

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования
Гомельский государственный технический университет
имени П.О. Сухого

Факультет автоматизированных и информационных систем

Кафедра «Информационные технологии»
1-40 05 01-01 Информационные системы и технологии (в проектировании и
производстве)
Отчёт по преддипломной практике

на тему «Автоматизированное рабочее место оператора приема и переработки
молочной продукции ОАО «Туровский молочный комбинат»

Исполнитель: студент гр. ЗИТ-51
Москаленко И.Н.

Руководитель от предприятия:
Домнич И. А.

Руководитель от университета:
Доцент Стародубцев Е. Г.

Дата проверки: _____

Дата допуска к защите: _____

Дата защиты: _____

Оценка работы: _____

Подписи членов комиссии: _____

Гомель 2021

СОДЕРЖАНИЕ

Содержание	2
Введение	3
1 Описание предприятия ОАО «Туровский молочный комбинат»	4
1.1 Общие сведения об ОАО «Туровский молочный комбинат»	4
1.2 Должностная инструкция инженера-программиста ОАО «Туровский молочный комбинат»	6
1.3 Охрана труда предприятия ОАО «Туровский молочный комбинат»	7
2 Архитектура программного комплекса для автоматизации системы учёта работ с приезжающими молоковозами	9
2.1 Анализ существующих аналогов	9
2.2 Функциональная архитектура приложения	14
2.3 Логическая модель базы данных	14
2.4 Физическая модель базы данных	15
2.5 Программная архитектура приложения	19
2.6 Настройка программного комплекса	25
Заключение	27
Список использованных источников	28
Приложение А	30

ВВЕДЕНИЕ

Внедрение средств вычислительной техники, доступность информации, объём и скорость её обработки становятся решающими факторами развития производственных сил государства, науки, культуры, общественных институтов и всех сфер жизнедеятельности человека. Информация и данные все чаще рассматриваются как жизненно важные ресурсы, которые должны быть организованы таким образом, чтобы ими можно было легко пользоваться.

Основные идеи современных информационных технологий базируются на концепции, согласно которой данные должны быть организованы в базы данных, с целью адекватного отображения изменяющегося реального мира и удовлетворения информационных потребностей пользователей.

Любая информационная система представляет собой программный комплекс, функции которого состоят в поддержке надежного хранения информации в памяти компьютера, выполнении специфических для данного приложения преобразований информации и вычислений, предоставлении пользователям удобного и легко осваиваемого интерфейса [1].

Цель преддипломной практики – разработать информационную систему учета работ с молоковозами для поддержки работы работников молочного комбината.

Методы исследования и технология разработки: анализ деятельности ИТ-отдела, обзор и выбор средств разработки, объектно-ориентированное программирование, СУБД *Workbench MySQL*.

Средствами разработки являются *C#*, *WindowsForm*, инструментарий для разработки базы данных – *MySql Workbench*.

Для достижения поставленной цели были сформулированы следующие задачи:

- анализ деятельности отдела информационных технологий **ОАО «Туровский молочный комбинат»**;
- выявление существующих недостатков в текущей технологии управления;
- обоснование необходимости автоматизации;
- обоснование проектных решений по программному, техническому и информационному обеспечению;
- разработка базы данных и приложения для работы с ней.

Область применения – данная информационная система позволит автоматизировать процесс оформления и проведенных работ с молоковозами **ОАО «Туровский молочный комбинат»**. Использование данной системы позволит сократить временной цикл операции и обеспечить достоверность информации на выходе.

1 ОПИСАНИЕ ПРЕДПРИЯТИЯ **ОАО «ТУРОВСКИЙ МОЛОЧНЫЙ КОМБИНАТ»**

1.1 Общие сведения **об ОАО «Туровский молочный комбинат»**

Предприятие было основано в 2010 году. Строительство ОАО «Туровский молочный комбинат» комбината стало частью Государственной программы развития Припятского Полесья, и сегодня комбинат является одним из градообразующих предприятий города Турова и Житковичского района.

Запуск производства и первые отгрузки готовой продукции состоялись в мае 2013 года. В январе 2018 года был осуществлен запуск второй производственной очереди комбината [2].

На ОАО «Туровский молочный комбинат» создано более 500 рабочих мест. Кадровая политика предприятия основана на организации занятости местных жителей, а также привлечении уже состоявшихся специалистов и молодежи из всех областей республики. Для сотрудников созданы все необходимые условия для качественной, плодотворной и комфортной работы, а также профессионального роста.

В 2020 году на комбинате были увеличены мощности по производству сливочных сыров. Установлены линии по переработке и сгущению сыворотки. В планах на 2021 год – модернизация линии вытяжных сыров (Паста Филата).

Предприятие активно работает над развитием собственной сырьевой зоны и в настоящее время является учредителем в пяти хозяйствах Житковичского района Гомельской области – ОАО «Приозерское-Агро», ОАО «Дяковичи», ОАО «Урожайный», ОАО «Колпель-Агро» и КСУП «Комсомольск», учредителем в двух хозяйствах Минской области – ОАО «Долгиново» и ОАО «Новая Вилия», а также в двух Витебской области – ОАО «Торгуны» и ГП «Гнездилово-Агро».

ОАО «Туровский молочный комбинат» – современное успешное высокотехнологичное предприятие, выпускающее полутвердые и мягкие сыры по итальянским технологиям с 2013 года.

Сыры Туровского молочного комбината хорошо известны в Беларуси и за рубежом под ТМ Bonfesto и ТМ CooKing. В 2019 году предприятие вывело на рынок новую марку сыров «Басни о сыре» [3].

Предприятие выпускает широкий ассортимент продукции: моцарелла в заливке и полутвердая, маскарпоне, рикотта, провола, проволетта, скаморца, кремчиз, кремолле, эдам, гауда, сулугуни.

Натуральный, вкусный и полезный сыр на Туровском молочном комбинате производят благодаря применению проектных и технологических решений, разработанных совместно с итальянскими специализированными компа-

ниями. Технологии производства отрабатывались с участием опытных итальянских экспертов. К главным преимуществам технологий, используемых сегодня на Туровском молочном комбинате, относится максимальный выход готового продукта и отсутствие потерь белка в ходе технологического процесса.

Успех продукции предприятия во многом обусловлен использованием только высококачественного белорусского молока. В производстве качественных пищевых продуктов очень важное значение имеет также соблюдение стандартов безопасности и гигиены производства. Влияние человеческого фактора в производственном процессе минимизировано. Все установленное на комбинате оборудование относится к экологичным производствам и соответствует требованиям и нормам Евросоюза. Система менеджмента качества и безопасности пищевой продукции ОАО «Туровский молочный комбинат» соответствует требованиям международных стандартов, что подтверждено соответствующими сертификатами. Среди них сертификаты соответствия требованиям международных стандартов ISO 9001 и схеме FSSC 22000, ISO 14001, OHSAS 18001, сертификат «Халяль».

В 2019 году количество рынков сбыта продукции выросло до 16, среди них Российская Федерация, Казахстан, Кыргызстан, Азербайджан, Армения, Грузия, Узбекистан, Объединенные Арабские Эмираты, Туркменистан, Сингапур, Иордания, Гонконг, Молдова, Китай, Япония, Украина. Успешному старту продаж на рынках стран Азиатско-Тихоокеанского региона способствует в том числе и наличие международного сертификата Халяль.

Как коммерчески успешное предприятие, ОАО «Туровский молочный комбинат» уделяет самое серьезное внимание социальной политике и принимает участие в проектах, направленных на решение важнейших социальных вопросов в регионе и в стране в целом. Так, в 2018 году стартовал национальный проект «Bonfesto – детям». В рамках реализации проекта предусмотрено строительство полей с искусственным покрытием для мини-футбола в разных населенных пунктах Беларуси с целью создания необходимой инфраструктуры для развития детско-юношеского спорта и популяризации здорового образа жизни. По программе «Bonfesto – детям» в 2018 году открыли 9 новых полей по мини-футболу. В 2019 году было введено в эксплуатацию еще 12 современных футбольных площадок. Всего за период 2018-2022 гг. будет построено и введено в эксплуатацию 78 объектов

В ближайшей перспективе предприятие ставит перед собой амбициозные цели: дальнейшее развитие самой категории мягких и полутвердых сыров на рынке, а также изменение гастрономических и кулинарных предпочтений потребителей в сторону принципов правильного питания и более сбалансированного рациона, свойственного для европейской кухни.

1.2 Должностная инструкция инженера-программиста ОАО «Туровский молочный комбинат»

На инженера-программиста возлагаются следующие функции:

- разработка программ, направленных на разработку приложений;
- отладка программ;
- сопровождение внедренных программ и программных средств;
- участие в разработке форм документов, подлежащих машинной обработке [4].

Для выполнения возложенных на него функций инженер-программист обязан:

- разрабатывать технологию, этапы и последовательность написания приложения;
- осуществлять выбор языка программирования и перевод на него используемых моделей и алгоритмов задач;
- определять информацию, подлежащую обработке на ЭВМ, ее объемы, структуру, макеты и схемы ввода, обработки, хранения и выдачи информации, методы ее контроля;
- определять объем и содержание данных тестовых примеров, обеспечивающих наиболее полную проверку соответствия программ их функциональному назначению;
- выполнять работу по подготовке программ к отладке и проводить отладку;
- разрабатывать инструкции по работе с программами, оформлять необходимую техническую документацию;
- определять возможность использования готовых программных средств;
- осуществлять сопровождение внедренных программ и программных средств;
- определять совокупность данных, обеспечивающих решение максимального числа условий, включенных в программу, выполнять работу по ее подготовке к отладке;
- проводить отладку разработанных программ, корректировать их в процессе доработки.

Инженер-программист имеет право:

- знакомиться с проектами решений руководства предприятия, касающимися его деятельности;
- вносить на рассмотрение руководства предложения по совершенствованию работы, связанной с обязанностями, предусмотренными настоящей инструкцией;

- получать от руководителей структурных подразделений, специалистов информацию и документы, необходимые для выполнения своих должностных обязанностей;

- привлекать специалистов всех структурных подразделений предприятия для решения возложенных на него обязанностей (если это предусмотрено положениями о структурных подразделениях, если нет - с разрешения руководителя предприятия);

- требовать от руководства предприятия оказания содействия в исполнении своих должностных обязанностей и прав;

Инженер-программист несет ответственность:

- за неисполнение (ненадлежащее исполнение) своих должностных обязанностей, предусмотренных настоящей должностной инструкцией, в пределах, определенных действующим трудовым законодательством Республики Беларусь;

- за совершенные в процессе осуществления своей деятельности правонарушения – в пределах, определенных действующим административным, уголовным и гражданским законодательством Республики Беларусь;

- за причинение материального ущерба – в пределах, определенных действующим трудовым, уголовным и гражданским законодательством Республики Беларусь.

1.3 Охрана труда предприятия ОАО «Туровский молочный комбинат»

Охрана труда – это система обеспечения безопасности жизни и здоровья работников в процессе трудовой деятельности, включающая правовые, социально-экономические, организационные, технические, психофизиологические, санитарно-гигиенические, лечебно-профилактические, реабилитационные и иные мероприятия и средства [5].

Вся деятельность в области охраны труда на туровском молочном комбинате регламентирована действующим законодательством Республики Беларусь (РБ), санитарными нормами и правилами, гигиеническими нормативами, предписаниями надзорных органов.

Общие требования безопасности на предприятии:

- к самостоятельной работе инженера-программиста допускаются лица в возрасте не моложе 18 лет, прошедшие соответствующую подготовку, инструктаж по охране труда, не имеющие противопоказаний по состоянию здоровья и прошедшие проверку знаний по электробезопасности.

- Инженер-программист обязан соблюдать правила внутреннего трудового распорядка, установленные режимы труда и отдыха, правила техники безопасно-

сти в помещениях предприятия, правила по охране труда при работе с ЭВМ; выполнять только ту работу, которая определена инструкцией по эксплуатации оборудования и должностными инструкциями, утвержденными администрацией предприятия, и при условии, что безопасные способы ее выполнения хорошо известны.

- Соблюдение правил пожарной безопасности, знание места расположения первичных средств пожаротушения и направления эвакуации при пожаре.

- При несчастном случае немедленно сообщить об этом администрации учреждения.

Требования безопасности перед началом работы:

- Включить полностью освещение кабинета и убедиться в исправной работе светильников. Рабочее место по отношению к световым проемам должно располагаться так, чтобы естественный свет падал сбоку, преимущественно слева. В целях профилактики переутомления и перенапряжения при работе с дисплеями необходимо выполнять во время регламентных перерывов комплексы специальных упражнений.

- Рабочий стол должен регулироваться по высоте в пределах 680 – 760 мм; под столешницей рабочего стола должно быть свободное пространство для ног с размерами по высоте не менее 600 мм, по ширине 500 мм, по глубине – 650 мм.

- Убедиться в исправности электрооборудования кабинета: светильники должны быть надежно подвешены к потолку и иметь светорассеивающую арматуру; электрические коммутационные коробки должны быть закрыты, а электророзетки – фальшвилками; корпуса и крышки выключателей и розеток не должны иметь трещин и сколов, а также оголенных контактов.

Требования безопасности в аварийных ситуациях:

- при возникновении ситуации, которые могут привести к несчастным случаям, остановить работу и сообщить о возникшей ситуации руководителю предприятия;

- при появлении запаха гари, дыма или искр в токоведущих частях немедленно выключить компьютерную технику, при возникновении возгорания немедленно отключить оборудование, обесточить электросеть за исключением осветительной сети, сообщить о пожаре всем работающим и приступить к тушению очага загорания имеющимися средствами пожаротушения;

- при прекращении подачи электроэнергии отключить оборудование.

Требования безопасности по окончании работы:

- отключить от электросети технические средства;

- привести в порядок рабочее место;

- доложить руководителю обо всех недостатках, выявленных в процессе работы, и о мерах, принятых к их устранению.

2 АРХИТЕКТУРА ПРОГРАММНОГО КОМПЛЕКСА ДЛЯ АВТОМАТИЗАЦИИ СИСТЕМЫ УЧЁТА РАБОТ С ПРИЕЗЖАЮЩИМИ МОЛОКОВОЗАМИ

Архитектура программного обеспечения – совокупность важнейших решений об организации программной системы [6].

Архитектура включает:

- выбор структурных элементов и их интерфейсов, с помощью которых составлена система, а также их поведения в рамках сотрудничества структурных элементов;
- соединение выбранных элементов структуры и поведения во всё более крупные системы;
- архитектурный стиль, который направляет всю организацию – все элементы, их интерфейсы, их сотрудничество и их соединение.

Документирование архитектуры программного обеспечения упрощает процесс коммуникации между разработчиками, позволяет зафиксировать принятые проектные решения и предоставить информацию о них эксплуатационному персоналу системы, повторно использовать компоненты и шаблоны проекта в других.

2.1 Анализ существующих аналогов

2.1.1 1С:Предприятие 8. Бухгалтерия молокозавода

Отраслевое решение "1С:Предприятие 8. Бухгалтерия молокозавода" разработано на базе "1С:Предприятие 8. Бухгалтерия предприятия" и предназначено для автоматизации бухгалтерского и налогового учета, включая подготовку обязательной (регламентированной) и специализированной отчетности на молокоперерабатывающих предприятиях и обеспечивает автоматизацию [7].

Учет поступления молочного сырья от поставщиков. Учет поступления молочного сырья от поставщиков производится с регистрацией качественных показателей (критериев), на основании которых определяется сортность сырья с последующим автоматическим расчетом зачетного веса и закупочной цены сырья. Для расчета закупочной цены применяется специализированный алгоритм расчета, использующий базовые цены сырья по контрагентам, надбавки за качественные показатели (жирность, белок) и коэффициенты за сортность. Состав качественных показателей и сортов для каждого вида сырья определяется пользователем.

При приемке сырья предусмотрена возможность ведения учета поступления в трех единицах измерения: физический вес, зачетный вес, единица складского учета.

Учет выпускаемой продукции. Учет выпускаемой продукции производится, как в основных, так и в дополнительных единицах измерения, пересчитываемых через коэффициент. Параметры выпуска продукции имеют связь с качественными показателями сырья, например, % жира, % белка, что позволяет при расчете списания сырья на выпуск продукции корректно рассчитывать нормативы расхода. Нормативы могут быть заданы, как числовым методом, так и в виде формул или таблиц, зависящих от параметров. Расходовать сырье на производство можно как по нормативному методу, так и по фактическим данным.

2.1.2 Система учета молока Flexmilk

Автоматизированная система учета приемки и движения молока на производственном перерабатывающем предприятии Предлагаемый программно-аппаратный комплекс предназначен для объединения различных устройств учета приемки и движения молока на предприятии в единую сеть – под одним аппаратным и программным интерфейсом. Это позволяет проводить учет объема молока в реальном режиме времени со всеми необходимыми качественными параметрами, отслеживать его дальнейшее движение на переработку, и вести единый архив по приемке каждого молоковоза (секции молоковоза) [8].

Автоматизированная система учета энергоресурсов

- Производить запись в реальном режиме времени информации о количестве принятого молока
- Производить запись параметров принятого молока в одном отчете (в привязке к принятому объему)
- обеспечить оперативный доступ к информации о движении молока на предприятии различным категориям пользователей и мгновенно информировать о каких-либо отклонениях или сбоях в работе оборудования (отсутствии сигналов)

Предлагаемое решение предусматривает объединение всех устройств контроля приемки молока (анализатор молока, счетчики-расходомеры, самописец на пастеризаторе и пр.) в общую цифровую сеть (посредством протокола и контроллера приема импульсных сигналов с расходомера) для централизованного сбора и обработки данных.

Преимущества системы: интерфейс и мобильный интерфейс для работы с системой

Позволяет контролировать приемку молока (текущее состояние и архивы) с любого устройства в любой точке мира – достаточно иметь подключение к

интернет и быть зарегистрированным пользователем в системе с соответствующим правом доступа.

Централизованный сбор данных и облачная архитектура

Данные со всех устройств собираются в едином хранилище — в базе данных, которая установлена на сервере (как внешнем, так и внутреннем).

- Автоматическая обработка Система круглосуточно собирает и обрабатывает данные, устраняет человеческий фактор.
- Удалённый контроль Данные по приемке молока доступны пользователям для просмотра через сеть интернет в любое время суток.
- Мгновенное оповещение Сервер выполняет функцию оповещения, при каких либо отклонениях в работе системы в процессе приемки и\или движения молока. В таких ситуациях указанные в системе пользователи получают электронное сообщение.
- Низкая стоимость обслуживания. При нормальной работе не требуется ручных действий со стороны обслуживающего персонала.
- Гибкая настройка и расширение. Возможно добавление дополнительных устройств сбора данных, работающих с различными параметрами.

2.1.3 Автоматизированные системы учета молока "Поток Альфа"

Сырьевой цех является первичным подразделением любого молочного предприятия и один из наиболее важных участков этого цеха – приемка молока. Важность этого участка определяется тем, что здесь происходит входной учет принимаемого сырья, необходимый для расчета конечной себестоимости производимых продуктов [9].

Недостатки существующих методов учета сырья

1. Весовой метод

- необходимость двойного взвешивания (брутто и тара);
- наличие на автоцистерне при первом взвешивании льда и грязи после опорожнения и мойки автоцистерны вносят при повторном взвешивании существенную погрешность измерения;
- перерабатывающее предприятие расплачивается не за принятое в танки хранения сырье, а за ввезенное на его территорию (потери при сливе).

2. Поточный метод (механические расходомеры)

- низкая надежность;
- проблемы при мойке;
- невозможность прямого расчета в «жирокилограммах»;
- дополнительная значительная погрешность при пересчете в весовые единицы (до 2,0% от принимаемой партии сырья);

3. Поточный метод (электромагнитные расходомеры)

- невозможность прямого расчета в «жирокилограммах»;

– дополнительная значительная погрешность при пересчете в весовые единицы (до 1,5% от принимаемой партии сырья);

2.1.4 Автоматизированная система управления постом приема молока

Созданная ООО «АСУ-Техно» Система построена на базе клиент-серверной технологии и состоит из OPC-сервера, который опрашивает все данные с контроллеров, архиватора и программы АРМ. Количество компьютеров, находящихся в заводской сети, на которых можно открыть программу АРМ неограниченно [10].

В рамках этого проекта компания «АСУ-Техно» выполнила следующий комплекс работ:

- 1) Изготовление шкафа управления;
- 2) Разработка программного обеспечения нижнего уровня;
- 3) Разработка программного обеспечения верхнего уровня (АРМ-Лаборанта, АРМ-оператора, программа «технологический сервер»);
- 4) Электромонтажные работы, пусконаладочные работы, обучение персонала, ввод в эксплуатацию.

Шкаф управления выполнен на базе следующих комплектующих:

- 1) Конструктив – Rittal, степень защиты IP65 со встроенной решеткой и вентилятором охлаждения;
- 2) Контроллерное оборудование – Siemens;
- 3) Панель оператора – 10'', Weintek;
- 4) Преобразователь частоты – ABB;
- 5) Сигнальная колонна, источник питания, пускозащитная аппаратура – EATON;
- 6) Пневматика – SMC, пневмотрубка специального исполнения (хим- и термостойкая), тройники и фитинги с цангами из нержавеющей стали.

Шкаф управления соответствует требованиям ТР ТС 004/2011 «О безопасности низковольтного оборудования» и ТР ТС 020/2011 «Электромагнитная совместимость технических средств».

Созданное программное обеспечение состоит из:

1. Программы для ПЛК;
2. Программы для панели оператора;
3. Программы верхнего уровня (технологический сервер архивирования, АРМ-оператора, АРМ-лаборанта).

2.1.5 «ISOFT: Управление молокозаводом»

Специализированное решение «ISOFT: Управление молокозаводом» разработано специалистами компании «ЮКОЛА-ИНФО-Брест» на платформе

«1С:Предприятие 8» для автоматизации молочного производства. Использование данного программного продукта позволяет организовать единую информационную систему (ИС) учета на молокоперерабатывающем предприятии [11].

"1SOFT: Управление молокозаводом" позволяет автоматизировать следующие функции:

- Управление производством;
- Управление запасами и закупками;
- Управление продажами;
- Управление финансами;
- Управление техобслуживанием и ремонтами;
- Управление автотранспортом.

"1SOFT: Управление молокозаводом" включает в себя следующие подсистемы:

1. Подсистема управления транспортной логистикой
 - Формирование маршрутов доставки готовой продукции собственным и заказным транспортом, с учетом загрузки транспорта;
 - Интеграция с системами оптимизации маршрутов;
 - Контроль веса автотранспорта на въезде/выезде с территории молокозавода.
2. Подсистема учета молочного сырья
 - Учет поступления молока с автоматическим пересчетом физического веса в зачетный вес, расчетом цены молока, расчетом доплаты за белок, доплаты за сбор молока от населения;
 - Учет взаиморасчетов с поставщиками молока;
 - Интеграция с электронными счетчиками молока;
 - Регистрация основных показателей сырья: органолептическая оценка, кислотность, жирность, температура, плотность, степень чистоты, соматические клетки, сорт и т.д.
 - Автоматический расчет закупочной стоимости молочного сырья;

Провести корректное сравнение данных программ не получится, поскольку они построены немного для разных целей, хоть и являются подобными. Первое приложение идет больше для учета данных по всему молокозаводу, чем по отдельному учету работы с молокозаводом, а вторая больше по внутреннему учету движения молока. Оставшиеся описанные программы являются комплексным решением для ферм, что не подходит к текущему случаю.

Таким образом, был сделан вывод о том, что приобретение готового программного продукта является нецелесообразным. Главным образом этот выбор основан на том, что имеющиеся системы обладают избыточным функционалом и, как следствие, имеют высокую стоимость, что противоречит условиям заказчика. Кроме того, **указанные программы** мало ориентированы под нашу задачу.

2.2 Функциональная архитектура приложения

Функциональная архитектура представляет собой набор функций и их подфункций, определяющих преобразования, осуществляемые системой при выполнении своего назначения [12].

Функциональная архитектура приложения описывает принцип работы, а также демонстрирует его возможности:

а) безопасность – представлена в виде ограниченного доступа к данным, что не позволит не пользователю без необходимого права доступа работать с данными базой данных (БД);

б) разделение на роли – представлено в виде предоставления допуска к данным для роли менеджера программы;

в) справочная информация – представлена в виде отдельных форм, на которых показана информация о сотрудниках, постах и молоковозах;

г) бизнес-логика – представлена несколькими таблицами, а именно: "сотрудник", "пост", "молоковоз", "приезд молоковоза", "обслуживание", каждая из которых реализует связанную между собой цепочку алгоритма;

д) доступ к данным – отображает, как происходит обмен данными в приложении, откуда они берутся и какой исходный результат получает пользователь при работе с полученными данными;

е) интерфейс – предоставляет информацию о том, какие модули приложения будут предоставлены пользователю при работе с приложением через браузер;

2.3 Логическая модель базы данных

Сущность - Это реальный или воображаемый объект, информация о котором представляет интерес. В диаграммах Ег-модели сущность представляется в виде прямоугольника, содержащего имя сущности. При этом имя сущности - это имя типа, а не конкретного объекта - экземпляра этого типа. Каждый экземпляр сущности должен быть отличим от любого экземпляра той же сущности. В зависимости от режима представления диаграммы прямоугольник может содержать имя сущности, ее описание, список ее атрибутов и другие требования [13].

В рамках решаемой задачи в процессе разработки логической модели было выделено 5 сущностей: "сотрудник", "пост", "молоковоз", "приезд молоковоза", "обслуживание".

– "сотрудник", содержит информацию о сотрудниках предприятия, которые работают с молоковозами

- "пост", содержит информацию о постах, на которые приезжают молоковозы
- "молоковоз", содержит информацию о машине, которая привозит молоко на пост
- "приезд молоковоза", содержит информацию о том, когда куда и какой молоковоз приехал, сколько молока привез и что с ним стало
- "обслуживание", содержит информацию об сотрудниках, которые обслуживали выбранный приезд молоковоза

Между сущностями определена связь «один ко многим».

Логическая структура базы данных, разработанного приложения представлена на рисунке 2.1.

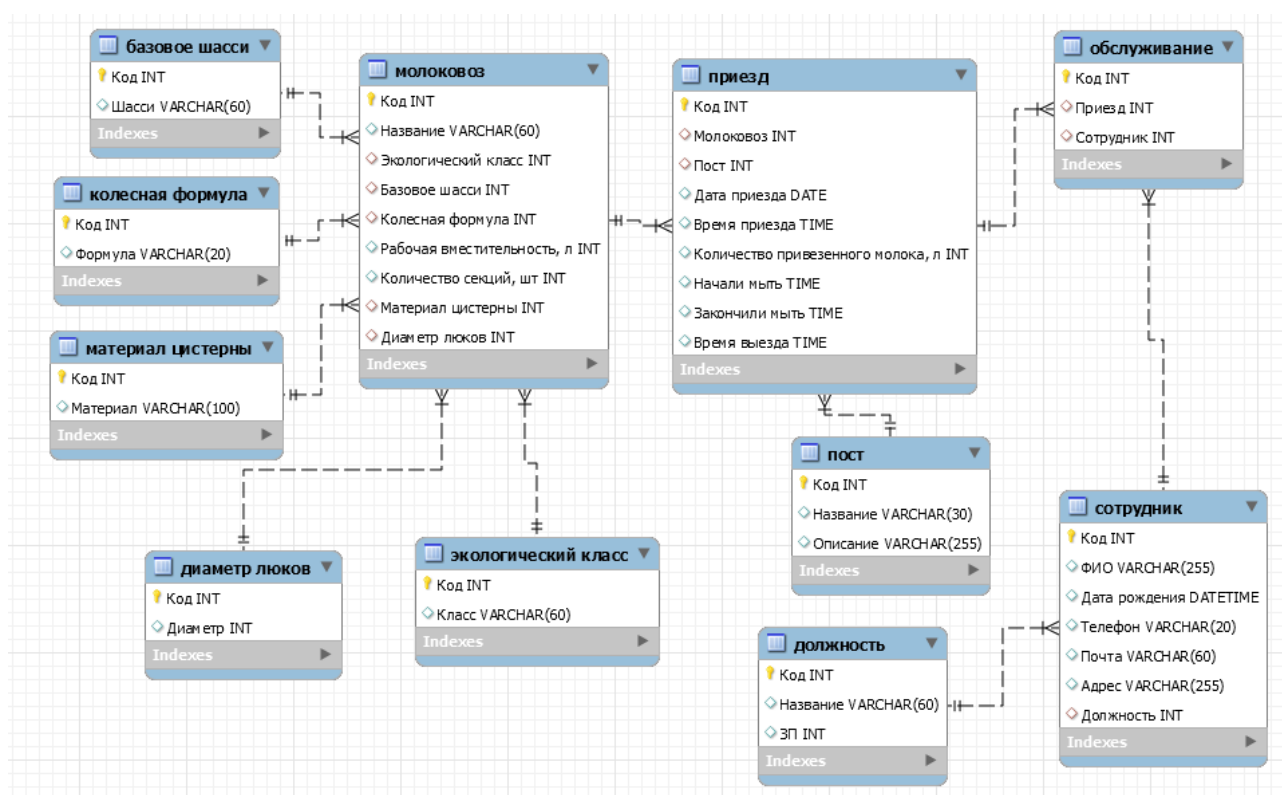


Рисунок 2.1 – Логическая структура базы данных

2.4 Физическая модель базы данных

На основе логической модели описанной выше была разработана физическая модель.

База данных содержит 11 таблиц: "сотрудник", "пост", "молоковоз", "приезд молоковоза", "обслуживание", "базовое шасси", "колесная формула", "материал цистерны", "диаметр люков", "экологический класс", "должность".

Структуры таблиц разработанной БД приведены в таблицах 2.1 – 2.11.

В таблице 2.1 представлена информация о полях и типах данных таблицы «сотрудник».

Таблица 2.1 – Таблица «сотрудник»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>ФИО</i>	<i>varchar(255)</i>
<i>Дата рождения</i>	<i>date</i>
<i>Телефон</i>	<i>varchar(20)</i>
<i>Почта</i>	<i>varchar(60)</i>
<i>Адрес</i>	<i>varchar(255)</i>
<i>Должность</i>	<i>int(11)</i>

В таблице 2.2 представлена информация о полях, типах данных таблицы «пост».

Таблица 2.2 – Таблица «пост»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Название</i>	<i>varchar(30)</i>
<i>Описание</i>	<i>varchar(255)</i>

В таблице 2.3 представлена информация о полях, типах данных таблицы «молоковоз».

Таблица 2.3 – Таблица «молоковоз»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Название</i>	<i>varchar(60)</i>
<i>Экологический класс</i>	<i>int(11)</i>
<i>Базовое шасси</i>	<i>int(11)</i>
<i>Колесная формула</i>	<i>int(11)</i>
<i>Рабочая вместительность, л</i>	<i>int(11)</i>
<i>Количество секций, шт</i>	<i>int(11)</i>
<i>Материал цистерны</i>	<i>int(11)</i>
<i>Диаметр люков</i>	<i>int(11)</i>

В таблице 2.4 представлена информация о полях, типах данных таблицы «приезд молоковоза».

Таблица 2.4 – Таблица « приезд молоковоза »

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Молоковоз</i>	<i>int(11)</i>
<i>Пост</i>	<i>int(11)</i>
<i>Дата приезда</i>	<i>date</i>
<i>Время приезда</i>	<i>time</i>
<i>Количество привезенного молока, л</i>	<i>int(11)</i>
<i>Начали мыть</i>	<i>time</i>
<i>Закончили мыть</i>	<i>time</i>
<i>Время выезда</i>	<i>time</i>

В таблице 2.5 представлена информация о полях, типах данных таблицы «обслуживание».

Таблица 2.5 – Таблица « обслуживание »

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Приезд</i>	<i>int(11)</i>
<i>Сотрудник</i>	<i>int(11)</i>

В таблице 2.6 представлена информация о полях, типах данных таблицы «Экологический класс».

Таблица 2.6 – Таблица « Экологический класс»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Класс</i>	<i>varchar(60)</i>

В таблице 2.7 представлена информация о полях, типах данных таблицы «Базовое шасси».

Таблица 2.7 – Таблица «Базовое шасси»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Шасси</i>	<i>varchar(60)</i>

В таблице 2.8 представлена информация о полях, типах данных таблицы «Колесная формула».

Таблица 2.8 – Таблица «Колесная формула»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Формула</i>	<i>varchar(20)</i>

В таблице 2.9 представлена информация о полях, типах данных таблицы «Материал цистерны».

Таблица 2.9 – Таблица «Материал цистерны»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Материал</i>	<i>varchar(100)</i>

В таблице 2.10 представлена информация о полях, типах данных таблицы «Диаметр люков».

Таблица 2.10 – Таблица «Диаметр люков»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Диаметр</i>	<i>int(11)</i>

В таблице 2.11 представлена информация о полях, типах данных таблицы «Должность».

Таблица 2.11 – Таблица «Должность»

Имя поля	Тип данных
<i>Код</i>	<i>int(11)</i>
<i>Название</i>	<i>varchar(60)</i>
<i>ЗП</i>	<i>int(11)</i>

2.5 Программная архитектура приложения

Для представления архитектуры, а точнее – различных входящих в нее структур, удобно использовать графические языки. На настоящий момент наиболее проработанным и наиболее широко используемым из них является унифицированный язык моделирования (*Unified Modeling Language, UML*) [14].

Основная цель создания любой программной системы – создание такого программного продукта, который помогает пользователю выполнять свои повседневные задачи.

Для создания таких программ первым делом определяются требования, которым должна удовлетворять система. Для того чтобы более точно понять как должна работать система, используется описание функциональности системы через варианты использования.

Варианты использования отражают функциональность системы с точки зрения получения значимого результата для пользователя.

Варианты использования предназначены в первую очередь для определения функциональных требований к системе и управляют всем процессом разработки.

На рисунке 2.2 представлена диаграмма вариантов использования разработанной системы.



Рисунок 2.2 – Диаграмма вариантов использования

На диаграмме вариантов использования актерами являются люди в лице ответственного за оформления работ с приезжающими молоковозами.

В функции ответственного входят варианты использования:

- просмотр информации о приездах, в которых участвовал выбранный сотрудник;
- добавление, редактирование, удаление данных о сотрудниках;
- добавление, редактирование, удаление данных о сотрудниках, работавших в определенный приезд;
- добавление, редактирование, удаление данных о молоковозах;
- добавление, редактирование, удаление данных о приезде молоковоза и работ с ним;
- добавление, редактирование, удаление данных о постах;
- добавление, редактирование, удаление данных о должностях;
- добавление, редактирование, удаление данных об экологических классах;

- добавление, редактирование, удаление данных о материалах цистерны;
- добавление, редактирование, удаление данных о базовых шасси;
- добавление, редактирование, удаление данных о диаметре люков;
- добавление, редактирование, удаление данных о колесных формулах.

Диаграмма классов определяет типы классов системы и различного рода статические связи, которые существуют между ними. На диаграммах классов изображаются также атрибуты классов, операции классов и ограничения, которые накладываются на связи между классами. Вид и интерпретация диаграммы классов существенно зависит от точки зрения (уровня абстракции): классы могут представлять сущности предметной области или элементы программной системы.

Основными элементами являются классы и связи между ними. Классы характеризуются при помощи атрибутов и операций.

Атрибуты описывают свойства объектов класса. Имя атрибута должно быть уникально в пределах класса. Каждый класс имеет свое название и свои атрибуты и операции, которые впоследствии будут реализованы в программном коде. За именем атрибута следует его тип.

На основе анализа предметной области, были выявлены следующие классы-сущности, рассмотренные выше (подраздел 2.2). Модель предметной области представлена в виде диаграммы классов на рисунке 2.3.

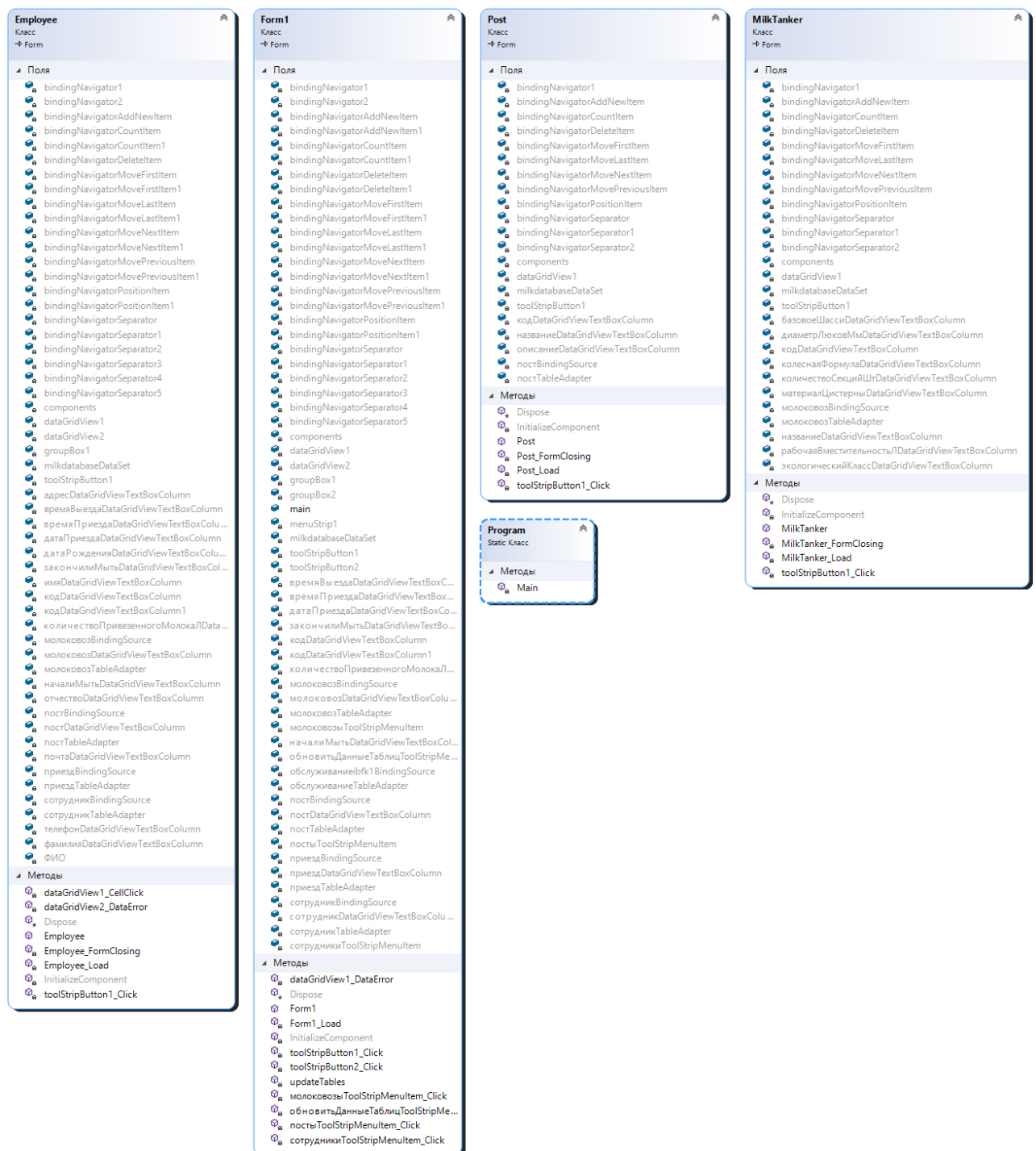


Рисунок 2.3 – Диаграмма классов

Для моделирования процесса выполнения операций в языке *UML* используются диаграммы деятельности.

Диаграмма деятельности – блок-схема, которая показывает, как поток управления переходит от одной деятельности к другой, при этом внимание фиксируется на результате деятельности. Результат может привести к изменению состояния системы или возвращению некоторого значения.

Графически диаграмма деятельности представляется в форме графа деятельности, вершинами которого являются состояния действия или состояния деятельности, а дугами – переходы от одного состояния действия/деятельности к другому [15]. Каждая диаграмма деятельности должна иметь единственное начальное и единственное конечное состояния (на практике иногда можно видеть несколько конечных состояний на одной диаграмме, но это одно и то же состояние, изображенное несколько раз для лучшей читабельности диаграммы). Саму диаграмму деятельности принято располагать таким образом, чтобы действия следовали сверху вниз. В этом случае начальное состояние будет изображаться в верхней части диаграммы, а конечное – в ее нижней части.

На рисунке 2.4 представлена диаграмма деятельности бизнес процесса для потока прецедента «Работа по приезду молоковоза».

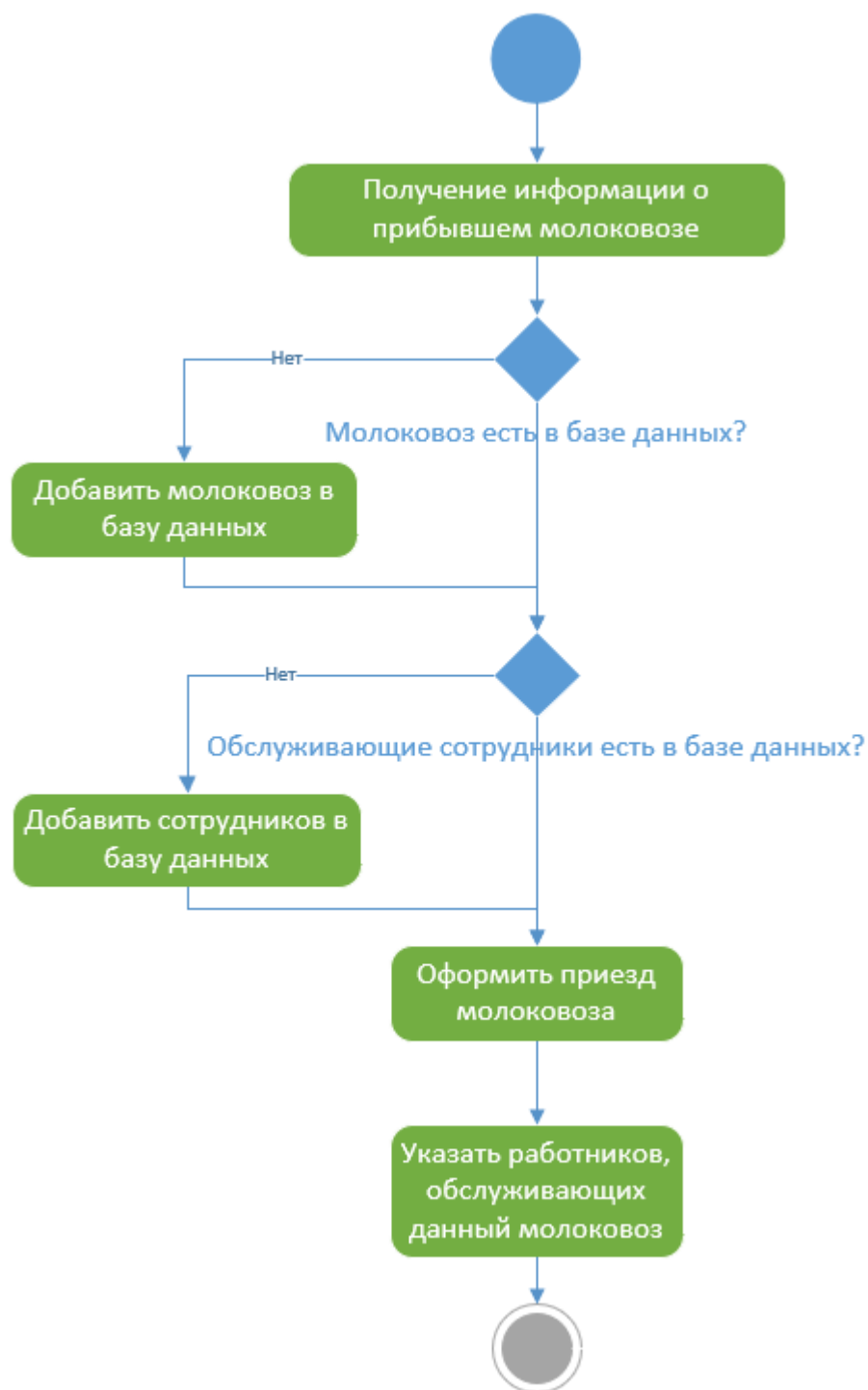


Рисунок 2.4 – Диаграмма деятельности бизнес процесса

Диаграмма деятельности 2.4 используется для получения представления о последовательности состояний (страниц) и переходов приложения. Данная диаграмма позволяет помочь определиться с навигацией программного обеспечения (ПО).

Ответственный первым делом получает информацию о молоковозе, который приехал, и при необходимости заносит его в справочник молоковозов. Также, заносит в систему информацию о тех сотрудниках, которые еще не были

внесены в систему. Наконец, после проведенных с молоковозом работ о его приезде вносится информация в систему, после чего к текущему приезду вносятся связи всех сотрудников, которые участвовали в обслуживанию данного молоковоза.

2.6 Настройка программного комплекса

Для работы приложения необходим персональный компьютер, на котором будет располагаться само приложение.

Минимальные требования следующие: минимум 1024 МБ ОЗУ (рекомендуется 4 Гб или выше), 64-разрядная система, оборудованная процессором *AMDATAHLON4800*, *INTELCoreI3-3110* и выше. Установленная база данных MySQL 8, .NET Framework 4.6.

Действия для установки и настройки сервера СУБД:

- загрузить актуальную версию дистрибутива сервера СУБД *Workbench MySQL*;

- используя, мастер установки загруженного дистрибутива установить СУБД, указав пароль и логин для системных пользователей. В приложении используется следующая информация:

Логин – root

Пароль - 1234

Далее запустить установленный программный продукт и ввести логин и пароль, который был задан в процессе установки.

Для подключения к БД необходимо использовать файл-конфигурации с именем «*App.config*», который располагается в файлах проекта.

Зайдя в данный файл можно изменить настройки подключения к базе данных, открыв его в любом текстовом редакторе, например в блокноте.

Пример конфигурационного файла, разработанного приложения, представлен на рисунке 2.5.

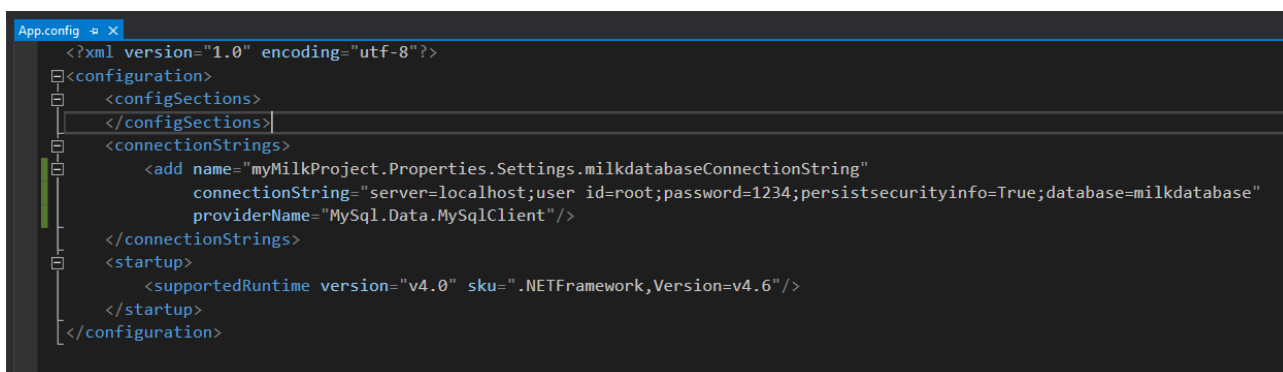


Рисунок 2.5 – Конфигурационный файл

В рассматриваемом примере показано, как указать строку подключения.

Эти параметры задают имя строки подключения, сервер, на котором размещена база данных, имя базы данных и учетные данные пользователя для доступа к базе данных.

Для запуска приложения необходимо создать ярлык с адресом exe файла, на рабочий стол и запустить его.

ЗАКЛЮЧЕНИЕ

За время прохождения преддипломной практики были приобретены навыки в практической работе на предприятии ОАО «Туровский молочный комбинат», а также изучена структура данного предприятия, применяемые технологии, принципы управления и работы.

По прибытию на предприятие был пройден вводный инструктаж по охране труда и технике безопасности на рабочем месте.

При прохождении практики были выполнены все поставленные задачи, все задачи прошли проверку у руководителя по практике от предприятия, это подтверждает, что задание было выполнено верно.

В результате прохождения преддипломной практики была создана автоматизированная система учёта оформления приезда и проведенных работ с молоковозами.

Разработанное приложение имеет простой и понятный для работы интерфейс.

Также были расширены и закреплены знания объектно-ориентированного языка программирования *C#* и базы данных *Workbench MySQL*.

При реализации работы были пройдены этапы описания и постановки задачи, кодирование программы на алгоритмический язык и тестирование полученного приложения.

Благодаря получившемуся приложению, люди смогут получать информацию об интересующей их информации в кратчайшие сроки и в удобной для них форме.

Разработанный проект даёт возможность глубже изучить пройденный материал, позволяет закрепить навыки решения поставленных задач и научиться поиску необходимой для этого информации.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Общая характеристика информационных систем [Электронный ресурс]. – Режим доступа: <https://mydocx.ru/3-57165.html>. – Дата доступа: 13.03.2021.
2. История ОАО «Туровский молочный комбинат» [Электронный ресурс]. – Режим доступа: <https://turovmilk.by/about/history/>. – Дата доступа: 13.03.2021.
3. Туровский молочный комбинат ОАО [Электронный ресурс]. – Режим доступа: <https://www.produktgoda.by/2013-10-21-13-19-04/item/turovskij-molochnyj-kombinat-oao-3>. – Дата доступа: 13.03.2021.
4. Должностная инструкция программиста на предприятии [Электронный ресурс]. – Режим доступа: <https://hr-portal.info/job-description/dolzhnostnaya-instruktsiya-programmista>. – Дата доступа: 13.03.2021.
5. Охрана труда [Электронный ресурс]. – Режим доступа: <http://multilang.pravo.by/ru/Term/Index/525?langName=ru&ch=B&size=25&page=81&type=0>. – Дата доступа: 13.03.2021.
6. Гагарина, Л. Г. Введение в архитектуру программного обеспечения. Учебное пособие / Л.Г. Гагарина, А.Р. Федоров, П.А. Федоров. - М.: Инфра-М, Форум, 2016. - 320 с.
7. 1С:Предприятие 8. Бухгалтерия молокозавода [Электронный ресурс]. – Режим доступа: <https://solutions.1c.ru/catalog/buh-molzavod/features>. – Дата доступа: 13.03.2021.
8. Система учета молока FlexMilk [Электронный ресурс]. – Режим доступа: <https://flexsys.com.ua/works/cistema-ucheta-moloka-flexmilk/>. – Дата доступа: 13.03.2021.
9. Автоматизированные системы учета молока "Поток Альфа" [Электронный ресурс]. – Режим доступа: <http://www.ib-a.ru/article/автоматизированные-системы-учета-молока-поток-альфа>. – Дата доступа: 17.03.2021.
10. Автоматизированная система управления постом приема молока [Электронный ресурс]. – Режим доступа: <https://asu-techno.by/ru/molochnaya/69-avtomatizirovannaya-sistema-upravleniya-postom-priema-moloka.html>. – Дата доступа: 17.03.2021.
11. 1SOFT: Управление молокозаводом [Электронный ресурс]. – Режим доступа: <https://1soft.by/product/1soft-upravlenie-molokozavodom.html>. – Дата доступа: 17.03.2021.
12. Энкарначчо, Ж. Автоматизированное проектирование: Основные понятия и архитектура систем / Ж. Энкарначчо, Э. Шлехтендаль. - М.: Радио и связь, 2013. - 288 с

13. Кузнецов, С.Д. Основы баз данных: учебное пособие[Текст]/.- М.:Бином, 2007-484 с.

14. Диаграммы UML [Электронный ресурс]. – Режим доступа: <https://bstudy.net/915235/tehnika/diagrammy>. – Дата доступа: 13.03.2021.

15. Элементы графической нотации диаграммы деятельности [Электронный ресурс]. – Режим доступа: <https://intuit.ru/studies/courses/32/32/lecture/1020>. – Дата доступа: 13.03.2021.

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг программных модулей

```
// подключение библиотек для работы кода
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;
//определение зоны видимости
namespace myMilkProject
{
    //создание класса с точкой входа в программу
    static class Program
    {
        /// <summary>
        /// Главная точка входа для приложения.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            // запуск главной формы приложения
            Application.Run(new Form1());
        }
    }
}
//подключение библиотек для работы кода
using System;
using System.Windows.Forms;
//определение зоны видимости
namespace myMilkProject
{
    //создание класса наследника формы для работы с постами
    public partial class Post : Form
    {
        //создание публичного конструктора без полей
        public Post()
        {
            //вызов инициализации всех компонентов формы
            InitializeComponent();
        }
        //метод, работающий при первом отображении формы
        private void Post_Load(object sender, EventArgs e)
        {
            //заполнение таблицы постов данными из базы данных
            this.postTableAdapter.Fill(this.milkdatabaseDataSet.пост);
        }
        //метод, отрабатывающий при закрытии формы
        private void Post_FormClosing(object sender, FormClosingEventArgs e)
        {
            // при закрытии формы отобразить на экране главную форму приложения
            Form1.main.Show();
        }
        // метод нажатия кнопки сохранить
        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            //написание обработчика ошибок, чтоб в случае чего вывести пользователю сообщение
            try
            {
                //обновление данных в таблице постов, беря данные из таблицы, отображаемой на форме
                postTableAdapter.Update(milkdatabaseDataSet.пост);
                //вывод пользователю сообщения об успешном обновлении данных
                MessageBox.Show("Изменения сохранены");
            }
            catch { }
        }
    }
}
```

```

//обновление данных в таблице на форме приложения
        this.постTableAdapter.Fill(this.milkdatabaseDataSet.пост);
    } //часть кода, вызываемая в случае ошибки при сохранении данных
    catch (Exception ex) {
        //вывести пользователю сообщение об ошибке и код этой ошибки
        MessageBox.Show("Ошибка при сохранении данных.\n"+ex.Message); }
    }
}

//подключение библиотек для работы кода
using System;
using System.Windows.Forms;
//определение зоны видимости
namespace myMilkProject
{
    //создание класса наследника формы для работы с молоковозами
    public partial class MilkTanker : Form
    {
        //создание конструктора без параметров
        public MilkTanker()
        {
            //инициализация всех полей формы
            InitializeComponent();
        }

        //метод, обрабатываемый при первом отображении формы
        private void MilkTanker_Load(object sender, EventArgs e)
        {
            //заполнить таблицу данными из базы данных
            this.молоковозTableAdapter.Fill(this.milkdatabaseDataSet.молоковоз);
        }

        //метод, обрабатываемый при нажатии кнопки сохранить
        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            //обработчик ошибок, чтоб в случае чего вывести пользователю соответствующее уведомление
            try
            {
                //обновление данных в таблице молоковоза из базы данных
                молоковозTableAdapter.Update(milkdatabaseDataSet.молоковоз);
                //вывод пользователю сообщения об успешном обновлении данных
                MessageBox.Show("Изменения сохранены");
                //обновить таблицу молоковозов, выведя в нее все данные из таблицы молоковозов
                this.молоковозTableAdapter.Fill(this.milkdatabaseDataSet.молоковоз);
            }
            catch (Exception ex) {
                //код, обрабатываемый в случае ошибки при обновлении данных
                //выводит пользователю сообщение об ошибке и код этой ошибки
                MessageBox.Show("Ошибка при сохранении данных.\n" + ex.Message); }
        }

        //метод, обрабатываемый при закрытии формы
        private void MilkTanker_FormClosing(object sender, FormClosingEventArgs e)
        {
            //отдает команду показать пользователю главную форму приложения
            Form1.main.Show();
        }
    }
}

//подключение библиотек для работы кода
using System;
using System.Windows.Forms;
//определение зоны видимости
namespace myMilkProject
{
    //создание класса наследника формы для отображения пользователю главной формы приложения
    public partial class Form1 : Form
    {
        //публичная переменная типа текущей формы, чтобы из других форм можно было к ней обратиться
        //и вызвать метод отображения формы на экране
        public static Form1 main;
        //определение конструктора без аргументов

```

```

public Form1()
{
    //инициализация всех полей формы
    InitializeComponent();
    //определение переменной типа текущей формы, присвоив ей значение текущей формы
    main = this;
}
//метод, обрабатываемый при первом появлении формы на экране
private void Form1_Load(object sender, EventArgs e)
{
    //вызвать метод обновления таблиц
    updateTables();
}
//метод, обрабатываемый при нажатии кнопки показа молоковозов
private void молоковозыToolStripMenuItem_Click(object sender, EventArgs e)
{
    //создать экземпляр класса формы молоковозов и отобразить его на экране
    new MilkTanker().Show();
    //скрыть текущую форму, чтоб на экране отображалась только одна
    this.Hide();
}
//метод, вызываемый при нажатии кнопки отобразить сотрудников
private void сотрудникиToolStripMenuItem_Click(object sender, EventArgs e)
{
    //создать экземпляр класса формы сотрудников и показать его на экране
    new Employee().Show();
    //скрыть текущую форму, чтоб за раз на экране отображалась только одна форму
    this.Hide();
}
//метод, вызываемый при нажатии кнопки показа всех постов
private void постыToolStripMenuItem_Click(object sender, EventArgs e)
{
    //создать экземпляр класса формы постов и отобразить его на экране
    new Post().Show();
    //скрыть текущую форму, чтоб в один момент времени на экране отображалась только одна форма
    this.Hide();
}
//метод, вызываемый при нажатии кнопки сохранить на таблице приезда
private void toolStripButton1_Click(object sender, EventArgs e)
{
    //конструкция обработки запроса, чтоб в случае чего вывести пользователю сообщение об ошибке
    try
    {
        //обновление данных в таблице приезда
        приездTableAdapter.Update(milkdatabaseDataSet.приезд);
        //вывести пользователю сообщение об удачном обновлении данных
        MessageBox.Show("Изменения сохранены");
        //заполнить таблицу актуальными данными из базы данных
        this.приездTableAdapter.Fill(this.milkdatabaseDataSet.приезд);
    } //эта часть кода вызывается в случае ошибки при обновлении данных в таблице
    базы данных
    catch (Exception ex) {
        //выводим пользователю сообщение об ошибке и текст этой ошибки
        MessageBox.Show("Ошибка при сохранении данных.\n" + ex.Message); }
    }
    //метод обработки неявных ошибок в таблице на форме
    private void dataGridView1_DataError(object sender, DataGridViewDataErrorEventArgs e)
    {
        //оставляем ее реализацию пустым, так как ошибки могут вызываться в случае простого
        косания
    }
    //метод, вызываемый при нажатии кнопки сохранить в таблице обслуживания
    private void toolStripButton2_Click(object sender, EventArgs e)
    {
        //конструкция обработки исключений, чтоб в случае чего мы могли вызвать пользовате-
        лю соответствующее окно
        try

```



```

        { //сохранение измененных данные в базу данных
            обслуживаниеTableAdapter.Update(milkdatabaseDataSet.обслуживание);
        } //вывод пользователю сообщения об успешном сохранении
        MessageBox.Show("Изменения сохранены");
        //заполнение таблицы актуальными данными из таблицы в базе данных
        this.обслуживаниеTableAdapter.Fill(this.milkdatabaseDataSet.обслуживание);
    } //эта часть кода вызывается в случае обнаружения ошибок при попытке обновить
    данные в базе данных
    catch (Exception ex) {
        //выводим пользователю уведомление об ошибке и код этой ошибки
        MessageBox.Show("Ошибка при сохранении данных.\n" + ex.Message); }
    }
    //метод, обрабатываемый при нажатии кнопки обновления
    private void обновитьДанныеТаблицToolStripMenuItem_Click(object sender, EventArgs e)
    {
        //вызвать метод обновления таблиц
        updateTables();
    }
    //метод обновления таблиц
    private void updateTables() {
        //для каждой таблицы и выпадающего списка на формы обновляет данные, заполняя их контейнера
        актуальными данными из базы данных
        this.сотрудникTableAdapter.Fill(this.milkdatabaseDataSet.сотрудник);
        this.обслуживаниеTableAdapter.Fill(this.milkdatabaseDataSet.обслуживание);
        this.постTableAdapter.Fill(this.milkdatabaseDataSet.пост);
        this.молоковозTableAdapter.Fill(this.milkdatabaseDataSet.молоковоз);
        this.приездTableAdapter.Fill(this.milkdatabaseDataSet.приезд);
    }
}

//подключение классов для работы с кодом
using System;
using System.Windows.Forms;
//определение зоны видимости
namespace myMilkProject
{
    //создание класса наследника формы для работы с сорудниками
    public partial class Employee : Form
    {
        //создание конструктора класса без параметров
        public Employee()
        {
            //инициализация всех полей формы для отображения
            InitializeComponent();
        }
        //метод, вызываемый при первом показе формы на экране
        private void Employee_Load(object sender, EventArgs e)
        {
            //методы ниже заполняют все таблицы формы соответствующими данными из базы данных
            this.постTableAdapter.Fill(this.milkdatabaseDataSet.пост);
            this.молоковозTableAdapter.Fill(this.milkdatabaseDataSet.молоковоз);
            this.сотрудникTableAdapter.Fill(this.milkdatabaseDataSet.сотрудник);
        }
        //метод, обрабатываемый при закрытии формы
        private void Employee_FormClosing(object sender, FormClosingEventArgs e)
        {
            //отобразить главную форму приложения
            Form1.main.Show();
        }
        //метод, обрабатываемый при нажатии кнопки созранить на таблице
        private void toolStripButton1_Click(object sender, EventArgs e)
        {
            //конструкция обработки ошибок, чтобы в случае чего вывести пользователю соответствующее
            окно
            try
            {
                //сохранение измененных данных в базе данных
                сотрудникTableAdapter.Update(milkdatabaseDataSet.сотрудник);
            }
            catch { }
        }
    }
}

```

```

//вывод пользователю сообщения об успешном сохранении
    MessageBox.Show("Изменения сохранены");
//обновление текущей таблицы, выводя в нее обновленные данные из базы данных
    this.сотрудникTableAdapter.Fill(this.milkdatabaseDataSet.сотрудник);
} //эта часть кода работает, если при обновлении произошли какие-либо ошибки
catch (Exception ex) {
//вывести пользователю сообщение об ошибке и текст этой ошибки
    MessageBox.Show("Ошибка при сохранении данных.\n" + ex.Message); }
}
//метод, вызываемый при неявных ошибках таблиц
private void dataGridView2_DataError(object sender, DataGridViewDataErrorEventArgs
e)
{ //реализацию метода оставляем пустой, поскольку это заглушка ошибок, которые появ-
ляются просто при закрытии формы
}
}
}
//конфигуратор данных
<?xml version="1.0" encoding="utf-8"?>
<configuration>
    <configSections>
    </configSections>
    <connectionStrings>
//на этой строке автоматически генерируется строка подключения к базе данных .При смене ло-
гина или пароля подключения или же названия самой базы данных, все новые значения вносятся в
эту строку
        <add name="myMilkProject.Properties.Settings.milkdatabaseConnectionString"
            connectionString="server=localhost;user
id=root;password=1234;persistsecurityinfo=True;database=milkdatabase"
            providerName="MySQL.Data.MySqlClient"/>
    </connectionStrings>
    <startup>
//используемая версия фреймворка
        <supportedRuntime version="v4.0" sku=".NETFramework,Version=v4.6"/>
    </startup>
</configuration>

```