

СОДЕРЖАНИЕ

| | |
|---|----|
| Введение..... | 3 |
| 1 Анализ актуальности АРМ..... | 4 |
| 1.1 Обзор существующих решений..... | 4 |
| 1.1.1 БИТ. Управление автосервисом и автосалоном..... | 5 |
| 1.1.2 Система «АвтоДилер»..... | 5 |
| 1.1.3 Система «AutoService»..... | 6 |
| 1.1.4 Система «CarRepair»..... | 6 |
| 1.1.5 Система «Ваш автосервис»..... | 7 |
| 1.2 Краткая концепция программного средства..... | 8 |
| 2 Проектирование программного продукта..... | 10 |
| 2.1 Выбор языка программирования..... | 10 |
| 2.2 Выбор базы данных..... | 11 |
| 3 Описание и разработка программного продукта..... | 12 |
| 3.1 Описание программы..... | 12 |
| 3.2 Описание функций программного продукта..... | 13 |
| 3.2.1 Редактирование БД – Клиенты..... | 14 |
| 3.2.2 Редактирование БД – Сотрудники..... | 15 |
| 3.2.3 Редактирование БД – Список типов работ..... | 16 |
| 3.2.4 Редактирование БД – График записи на ремонт..... | 16 |
| 3.2.5 Редактирование БД – Список выполненных работ..... | 17 |
| 3.2.6 Вывод информации – Зарплата сотрудников..... | 18 |
| 3.2.7 Вывод информации – Денежный баланс..... | 20 |
| 3.2.8 Отчеты – Общий отчет..... | 21 |
| 3.2.9 Отчеты – Отчет по сотруднику..... | 22 |
| 4 Внедрение программного продукта..... | 24 |
| Заключение..... | 26 |
| СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ..... | 27 |
| ПРИЛОЖЕНИЕ А..... | 28 |
| ПРИЛОЖЕНИЕ Б..... | 32 |

ВВЕДЕНИЕ

В наши дни автоматизация производственных процессов становится одним из основных направлений технического прогресса. Одним из решений по автоматизации производственных процессов является создание автоматизированных рабочих мест.

Автоматизированное рабочее место (АРМ) – индивидуальный комплекс технических или программных средств, предназначенный для автоматизации профессионального труда и обеспечивающий оператора всеми средствами, необходимыми для выполнения определенных функций.

Автоматизированные рабочие места создаются строго в соответствии с их предполагаемым функциональным назначением, а поэтому позволяют обеспечить существенную пользу при их использовании.

В небольших автосервисах существует проблема с учетом деятельности компании. Простые операции, например, расчет заработных плат сотрудников или подсчет доходов автосервиса, требуют много времени, потому что компания не имеет специальной программы для управления всеми процессами в автосервисе.

На личном опыте столкнувшись с проблемой поиска качественного программного обеспечения для управления небольшим автосервисом, возникла необходимость в новой системе, которая будет отвечать определенным требованиям.

1 Анализ актуальности АРМ

1.1 Обзор существующих решений

Существует много продуктов для управления автосервисом, но все из них относятся к одной из двух групп, каждая из которых имеет свои серьезные недостатки:

Качественное, но платное программное обеспечение.

Преимущества:

- Полный спектр функций (полноценный учет, бухгалтерские и экономические отчеты, ведение склада запчастей и прочее)
- Высокая отказоустойчивость и стабильность работы (отсутствие программных сбоев, резервное копирование информации)

Недостатки:

- Высокая стоимость продукта (зачастую превышает 60-70 тысяч рублей в год)
- Высокая сложность освоения программы («переполненный» интерфейс, слишком большое обилие функций)
- Высокие требования к навыкам оператора

Доступное программное обеспечение с ограниченным функционалом.

Преимущества:

- Низкая цена (бесплатное программное обеспечение)

Недостатки:

- Очень ограниченный спектр функций (зачастую недостает необходимых функций по учету и ведению денежного баланса)
- Низкая стабильность в работе программы (частые программные сбои и прочее)

Ниже рассмотрены наиболее популярные программные продукты.

1.1.1 БИТ. Управление автосервисом и автосалоном

Основные функции:

- 1) Работы.
- 2) Сервис:
 - Автоматизация и контроль над движением запчастей;
 - Распределение и учет работ
 - Подключение внешних баз нормо-часов;
 - Все необходимые печатные формы.
 - Расчет зарплаты работников
 - Банк-касса
 - Расчеты с подотчётными лицами
 - Управление финансами
- 3) Особенности процессов в компаниях.
- 4) Планирование, контроль и учет работ исполнителей.
- 5) Подбор и логистика запчастей.

Краткая характеристика: система имеет отличный набор функций, высокую стабильность. Минусы – распространяется на платной основе (от 14000 рублей за минимальную версию для одного рабочего места), имеет сложный интерфейс, требует от оператора навыков работы с 1С-подобными программными средствами.

1.1.2 Система «АвтоДилер»

Одна из самых оптимальных систем для управления автосервисом. Имеет большой набор функций, необходимых для управления небольшим автосервисом. Проста в изучении и использовании, что позволяет работать с ней практически любому человеку.

Главный недостаток – система разделена на модули, каждый из которых можно приобрести отдельно. Для того, чтобы получить минимально

необходимый функционал для управления автосервисом, нужны как минимум три или четыре модуля, но стоимость каждого из модулей превышает 15 тысяч рублей в год.

1.1.3 Система «AutoService»

Зарубежный аналог системы «АвтоДилер». Имеет ряд преимуществ перед последней:

- Нет излишнего функционала
- Нет деления на модули
- Есть полноценная техническая поддержка
- Более «дружелюбный» интерфейс
- Присутствие базы данных различных автопроизводителей

Наряду со всеми преимуществами, которая предлагает эта система по управлению автосервисом, она ориентирована на использование крупными предприятиями, официальными дилерами и так далее. Она не подходит нам по нескольким критериям:

- Отсутствие русскоязычной версии
- Очень высокая стоимость (более 1000\$ в год)

Так как программное средство должно лишь облегчить ведение учета в автосервисе, при этом не требуя от предпринимателя каких-либо затрат, эти два минуса делают её совершенно неподходящей.

1.1.4 Система «Car Repair»

Есть все функции, которые нужны для управления автосервисом. Преимущество – система совершенно бесплатна. Первый минус – интерфейс на английском языке. В ходе использования программы становится очевидным наличие большого числа функциональных ошибок. Итог – программа очень «сырая», что делает ее непригодной для использования.

1.1.5 Система «Ваш автосервис»

Интерфейс на русском языке добавляет системе привлекательности, но минусы те же – недостаточный функционал и высокая вероятность того, что данные будут потеряны либо искажены.

1.2 Оценка существующих систем

В течение анализа существующих программных решений мной был организован поиск систем, с помощью которых можно управлять автосервисом. Было изучено много информации, отзывов других автосервисов о существующих системах, и на основании этого были отобраны несколько систем, которые были изучены более подробно.

Таких решений оказалось немного, и все они были описаны выше.

Для сравнения, все изученные мной системы по управлению автосервисом представлены в таблице 1.

Таблица 1 - Сравнение существующих систем

| | «БИТ» | «АвтоДилер» | «AutoService» | «CarRepair» | «Ваш автосервис» |
|---------------------------------------|-------|-------------|---------------|-------------|------------------|
| Необходимый функционал | ✓ | ✓ | ✓ | ✓ | |
| Отсутствие переполненного функционала | | | | ✓ | ✓ |
| Реализация на бесплатной основе | | | | ✓ | ✓ |
| Легкость освоения | | | | ✓ | ✓ |
| Низкие требования к навыкам | | | | ✓ | ✓ |
| Высокая надежность | ✓ | ✓ | ✓ | | |
| Русскоязычный интерфейс | ✓ | ✓ | | | ✓ |

Это все основные системы, которые используются для управления автосервисом. Все они относятся к одной из двух вышеописанных групп. Таким образом, на данный момент не существует продукта, который бы подходил по определенным параметрам.

Исходя из своего опыта, возможностей и пожеланий, были сформулированы основные принципы, которым должен отвечать новый продукт:

- Достаточный спектр функций для управления автосервисом;
- Простой, интуитивный интерфейс;
- Высокая стабильность программы;
- Низкие требования к навыкам оператора;
- Бесплатное распространение.

После формулировки основных принципов, которым будет отвечать будущий продукт, было произведено исследование, которое представляло собой опрос коллег, партнеров и даже конкурентов. Была выслушана критика

существующих продуктов, а также пожелания и предложения того, что нужно в «идеальном» программном средстве для управления автосервисом. Итогом исследования стала краткая концепция, показывающая, что из себя будет представлять конечный продукт.

1.3 Краткая концепция программного средства

В ходе изучения производственных процессов предприятия был сформулирован следующий список необходимых функций:

1) Хранение информации:

- Список сотрудников;
- Список клиентов;
- Список видов работ;
- Список выполненных работ;
- График записи на ремонтные работы;

2) Ведение бухгалтерии, расчеты и прочее:

- Ведение денежного баланса автосервиса;
- Расчет итоговой стоимости отдельного ремонта;
- Расчет заработной платы сотрудников.

3) Подготовка отчетов:

- Подготовка общих отчетов о работе автосервиса по выбранному временному отрезку (дневной, недельный, месячный и т.д.);
- Отчеты по различным критериям (виды работ, производители автомобилей, суммы выполненных ремонтов и т.д.);
- Отчет о выполненной работе отдельным сотрудником по выбранному временному отрезку.

В процессе создания программного средства будут добавлены новые возможности, а существующие возможности будут уточнены или изменены. Но основная концепция программы представляет собой действительно

необходимый функционал, достаточный для управления небольшим автосервисом, с интуитивно понятным интерфейсом, который сможет освоить практически любой пользователь.

1.4 Вывод

В процессе анализа актуальности разработки АРМ для управления автосервисом были проанализированы существующие аналоги. В них был выявлен ряд недостатков, не позволяющих их использование в нашем автосервисе.

На основании этого была проведена работа с руководством и сотрудниками компании, в ходе которой были выслушаны все пожелания. Итогом этой работы стал четкий список требований, которые будут предъявляться к разрабатываемой программе.

2 Проектирование программного продукта

2.1 Выбор языка программирования

При разработке нового программного средства, выбор языка программирования – один из важнейших этапов. Я выделил несколько критериев, по которым можно судить о том или ином языке программирования [2]:

- Скорость разработки программы.

Это требование весьма важно, т.к. разработка программы на малознакомом языке занимает весьма ощутимое время. В таких случаях программист склоняется в пользу высокоуровневых языков с максимально простым синтаксисом.

- Ориентированность программы

В этом критерии стоит подумать – с кем будет работать программа? С компьютером, или с живым человеком? Если программа предназначена для работы с пользователем, она должна обладать графической оболочкой.

Учитывая характер поставленной передо мной задачи, я сделал выбор в пользу языка программирования C# [5].

К такому выбору привело следующее:

- Возможности C# позволяют выполнить поставленную задачу
- Самый известный для меня язык программирования

2.2 Выбор базы данных

Так как приложению требуется работа только с локальной базой данных, был выбран простой и подходящий вариант – SQL Express LocalDB. LocalDB это вариант SQL сервера [4], доступный в рамках SQL Server 2012 Express, а именно альтернатива Express версии, которая не является постоянно

запущенным сервисом и позволяет только локальные подключения. При этом, в отличие от Compact, она поддерживает весь функционал Express.

2.3 Архитектура базы данных

База данных для разрабатываемой системы будет состоять из шести таблиц. Общий вид базы данных можно увидеть на рисунке 1.

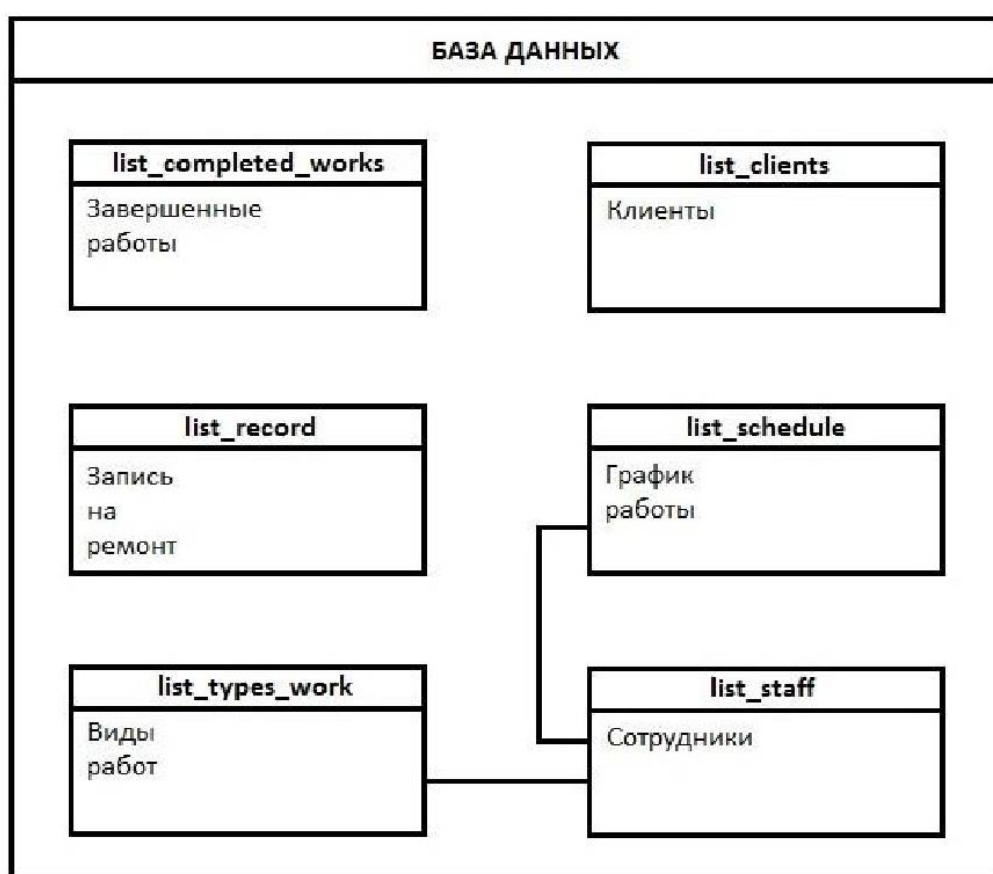


Рисунок 1 – Архитектура базы данных

В базе данных есть связи между таблицами. Таблица «Сотрудники» связана с таблицей «График работы», а таблица «Виды работ» связана с таблицей «Сотрудники».

2.4 Вывод

Во второй главе были рассмотрены средства, с помощью которых будет происходить разработка АРМ для управления автосервисом. Были выбраны подходящие для выполнения поставленной задачи язык программирования и базы данных.

Выбранные средства позволяют реализовать все функции, которые были изложены в концепции программного средства.

Также была сформирована архитектура базы данных, с которой будет работать разрабатываемая АРМ. С данной архитектурой базы данных система сможет выполнять все задачи, поставленные перед ней. В случае необходимости расширения, данная архитектура может быть пересмотрена, в ней могут появиться новые таблицы и новые связи.

3 Описание и разработка программного продукта

Ранее мной было выбрано несколько основных критериев, которым должна соответствовать программа для управления автосервисом. Исходя из этих критериев, становится понятно, что программа ориентирована на работу с человеком, который сможет без специальной подготовки и обучения пользоваться всеми доступными в ней возможностями.

Основная трудность – максимально сократить функционал, но оставить его достаточным для того, чтобы гарантировать охват всех функций по учету в автосервисе. Максимально сжатый функционал снизит требования к навыкам оператора программы. Высокой надежности можно достигнуть за счет хорошо структурированного кода программы и хорошо проделанной работы по тестированию программы и исправлению ошибок.

3.1 Описание программы

База данных системы состоит из нескольких таблиц:

- График записи на ремонт;
- Клиенты;
- Рабочий график сотрудников;
- Сотрудники;
- Список видов работ;
- Список выполненных работ.

Общий вид структуры программы можно увидеть на рисунке 2.



Рисунок 2 – Структура программы

Для отображения данных в окне системы используется элемент управления DataGridView.

DataGridView предоставляет собой мощный и гибкий способ отображения данных в табличном формате. Этот элемент управления можно использовать для отображения представлений небольшого объема данных только для чтения, либо можно масштабировать его для отображения редактируемого представления очень больших наборов данных.

В качестве связи между базой данных и элементом управления используется объект DataSet. Он используется в программе для передачи необходимых данных из базы данных в элемент управления DataGridView.

При необходимости произвести обработку данных из базы и провести над ними нужные вычисления, также используется связка DataSet – DataGridView. Внутри одного объекта DataSet может храниться сразу несколько таблиц из базы данных, помещенных в соответствующие объекты DataTable, что позволяет при необходимости произвести их обработку и провести нужные вычисления, используя данные сразу из нескольких таблиц. После всех действий с данными, конечная информация передается в окно при помощи DataGridView.

3.2 Интерфейс программы

Одним из главных требований к разрабатываемой системе был простой интерфейс.

Программа содержит 3 основных пункта меню:

1) Редактирование базы данных

В этом пункте происходит заполнение базы данных и ее редактирование.

2) Вывод информации

В пункте вывода информации, в основном, находятся данные о финансах, расчеты заработной платы сотрудников и прочее.

3) Отчеты

В меню «Отчеты» можно подготовить отчеты по работе автосервиса за определенный промежуток времени

Такое количество пунктов позволяет с первого взгляда разобраться в программе и пользоваться ей полноценно с первой минуты (Рисунок 3):

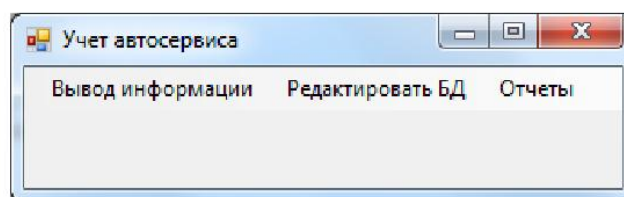


Рисунок 3 – Главное меню

Во всех меню редактирования базы данных присутствует идентичная панель управления. Ее можно увидеть на рисунке 4.



Рисунок 4 – Панель управления

При создании новой записи в таблице оператору необходимо нажать кнопку «Добавить запись» в панели управления. Далее следует заполнить все поля данных. Если нужно добавить сразу несколько строк, записи добавляются последовательно одна за другой. После добавления всех необходимых записей оператор нажимает кнопку «Сохранить изменения». База данных сохраняется.

Для удаления какой-либо записи в базе данных оператору нужно мышкой выбрать нужную запись, нажать кнопку «Удалить запись» в панели управления. Запись удалится из базы данных. Для сохранения всех действий оператору необходимо по завершению нажать кнопку «Сохранить изменения».

В случае, если оператор редактирует базу данных, но в конце по какой-либо причине не нажимает кнопку «Сохранить изменения», все внесенные изменения пропадают, база данных остается в изначальном виде.

3.3 Описание функций программного продукта

3.3.1 Редактирование БД – Клиенты

Меню «Клиенты» отвечает за базу данных клиентов автосервиса. Функционал меню прост: в верхней части окна присутствует панель управления, с помощью которой можно перелистывать страницы базы данных, добавлять клиента, удалять клиента, и сохранять изменения, которые были внесены в базу данных. Вид меню «Клиенты» можно увидеть на рисунке 5.

Каждая запись о клиенте содержит 6 полей:

- ID (идентификационный номер) клиента;
- Фамилия, Имя и Отчество клиента;
- Марка и модель автомобиля клиента;
- Год выпуска автомобиля клиента;
- Дата первого обслуживания данного клиента;
- Скидка, которая предоставляется данному клиенту.

| id | ФИО | Марка автомобиля | Год выпуска | Наш клиент с | Скидка |
|----|----------------------------|------------------|-------------|--------------|--------|
| 1 | Петров Иван Иванович | Toyota Camry | 2007 | 01.03.2016 | 0 |
| 2 | Попов Евгений Олегович | Hyundai Sonata | 2003 | 26.04.2015 | 10 |
| 3 | Микрюков Максим Петрович | Toyota Caldina | 1998 | 24.03.2016 | 5 |
| 4 | Соломатина Мария Андреевна | Ford Focus | 2011 | 14.01.2016 | 0 |
| 5 | Соснова Алена Алексеевна | Honda Civic | 2008 | 21.09.2015 | 0 |
| 6 | Петренко Федор Михайлович | Nissan Almera | 2012 | 03.03.2016 | 0 |

Рисунок 5 – Редактирование БД – Клиенты

3.3.2 Редактирование БД – Сотрудники

Пункт меню «Сотрудники» отвечает за редактирование базы данных работников автосервиса. Вверху страницы находится панель управления, чуть ниже отображается база данных сотрудников. Вид меню «Сотрудники» можно увидеть на рисунке 6.

Каждая запись о сотруднике содержит 7 полей:

- ID сотрудника;
- ФИО сотрудника;
- Должность сотрудника;
- Дата рождения;

- Дата приема на работу;
- Оплачиваемая процентная доля от выполнения работы;
- Образование сотрудника.

The screenshot shows a window titled 'Сотрудники' with a table containing the following data:

| | Id | ФИО | Должность | Дата рождения | Дата приема на работу | Зарплата процент от выполнения | Образование |
|---|----|------------------------|------------------|---------------|-----------------------|--------------------------------|--------------------|
| ▶ | 1 | Иванов Иван Иванович | Механик | 20.04.1976 | 20.04.2000 | 50 | Средне-специальное |
| | 2 | Сидоров Сидр Сидорович | Электрик | 10.05.1977 | 21.04.2000 | 40 | Средне-техническое |
| | 3 | Петров Петр Петрович | Жестянщик | 05.06.1978 | 22.04.2000 | 35 | Среднее |
| | 4 | Иванов Сидр Петрович | Слесарь-электрик | 25.07.1980 | 23.04.2000 | 60 | Средне-техническое |
| | 5 | Сидоров Петр Иванович | Диагност | 30.08.1982 | 25.04.2000 | 55 | Высшее |
| * | | | | | | | |

Рисунок 6 – Пункт меню «Сотрудники»

3.3.3 Редактирование БД – Список типов работ

В этом пункте меню можно редактировать список наименований работ. Записи из этого меню используются в дальнейших пунктах, а также в составлении отчетов.

Каждая запись о типе работы состоит из 2 полей:

- ID типа работы;
- Наименование типа работы.

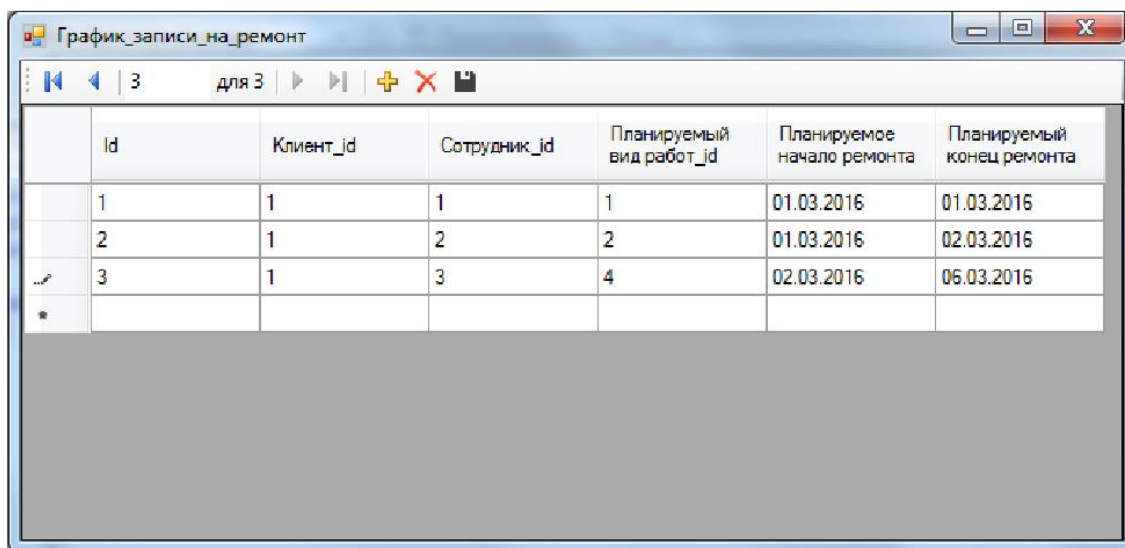
3.3.4 Редактирование БД – График записи на ремонт

Данный пункт меню служит для того, чтобы вести график предварительных записей клиентов на ремонт своего автомобиля. Если клиент обратился в автосервис для какого-либо ремонта, но расчетное время ремонта слишком большое, то клиент просит записать его на удобное ему время. В этом

случае оператор заходит в меню «График записи на ремонт» и производит запись данного клиента. Вид меню «График записи на ремонт» можно увидеть на рисунке 7.

Панель управления ничем не отличается от других меню редактирования баз данных. Каждая запись о предварительной договоренности содержит 6 полей:

- ID записи на ремонт;
- ФИО клиента;
- ФИО сотрудника;
- Планируемый вид ремонта;
- Планируемое начало ремонта;
- Планируемое окончание ремонта;



| | id | Клиент_id | Сотрудник_id | Планируемый вид работ_id | Планируемое начало ремонта | Планируемый конец ремонта |
|-----|----|-----------|--------------|--------------------------|----------------------------|---------------------------|
| | 1 | 1 | 1 | 1 | 01.03.2016 | 01.03.2016 |
| | 2 | 1 | 2 | 2 | 01.03.2016 | 02.03.2016 |
| ... | 3 | 1 | 3 | 4 | 02.03.2016 | 06.03.2016 |
| * | | | | | | |

Рисунок 7 – Пункт меню «График записи на ремонт»

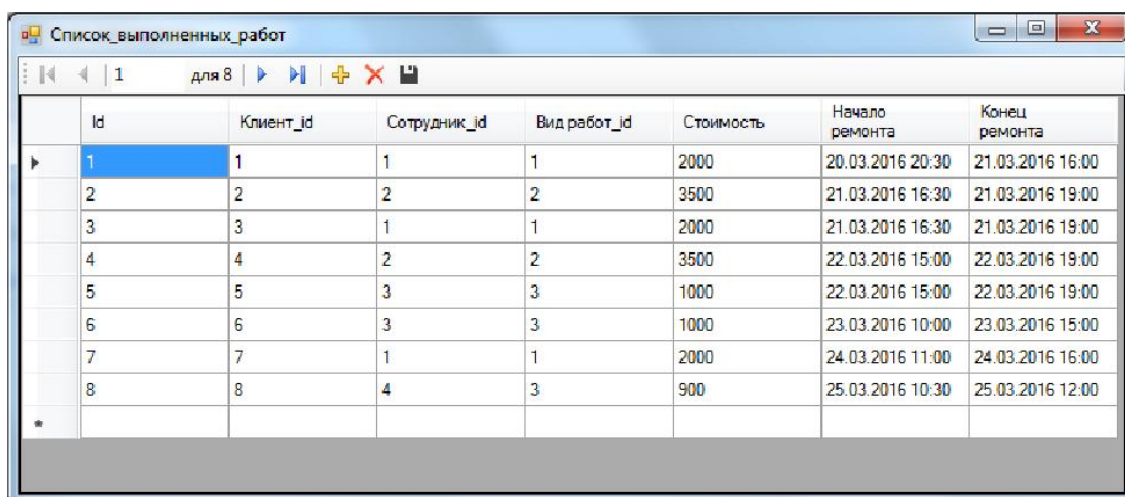
3.3.5 Редактирование БД – Список выполненных работ

После окончания ремонта оператору нужно внести уже фактические данные о проделанном ремонте в базу данных. Это нужно для дальнейших

расчетов, составления различных отчетов и так далее. Внешний вид меню «Список выполненных работ» можно посмотреть на рисунке 8.

Для ведения учета по выполненным работам и создан этот пункт. Каждая запись о проведенном ремонте содержит 7 полей:

- ID выполненной работы;
- ФИО клиента;
- ФИО сотрудника, выполнявшего ремонт;
- Вид ремонта;
- Итоговая стоимость ремонта;
- Время начала ремонта;
- Время окончания ремонта.



| | Id | Клиент_id | Сотрудник_id | Вид работ_id | Стоимость | Начало ремонта | Конец ремонта |
|---|----|-----------|--------------|--------------|-----------|------------------|------------------|
| ▶ | 1 | 1 | 1 | 1 | 2000 | 20.03.2016 20:30 | 21.03.2016 16:00 |
| | 2 | 2 | 2 | 2 | 3500 | 21.03.2016 16:30 | 21.03.2016 19:00 |
| | 3 | 3 | 1 | 1 | 2000 | 21.03.2016 16:30 | 21.03.2016 19:00 |
| | 4 | 4 | 2 | 2 | 3500 | 22.03.2016 15:00 | 22.03.2016 19:00 |
| | 5 | 5 | 3 | 3 | 1000 | 22.03.2016 15:00 | 22.03.2016 19:00 |
| | 6 | 6 | 3 | 3 | 1000 | 23.03.2016 10:00 | 23.03.2016 15:00 |
| | 7 | 7 | 1 | 1 | 2000 | 24.03.2016 11:00 | 24.03.2016 16:00 |
| | 8 | 8 | 4 | 3 | 900 | 25.03.2016 10:30 | 25.03.2016 12:00 |
| * | | | | | | | |

Рисунок 8 – Пункт меню «Список выполненных работ»

3.3.6 Вывод информации – Зарплата сотрудников

Одна из удобных функций программы по управлению автосервисом – автоматический расчет заработных плат сотрудников. Это избавляет управляющего автосервисом от длительных и рутинных расчетов, позволяет видеть всю необходимую информацию в уже обработанном виде в одном окне.

После нажатия кнопки меню «Зарплата сотрудников» на экране появляется окно, формально поделенное на две части. В соответствии с рисунком 9, в нижней части окна отображается таблица со всеми выполненными работами. Для удобства восприятия в таблице отображаются следующие данные о каждом ремонте:

- ФИО сотрудника;
- Дата и время начала ремонта;
- Вид работ;
- Стоимость работ;
- Оплачиваемая процентная доля от выполнения работы;
- Итоговая сумма, которую заработал сотрудник.

The screenshot shows a window titled "Зарплата" (Salary) with two main sections. The top section, titled "ОБЩИЙ РАСЧЕТ" (General Calculation), contains a table with two columns: "ФИО" (Full Name) and "Итоговая_зарплата" (Total Salary). The bottom section, titled "ПОДРОБНЫЙ РАСЧЕТ" (Detailed Calculation), contains a table with seven columns: "ФИО", "Начало_ремонта" (Start of Repair), "Вид_работ" (Type of Work), "Стоимость_работ" (Cost of Work), "Процент_ЗП_от_выполнения" (Percentage of Salary from Completion), and "Сумма_выд" (Sum of Issued). The data in the tables is as follows:

| ФИО | Итоговая_зарплата |
|------------------------|-------------------|
| Иванов Иван Иванович | 3000 |
| Сидоров Сидр Сидорович | 2800 |
| Петров Петр Петрович | 700 |
| Иванов Сидр Петрович | 540 |
| Сидоров Петр Иванович | |

| ФИО | Начало_ремонта | Вид_работ | Стоимость_работ | Процент_ЗП_от_выполнения | Сумма_выд |
|------------------------|------------------|-----------------|-----------------|--------------------------|-----------|
| Иванов Иван Иванович | 20.03.2016 20:30 | Замена свечей | 2000 | 50 | 1000 |
| Сидоров Сидр Сидорович | 21.03.2016 16:30 | Замена проводки | 3500 | 40 | 1400 |
| Иванов Иван Иванович | 21.03.2016 16:30 | Замена свечей | 2000 | 50 | 1000 |
| Сидоров Сидр Сидорович | 22.03.2016 15:00 | Замена проводки | 3500 | 40 | 1400 |
| Петров Петр Петрович | 22.03.2016 15:00 | Замена масла | 1000 | 35 | 350 |

Рисунок 9 – Пункт меню «Зарплата сотрудников»

Таким образом, если у сотрудника возникают вопросы по заработной плате, управляющий автосервисом сможет в подробностях изучить всю информацию и, при необходимости, предоставить ее обратившемуся сотруднику.

В верхней части окна расположена таблица с уже рассчитанной заработной платой для каждого сотрудника. Эта таблица показывает лишь два типа данных:

- ФИО сотрудника;
- Итоговая сумма, которую заработал сотрудник.

Эта таблица позволяет управляющему мгновенно получить итоговую информацию.

3.3.7 Вывод информации – Денежный баланс

Эта функция позволяет управляющему автосервисом мгновенно получить информацию о прибыли автосервиса за тот или иной период времени.

При нажатии на кнопку «Денежный баланс» на экране появляется окно, которое можно увидеть на рисунке 10. Оно делится на три части:

- Календарь для выбора периода;
- Итоговая прибыль за выбранный период;
- Подробный отчет за выбранный период.

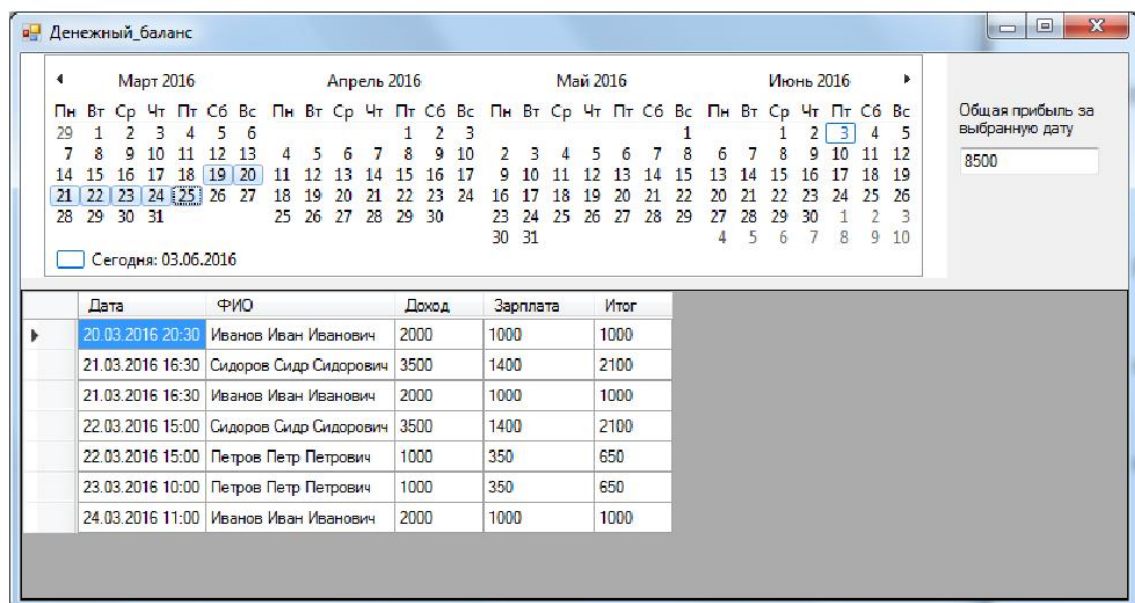


Рисунок 10 – Пункт меню «Денежный баланс»

Календарь помогает быстро и удобно выбрать период, за который управляющий или оператор хочет получить необходимую информацию. Можно выбрать любой период в рамках от 1 дня до 1 месяца.

В части окна, расположенной правее от календаря, показывается итоговая прибыль автосервиса за выбранный период. Отображается именно чистая прибыль, уже учтены будущие вычеты на заработную плату сотрудникам. Таким образом, управляющий может быстро получить информацию о том, насколько эффективно работал автосервис в тот или иной период времени.

Если управляющему или оператору нужна более подробная информация, то она отображается ниже в том же окне. Благодаря этой информации управляющий может провести анализ не только в целом по автосервису, но и по отдельным работникам. Это позволяет делать выводы об эффективности тех или иных сотрудников и сравнивать их с другими сотрудниками.

3.3.8 Отчеты – Общий отчет

Данный пункт меню состоит из двух областей:

- Календарь для выбора периода;
- Таблица с ежедневными отчетами внутри выбранного периода.

Каждая запись в таблице с ежедневными отчетами содержит следующие поля:

- Дата;
- Количество выполненных ремонтов;
- Доход;
- Расход;
- Средняя стоимость работ.

Как видно на рисунке 11, эта таблица позволяет управляющему получить отчетность за каждый рабочий день, анализировать количество обращений клиентов и полученные доходы. Вся эта информация за большой промежуток

времени позволяет проследить за тем, развивается ли автосервис, и если да, то насколько быстро это происходит.

| Дата | Количество выполненных ремонтов | Доход | Расход | Средняя стоимость одного ремонта |
|------------|---------------------------------|-------|--------|----------------------------------|
| 20.03.2016 | 1 | 2000 | 1000 | 2000 |
| 21.03.2016 | 2 | 5500 | 2400 | 2750 |
| 22.03.2016 | 2 | 4500 | 1750 | 2250 |
| 23.03.2016 | 1 | 1000 | 350 | 1000 |
| 24.03.2016 | 1 | 2000 | 1000 | 2000 |
| 25.03.2016 | 1 | 900 | 540 | 900 |

Рисунок 11 – Пункт меню «Общий отчет»

3.3.9 Отчеты – Отчет по сотруднику

Этот пункт меню позволяет управляющему получить ежедневный отчет по любому сотруднику за выбранный период времени. Как можно увидеть на рисунке 12, окно состоит из трех областей:

- Календарь для выбора периода;
- Выбор сотрудника;
- Таблица с ежедневными отчетами внутри выбранного периода по выбранному сотруднику.

Каждая запись в таблице содержит следующие поля:

- ФИО сотрудника;

- Дата;
- Количество выполненных работ;
- Доход;
- Расход;
- Средняя стоимость ремонта.

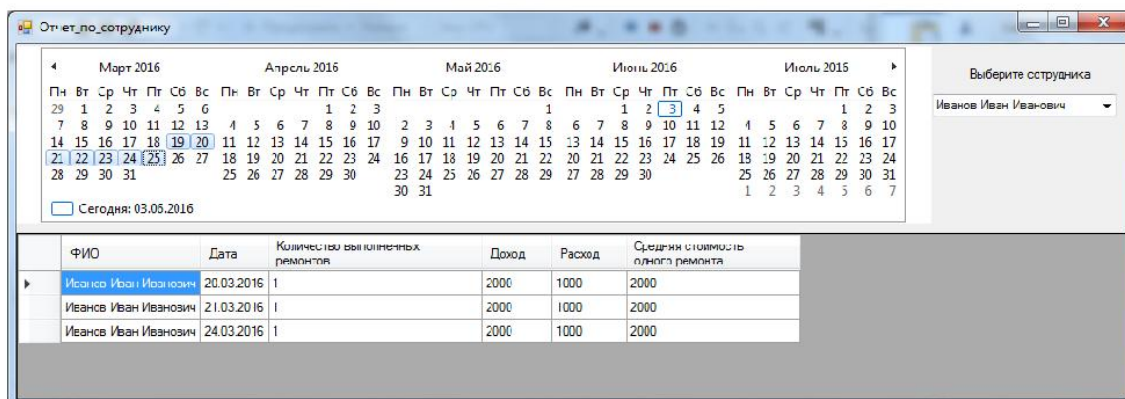


Рисунок 12 – Пункт меню «Отчет по сотруднику»

Эта таблица позволяет управляющему проанализировать работу сотрудника «на длинной дистанции», то есть за большой период времени. По результатам анализа можно сделать выводы о том, насколько эффективно работает тот или иной сотрудник, сравнить сотрудников, а также получить информацию о том, заслуживает ли сотрудник повышения (или понижения) процента от выполненных работ.

3.4 Внедрение программного продукта

Разработанная система ограничена небольшим набором функций, и это позволяет быстро протестировать каждую функцию. План тестирования не предусматривал каких-либо специальных тестов. Каждая таблица системы была последовательно проверена на корректность сохранения и удаления записей.

Каждая математическая функция была проверена несколькими способами, по итогам проверялась корректность вычисляемых и отображаемых данных.

После тестирования программы был введен тестовый период, в который вся отчетность велась в двух видах: так, как она велась ранее в автосервисе, и через разработанную систему учета. Длительность тестового периода была заранее ограничена одним месяцем, так как ведение документации сразу в двух видах занимает много времени и людских ресурсов.

В течение тестового периода система работала корректно. Было выявлено несколько ошибок, но все серьезные недочеты были выявлены на стадии тестирования.

По истечении тестового периода руководством компании была проанализирована работа системы, оценена польза от ее использования. Было выявлено несколько основных преимуществ данной системы по сравнению с тем, в каком виде велась вся документация ранее:

- Документация и отчетность собраны в одном месте
- Быстрое нахождение заказа, независимо от его давности
- Возможность оценки эффективности работников
- Возможность быстрого расчета заработной платы
- Возможность по нужным периодам оценить темпы развития автосервиса
- Возможность быстро получить нужные отчеты о работе автосервиса

Как видно, в условиях конкретного автосервиса, введение этой системы помогло значительно ускорить некоторые процессы, а также позволило получить некоторые новые возможности по оценке и анализу работы автосервиса. Система получила хорошую оценку от руководства компании и была введена в эксплуатацию на постоянной основе, о чем свидетельствует приказ руководителя компании.

Также, руководством компании уже были сделаны предложения по дополнению системы. В данный момент эти предложения обсуждаются.

3.5 Вывод

Было разработано автоматизированное рабочее место для управления автосервисом, которое полностью отвечает поставленным руководством компании требованиям. Функционал АРМ был испытан на рабочем месте несколькими сотрудниками, серьезных нареканий выявлено не было. По истечению пробного периода руководством компании было принято решение о том, что система оказалась удачным решением, подходящим для использования в нашей компании. Итогом разработки АРМ для управления автосервисом стал приказ руководства компании о введении разработанной системы в эксплуатацию.

ЗАКЛЮЧЕНИЕ

В процессе выполнения бакалаврской работы были изучены основные характеристики различных сред программирования. По завершению анализа готовых сред, были сформированы требования к разрабатываемому продукту.

Для разработки интегрированной среды был выбран оптимальный для данного случая набор инструментов, в который вошли: язык программирования C++ и набор библиотек .Net Framework. Этот набор позволил в полной мере реализовать все требования к разрабатываемой системе.

В процессе разработки были изучены ранее неизвестные возможности библиотек .Net, спроектирована архитектура разрабатываемого приложения и выполнена реализация приложения в виде программы.

Разработанная система была испытана в реальных условиях функционирования автосервиса. Во время испытаний система зарекомендовала себя как отличный помощник администратору автосервиса в ведении отчетности. Было замечено повышение скорости подготовки различных отчетов, а также возможность анализировать процесс развития автосервиса. По итогам испытания система была одобрена руководством и принята на постоянное использования взамен ранее используемым средствам учета.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Фленов, Михаил. Библия С#. 3 изд. ; книга / Михаил Фленов. – Санкт-Петербург : БХВ-Петербург, 2016. – 544 с.
2. Климов, Александр. С#. Советы программистам : книга / Александр Климов. – Санкт-Петербург : БХВ-Петербург, 2012. – 544 с.
3. Руководство по программированию на С# [Электронный ресурс] : учебные пособия. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library/67ef8sbd.aspx>
4. Работа с базами данных на языке С#. Технология ADO .NET [Электронный ресурс] : учебное пособие. – Режим доступа: <http://window.edu.ru/resource/225/65225>
5. Гриффитс, Иэн. Программирование на С# 5.0 : учебник / Иэн Гриффитс. – Москва : Эксмо, 2015. – 1136 с.

ПРИЛОЖЕНИЕ А

Листинг программы

```
{
public partial class Зарплата : Form
{
    public Зарплата()
    {
        InitializeComponent();
    }

    private void Зарплата_Shown(object sender, EventArgs e)
    {
        Список_выполненных_работ list_completed_works = new
Список_выполненных_работ();
        list_completed_works.Show();
        list_completed_works.Visible = false;
        Сотрудники list_staff = new Сотрудники();
        list_staff.Show();
        list_staff.Visible = false;
        Список_видов_работ list_types_work = new Список_видов_работ();
        list_types_work.Show();
        list_types_work.Visible = false;
        dataGridView1.AllowUserToAddRows = false;
        dataGridView2.AllowUserToAddRows = false;

list_completed_works.list_completed_worksDataGridView.AllowUserToAddRows =
false;
        list_types_work.list_types_workDataGridView.AllowUserToAddRows =
false;
        list_staff.list_staffDataGridView.AllowUserToAddRows = false;

        for (int I = 0; I <
list_completed_works.list_completed_worksDataGridView.Rows.Count; i++)
        {
            dataGridView1.Rows.Add();
            dataGridView1.Rows[i].Cells[1].Value =
list_completed_works.list_completed_worksDataGridView.Rows[i].Cells[5].Value;
            for (int j = 0; j <
list_completed_works.list_completed_worksDataGridView.Rows.Count; j++)
            {
```

```

        if (list_staff.list_staffDataGridView.Rows.Count <= j)
        {
            break;
        }
        if
(list_completed_works.list_completed_worksDataGridView.Rows[i].Cells[2].Value.
ToString().Contains(list_staff.list_staffDataGridView.Rows[j].Cells[0].Value.ToStrin
g()))
        {
            dataGridView1.Rows[i].Cells[0].Value =
list_staff.list_staffDataGridView.Rows[j].Cells[1].Value;
        }
        for (int j = 0; j <
list_completed_works.list_completed_worksDataGridView.Rows.Count; j++)
        {
            int v =
int.Parse(list_completed_works.list_completed_worksDataGridView.Rows[i].Cells[3
].Value.ToString());
            if
(list_completed_works.list_completed_worksDataGridView.Rows[i].Cells[3].Value.
ToString().Contains(list_types_work.list_types_workDataGridView.Rows[v -
1].Cells[0].Value.ToString()))
            {
                dataGridView1.Rows[i].Cells[2].Value =
list_types_work.list_types_workDataGridView.Rows[v - 1].Cells[1].Value;
            }
        }
        dataGridView1.Rows[i].Cells[3].Value =
list_completed_works.list_completed_worksDataGridView.Rows[i].Cells[4].Value;
        for (int j = 0; j < list_staff.list_staffDataGridView.Rows.Count; j++)
        {
            if
(list_staff.list_staffDataGridView.Rows[j].Cells[1].Value.ToString().Contains(dataGr
idView1.Rows[i].Cells[0].Value.ToString()))
            {
                dataGridView1.Rows[i].Cells[4].Value =
list_staff.list_staffDataGridView.Rows[j].Cells[5].Value;
            }
        }
        double z =
double.Parse(dataGridView1.Rows[i].Cells[3].Value.ToString());
        double x =
double.Parse(dataGridView1.Rows[i].Cells[4].Value.ToString());

```



```

        dataGridView1.Rows[i].Cells[5].Value = z * x / 100;
    }
    List<int> list_ZP = new List<int>();
    List<string> list_FIO = new List<string>();
    int searcher = 0;
    for (int I = 0; I < list_staff.list_staffDataGridView.Rows.Count; i++)
    {
        dataGridView2.Rows.Add();
        dataGridView2.Rows[i].Cells[0].Value =
list_staff.list_staffDataGridView.Rows[i].Cells[1].Value;

        if
(dataGridView2.Rows[i].Cells[0].Value.ToString().Contains(dataGridView1.Rows[i]
.Cells[0].Value.ToString()))
        {

//list_ZP.Add(Convert.ToInt32(dataGridView1.Rows[i].Cells[5].Value));
        //list_ZP[0] += list_ZP[1];
        }

        }
    for (int j = 0; j < dataGridView1.Rows.Count; j++)
    {
        if (j == 0)
        {

list_FIO.Add(Convert.ToString(dataGridView1.Rows[j].Cells[0].Value));
        list_ZP.Add(Convert.ToInt32(dataGridView1.Rows[j].Cells[5].Value));
        continue;
        }
        if (list_FIO.Contains(dataGridView1.Rows[j].Cells[0].Value.ToString()))
        {
            for (int z = 0; z < list_FIO.Count; z++)
            {
                if
(list_FIO[z].Contains(dataGridView1.Rows[j].Cells[0].Value.ToString()))
                {
                    searcher = z;
                    break;
                }
            }
            list_ZP[searcher] +=
(Convert.ToInt32(dataGridView1.Rows[j].Cells[5].Value));
        }
    }

```

```

    if (!list_FIO.Contains(dataGridView1.Rows[j].Cells[0].Value.ToString()))
    {
list_FIO.Add(Convert.ToString(dataGridView1.Rows[j].Cells[0].Value));
        list_ZP.Add(Convert.ToInt32(dataGridView1.Rows[j].Cells[5].Value));
    }
}

for (int I = 0; I < dataGridView1.Rows.Count; i++)
{
    if (dataGridView1.Rows[i].Cells[0].Value == null)
    {
        dataGridView1.Rows.RemoveAt(i);
        i--;
    }
}
for (int I = 0; I < dataGridView2.Rows.Count; i++)
{
    if (dataGridView2.Rows[i].Cells[0].Value == null)
    {
        dataGridView2.Rows.RemoveAt(i);
        i--;
    }
}
for (int I = 0; I < dataGridView2.Rows.Count; i++)
{
    if (list_ZP.Count <= i)
    {
        break;
    }
    dataGridView2.Rows[i].Cells[1].Value = list_ZP[i];
}
}
}
}

```