)

{

switch (s)

{

case 0:

вагоныTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

вагоныTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

break;

case 1:

ремонтные\_РаботыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

ремонтные\_РаботыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

break;

case 2:

железные\_ДорогиTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

железные\_ДорогиTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

break;

case 3:

ремонтыTableAdapter.Update(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

ремонтыTableAdapter.Fill(уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

break;

}

MessageBox.Show("Данные внесены.", "Уводмление");

}

private void Вернуться\_Click(object sender, EventArgs e)

{

Hide();

Авторизация UserForm = new Авторизация();

UserForm.Show();

}

private void Вагоны\_Click(object sender, EventArgs e)

{

Просмотр(0);

Фото\_Вагона.Visible = true;

Таблица.Size = new Size(826, 438);

}

private void Ремонтные\_Работы\_Click(object sender, EventArgs e)

{

Просмотр(1);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Железные\_Дороги\_Click(object sender, EventArgs e)

{

Просмотр(2);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Ремонты\_Click(object sender, EventArgs e)

{

Просмотр(3);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Выйти\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void Поиск\_TextChanged(object sender, EventArgs e)

{

for (int i = 0; i < Таблица.RowCount; i++)

{

Таблица.Rows[i].Selected = false;

for (int j = 0; j < Таблица.ColumnCount; j++)

if (Таблица.Rows[i].Cells[j].Value != null)

if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Поиск.Text))

{

Таблица.Rows[i].Selected = true;

break;

}

}

}

Point Mouse;

private void Сотрудник\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

Left += e.X - Mouse.X;

Top += e.Y - Mouse.Y;

}

}

private void Сотрудник\_MouseDown(object sender, MouseEventArgs e)

{

Mouse = new Point(e.X, e.Y);

}

private void Сотрудник\_Load(object sender, EventArgs e)

{

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты". При необходимости она может быть перемещена или удалена.

this.ремонтыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонты);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги". При необходимости она может быть перемещена или удалена.

this.железные\_ДорогиTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Железные\_Дороги);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы". При необходимости она может быть перемещена или удалена.

this.ремонтные\_РаботыTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Ремонтные\_Работы);

// TODO: данная строка кода позволяет загрузить данные в таблицу "уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны". При необходимости она может быть перемещена или удалена.

this.вагоныTableAdapter.Fill(this.уП01\_ИСПП5\_Макуров\_ВАDataSet.Вагоны);

}

private void Выбрать\_Фотографию\_Click(object sender, EventArgs e)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

OpenFileDialog load = new OpenFileDialog();

if (load.ShowDialog() == DialogResult.OK)

{

Фото\_Вагона.ImageLocation = load.FileName;

SqlDataAdapter a = new SqlDataAdapter($"UPDATE Вагоны SET Фотография = '{load.FileName}' WHERE Код\_Вагона = '{Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[0].Value}'", sqlConnect);

DataSet dt = new DataSet();

a.Fill(dt);

}

}

string picture;

private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();

Фото\_Вагона.Image = Image.FromFile(picture);

}

catch { }

}

}

}

Листинг кода формы “Гость”:

using System;

using System.Data;

using System.Data.SqlClient;

using System.Drawing;

using System.Windows.Forms;

namespace Ремонт\_пассажирских\_вагонов

{

public partial class Гость : Form

{

public Гость()

{

InitializeComponent();

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

public void Просмотр(int i)

{

SqlConnection sqlConnect = new SqlConnection("Data Source = sql; Initial Catalog = уП01\_ИСПП5\_Макуров\_ВА; Integrated Security = True");

sqlConnect.Open();

SqlDataAdapter[] sa = new SqlDataAdapter[3];

sa[0] = new SqlDataAdapter("SELECT \* FROM Вагоны", sqlConnect);

sa[1] = new SqlDataAdapter("SELECT \* FROM Ремонтные\_Работы", sqlConnect);

sa[2] = new SqlDataAdapter("SELECT \* FROM Бригады", sqlConnect);

DataSet ds = new DataSet();

sa[i].Fill(ds);

Таблица.DataSource = ds.Tables[0];

}

private void Выйти\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void Вагоны\_Click(object sender, EventArgs e)

{

Просмотр(0);

Фото\_Вагона.Visible = true;

Таблица.Size = new Size(826, 438);

}

private void Ремонтные\_Работы\_Click(object sender, EventArgs e)

{

Просмотр(1);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Бригады\_Click(object sender, EventArgs e)

{

Просмотр(2);

Фото\_Вагона.Visible = false;

Таблица.Size = new Size(1041, 438);

}

private void Вернуться\_Click(object sender, EventArgs e)

{

Hide();

Авторизация UserForm = new Авторизация();

UserForm.Show();

}

private void Поиск\_TextChanged(object sender, EventArgs e)

{

for (int i = 0; i < Таблица.RowCount; i++)

{

Таблица.Rows[i].Selected = false;

for (int j = 0; j < Таблица.ColumnCount; j++)

if (Таблица.Rows[i].Cells[j].Value != null)

if (Таблица.Rows[i].Cells[j].Value.ToString().Contains(Поиск.Text))

{

Таблица.Rows[i].Selected = true;

break;

}

}

}

Point Mouse;

private void Гость\_MouseMove(object sender, MouseEventArgs e)

{

if (e.Button == MouseButtons.Left)

{

Left += e.X - Mouse.X;

Top += e.Y - Mouse.Y;

}

}

private void Гость\_MouseDown(object sender, MouseEventArgs e)

{

Mouse = new Point(e.X, e.Y);

}

string picture;

private void Таблица\_CellClick(object sender, DataGridViewCellEventArgs e)

{

try

{

picture = Таблица.Rows[Таблица.SelectedCells[0].RowIndex].Cells[7].Value.ToString();

Фото\_Вагона.Image = Image.FromFile(picture);

}

catch { }

}

}}