МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

кафедра «Компьютерные технологии»

«Допущен к защите»
Заведующий кафедрой _____
Курмбаев З.К., профессор д.ф.м.-н.
(Ф.И.О., ученая степень, звание)
_____ «_____»_____ **20__ г.**
(подпись)

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной, аннуитетно кредитной системы "Ломбард"

Специальность Вычислительная техника и программное обеспечение

Выполнил (а) Жунусова Д.И. ВТ-12-2
(Фамилия и инициалы) группа

Научный руководитель Курмбаев З.К., профессор, д.ф.м.-н.
(Фамилия и инициалы, ученая степень, звание)

Консультанты:

по экономической части:
Бекешева А.И., доцент, к.э.н.
(Фамилия и инициалы, ученая степень, звание)
_____ «05» 05 2016 г.
(подпись)

по безопасности жизнедеятельности:
Тихонова И.Г. д.х.н., проф.
(Фамилия и инициалы, ученая степень, звание)
_____ «05» 05 2016 г.
(подпись)

по применению вычислительной техники:
Курмбаев З.К., профессор д.ф.м.-н.
(Фамилия и инициалы, ученая степень, звание)
_____ «25» 05 2016 г.
(подпись)

Нормоконтролер: Курмбаев З.К., профессор д.ф.м.-н.
(Фамилия и инициалы, ученая степень, звание)
_____ «25» 05 2016 г.
(подпись)

Рецензент: Байгунчалиев А.А., профессор, д.ф.м.-н.
(Фамилия и инициалы, ученая степень, звание)
_____ «_____»_____ 20___ г.
(подпись)

Алматы 2016 г.

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

Некоммерческое акционерное общество
АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ

Факультет Аэрокосмических и информационных технологий
Специальность Вычислительная техника и программное обеспечение
Кафедра Компьютерных технологий

ЗАДАНИЕ
на выполнение дипломного проекта

Студент Хизова Диана Исламовна
(фамилия, имя, отчество)

Тема проекта Разработка информационной, аннуитетно кредитной системы "Ломбард"

утверждена приказом ректора № 148 от «19» октября 2015 г.
Срок сдачи законченной работы «___» _____ 20___ г.
Исходные данные к проекту требуемые параметры результатов проектирования (исследования) и исходные данные объекта
разработать информационную, аннуитетно кредитную систему "Ломбард"

Перечень подлежащих разработке дипломного проекта вопросов или краткое содержание дипломного проекта:
1) Выбрать СУБД и язык программирования для разработки информационной системы
2) Осуществить постановку задачи
3) Разработать базу данных, определение атрибутов сущностей, создание функций, процедур и триггеров
4) Создать диаграммы, описывающие ТО
5) Разработать ПО
6) Написать требования к аппаратной части
7) Написать руководство пользователя

Перечень графического материала (с точным указанием обязательных чертежей)

0 рисунков в главе 1
17 рисунков в главе 2
37 рисунков в главе 3
1 рисунок в разделе Экономика
2 рисунка в разделе БЖД

Рекомендуемая основная литература

Г. Шилдт. Самоучитель C# – СПб: БХВ-Петербург, 2007. – 656 с.

Голд Т., Хейлсберг А. Язык программирования C#. Классика Computers Science. 4-е издание

Кренке Д. Теория и практика построения баз данных. Издание 9 – Питер, 2005.

И. Бен-Ган. Microsoft SQL Server 2012. Высокопроизводительный код T-SQL. – БХВ-Петербург, 2013. – 544 с.

Консультанты по проекту с указанием относящихся к ним разделов

Раздел	Консультант	Сроки	Подпись
Экономика	Бекишева А. И.	23.02 - 05.05.16	
БЖД	Приходько Н. Г.	29.02 - 05.05.16	
Глава 1	Жуматбаев З. К.	1.01.16 - 29.02.16	
Глава 2	Жуматбаев З. К.	2.03.16 - 29.03.16	
Глава 3	Жуматбаев З. К.	2.04.16 - 29.04.16	
Нормоконтроль	Жуматбаев З. К.	25.05.2016	

Г Р А Ф И К
подготовки дипломного проекта

№ п/п	Наименование разделов, перечень разрабатываемых вопросов	Сроки представления руководителю	Примечание
1	Глава 1	1.01.16-29.02.16	
2	Глава 2	2.03.16-29.03.16	
3	Глава 3	2.04.16-29.04.16	
4	БЖД	29.02.16-05.05.16	
5	Экономика	23.02.16-05.05.16	

Дата выдачи задания «19» октября 2015 г.

Заведующий кафедрой _____ Курамбаев З.К
(подпись) (Фамилия и инициалы)

Руководитель _____ Курамбаев З.К
(подпись) (Фамилия и инициалы)

Задание принял к исполнению
студент _____ Князова Д.И.
(подпись) (Фамилия и инициалы)

Аннотация

В данной дипломной работе представлена разработка моделей программного обеспечения и базы данных, также разработана база данных, связанные с ней функции, триггеры и процедуры; программного продукта на языке C# написан; описаны требования к аппаратной части; написана инструкция пользования.

В разделе безопасности жизнедеятельности произведён анализ условий труда, а также произведён расчёт искусственного освещения и кондиционирования.

В экономической части произведён расчёт затрат на разработку программного продукта, которые составили 248294.35, а также я рассчитал конечную стоимость продукта, которая составила 333707.61 тг.

Аңдатпа

Осы дипломдық жұмыста үлгілерді әзірлеу бағдарламалық қамтамасыз ету және деректер базасын ұсынылған, сондай-ақ дайындалды функциялары, рәсімдер мен триггерлер; бағдарламалық өнім C# тілінде жазылған, сипатталады қойылатын талаптар аппараттық бөлігін, нұсқаулық жазылған.

Тіршіліктің қауіпсіздігі тарауында еңбек шарттарына талдау жасалып, сонымен бірге жасанды жарықтандыру пен кондиционерлеу есебі жүргізілді.

Annotation

In this thesis work design of software models and data base were presented, also functions, triggers and procedures were designed; the software was written in C#; the requirements for hardware were described; the manual was written.

In the life safety section the analysis of working conditions was promoted, as well as the calculation of lamplight and conditioning were made.

In the economic part the calculation of development cost of software, which amounted 248294.35 KZT, were made, and I calculated the final cost of the software, which amounted 333707.61 KZT.

Содержание

Введение

Современная Казахстанская экономика характеризуется своим низким уровнем развития рынка розничных кредитных услуг банковских учреждений, малочисленностью и разрозненностью кредитных кооперативов и товариществ. В таких условиях особенно возрастает роль ломбардов как ведущих кредиторов населения, призванных поддерживать и повышать платежеспособность населения, активизировать совокупный спрос в экономике и служить дополнительным импульсом экономического роста страны. Для интенсивного развития кредитных отношений с населением в стране должна быть создана широкая сеть ломбардов [1)].

Многие задаются вопросом, как работает система в ломбардах. Поскольку ломбард мы можем считать одним из основных кредиторов населения, который призван поддерживать и повышать платежеспособность народа, мы можем сделать для себя вывод как работает система ломбард.

Населению иногда приходится сталкиваться с ситуацией, когда срочно нужны деньги на какую–либо срочную вещь или дело. Большинство занимают деньги у друзей или берут кредит в банке, но и не маленькая часть населения пользуется услугами ломбарда.

Ломбарды принадлежат к финансовому учреждению, предоставляют финансовые услуги, а именно, средства в заем на условиях кредита. Финансовый кредит предоставляется под залог какого–либо имущества клиентом ломбарда.

Ломбарды бывают разные. Одни ломбарды принимают только ювелирные изделия, другие к примеру, автомобили, их называют авто ломбардами.

Программа «Ломбард» – это программа для автоматизации работы ломбарда.

В данной дипломной работе применятся аннуитетно кредитная система платежа. Поскольку это аннуитетно кредитная система платежа, клиент может погашать ее частично, а может сразу внести всю сумму долга. Аннуитетный платеж – это вариант ежемесячного платежа по кредиту, когда размер ежемесячного платежа остаётся постоянным на всём периоде кредитования.

В данной программе возможна работа с сотрудниками, клиентами, договорами, справками. Можно проделывать различные расчеты, следить за статистикой.

Поскольку в данной программе хранится большой ассортимент справок и видов вещей, также проводится полный расчет, сотрудник ломбарда может наблюдать за статистикой договоров и вещей, заложенных в ломбард, и делать различные функции с договорами, систему «Ломбард» можно считать информационной системой.

Актуальность данной темы заключается в том, что нет аналогов программного продукта в Казахстане, который позволял бы автоматизировать работу ломбарда, обеспечить хранение важной информации в цифровом виде, производить расчёты быстро и удобно, а также делать СМС–рассылку.

В данном случае программное обеспечение для ломбардов поможет организовать быструю и оптимальную работу ломбардов. Обеспечить хранение данных и важной информации. А также произвести удобные и быстрые расчеты.

Цель разработки ПО: разработать информационную аннуитетно кредитную систему, а также программный продукт, который позволяет автоматизировать работу ломбарда. Программа «Ломбард» – это решение, которое позволяет увеличить количество клиентов, обслуживаемых ломбардами, сократив издержки на обслуживание, уменьшить время, затрачиваемое на работу с одним клиентом, а главное вести учет и контроль над финансами и хранимыми материальными ценностями.

1 Обзор используемых технологий

1.1 Язык программирования

Во время реализации данной работы применялся язык программирования C#. C++ является прародителем C#. Этот язык простой, объектно–ориентированный. Он более гибок и ясен для написания кода. Особенностями языка программирования C# являются:

1) файл является единицей компиляции. Он внутри себя содержит типовые описания перечислений, классов, структур, интерфейса с описанием их именного пространства;

2) видимостью программных объектов контролируется именным пространством. Именное пространство может вкладывать в другое. Также можно работать и без определения именного пространства. Можно использовать алиасы для наименования именного пространства;

3) применение всевозможных типов данных от целочисленных типов и до логических типов;

4) наличие структурных типов, как интерфейсы, классы, многомерные и одномерные массивы, структуры (которые размещаются в куче), строки, перечисления и т.д.;

5) подразделение типов на типы-значения и ссылочные. Объекты ссылочных типов находятся в куче, а их переменные есть указатели на объекты. Переменные типов-значений являются значениями, а не самими указателями. Разрешается неявное преобразование, если нет нарушения системы безопасности типов и отсутствует потеря информации. Объектный тип является основным классом остальных типов. Есть возможность неявно преобразовать в тип object и явное обратное;

6) в методах существует возможность указывать значения параметров. Существует два параметра: ref (ссылочный) и out (выходной). Параметры необходимо указывать при выходе и при реализации метода. Позволяют осуществлять контроль выполнения определяющих присваиваний;

7) также существуют операторы управления, такие как do, if, for, while, continue, switch, break. Также существует цикл foreach, который работает с коллекциями;

8) наличие try и catch для обработки исключающих ситуаций;

9) к свойствам элементов класса можно осуществлять доступ, как и доступ к полям, но реализация неявная. Есть встроенный get и set для доступа и назначения параметров;

10) также присутствуют индексаторы, которые являются элементами класса, позволяя обращаться к объектам подобно массивам. Реализация осуществляется неявно с помощью get и set;

11) одним из важных элементов класса является событие, которое позволяет добавлять или убирать методы обработки событий [2)].

Преимуществом C# является скорость разработки, сравнивая с остальными языками. Разработка оставляется быстрее для неготовой инфраструктуры, когда проект только создаётся. Но минусы языка - отсутствие кроссплатформенности и работа приложения возможна на ОС Windows, где создана полная инфраструктура. На C# удобнее писать код, в последствии соответствующий требованиям. Плюсом является большие возможности стандартных библиотек, не приходится тратить время на поиск сторонних библиотек.

1.2 Среда программирования

В качестве среды программирования я использовала Microsoft Visual Studio Ultimate 2013. Microsoft Visual Studio – это продукт компании Microsoft, включающий в себя среды разработки ПО и другие разные средства для поддержания продукта. Продукт позволяет разрабатывать как GUI-приложения, так и простые консольные программы. Существует возможность разработки программ с использованием технологии WPF. Также есть возможность создания веб-сайтов и веб-приложений с использование asp.net, python, javascript и html. Так есть поддержка мобильных ОС, таких как WIndows Phone и Android. Для написания приложения для Android используется Xamarin. Visual Studio состоит из редактора исходного кода, поддерживая технологии для удобного его отображения и читаемости, а также быстрого написания. Также есть встроенный отладчик, который работает в двух режимах: уровня исходных кодов или машинного уровня. Остальные инструменты – это редактор форм для более удобного создания GUI–приложения, веб–редактор, дизайнер классов и схем БД. Существует возможность подключения к Visual Studio сторонних расширений, плагинов, позволяющих расширить

функционал среды. также был добавлен GIT, позволяющий контролировать версии выпускаемого продукта.

В Visual Studio встроен магазин Windows, что позволяет в кратчайшие сроки. Также существует комплекс инструментов для разработки, которые позволяют осуществлять совместную разработку, создавать отчёты и метрики [4].

1.3 СУБД

В качестве СУБД я выбрана MS SQL Server 2012. Microsoft SQL Server – это система управления реляционными базами данных, разработанная корпорацией Microsoft. Языком запросом является T-SQL, который используется в Sybase. T-SQL – реализация стандарта ANSI/ISO с расширениями. Данная СУБД используется от небольших БД до БД промышленного масштаба.

Одним плюсом MS SQL Server 2012 является масштабируемость, т.е. данная СУБД может работать с большим количеством процессоров, что оказывает влияние на производительность. MS SQL Server является Windows совместимой, т.е. предназначена для работы на платформе Windows, т.е. не надо дополнительно производить установку иной ОС для работы с ней.

Важным нововведение работы с реляционными БД являются возможности управлять большим количеством серверов. Раньше не было возможности управления группой серверов, она была ограниченна. Можно было добавить несколько серверов в Management Studio, но такое не возможно использовать для задания одинаковых задач большому количеству серверов, т.е. нет объединения серверов в группы. Для этого реализован новейший компонент – Utility Explorer. С помощью его создаётся точка управления, где хранятся списки с экземплярами всех добавленных серверов. Благодаря обозревателю, появилась возможность управлять экземплярами в количестве 25 штук.

Обозреватель показывает общую производительность, информацию о ресурсах и емкость добавленных серверов. В первом выпуске есть только возможность осуществлять управление только экземплярами. Но в первоначальном выпуске возможно управление только экземплярами БД. Управлять несколькими серверами возможно только в выпусках Enterprise Edition и Datacenter Edition.

Также сервер может осуществлять шифрование данные всей БД, файлов данных и журналы.

Сервер БД обеспечивает высокую надежность:

– осуществлять шифрование данные всей БД, файлов данных и журналы;

– возможность управлять аппаратными модулями безопасности и ключами;

– выполнять аудит. Аудит позволяет отслеживать и заносить в журналы события, произошедшие с разными компонентами БД.

MDS обеспечивает создание определения всех примерных данных на предприятии. У большинства предприятий существуют различные БД, использующие приложения. Между этими БД есть различия в представлении однотипных данных и схемах. Из-за чего появляются проблемы отсутствия унифицированного критерия истины, при этом компании пытаются объединять разные данные, чтобы вести централизованную отчётность, аналитику и поиск.

Благодаря MSD есть возможность создавать главные определения данных предприятия для удобного сопоставления и преобразования данных из различных ресурсов в единый депозитарий. Служба MSD может стать центральным источником корпоративных данных. Для управления MDS можно использовать веб-клиент для формирования потока операций, которые извещают хозяев данных о разных нарушениях. В редакциях Enterprise Edition и Datacenter Edition присутствует служба MSD.

Администратор БД может выполнить гибкую настройку прав доступа, т.е. возможность управлять доступа не только пользователя, но и группы, в которых он состоит. SQL Server имеет удобную консоль управления – Management Studio, что позволяет работать с БД не только из GUI, но из консоли.

В SQL Server обновился компонент Report Builder, позволяющий строить отчёты, которые можно дополнить его пространственной информацией. Также можно добавить различные графики и гистограммы для более наглядной отчётности.

В SQL Server 2012 имеется большое количество возможностей. Выделю основные улучшения:

– возможность создавать образы на сменном носителе продукта вместе с обновлениями;

– возможность настраивать виртуальные кластеры "на лету" вместе с новыми обновлениями;

– возможность подключаться к облаку azure, а также управлять экземплярами;

– поддержка Sharepoint 2013;

– возможность создавать элементы, которые могут быть использованы в нескольких отчётах[3)].

2 Разработка

2.1 Постановка задачи

На основе полученного навыка проектирования реляционных БД и изучения методов разработки СУБД, основываясь на них, создать свою базу данных, которая будет ориентирована на предметную область Ломбард. Программ «Ломбард» будет разработана с целью автоматизации ломбарда, и упрощения работы сотрудника с ломбардом. Полное название программного обеспечения и ее обозначение «Ломбард». В данной программе необходимо авторизоваться сотруднику для работы с ней. Любой сотрудник может добавлять клиентов, договора, вещи. Возможна работа со справочным материалом вещей, которые уже хранятся в таблицах БД Ломбард. Возможна работа с документацией в Excel, распечатка. Главной фишкой программы является – СМС–рассылка. Таким образом, любой клиент, который заложил вещь в данный ломбард, уведомляется, что у него должен будет произойти какой–либо взнос по его долгу.

Основное переназначение программы, максимально упростить работы с платежами, расчеты над суммами, которые хранятся в ломбарде, и автоматизировать работу с документацией и клиентами. Удобный интерфейс, использован нейтральный цвет программы – черный, чтобы не рябило и не бросались в глаза яркие цвета в интерфейсе программы. Расположение информации и кнопок не хаотично, компактно все находится в местах, пригодных для восприятия пользователя программы.

2.2 Разработка базы данных

Самое распространённое определение реляционной модели, принадлежит Диету, который репродуцирует ее во всех книгах, написанных им. Ссылаясь на его книги, стоит понимать, что в реляционную модель входят три части, которые описывают всевозможные аспекты структурной части реляционного подхода, целостной части и манипуляционной части.

В части модели, называемой структурной, понимается, что унарной структурой данных, использующейся в реляционных БД, есть типичное n–барное отношение.

В части модели, называемой манипуляционной, устанавливаются два фундаментальных механизма, которые

манипулируют реляционными БД – реляционные алгебра и исчисление. На теории множеств основывается первая часть, а на классическом аппарате логики исчисления предикатов первого порядка – вторая.

Защитой данных от повреждений и некорректных изменений является целостность данных, которая обеспечивает выше сказанное.

Построение и разработка ER–диаграммы производится, базируясь на бизнес–правилах. Один сотрудник имеет только одну должность. Бизнес–правило 1 показано на рисунке 2.1.

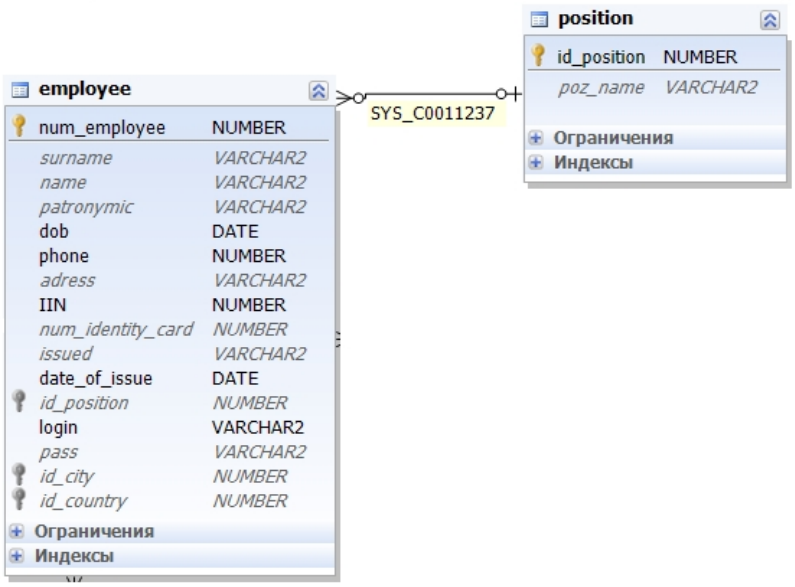


Рисунок 2.1 – Бизнес–правило 1

Бизнес–правило 2 гласит, что единственный клиент имеет множество различных договоров. Бизнес–правило 2 показано на рисунке 2.2.

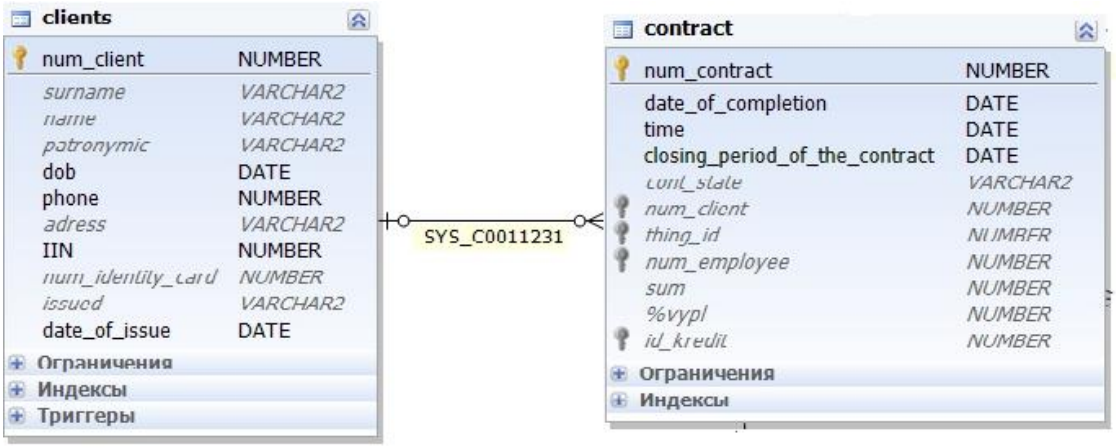


Рисунок 2.2 – Бизнес–правило 2

Бизнес–правило 3 – один договор заключается только на одну вещь. На рисунке 2.3 показано бизнес–правило 3.

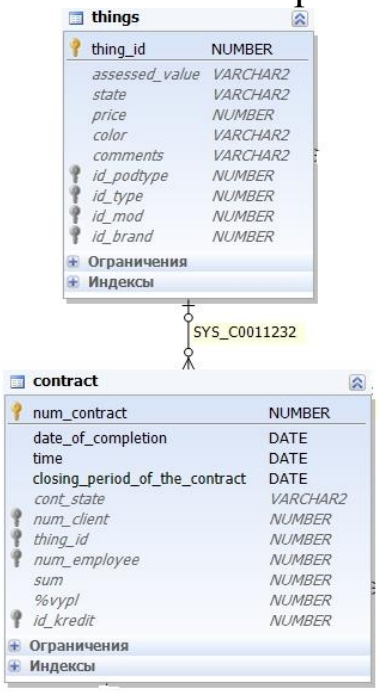


Рисунок 2.3 – Бизнес–правило 3

Бизнес–правило 4 гласит, что процент от суммы заложенных вещей начисляется сотруднику через договор. За каждый договор, на который оформлена заложенная вещь, сотруднику начисляется процент с ее стоимости. Бизнес–правило 4 показано на рисунке 2.4.

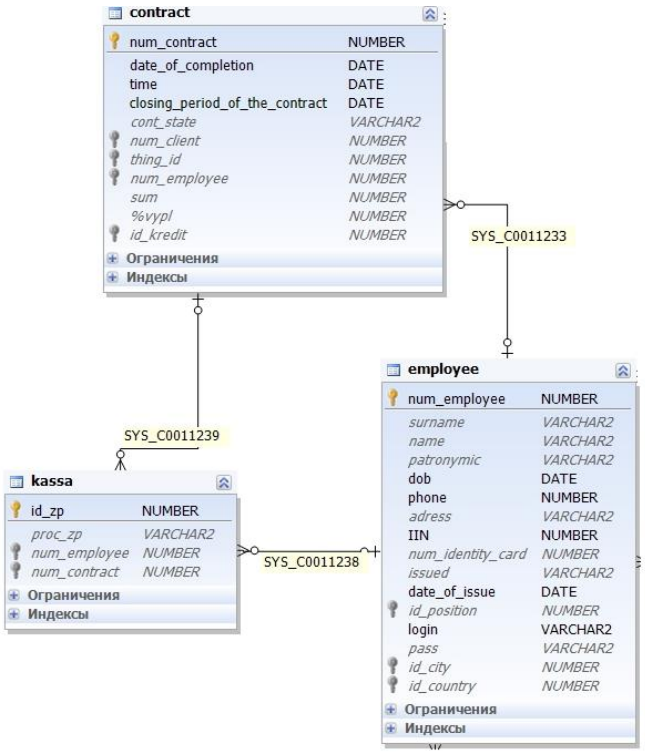


Рисунок 2.4 – Бизнес–правило 4

Бизнес–правило 5 – одна вещь может являться только вещью одной разновидности. На рисунке 2.5 показано бизнес–правило 5.

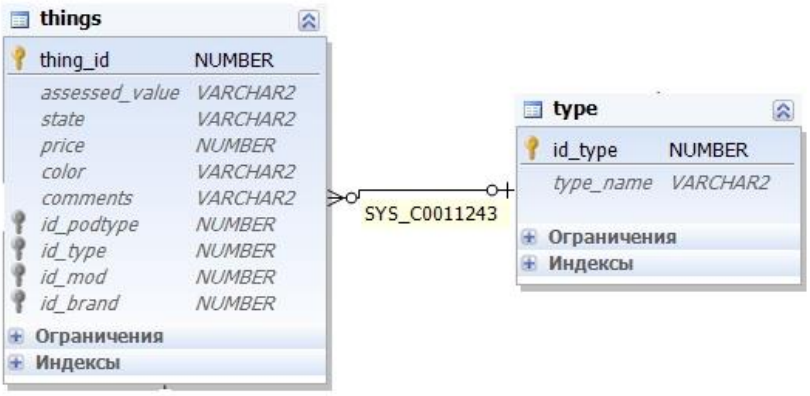


Рисунок 2.5 – Бизнес–правило 5

Бизнес–правило 6 гласит, что один тип может иметь один подтип. Бизнес–правило 6 показано на рисунке 2.6.

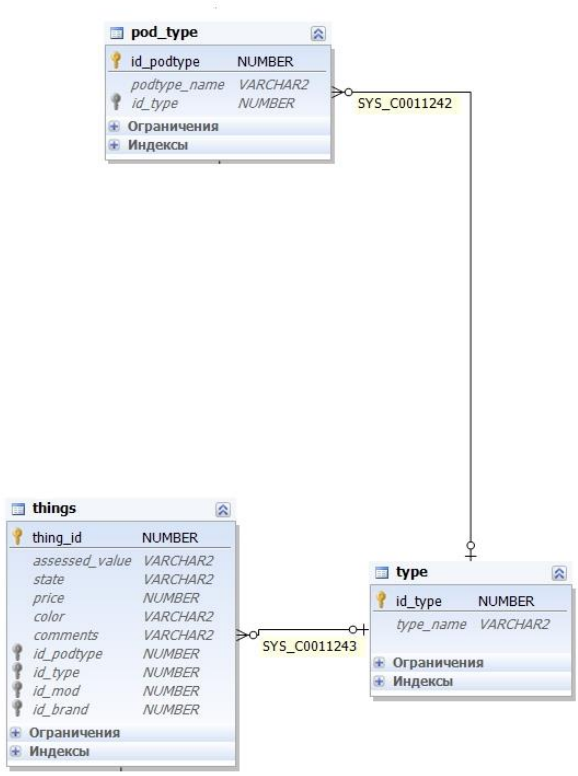


Рисунок 2.6 – Бизнес–правило 6

Бизнес–правило 7 – подтип связан с брэндами ассоциативной таблицей assoc_brand, т.е. подтип имеет разные брэнды, как и брэнды могут связываться с разными подтипами. На рисунке 2.7 показано бизнес–правило 7.

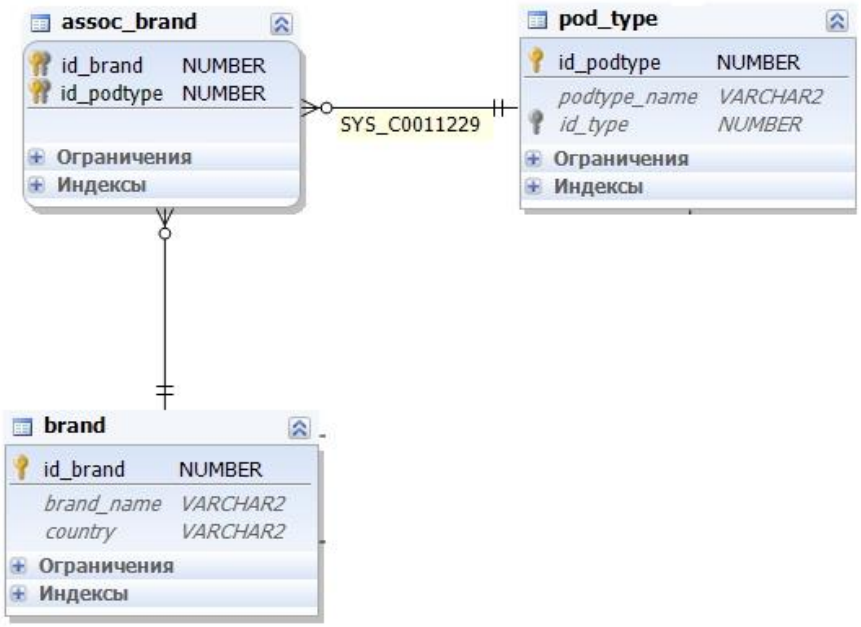


Рисунок 2.7 – Бизнес–правило 7

Бизнес–правило 8 гласит, что один брэнд имеет только одну модель, которая имеет свой один подтип. Бизнес–правило показано на рисунке 2.8.

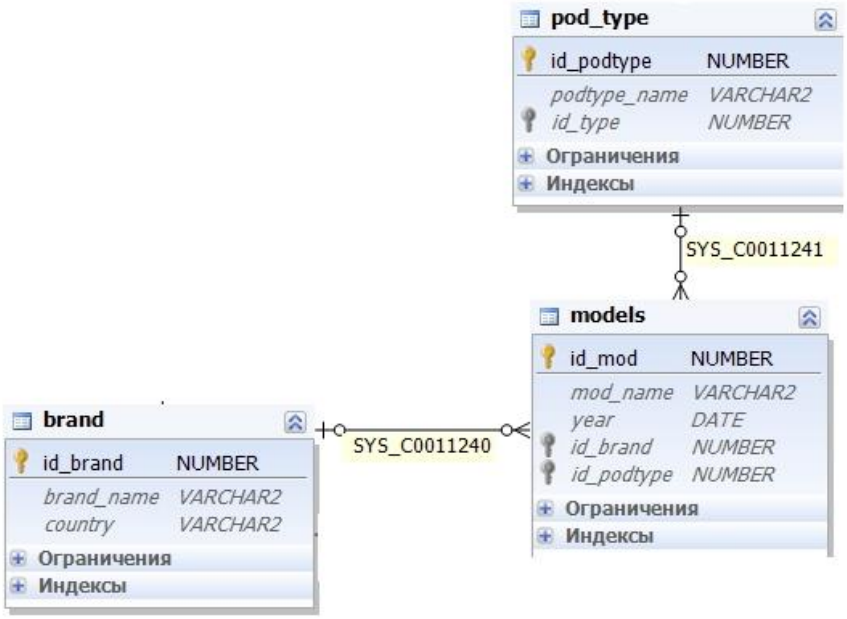


Рисунок 2.8 – Бизнес–правило 8

Бизнес–правило 9 – одна вещь имеет только один тип, одну модель, подтип, и брэнд. На рисунке 2.9 показано бизнес–правило 9.

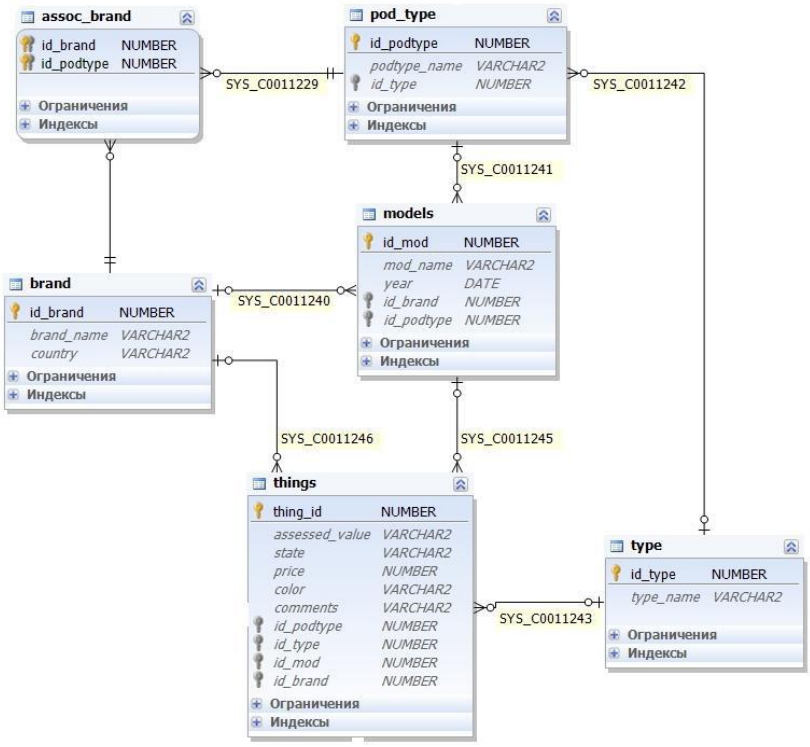


Рисунок 2.9 – Бизнес–правило 9

Бизнес–правило 10 гласит, что зарплата сотрудникам начисляется через кассу. Бизнес–правило 10 показано на рисунке 2.10.

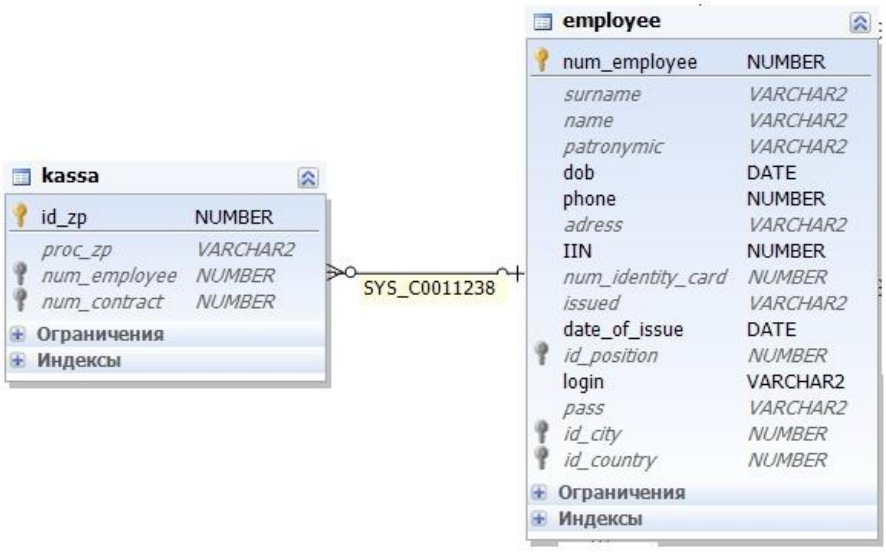


Рисунок 2.10 – Бизнес–правило 10

Бизнес–правило 11 гласит, что каждый сотрудник проживает в одной только стране и одном только городе. На рисунке 2.11 показано бизнес–правило 11.

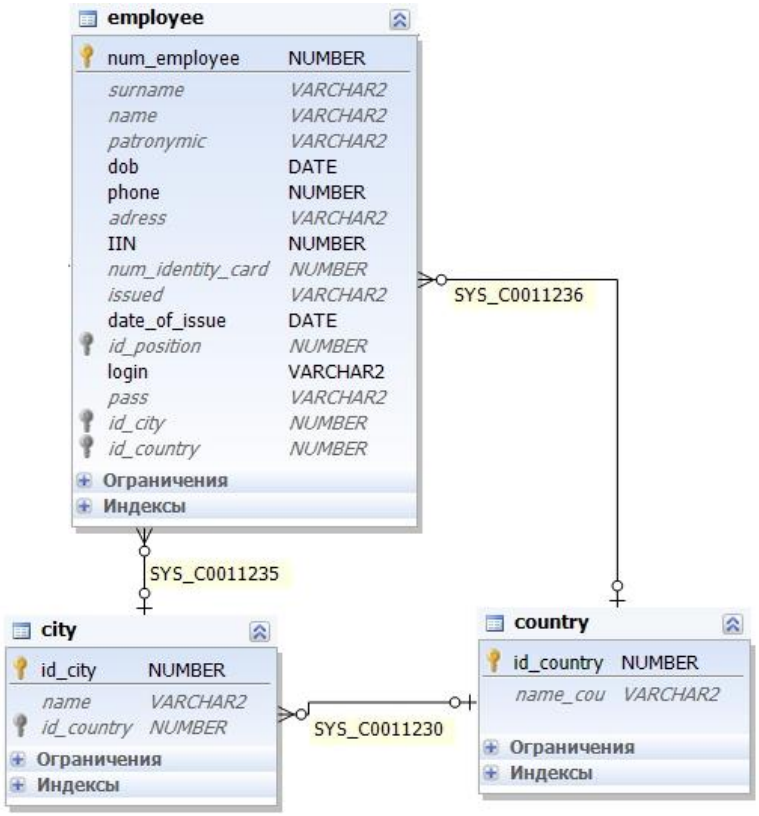


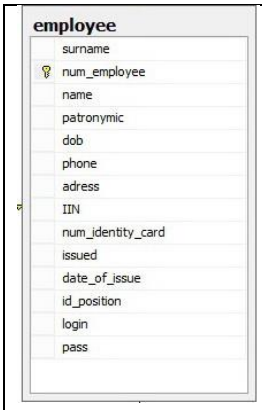
Рисунок 2.11 – Бизнес–правило 11

2.3 Определение атрибутов каждой сущности

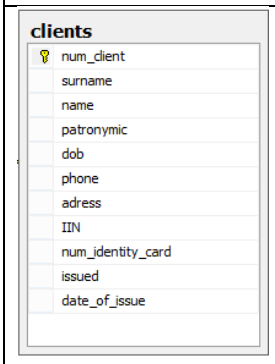
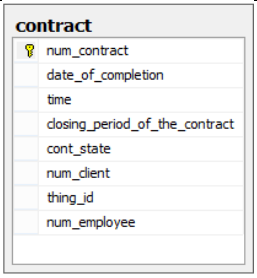
Сначала проведу описание сегмента ER модели. Сегмент ER модели и есть таблица, ее поля и описание этих полей. Например, таблица «employee» содержит следующие данные: 14 полей и одно поле из них, а именно num_employee – идентификатор должности. Подробное описание сегментов ER–модели указано в таблице 2.1.

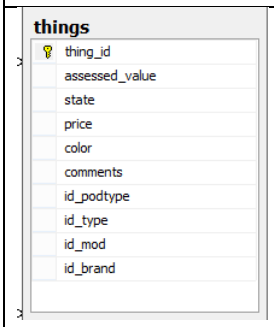
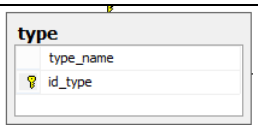
Таблица 2.1 – Описание сегментов ER–модели

Сегмент ER модели	Описание
1	2

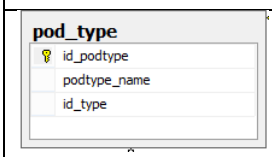
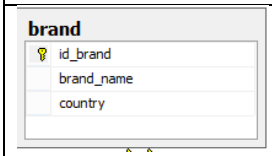
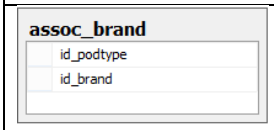
	Таблица сотрудники содержит следующие данные: num_employee – идентификатор должности surname – фамилия name – имя patronymic – отчество dob – дата рождения phone – телефон address – адрес iin – иин num_identity_card – номер удостоверения

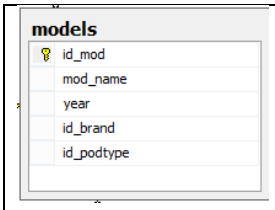
Продолжение таблицы 2.1

1	2
	issued – кем выдано date_of_issue – дата выдачи id_position – идентификатор должности login – логин pass – пароль
	Таблица клиенты содержит следующие данные: num_client – идентификатор клиента surname – фамилия name – имя patronymic – отчество dob – дата рождения phone – телефон address – адрес iin – иин num_identity_card – номер удостоверения issued – кем выдано date_of_issue – дата выдачи
	Таблица договоры содержит следующие данные: num_contract – идентификатор договора date_of_completion – дата заполнения time – срок closing_period_of_the_contract – закрытие срока договора cont_state – состояние договора num_client – идентификатор клиента thing_id – идентификатор вещи

	num_employee – идентификатор сотрудника
	Таблица вещи содержит следующие данные: thing_id – идентификатор вещи assessed_value – оценочная стоимость state – состояние вещи price – цена color – цвет comments – комментарии id_podtype – идентификатор подтипа id_type – идентификатор типа id_mod – идентификатор модели id_brand – идентификатор брэнда
	Таблица типы содержит следующие данные: id_type – идентификатор типа type_name – тип

Продолжение таблицы 2.1

1	2
	Таблица подтипы содержит следующие данные: id_podtype – идентификатор подтипа id_type – идентификатор типа podtype_name – подтип
	Таблица брэнды содержит следующие данные: id_brand – идентификатор брэнда brand_name – брэнд country – страна производитель
	Ассоциативная таблица брэнды содержит следующие данные: id_podtype – идентификатор подтипа id_brand – идентификатор брэнда

	Таблица модели содержит следующие данные: id_mod – идентификатор модели mod_name – модель year – год выпуска id_podtype – идентификатор подтипа id_brand – идентификатор брэнда

В результате исполнения запросов, получу реальную БД. Концептуальная модель представлена на рисунке 2.12.

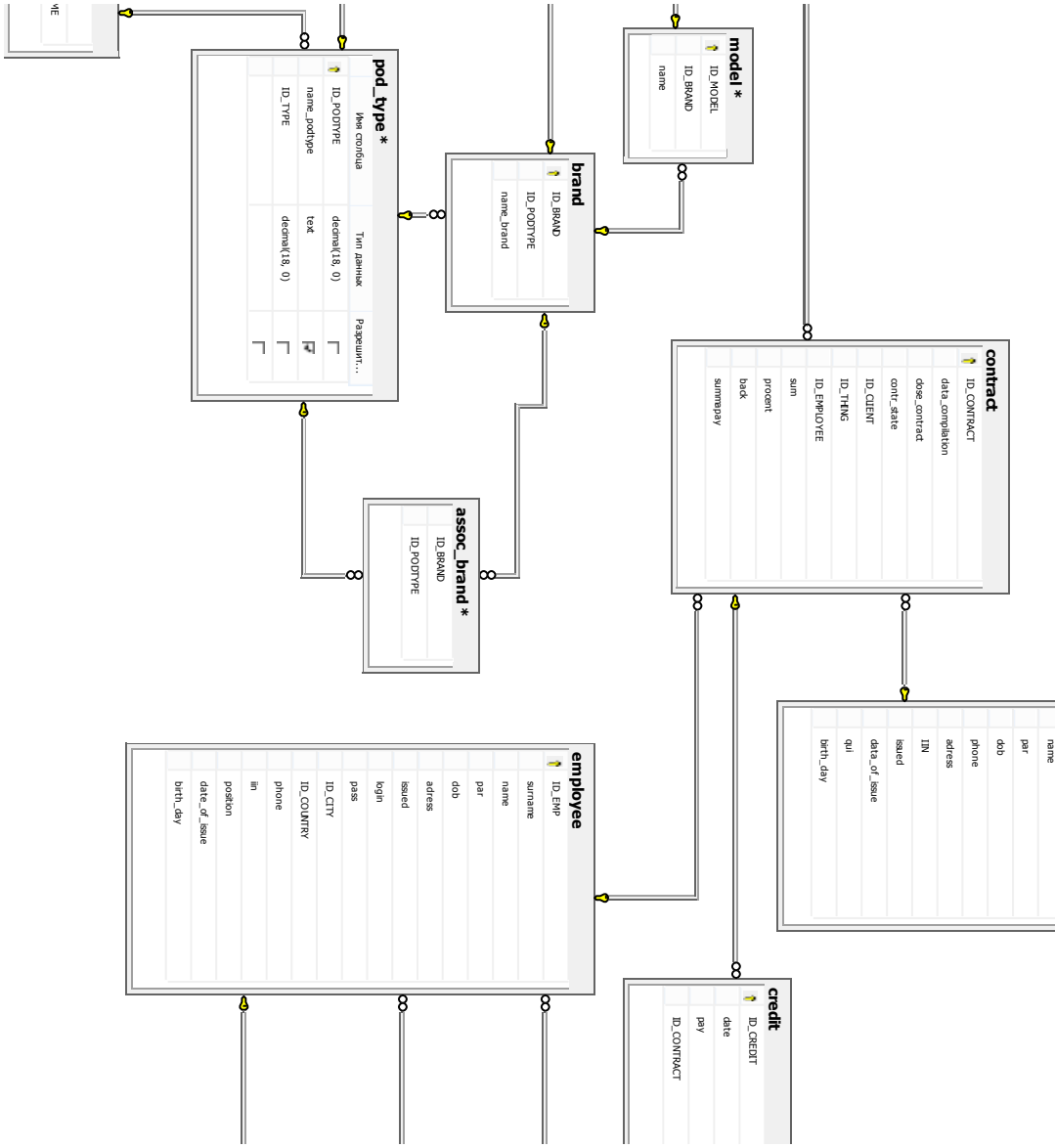


Рисунок 2.12 – Концептуальная модель базы данных

2.4 Функции, процедуры и триггеры

Процедура вывода контракта, суммы его и кем был подписан, а также клиента, на которого он оформлен:

```
create or replace procedure contract (num number):
is
cursor asd(num1 number)
is
select c."num_contract",c."sum",e."surname" as "empsur",cl."surname" from
"contract" c
inner join "employee" e
on c."num_employee"=e."num_employee"
inner join "clients" cl
on c."num_client"=cl."num_client"
where c."num_employee"=num1;
begin
for emp_1 in asd(num)
loop
DBMS_OUTPUT.PUT_LINE (lpad(emp_1."num_contract",20,' ')||' cost
'||emp_1."sum"||' client '||emp_1."surname"|| ' emp '||emp_1."empsur" );
end loop;
end;
```

Функция считает полную стоимость прибыли клиента от заказа:

```
create or replace function allcost (num number) return number
is
vip number;
begin
select sum("sum"·0.01·"%vypl") into vip from "contract" where
"num_client"=num;
return vip;
end;
```

Процедура считает полную стоимость прибыли всех сотрудников от заказа:

```
create or replace procedure allcost2
is
cursor cur
is
```

```
select sum("sum"·0.01·"%vypl") "ad","num_employee"
from "contract"
group by "num_employee";
begin
for ssd in cur
loop
DBMS_OUTPUT.PUT_LINE (ssd."num_employee"||' summ '||ssd."ad");
end loop;
end;
```

Функция считает прибыль от продажи вещи:

```
create or replace function pribil (num number) return number
is
vip number;
begin
select c."sum"–t."price" into vip from "contract" c
inner join "things" t
on t."thing_id"=c."thing_id"
where "num_contract"=num;
return vip;
end;
```

Триггер, запрещающий повторять ИИН:

```
create or replace TRIGGER mol
before  insert or update of "IIN" ON "clients"
for each  row
DECLARE
clientsn number;
BEGIN
select count(·) into clientsn from "clients" s where s."IIN"=:new."IIN";
IF (clientsn=1) then
RAISE_APPLICATION_ERROR(–20323,'Oshibka pri vvode IIN. Nelyazya povtiryt');
end if;
end;
```

2.5 Моделирование программного обеспечения

В начале разработки по следует создать модели, которые будут описывать поведение программы, описывать какие роли

будут, т.е. какие привилегии. Это всё описывают диаграммы. Для этого составлю следующие диаграммы:

– диаграмма компонентов;
– диаграмма развёртывания;
– диаграмма последовательностей;
– диаграмма активности;
– диаграмма состояний.

Для начала составлю диаграмму компонентов, чтобы иметь представление о программе: какие компоненты, что каждый элемент будет делать. Диаграмма компонентов – это элементное разбиение системы и связь между элементами. Элементами могут выступать файлы, пакеты, библиотеки, и т.д. Lambard.exe является основным элементом системы, который связывает базы данных. На рисунке 2.13 показана диаграмма компонентов.

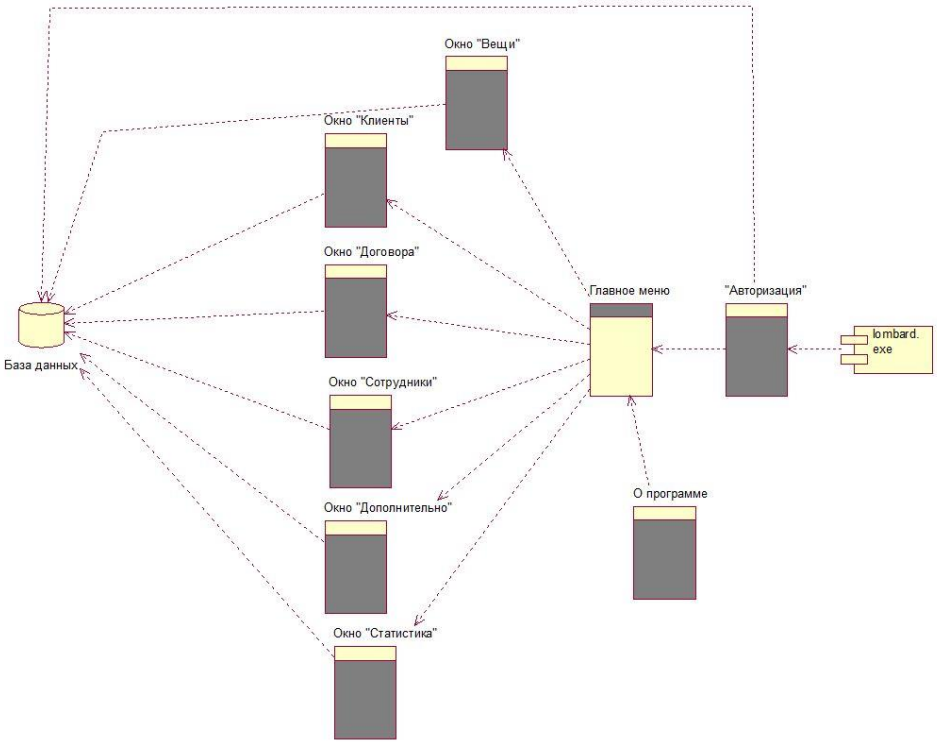


Рисунок 2.13 – Диаграмма компонентов

Программа состоит из окна авторизации, главного меню, информации о программе, окна «Вещь», окна «Клиенты», окна «Сотрудники», окна «Дополнительно» и окна «Статистика». Существует БД, хранящая информация о вещах, сотрудниках, клиентах и, основываясь на них, ведётся статистика.

Описание действий осуществляется с использованием диаграммы последовательностей. Диаграмма последовательностей показана на рисунке 2.14. Возьму, например, действие добавить клиента. Для этого нужно открыть программу, выбрать в меню «Клиенты», в появившемся окне нажать кнопку «Добавить клиента». Далее необходимо ввести ИИН и нажать кнопку «Проверить», чтобы осуществить проверку ИИН. Если ИИН нет в базе, то необходимо перед нажатием кнопки «Добавить» ввести данные в пустые поля. Если ИИН в базе имеется, то необходимо ввести его повторно.

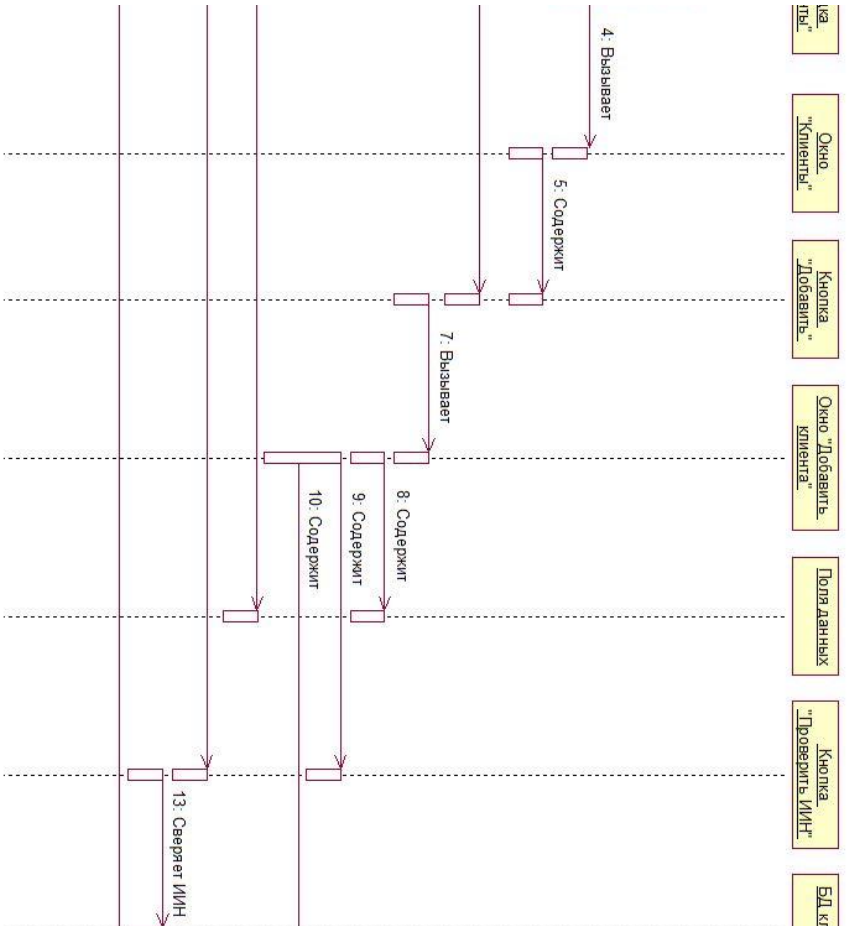


Рисунок 2.14 – Диаграмма последовательности

Диаграмма прецедентов – диаграмма, отображающая актёров, взаимоотношение между ними и набором действий, которые актёры исполняют или нет. Позволяет иллюстративно показать взаимоотношение системы и пользователя, его привилегии. В конкретном случае показано, что могут делать гость и администратор. На рисунке 2.15 показана диаграмма прецедентов.

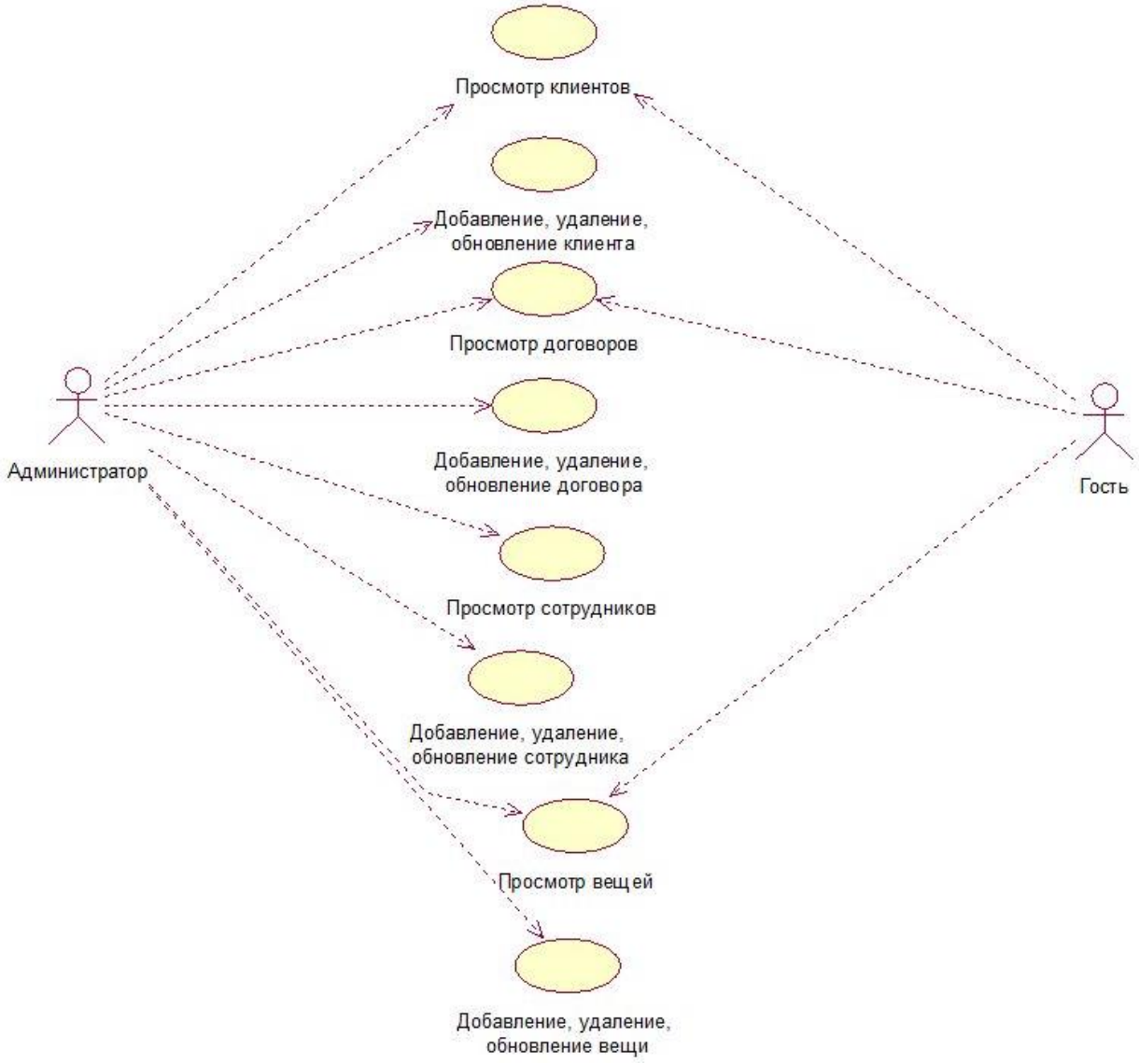


Рисунок 2.15 – Диаграмма прецедентов

Диаграмма активности описывает продолжительный этап выполнения. Деятельность сводится к исполнению некоторого действия. Например, возьму активность «Добавить клиента». Для добавления клиента нужно следовать следующим ударам:

– открыть программу;

– нажать «Клиенты» в главном меню;
– нажать кнопку «Добавить»;
– ввести ИИН;
– нажать кнопку «Проверка»;
– в зависимости от результата проверки ввести ИИН заново или ввести данные в поля;
– нажать «Добавить»;
– закрыть окно «Клиент».

Диаграмма активности показана на рисунке 2.16.

Рисунок 2.16 – Диаграмма активности

Диаграмма состояний используется для описания состояния в момент какого–то действия, описывает состояние объекта. Например, возьму операцию «Добавление клиента». На рисунке 2.17 показана диаграмма состояний.

Рисунок 2.17 – Диаграмма состояний

При выполнении каждого действия программа переходит в какое–то событие, вызывавшее это действие. При добавление клиента программа открывается окно «Добавление Клиента» и происходит заполнение полей, в последствии осуществляется

проверка ИИН, которая позволит определить существует ИИН в базе или отсутствует.

2.6 Разработка программного обеспечения

Создание программного обеспечения началось с формы входа, а именно окна авторизации. В данную форму добавлены текстовые поля ля «Логин» и «Пароль». В поле «Пароль» текст обозначен звездочками и запрещено копирование данных с данного поля. При нажатии на кнопку вход, производится запрос в базу данных таблицы «employee», который возвращает единицу при корректном вводе пароля и логина. Данная форма связана с формой «Главное меню», в которой мы можем работать с договорами, клиентами, сотрудниками, вещами, справками, а также просматривать статистику договоров и вещей. Были разделены пользовательские привилегии на четыре типа, а именно: администратор, бухгалтер, менеджер, гость. Привилегии помогают разграничить права доступа к разным функциям программы. Допустим, администратор системы может делать все, и он имеет неограниченные возможности. Бухгалтер может заходить в меню «Договоры» и просматривать договоры, сумму, а также делать расчеты с заложенными суммами. Менеджер может работать с клиентами, добавлять, удалять их, а также оформлять на них договоры, работать с расчетами, поскольку ему необходимо контролировать бюджет ломбарда и сдавать отчеты в формате Excel бухгалтеру, который в последующем также, как и менеджер может просматривать статистику договоров и вещей, которые были заложены в ломбард. Гость может только просматривать информацию в любом месте главного меню. Далее были созданы формы работы с договорами, клиентами, справками. В каждой такой форме добавлены методы добавления данных, редактирования их, удаления, обновления и возможность экспорта данных в формат Excel, где можно произвести распечатку нужных нам данных. Также был написан метод поиска по различным полям.

Окно договоры состоит из таблицы, в которой показаны все заключенные договоры. Список этих договоров заполняется из таблицы «contract» и отображается в элементе DataGridView. Можно произвести выбор договоров на основании даты заключения и даты закрытия договоров, выбрав диапазон дат, нажав на CheckBox для выбора. В выборе даты встроен ComboBox,

с выпадающем календарем, где пользователь может выбрать нужные ему даты заключения и закрытия договора. На форме добавлена кнопка добавления договора, которая позволяет добавить новый договор существующему клиенту. На форме располагаются кнопки: «Распечатать» и «Распечатать все». Кнопка «Распечатать все» позволяет произвести экспорт всего списка договоров из таблицы «contract» в формат Excel. Кнопка «Распечатать» позволяет сформировать приходной ордер на клиента по заложенной вещи и квитанцию по выплате кредита за заложенную вещь, которые в последствии можно распечатать также имеется метод, позволяющий осуществить СМС–рассылку для каждому клиенту индивидуально, по введенному его номеру в TextBox. Клиенту получает СМС–оповещение на его номер, что ему необходимо произвести оплату своего кредита на ту или определенную вещь. На форме расположены индикаторы текущей суммы долга по всем договорам и суммы по текущему договору. Здесь были использованы Label для удобного представления информации.

Далее рассмотрим основные фишки программы, точнее работа в Excel, распечатка, приходной кассовый ордер, СМС-рассылка.

Код кнопки «Распечатать»:

```
private void button3_Click(object sender, EventArgs e)
      {
         clientsBindingSource1.Filter = "";
         clientsBindingSource1.Filter = " ID_CLIENT = " +
dataGridView1[4, dataGridView1.CurrentRow.Index].Value.ToString();
         dataGridView3.Refresh();
         thingsBindingSource1.Filter = "";
         thingsBindingSource1.Filter = " ID_THING = " +
dataGridView1[5, dataGridView1.CurrentRow.Index].Value.ToString();
         dataGridView4.Refresh();
         employeeBindingSource1.Filter = "";
         employeeBindingSource1.Filter = " ID_EMP = " +
dataGridView1[6, dataGridView1.CurrentRow.Index].Value.ToString();
         dataGridView5.Refresh();
         typeBindingSource.Filter = "";
         typeBindingSource.Filter = " ID_TYPE = " + dataGridView4[8,
dataGridView1.CurrentRow.Index].Value.ToString();
         dataGridView6.Refresh();
```

```
        podtypeBindingSource.Filter = "";
        podtypeBindingSource.Filter = " ID_PODTYPE = " +
dataGridView4[5, dataGridView1.CurrentRow.Index].Value.ToString();
        dataGridView7.Refresh();
        try
        {
            Microsoft.Office.Interop.Excel.Application ObjExcel = new
Microsoft.Office.Interop.Excel.Application();
            Microsoft.Office.Interop.Excel.Workbook ObjWorkBook;
            Microsoft.Office.Interop.Excel.Worksheet ObjWorkSheet;
            ObjWorkBook                                                =
ObjExcel.Workbooks.Open(Environment.CurrentDirectory + "\\" +
"Order.xls");
            ObjWorkSheet = ObjWorkBook.Sheets[1];
            ObjWorkSheet.Cells[10, 2] = dataGridView3[1,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[8, 11] = dataGridView3[1,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[11, 2] = dataGridView7[1,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[9, 10] = dataGridView7[1,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[11, 4] = dataGridView6[1,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[9, 12] = dataGridView6[1,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[12, 2] = dataGridView1[7,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[10, 10] = dataGridView1[7,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[14, 2] = dataGridView1[10,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjWorkSheet.Cells[12, 11] = dataGridView1[10,
dataGridView1.CurrentRow.Index].Value.ToString();
            ObjExcel.Visible = true;
            ObjExcel.UserControl = true;
            ObjWorkBook.SaveAs(Environment.CurrentDirectory + "\\" +
"Order3.xls");
            ObjExcel.Quit();
```

```
        }
        catch (Exception ex)
        {
            MessageBox.Show("Error:\n" + ex.Message);
        }
    }
```

Код кнопки «Распечатать все»:

```
private void button7_Click_1(object sender, EventArgs e)
    {
        Microsoft.Office.Interop.Excel.Application Exl = new Microsoft.Office.Interop.Excel.Application();
        Microsoft.Office.Interop.Excel.Workbook wb;
        XlReferenceStyle RefStyle = Exl.ReferenceStyle;
        Exl.Visible = true;
        String TemplatePath = System.Windows.Forms.Application.StartupPath + @"\export.xls";
        try
        {
            wb = Exl.Workbooks.Add(TemplatePath); // !!!
        }
        catch (System.Exception ex)
        {
            throw new Exception("Шаблон не найден" + TemplatePath + "\n" + ex.Message);
        }
        Worksheet ws = wb.Worksheets.get_Item(1) as Worksheet;
        for (int j = 0; j < dataGridView1.Columns.Count; ++j)
        {
            (ws.Cells[1, j + 1] as Range).Value2 = dataGridView1.Columns[j].HeaderText;
            for (int i = 0; i < dataGridView1.Rows.Count; ++i)
            {
                object Val = dataGridView1.Rows[i].Cells[j].Value;
                if (Val != null)
                    (ws.Cells[i + 2, j + 1] as Range).Value2 = Val.ToString();
            }       }
        ws.Columns.EntireColumn.AutoFit();
        Exl.ReferenceStyle = RefStyle;
        Exl.Quit();
```

```
        MessageBox.Show("Экспорт успешно произведен");
    }
```

Код кнопки «СМС–рассылка»:

```
private void button5_Click(object sender, EventArgs e)
    {
        using (System.Net.WebClient client = new System.Net.WebClient())
        {
            try
            {
                string url = "http://smsc.vianett.no/v3/send.ashx?" +
                    "dst=+"+textBoxSms+" +
                    "&msg=Срок оплаты истек" +
                    "&username=zendarol@mail.ru" +
                    "&password=dsim";
                string result = client.DownloadString(url);
                if (result.Contains("OK"))
                    MessageBox.Show("Все задолжники уведомлены");
                else
                    MessageBox.Show("Sorry");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message, "Message");
            }       }       }
```

3 Описание работы программного обеспечения

3.1 Требования к аппаратной части

Для определения требований к аппаратной части необходимо сначала определить из каких модулей состоит программа, разработав диаграмму развёртывания. Диаграмма развертывания, содержит графическое отображение модулей всей системы. Основной модуль MS SQL Server 2012 связан с базой данных. Далее к ней привязаны: клиентское приложение, клиентский компьютер, локальная сеть, сервер к которому привязан модуль СМС–оповещения. На рисунке 3.1 показана диаграмма развертывания.

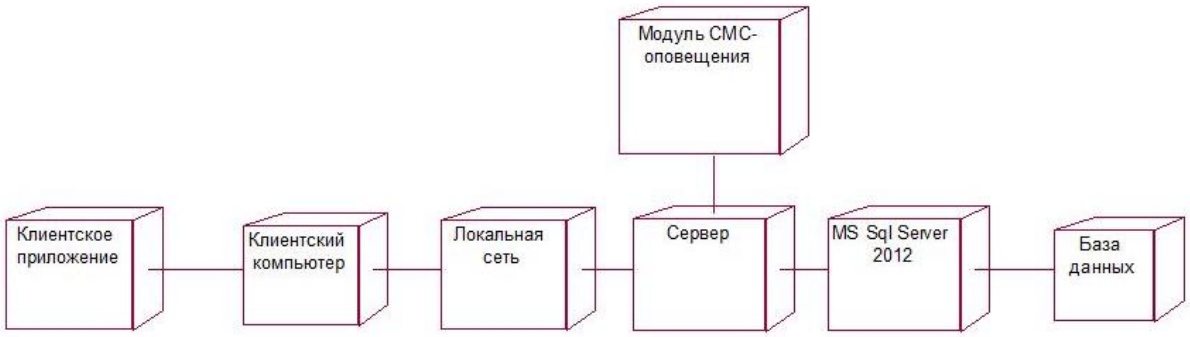


Рисунок 3.1 – Диаграмма развертывания

Появляется модульным, т.е. состоит из модулей: приложения на клиентском компьютере и серверной части. Для безотказной работы серверной части программного обеспечения необходимо иметь один сервер с характеристиками: двух ядерный процессор, более 8 Гб ОЗУ, двух винчестеров размером более 1 Тб и Ethernet интерфейса с пропускной способностью 1 Гб. Такие требования переопределяются тем, что на серверной стороне будет произведена установка серверной ОС Windows Server 2012 R2. Затем будет установлен СУБД MS SQL Server 2012. ОС Windows Server 2012 R2 требуется минимально ОЗУ 2 Гб, а СУБД MS SQL Server 2012 требуется минимально ОЗУ 4 Гб. В системе будут работать единовременно два человека, и база ежедневно будет увеличиваться, увеличивается нагрузка на сеть, поэтому требуется такой интерфейс. Существование двух винчестеров в системе объемом более 1 Тб определено тем, что на одном винчестере будет находиться ОС и БД, а на втором винчестере – резервный образ ОС и бекасы базы данных. Это позволит в наименьшие сроки осуществить восстановление работы серверной части, и работа системы продолжиться.

3.2 Руководство пользователя

Для использования программы требуется пройти авторизацию. Форма входа представлена на рисунке 3.2. Требуется заполнить поля «Пароль» и «Логин». Данные, для которых располагаются в таблице «employee». Таблица «employee» показана на рисунке 3.3.

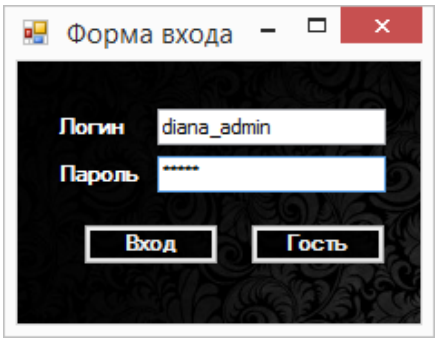


Рисунок 3.2 – Окно авторизации пользователя в программу

Результаты | Сообщения

	ID_EMP	sumame	name	par	dob	adress	issued	login	pass
1	1	Нургалиев	Жангирхан	Сапарулы	27.03.2016	Горпитомник 1 кв 3	МЮ РК	admin	admin
2	3	Хнизова	Диана	Исламовна	29.03.2016	Айэзовский район, Жетысу-2	МЮ РК	diana_admin	diana
3	4	Емелин	Тимур	Викторович	28.04.2016	Ауэзовский район дом 38	МЮ	timur	timur
4	5	Туренко	Елена	Сергеевна	01.05.2016	Орбита -2 дом 56	мю рк	elena	elena

Рисунок 3.3 – Данные с таблицы «employee» для авторизации

После ввода необходимо произвести нажатие кнопки «Вход». Попадаем в главное окно. Главное окно показано на рисунке 3.4.

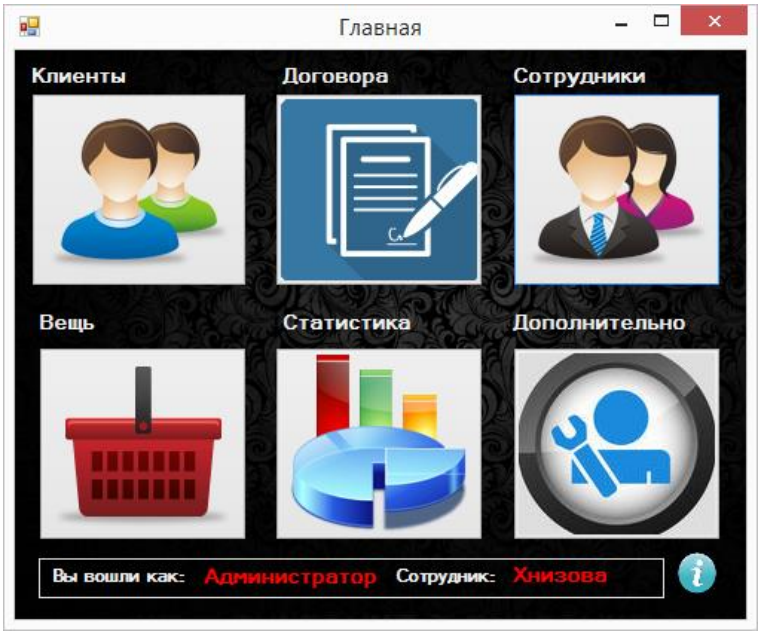
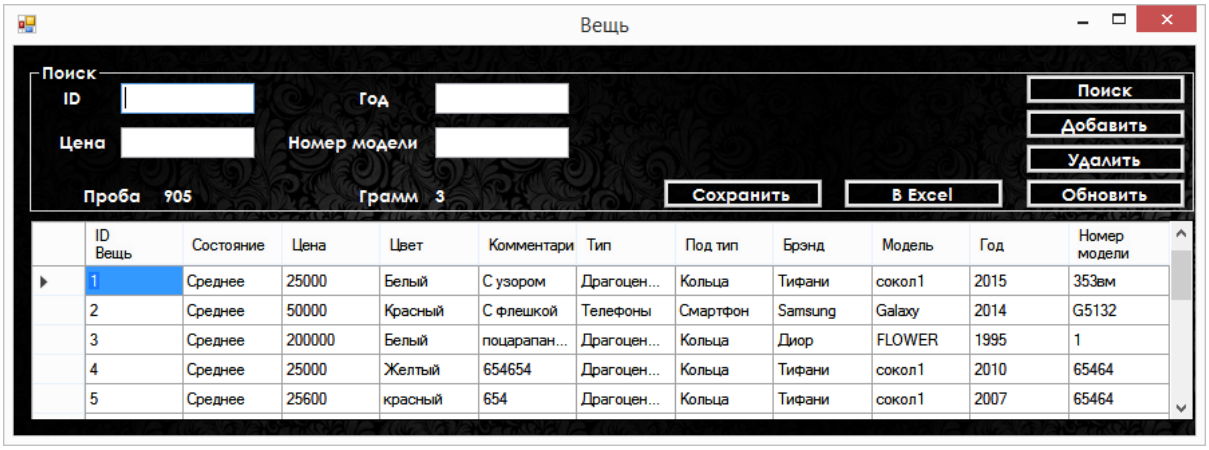


Рисунок 3.4 – Главное меню, где пользователь может работать с договорами, клиентами, вещами

Далее необходимо войти в систему как Администратор Хнизова, здесь позволено выполнить определенные действия. Можно добавлять клиентов, составлять договора на них, добавлять вещи, которые закладывает клиент. Также можно работать с сотрудниками, смотреть статистику ломбарда и в меню «Дополнительно» можно просматривать справки, какие вещи вообще существуют и могут быть заложены в ломбард.

Чтобы начать работу с программой, необходимо определиться с каким клиентом необходимо работать допустим, к нам пришел клиент Парамонов Владимир Алексеевич. Клиент принес в ломбард вещь, а именно золотые серьги. Для начала необходимо просмотреть в меню «Дополнительно» есть ли данные серьги в нашей таблице справок, если есть, то нужно оформить договор на данную вещь данному клиенту. Если таких серёг не оказалось, нам необходимо добавить их в нашу таблицу справок. Допустим, серег, которые принес Парамонов Алексей, не оказалось в нашей таблице справок. Добавим данный вид серег, нажав на кнопку «Вещь», и добавим эту вещь. На рисунке 3.5 показано окно с информацией о заложенных вещах.



ID Вещь	Состояние	Цена	Цвет	Комментари	Тип	Под тип	Брэнд	Модель	Год	Номер модели
1	Среднее	25000	Белый	С узором	Драгоцен...	Кольца	Тифани	сокол1	2015	353вм
2	Среднее	50000	Красный	С флешкой	Телефоны	Смартфон	Samsung	Galaxy	2014	G5132
3	Среднее	200000	Белый	поцарапан...	Драгоцен...	Кольца	Диор	FLOWER	1995	1
4	Среднее	25000	Желтый	654654	Драгоцен...	Кольца	Тифани	сокол1	2010	65464
5	Среднее	25600	красный	654	Драгоцен...	Кольца	Тифани	сокол1	2007	65464

Рисунок 3.5 – Окно с информацией о заложенных вещах клиентов

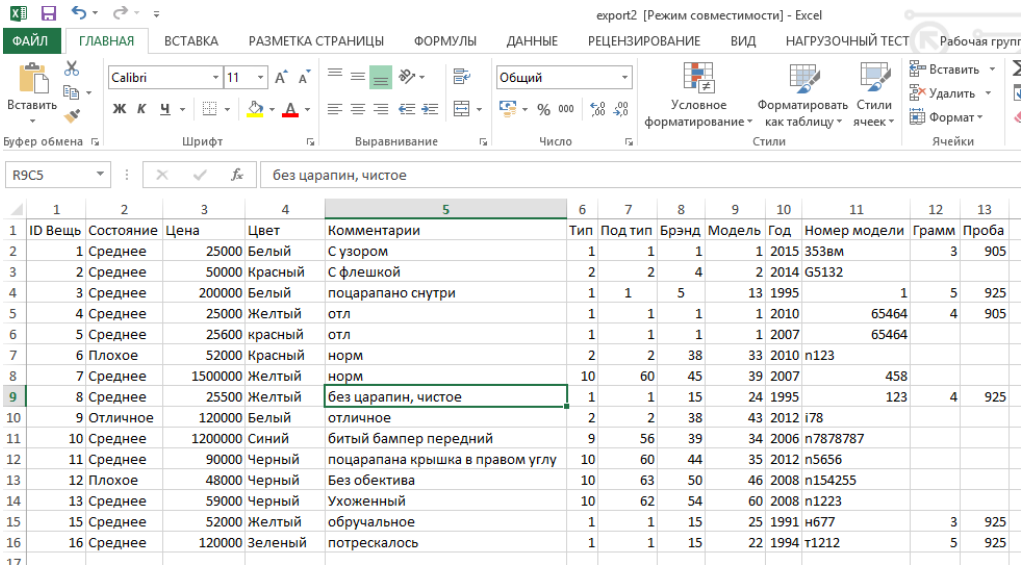
Открылось окно «Вещь». Здесь отображается список вещей, которые уже заложены в ломбард другими клиентами. Как видно, здесь можно произвести поиск вещей по ID, по цене, по номеру модели, по году. Также можно произвести изменения, удаление над данными и сохранить новую информацию в таблицу «things». На рисунке 3.6 показаны данные с таблицы «things».

Результаты | Сообщения

	ID_THING	state	price	color	comments	ID_PODTYPE	year	model_number	ID_MODEL	ID_BRAND	gramm	proba	ID_TYPE
1	1	Среднее	25000	Белый	С узором	1	2015	353вм	1	1	3	905	1
2	2	Среднее	50000	Красный	С флешкой	2	2014	G5132	2	4			2
3	3	Среднее	200000	Белый	поцарапано снутри	1	1995	1	13	5	5	925	1
4	4	Среднее	25000	Желтый	654654	1	2010	65464	1	1	4	905	1
5	5	Среднее	25600	красный	654	1	2007	65464	1	1			1

Рисунок 3.6 – Данные с таблицы «things» для хранения вещей, существующих в ломбарде

Все эти данные возможно вывести в Excel таблицу, и уже работать с ними там (изменять и сохранять в Excel, распечатывать). Экспортированные данные показаны на рисунке 3.7.



export2 [Режим совместимости] - Excel

ФАЙЛ | ГЛАВНАЯ | ВСТАВКА | РАЗМЕТКА СТРАНИЦЫ | ФОРМУЛЫ | ДАННЫЕ | РЕЦЕНЗИРОВАНИЕ | ВИД | НАГРУЗОЧНЫЙ ТЕСТ

R9C5: без царапин, чистое

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	ID Вещь	Состояние	Цена	Цвет	Комментарии	Тип	Под тип	Брэнд	Модель	Год	Номер модели	Грамм	Проба
2	1	Среднее	25000	Белый	С узором	1	1	1	1	2015	353вм	3	905
3	2	Среднее	50000	Красный	С флешкой	2	2	4	2	2014	G5132		
4	3	Среднее	200000	Белый	поцарапано снутри	1	1	5	13	1995	1	5	925
5	4	Среднее	25000	Желтый	отл	1	1	1	1	2010	65464	4	905
6	5	Среднее	25600	красный	отл	1	1	1	1	2007	65464		
7	6	Плохое	52000	Красный	норм	2	2	38	33	2010	n123		
8	7	Среднее	1500000	Желтый	норм	10	60	45	39	2007	458		
9	8	Среднее	25500	Желтый	без царапин, чистое	1	1	15	24	1995	123	4	925
10	9	Отличное	120000	Белый	отличное	2	2	38	43	2012	i78		
11	10	Среднее	1200000	Синий	битый бампер передний	9	56	39	34	2006	n7878787		
12	11	Среднее	90000	Черный	поцарапана крышка в правом углу	10	60	44	35	2012	n5656		
13	12	Плохое	48000	Черный	Без обектива	10	63	50	46	2008	n154255		
14	13	Среднее	59000	Черный	Ухоженный	10	62	54	60	2008	n1223		
15	15	Среднее	52000	Желтый	обручальное	1	1	15	25	1991	н677	3	925
16	16	Среднее	120000	Зеленый	потрескалось	1	1	15	22	1994	т1212	5	925

Рисунок 3.7 – Экспорт данных в Excel

По необходимости можно обновить сохранённые данные, нажав «Обновить». Но в текущий момент времени необходимо произвести добавление вещи, которую принес клиент Парамонов Владимир. Для этого необходимо нажать «Добавить». На рисунке 3.8 представлена форма добавления вещи.

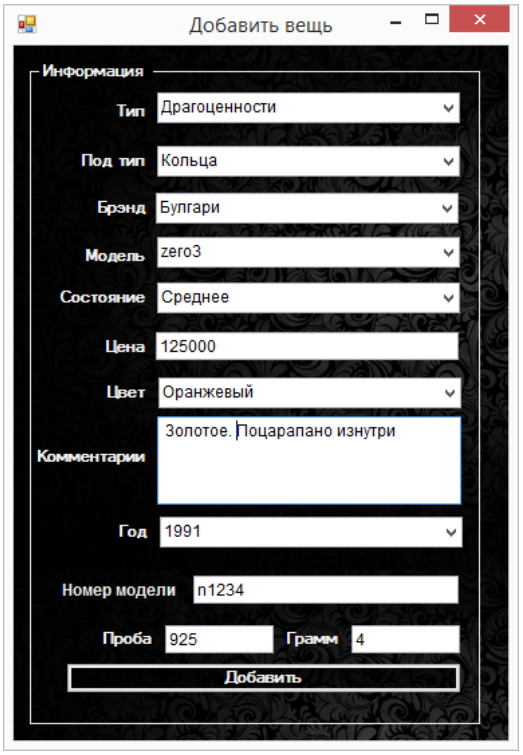


Рисунок 3.8 – Форма добавления вещи

Открылось окно добавления вещи. Теперь нужно выбрать необходимые характеристики вещи, которую закладывают в

ломбард. Допустим, это будет золотое кольцо «Булгари zero3», фактическая цена которого 125000 тг. Выбрав нужные параметры, необходимо нажать «Добавить». На рисунке 3.9 показано сообщение об успешном добавлении.

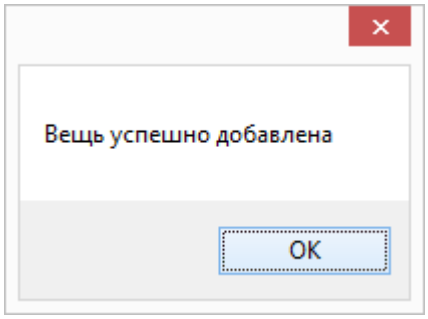
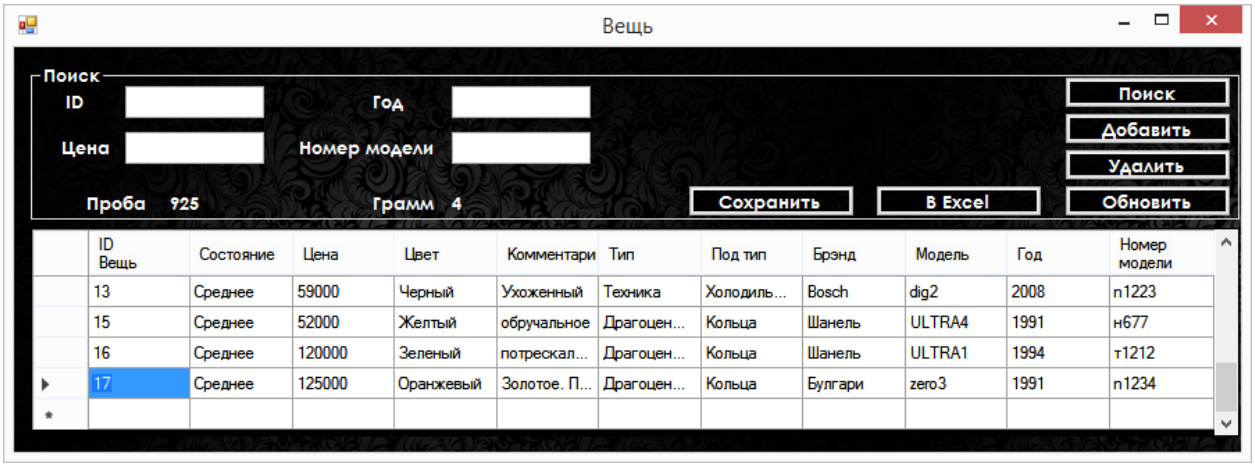


Рисунок 3.9 – Окно сообщения об успешном добавлении вещи

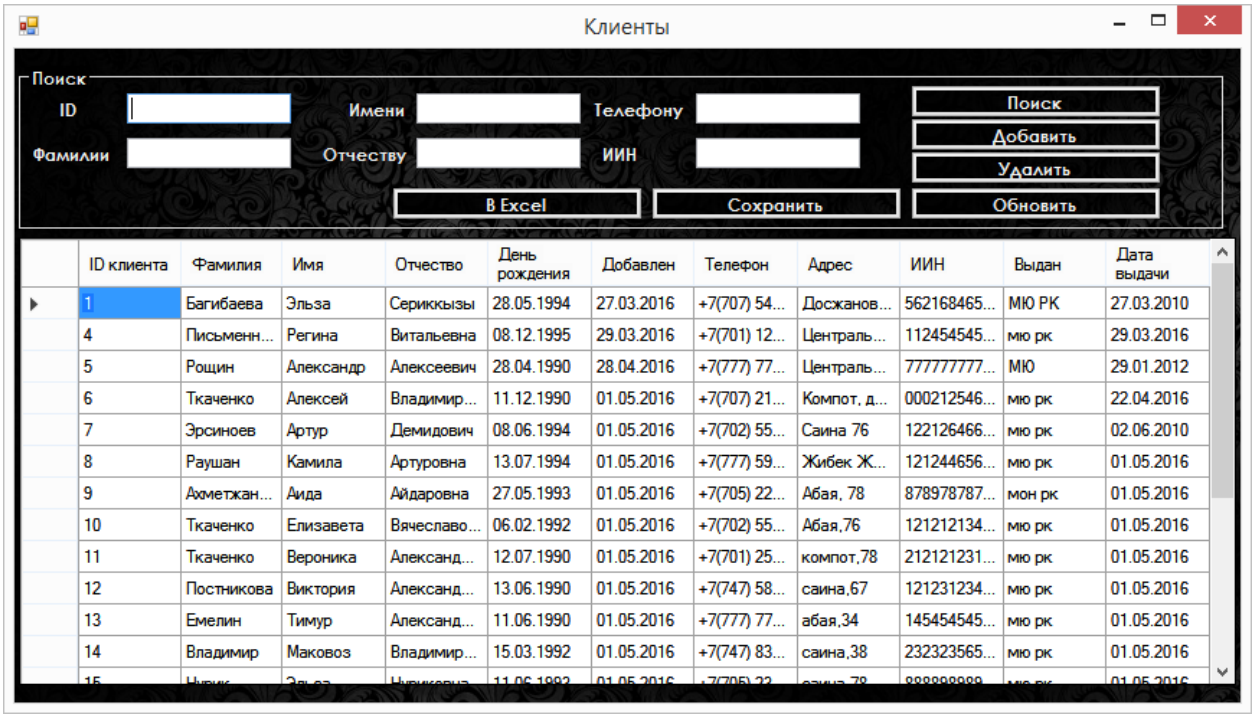
После открывается окно «Вещь», где уже показана добавленная ранее вещь. На рисунке 3.10 окно «Вещь» с добавленной ранее вещью.



ID Вещь	Состояние	Цена	Цвет	Комментари	Тип	Под тип	Брэнд	Модель	Год	Номер модели
13	Среднее	59000	Черный	Ухоженный	Техника	Холодиль...	Bosch	dig2	2008	n1223
15	Среднее	52000	Желтый	обручальное	Драгоцен...	Кольца	Шанель	ULTRA4	1991	н677
16	Среднее	120000	Зеленый	потрескал...	Драгоцен...	Кольца	Шанель	ULTRA1	1994	т1212
17	Среднее	125000	Оранжевый	Золотое. П...	Драгоцен...	Кольца	Булгари	zero3	1991	n1234

Рисунок 3.10 – Окно «Вещь» с добавленной ранее вещью

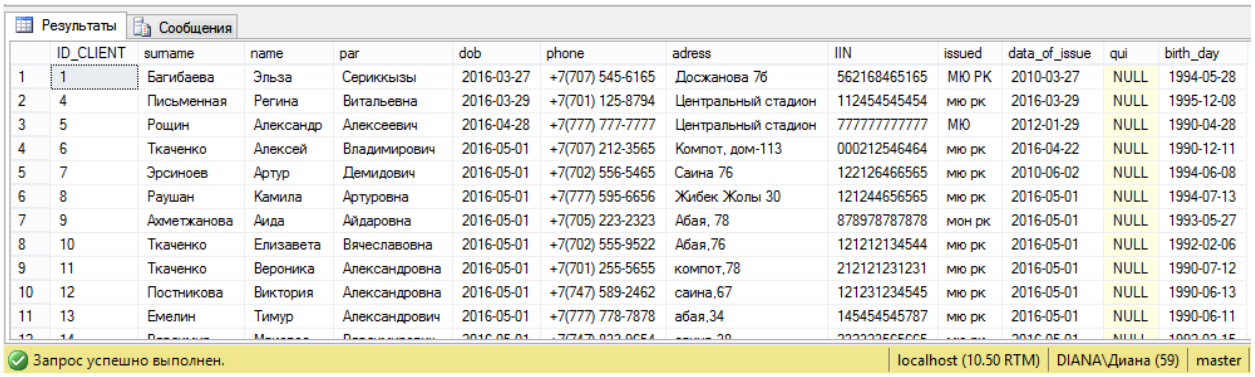
Видно, что необходимая вещь, добавилась. Теперь, надо добавить клиента, а именно Парамонова Владимира Алексеевича. Для этого нужно вернуться в главное окно, и вызвать меню «Клиенты». На рисунке 3.11 показано окно «Клиенты».



ID клиента	Фамилия	Имя	Отчество	День рождения	Добавлен	Телефон	Адрес	ИИН	Выдан	Дата выдачи
1	Багибаева	Эльза	Сериккызы	28.05.1994	27.03.2016	+7(707) 54...	Досжанов...	562168465...	МЮ РК	27.03.2010
4	Письменн...	Регина	Витальевна	08.12.1995	29.03.2016	+7(701) 12...	Централь...	112454545...	мю рк	29.03.2016
5	Рощин	Александр	Алексеевич	28.04.1990	28.04.2016	+7(777) 77...	Централь...	777777777...	МЮ	29.01.2012
6	Ткаченко	Алексей	Владимир...	11.12.1990	01.05.2016	+7(707) 21...	Компот, д...	000212546...	мю рк	22.04.2016
7	Эрсиноев	Артур	Демидович	08.06.1994	01.05.2016	+7(702) 55...	Саина 76	122126466...	мю рк	02.06.2010
8	Раушан	Камила	Артуровна	13.07.1994	01.05.2016	+7(777) 59...	Жибек Ж...	121244656...	мю рк	01.05.2016
9	Ахметжан...	Аида	Айдаровна	27.05.1993	01.05.2016	+7(705) 22...	Абая, 78	878978787...	мон рк	01.05.2016
10	Ткаченко	Елизавета	Вячеславо...	06.02.1992	01.05.2016	+7(702) 55...	Абая,76	121212134...	мю рк	01.05.2016
11	Ткаченко	Вероника	Александ...	12.07.1990	01.05.2016	+7(701) 25...	компот,78	212121231...	мю рк	01.05.2016
12	Постникова	Виктория	Александ...	13.06.1990	01.05.2016	+7(747) 58...	саина,67	121231234...	мю рк	01.05.2016
13	Емелин	Тимур	Александ...	11.06.1990	01.05.2016	+7(777) 77...	абая,34	145454545...	мю рк	01.05.2016
14	Владимир	Маковоз	Владимир...	15.03.1992	01.05.2016	+7(747) 83...	саина,38	232323565...	мю рк	01.05.2016

Рисунок 3.11 – Окно «Клиенты»

Откроется список клиентов, которые содержатся в таблице «clients». На рисунке 3.12 показаны данные из таблицы «clients».



	ID_CLIENT	sumame	name	par	dob	phone	adress	IIN	issued	data_of_issue	qui	birth_day
1	1	Багибаева	Эльза	Сериккызы	2016-03-27	+7(707) 545-6165	Досжанова 7б	562168465165	МЮ РК	2010-03-27	NULL	1994-05-28
2	4	Письменная	Регина	Витальевна	2016-03-29	+7(701) 125-8794	Центральный стадион	112454545454	мю рк	2016-03-29	NULL	1995-12-08
3	5	Рощин	Александр	Алексеевич	2016-04-28	+7(777) 777-7777	Центральный стадион	777777777777	МЮ	2012-01-29	NULL	1990-04-28
4	6	Ткаченко	Алексей	Владимирович	2016-05-01	+7(707) 212-3565	Компот, дом-113	000212546464	мю рк	2016-04-22	NULL	1990-12-11
5	7	Эрсиноев	Артур	Демидович	2016-05-01	+7(702) 556-5465	Саина 76	122126466565	мю рк	2010-06-02	NULL	1994-06-08
6	8	Раушан	Камила	Артуровна	2016-05-01	+7(777) 595-6656	Жибек Жолы 30	121244656565	мю рк	2016-05-01	NULL	1994-07-13
7	9	Ахметжанова	Аида	Айдаровна	2016-05-01	+7(705) 223-2323	Абая, 78	878978787878	мон рк	2016-05-01	NULL	1993-05-27
8	10	Ткаченко	Елизавета	Вячеславовна	2016-05-01	+7(702) 555-9522	Абая,76	121212134544	мю рк	2016-05-01	NULL	1992-02-06
9	11	Ткаченко	Вероника	Александровна	2016-05-01	+7(701) 255-5655	компот,78	212121231231	мю рк	2016-05-01	NULL	1990-07-12
10	12	Постникова	Виктория	Александровна	2016-05-01	+7(747) 589-2462	саина,67	121231234545	мю рк	2016-05-01	NULL	1990-06-13
11	13	Емелин	Тимур	Александрович	2016-05-01	+7(777) 778-7878	абая,34	145454545787	мю рк	2016-05-01	NULL	1990-06-11

Рисунок 3.12 – Данные из таблицы «clients»

Аналогичным методом производится выборка по полям фамилии, отчеству, ID, имени, телефону, ИИН. Можно изменять, удалять, обновлять данные в таблице «clients», затем сохранять эти изменения и работать с ними в Excel. В данном случаем нужно добавить клиента. Для этого необходимо нажать «Добавить». На рисунке 3.13 показано окно добавление клиента.

Рисунок 3.13 – Окно добавления клиента

Вводим данные приходящего клиента. После ввода ИИН клиента, необходимо проверить его, поскольку ИИН не должен повторяться. Необходимо нажать на кнопку «Проверка» и программа проверит, есть ли такой же ИИН. Если таковой ИИН имеется в таблице «clients», то программа выведет ошибку, если ошибка отсутствует - необходимо заполнить данными пустые поля. На рисунке 3.14 показана проверка ИИН.

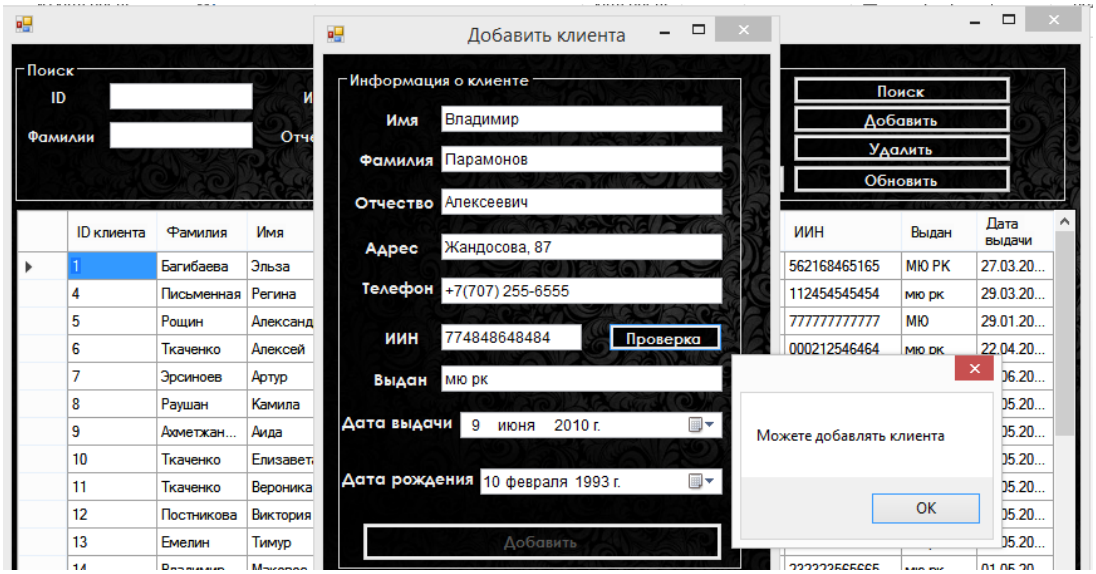
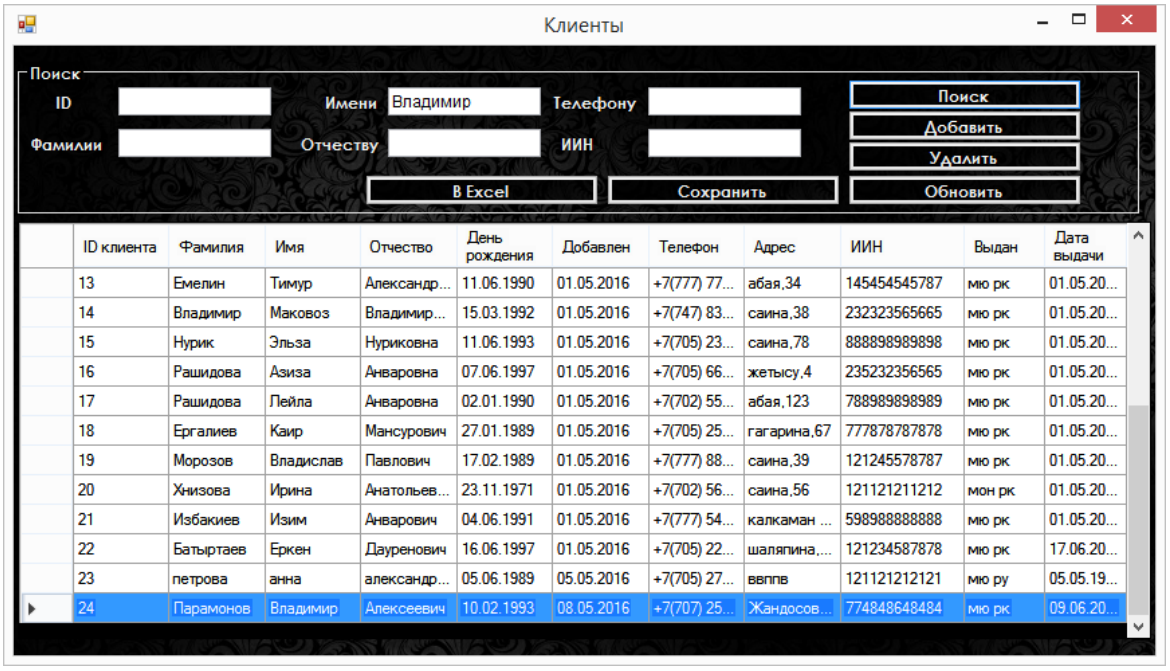


Рисунок 3.14 – Проверка ИИН

Необходимо заполнить данными пустые поля и нажать «Добавить». На рисунке 3.15 показано окно с добавленным клиентом.



ID клиента	Фамилия	Имя	Отчество	День рождения	Добавлен	Телефон	Адрес	ИИН	Выдан	Дата выдачи
13	Емелин	Тимур	Александр...	11.06.1990	01.05.2016	+7(777) 77...	абая,34	145454545787	мю рк	01.05.20...
14	Владимир	Маковоз	Владимир...	15.03.1992	01.05.2016	+7(747) 83...	саина,38	232323565665	мю рк	01.05.20...
15	Нурик	Эльза	Нуриковна	11.06.1993	01.05.2016	+7(705) 23...	саина,78	888898989898	мю рк	01.05.20...
16	Рашидова	Азиза	Анваровна	07.06.1997	01.05.2016	+7(705) 66...	жетысу,4	235232356565	мю рк	01.05.20...
17	Рашидова	Лейла	Анваровна	02.01.1990	01.05.2016	+7(702) 55...	абая,123	788989898989	мю рк	01.05.20...
18	Ергалиев	Каир	Мансурович	27.01.1989	01.05.2016	+7(705) 25...	гагарина,67	777878787878	мю рк	01.05.20...
19	Морозов	Владислав	Павлович	17.02.1989	01.05.2016	+7(777) 88...	саина,39	121245578787	мю рк	01.05.20...
20	Хнизова	Ирина	Анатольев...	23.11.1971	01.05.2016	+7(702) 56...	саина,56	121121211212	мон рк	01.05.20...
21	Избакиев	Изим	Анварович	04.06.1991	01.05.2016	+7(777) 54...	калкаман ...	598988888888	мю рк	01.05.20...
22	Батыртаев	Еркен	Дауренович	16.06.1997	01.05.2016	+7(705) 22...	шаляпина,...	121234587878	мю рк	17.06.20...
23	петрова	анна	александр...	05.06.1989	05.05.2016	+7(705) 27...	ввппв	121121212121	мю ру	05.05.19...
24	Парамонов	Владимир	Алексеевич	10.02.1993	08.05.2016	+7(707) 25...	Жандосов...	774848648484	мю рк	09.06.20...

Рисунок 3.15 – Окно с добавленным клиентом

Видно, что Парамонов добавлен в таблицу «clients». Проверим это, используя поиск по имени.

Далее, нужно оформить договор на этого клиента и вещь, которую он принес в ломбард. Для этого нужно вернуться в главное окно и войти в меню «Договора». На рисунке 3.16 показано окно «Договоры».

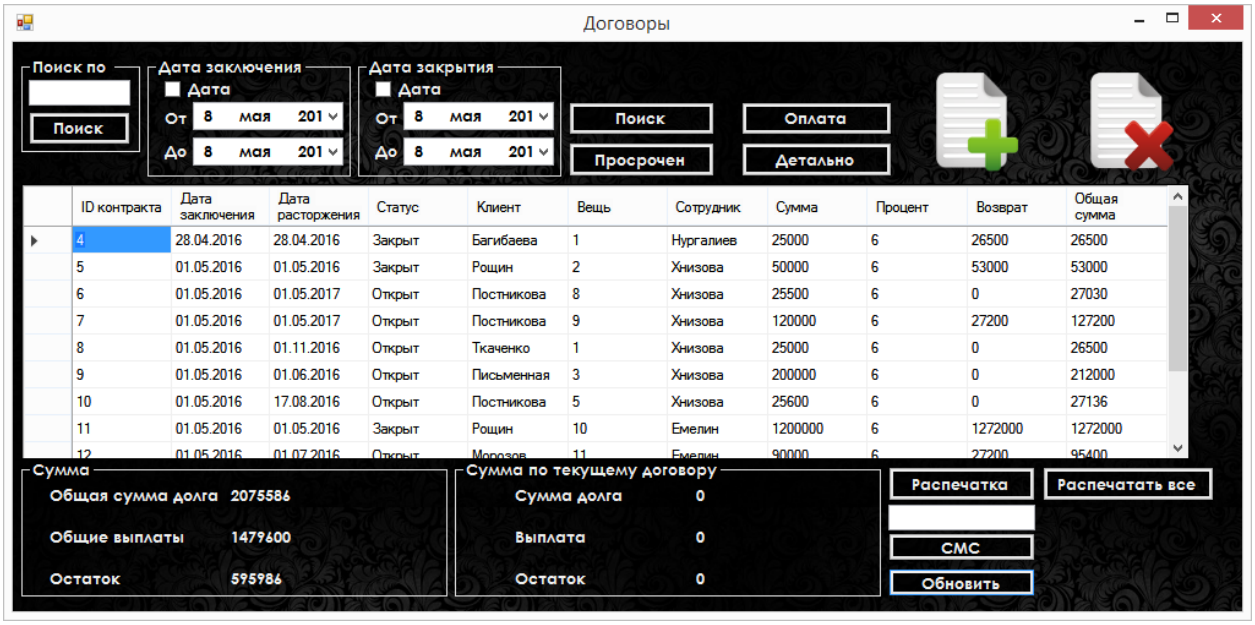


Рисунок 3.16 – Окно «Договоры»

Здесь видно множество возможностей работы с договорами. Выводится список договоров, которые содержатся в таблице «contract». На рисунке 3.17 показаны данные из таблицы «contract».

Результаты | Сообщения

	ID_CONTRACT	data_compilation	close_contract	contr_state	ID_CLIENT	ID_THING	ID_EMPLOYEE	sum	procent	back	summapay
1	4	2016-04-28	2016-04-28	Закрыт	1	1	1	25000	6	26500	26500
2	5	2016-05-01	2016-05-01	Закрыт	5	2	3	50000	6	53000	53000
3	6	2016-05-01	2017-05-01	Открыт	12	8	3	25500	6	0	27030
4	7	2016-05-01	2017-05-01	Открыт	12	9	3	120000	6	27200	127200
5	8	2016-05-01	2016-11-01	Открыт	10	1	3	25000	6	0	26500
6	9	2016-05-01	2016-06-01	Открыт	4	3	3	200000	6	0	212000

Рисунок 3.17 – Данные из таблицы «contract»

Для начала необходимо оформить договор на клиента. Для этого нажмем на значок документа с зеленым плюсом. На рисунке 3.18 показано добавление вещи к договору.

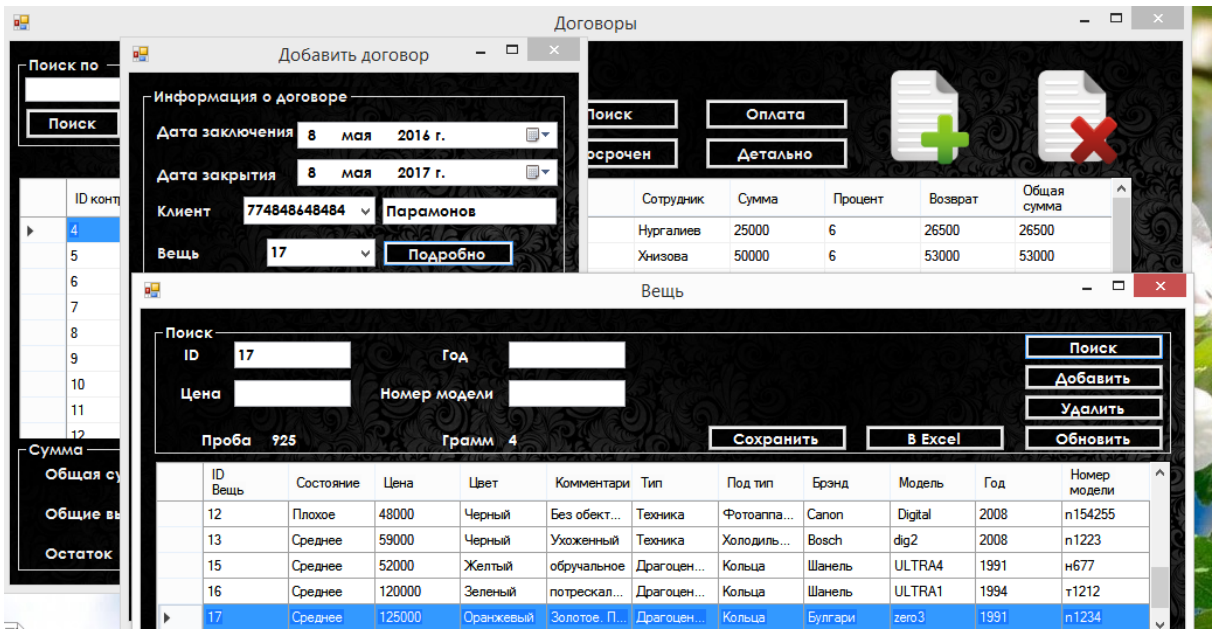


Рисунок 3.18 – Добавление вещи к договору

Оформляем договор на Парамонова Владимира Алексеевича. Дата заключения договора 8 мая 2016 года, дата закрытия 8 мая 2017 года. Таким образом, договор оформлен ровно на год, т.е. Владимиру Алексеевичу необходимо будет произвести выкуп заложенной вещи через год, а именно 8 мая 2017 года. Выберем вещь, которую недавно была добавлена в таблицу «things». На рисунке 3.19 показано окно «Добавить договор» с заполненными полями.

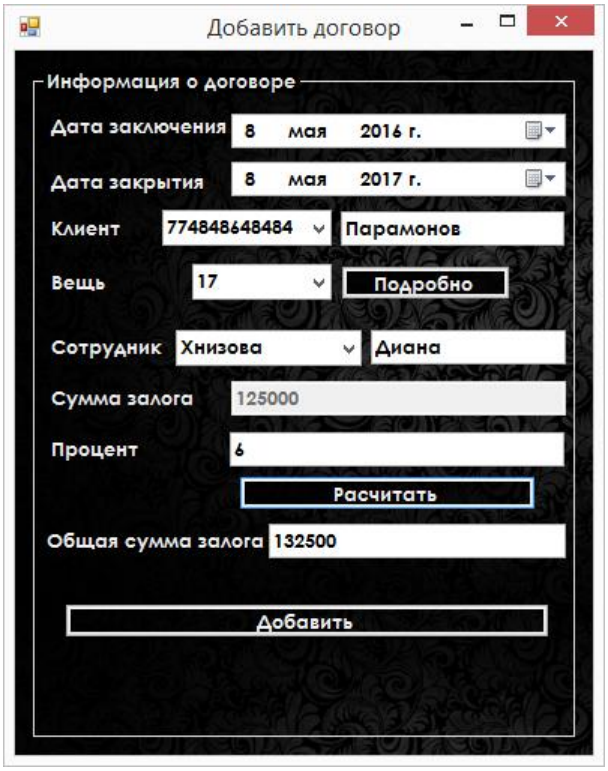


Рисунок 3.19 – Заполненные поля в окне «Добавить договор»

Сотрудник, который оформил данный договор – Хнизова Диана. Изначальная фактическая сумма залога вещи – 125000 тг. Поскольку у нас аннуитетно кредитная система платежа, ломбард берет свой процент – 6%. При аннуитетно кредитной схеме выплат по кредиту, ежемесячный платёж рассчитывается как сумма процентов, начисленных на текущий период и суммы идущей на погашения суммы кредита. Получается, что при 6% комиссии ломбарда, общая сумма залога получится 132500 тг – эта та сумма, которую клиент будет выплачивать в итоге. Поскольку это аннуитетно кредитная система платежа, то клиент может погашать ее частично, а может сразу внести всю сумму долга.

Итак, введя нужные данные, нажимаем на кнопку «Добавить». На рисунке 3.20 показано сообщение об успешном добавлении.

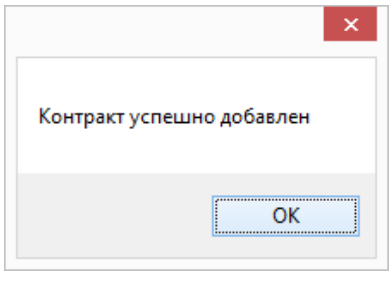


Рисунок 3.20 – Сообщение об успешном добавлении

После добавления видно, что на клиента Парамонова Владимира оформлен договор номер 16. На рисунке 3.21 показан добавленный ранее договор.

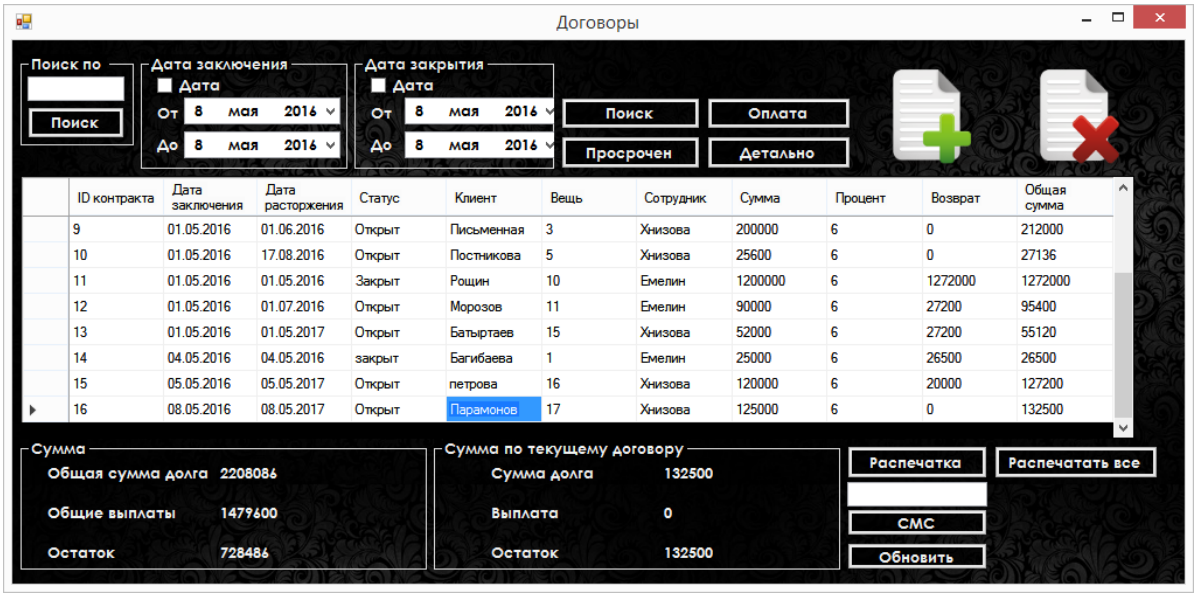
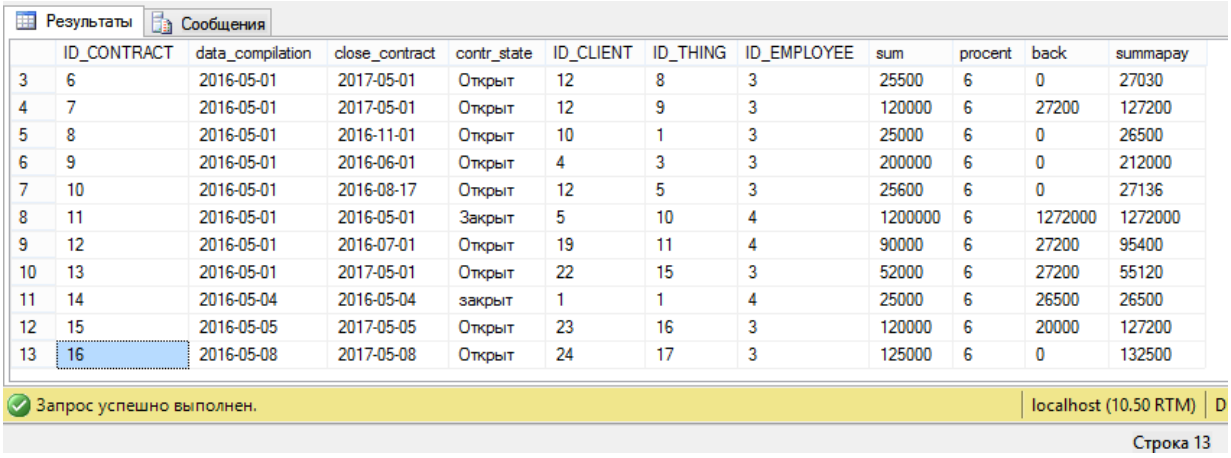


Рисунок 3.21 – Добавленный договор

Также видим, что в таблице «clients» добавился договор номер 16 (рисунок 3.22) на клиента 16 (рисунок 3.23), а именно Парамонова Владимира Алексеевича, который заложил вещь 17 (рисунок 3.24), а именно серьги «Булгари zero3».


Результаты | Сообщения

	ID_CONTRACT	data_compilation	close_contract	contr_state	ID_CLIENT	ID_THING	ID_EMPLOYEE	sum	procent	back	summapay
3	6	2016-05-01	2017-05-01	Открыт	12	8	3	25500	6	0	27030
4	7	2016-05-01	2017-05-01	Открыт	12	9	3	120000	6	27200	127200
5	8	2016-05-01	2016-11-01	Открыт	10	1	3	25000	6	0	26500
6	9	2016-05-01	2016-06-01	Открыт	4	3	3	200000	6	0	212000
7	10	2016-05-01	2016-08-17	Открыт	12	5	3	25600	6	0	27136
8	11	2016-05-01	2016-05-01	Закрыт	5	10	4	1200000	6	1272000	1272000
9	12	2016-05-01	2016-07-01	Открыт	19	11	4	90000	6	27200	95400
10	13	2016-05-01	2017-05-01	Открыт	22	15	3	52000	6	27200	55120
11	14	2016-05-04	2016-05-04	закрыт	1	1	4	25000	6	26500	26500
12	15	2016-05-05	2017-05-05	Открыт	23	16	3	120000	6	20000	127200
13	16	2016-05-08	2017-05-08	Открыт	24	17	3	125000	6	0	132500

Запрос успешно выполнен. | localhost (10.50 RTM) | D
Строка 13

Рисунок 3.22 – Данные из таблицы «contract»

Результаты | Сообщения

	ID_CLIENT	sumame	name	par	dob	phone	adress	IIN	issued	data_of_issue	qui	birth_day
12	14	Владимир	Маковоз	Владимирович	2016-05-01	+7(747) 833-9654	саина,38	232323565665	мю рк	2016-05-01	NULL	1992-03-15
13	15	Нурик	Эльза	Нуриковна	2016-05-01	+7(705) 236-5656	саина,78	888898989898	мю рк	2016-05-01	NULL	1993-06-11
14	16	Рашидова	Азиза	Анваровна	2016-05-01	+7(705) 665-8855	жетысу,4	235232356565	мю рк	2016-05-01	NULL	1997-06-07
15	17	Рашидова	Лейла	Анваровна	2016-05-01	+7(702) 556-5656	абая,123	788989898989	мю рк	2016-05-01	NULL	1990-01-02
16	18	Ергалиев	Каир	Мансурович	2016-05-01	+7(705) 258-9898	гагарина,67	777878787878	мю рк	2016-05-01	NULL	1989-01-27
17	19	Морозов	Владислав	Павлович	2016-05-01	+7(777) 889-8983	саина,39	121245578787	мю рк	2016-05-01	NULL	1989-02-17
18	20	Хнизова	Ирина	Анатольевна	2016-05-01	+7(702) 565-6888	саина,56	121121211212	мон рк	2016-05-01	NULL	1971-11-23
19	21	Избакиев	Изим	Анварович	2016-05-01	+7(777) 545-4487	калкаман 3, 78	598988888888	мю рк	2016-05-01	NULL	1991-06-04
20	22	Батыртаев	Еркен	Дауренович	2016-05-01	+7(705) 222-3564	шаляпина, 90	121234587878	мю рк	2010-06-17	NULL	1997-06-16
21	23	петрова	анна	александровна	2016-05-05	+7(705) 275-4241	ввппв	121121212121	мю ру	1991-05-05	NULL	1989-06-05
22	24	Парамонов	Владимир	Алексеевич	2016-05-08	+7(707) 255-6555	Жандосова, 87	774848648484	мю рк	2010-06-09	NULL	1993-02-10

Запрос успешно выполнен. | localhost (10.50 RTM) | DIANA\Диана (62) | master

Рисунок 3.23 – Данные из «clients»

Результаты | Сообщения

	ID_THING	state	price	color	comments	ID_PODTYPE	year	model_number	ID_MODEL	ID_BRAND	gramm	proba	ID_TYPE
6	6	Плохое	52000	Красный	норм	2	2010	n123	33	38			2
7	7	Среднее	1500000	Желтый	норм	60	2007	458	39	45			10
8	8	Среднее	25500	Желтый	без царапин, чистое	1	1995	123	24	15	4	925	1
9	9	Отличное	120000	Белый	отличное	2	2012	i78	43	38			2
10	10	Среднее	1200000	Синий	битый бампер передний	56	2006	n7878787	34	39			9
11	11	Среднее	90000	Черный	поцарапана крышка в правом углу	60	2012	n5656	35	44			10
12	12	Плохое	48000	Черный	Без обектива	63	2008	n154255	46	50			10
13	13	Среднее	59000	Черный	Ухоженный	62	2008	n1223	60	54			10
14	15	Среднее	52000	Желтый	обручальное	1	1991	н677	25	15	3	925	1
15	16	Среднее	120000	Зеленый	потрескалось	1	1994	т1212	22	15	5	925	1
16	17	Среднее	125000	Оранж...	Золотое. Поцарапано изнутри	1	1991	n1234	5	10	4	925	1

Запрос успешно выполнен. | localhost (10.50 RTM) | DIANA\Диана (63) | maste

Строка 16 Столбец 1

Рисунок 3.24 – Данные из таблицы «things»

Теперь произведем поиск, договоров, которые были оформлены 8 мая 2016 года. Открылся договор на Парамонова, который был оформлен недавно. На рисунке 3.25 показан поиск по номеру договора.

Рисунок 3.25 – Поиск по номеру договора

Теперь обратим внимание на дополнительные возможности работы над договорами. Заметно изменилась общая сумма долга ломбарду, когда был добавлен клиент Парамонов. Изначально общая сумма долга ломбарду составляла – 2075586 тг. Поскольку Парамонов Владимир взял в залог сумму 132500 тг, то обновился индикатор «Общая сумма долга» на сумму 2208086 тг. Таким образом, после оформления последнего договора на Парамонова увеличилась общая сумма долга со всех договоров ломбарду. Индикатор «Общие выплаты» показывает нам те выплаты, со всех договоров, которые уже произвели наши клиенты. Индикатор «Остаток» показывает, сколько еще осталось выплатить ломбарду со всех договоров, клиентам. Индикатор «Сумма по текущему договору», показывает сумму долгу по текущему договору, который мы выберем. Также здесь показывается, сколько уже выплатил клиент, и сколько ему еще нужно выплатить ломбарду. Произведём поиск по номеру договора номер 7 (рисунок 3.26).

Рисунок 3.26 – Поиск по договору номер 7

Здесь виден договор номер 7 на клиента Постникова. Видим, что общая сумма ее долга ломбарду – 127200 тг, выплатила она уже – 27200 тг, и осталось ей проплатить – 100000 тг.

Теперь, пришли к вопросу «а как же производятся выплаты клиента ломбарду?». Ответ прост. Необходимо нажать на кнопку «Оплата». На рисунке 3.27 показано окно «Оплата», в котором отображаются выплаты клиента ломбарду.

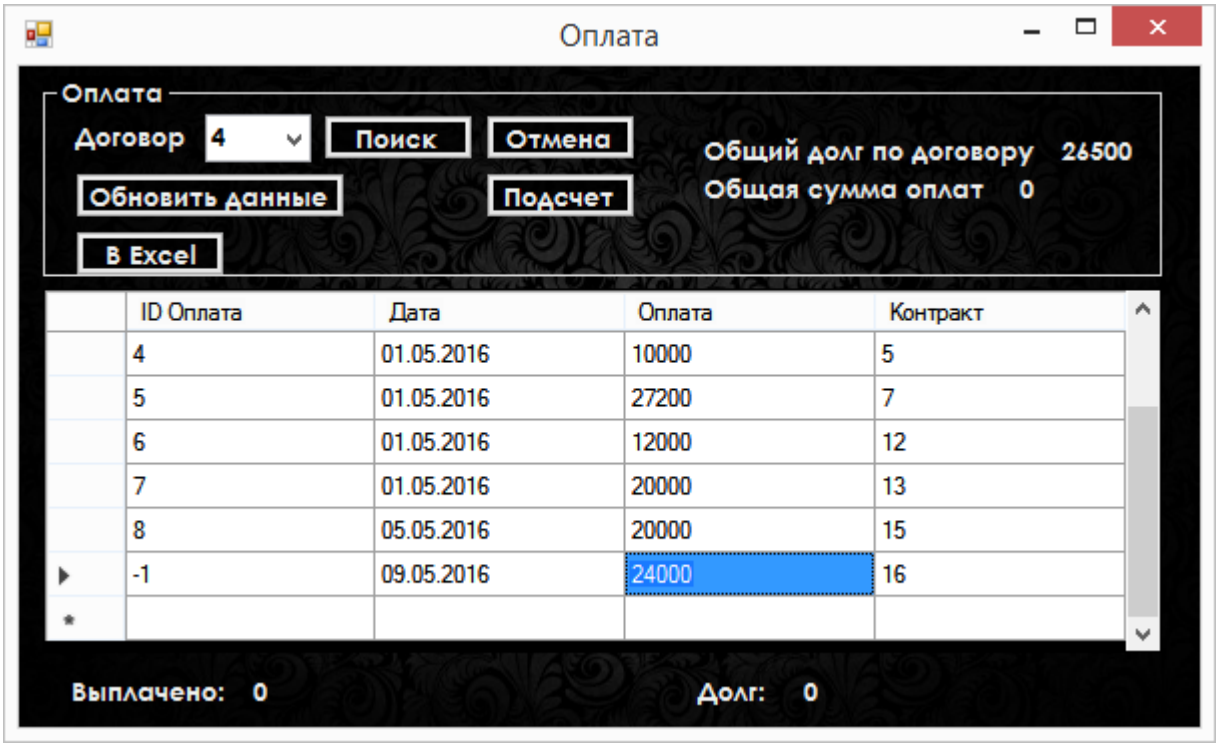


Рисунок 3.27 – Окно «Оплата» для отображения выплаты клиента

Здесь показан список оплат, которые были произведены клиентами по договору из таблицы «credit». На рисунке 3.28 показаны данные из таблицы «credit».

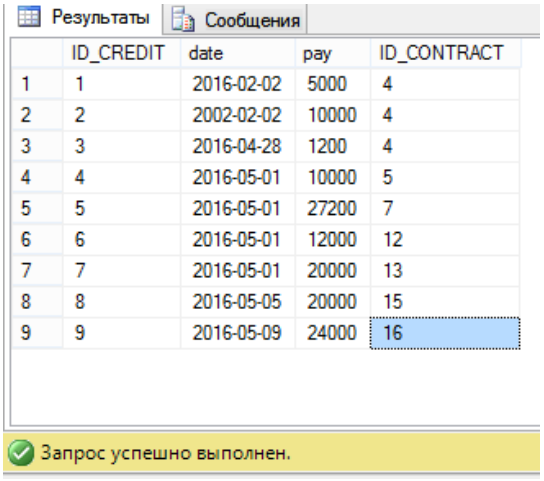


Рисунок 3.28 – Данные из таблицы «credit»

Допустим, Парамонов Владимир пришел 9 мая 2016 года внести некоторую сумму для погашения своего долга в размере – 24000 тг. Заполнив, эти данные можно наблюдать следующие действия. На рисунке 3.29 показана информация об оплате по договору 16.

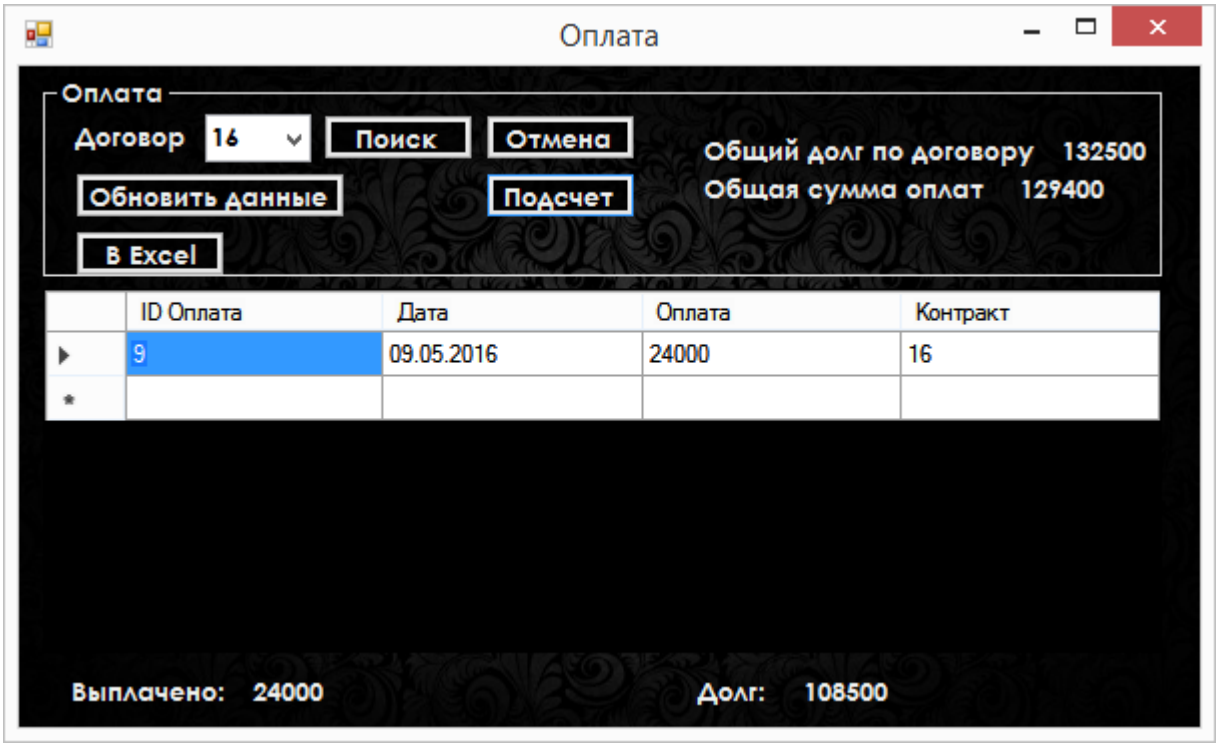


Рисунок 3.29 – Информация об оплате по договору 16

Выбрав в поиске по договору, договор номер 16, нажмем на кнопку «Подсчет». На индикаторе «Общий долг по договору» показывается та сумма, которую вообще должен Парамонов Владимир – 132500 тг, и на индикаторе «Общая сумма оплат» наблюдаем общую сумму оплат, которую произвели все клиенты ломбарду. В нижней части окна видим индикатор «Выплачено» – это та сумма, которую на данный момент выплатил Парамонов Владимир – 24000 тг, и на индикаторе «Долг» видим ту сумму, которую еще необходимо еще выплатить Владимиру в размере 108500 тг. Таким образом, общая сумма долга Парамонова Владимира ломбарду составит 132500 тг в общем. Также здесь можно обновить данные, работать с ними в Excel. На рисунке 3.30 показано окно «Договоры» с обновленными данными договора 16.

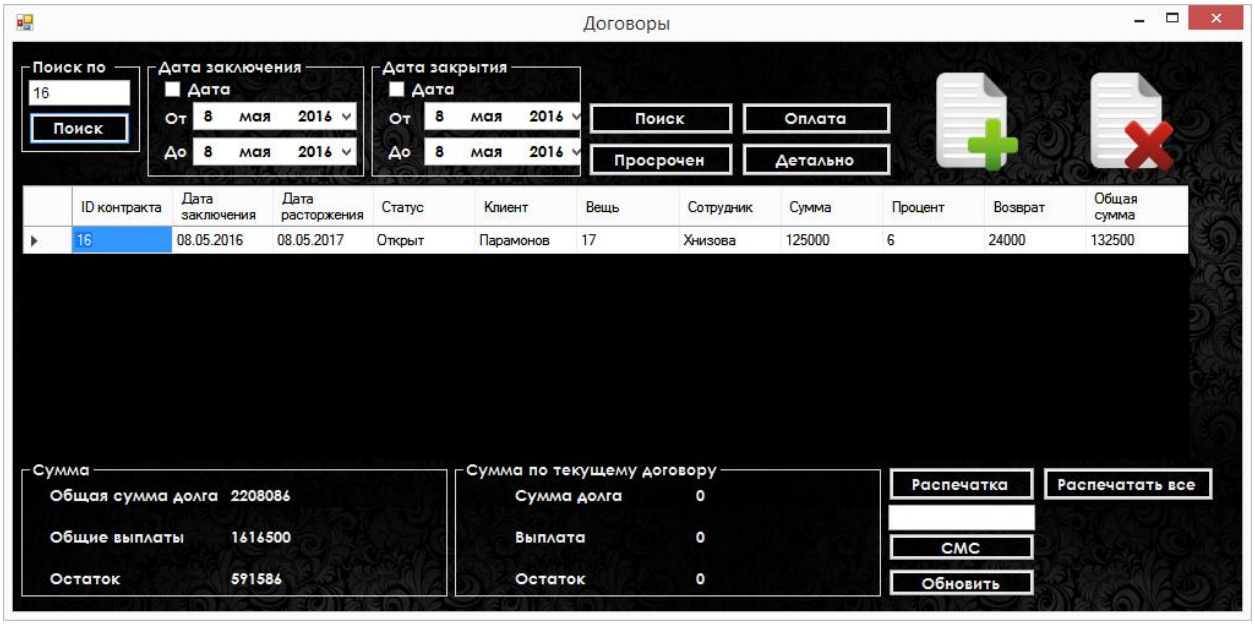
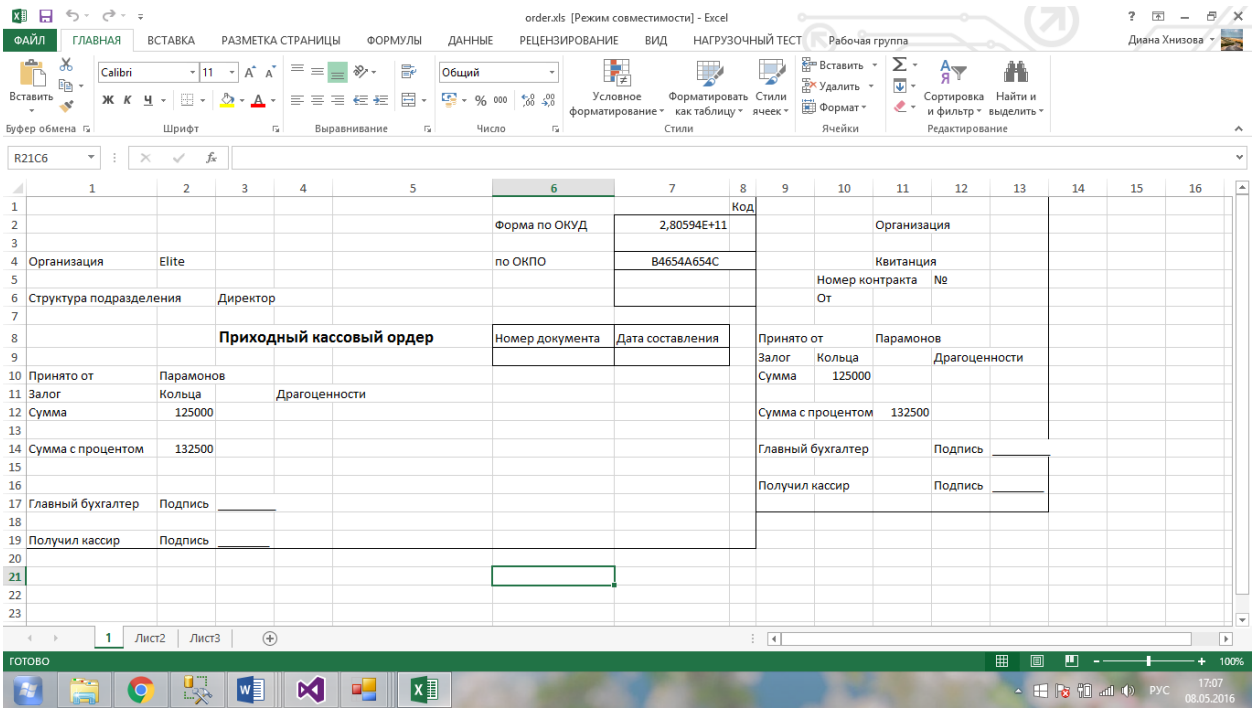


Рисунок 3.30 – Окно «Договоры» с обновлённой информацией договора номер 16

Теперь, произведя поиск по ID договору 16, видно в столбце «возврат» сумму, которую нам уже выплатил Парамонов, а именно 24000 тг.

Нажав на кнопку «Распечатка» мы можем распечатать квитанции для клиента, а именно – Приходный кассовый ордер, и квитанцию по выплате. На рисунке 3.31 показан приходный кассовый ордер и квитанция, которые автоматически заполняются во время экспорта.



							Код									
					Форма по ОКУД	2,80594E+11					Организация					
Организация	Elite				по ОКПО	B4654A654C					Квитанция					
										Номер контракта	№					
Структура подразделения		Директор								От						
		Приходный кассовый ордер			Номер документа	Дата составления		Принято от			Парамонов					
								Залог	Кольца			Драгоценности				
Принято от	Парамонов							Сумма	125000							
Залог	Кольца		Драгоценности													
Сумма	125000							Сумма с процентом	132500							
Сумма с процентом	132500							Главный бухгалтер			Подпись					
								Получил кассир			Подпись					
Главный бухгалтер	Подпись															
Получил кассир	Подпись															

Рисунок 3.31 – Экспортированные данные об оплате

Также в программе есть СМС – рассылка на нужный нам номер телефона клиента. Вводим номер клиента – 87052754341 и нажимаем на кнопку «СМС» и на этот номер телефона отправится СМС–сообщение о том, что наш задолжник должен произвести оплату в ломбард. На рисунке 3.32 показано СМС–уведомление.

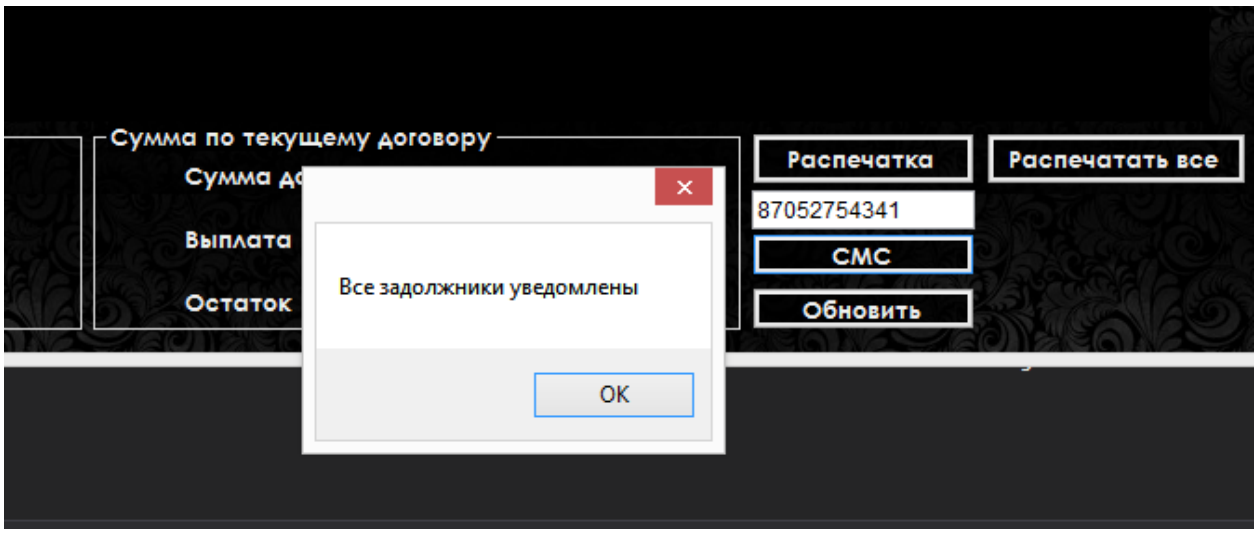
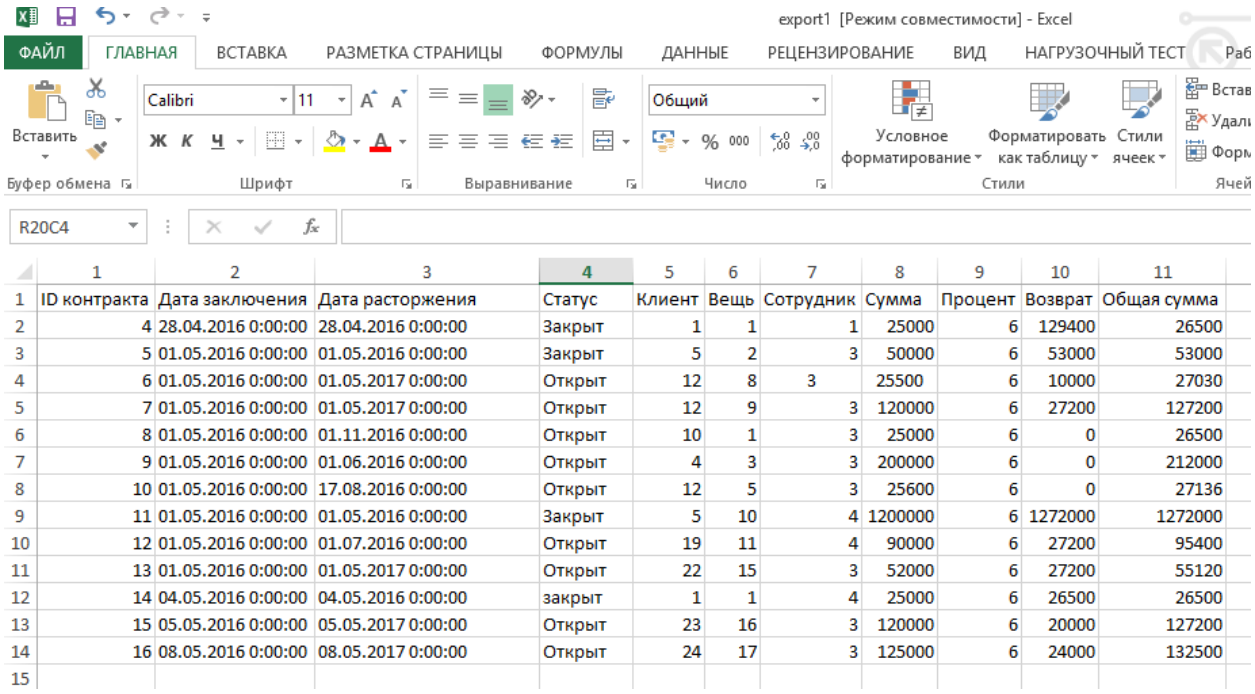


Рисунок 3.32 – СМС–уведомление

Нажав на кнопку «Распечатать все» мы можем работать со списком в Excel, печатать. На рисунке 3.33 показаны экспортированные данные всех договоров.



	1	2	3	4	5	6	7	8	9	10	11
1	ID контракта	Дата заключения	Дата расторжения	Статус	Клиент	Вещь	Сотрудник	Сумма	Процент	Возврат	Общая сумма
2	4	28.04.2016 0:00:00	28.04.2016 0:00:00	Закрыт	1	1	1	25000	6	129400	26500
3	5	01.05.2016 0:00:00	01.05.2016 0:00:00	Закрыт	5	2	3	50000	6	53000	53000
4	6	01.05.2016 0:00:00	01.05.2017 0:00:00	Открыт	12	8	3	25500	6	10000	27030
5	7	01.05.2016 0:00:00	01.05.2017 0:00:00	Открыт	12	9	3	120000	6	27200	127200
6	8	01.05.2016 0:00:00	01.11.2016 0:00:00	Открыт	10	1	3	25000	6	0	26500
7	9	01.05.2016 0:00:00	01.06.2016 0:00:00	Открыт	4	3	3	200000	6	0	212000
8	10	01.05.2016 0:00:00	17.08.2016 0:00:00	Открыт	12	5	3	25600	6	0	27136
9	11	01.05.2016 0:00:00	01.05.2016 0:00:00	Закрыт	5	10	4	1200000	6	1272000	1272000
10	12	01.05.2016 0:00:00	01.07.2016 0:00:00	Открыт	19	11	4	90000	6	27200	95400
11	13	01.05.2016 0:00:00	01.05.2017 0:00:00	Открыт	22	15	3	52000	6	27200	55120
12	14	04.05.2016 0:00:00	04.05.2016 0:00:00	закрыт	1	1	4	25000	6	26500	26500
13	15	05.05.2016 0:00:00	05.05.2017 0:00:00	Открыт	23	16	3	120000	6	20000	127200
14	16	08.05.2016 0:00:00	08.05.2017 0:00:00	Открыт	24	17	3	125000	6	24000	132500
15											

Рисунок 3.33 – Экспортированные данные всех договоров

Нажав на кнопку «Просрочен» видно все просроченный договора, т.е. те, по которым уже была произведена вся оплата. На рисунке 3.34 представлена информация о просроченных договорах.

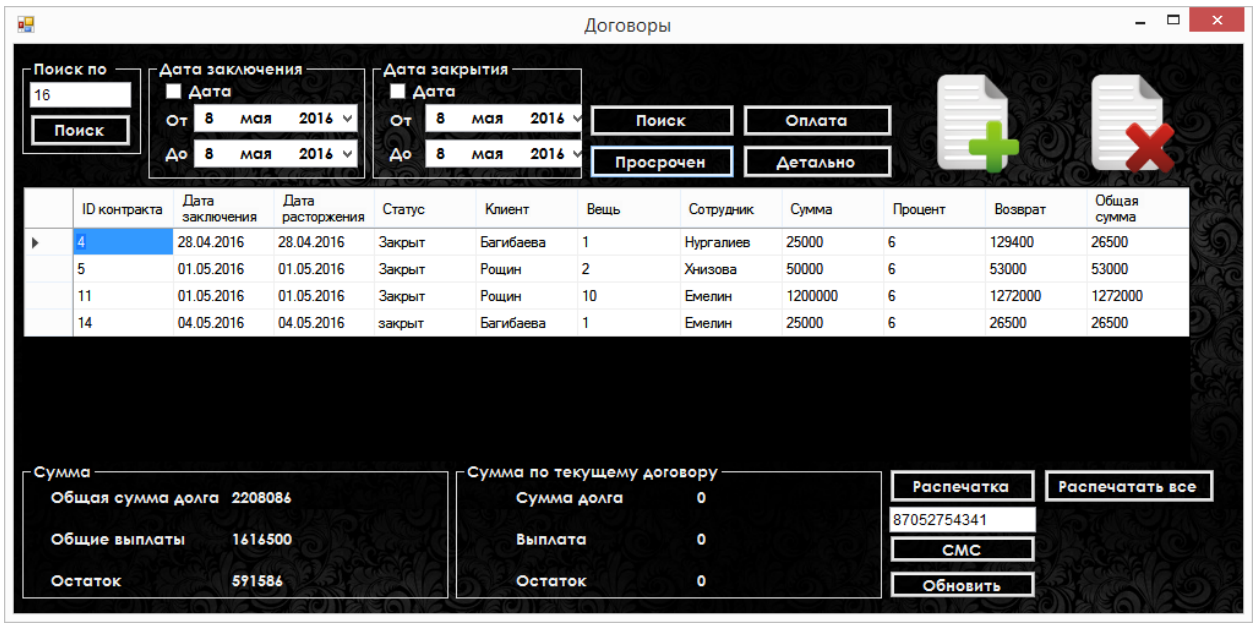


Рисунок 3.34 – Информация о просроченных договорах

Теперь вернемся в главное окно, и посмотрим статистику ломбарда. На рисунке 3.35 показана статистика договоров.

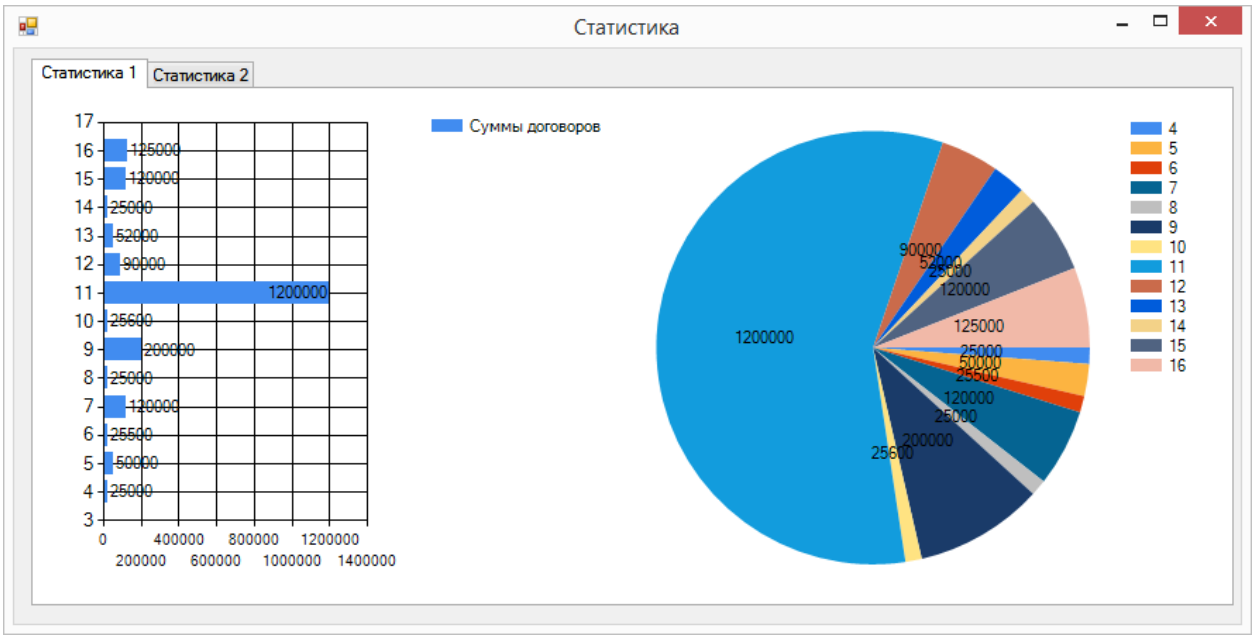


Рисунок 3.35 – Статистика договоров

Здесь отображается статистика договоров, какую прибыль они предоставляют ломбарду. Например, договор номер 4, приносит ломбарду сумму в размере 1200000 тг – это самая большая сумма, которая хранится в кассе ломбарда.

Также можно просмотреть статистику по товарам, которые хранятся в ломбарде. На рисунке 3.36 показана информация о количестве товаров, хранящихся в ломбарде.

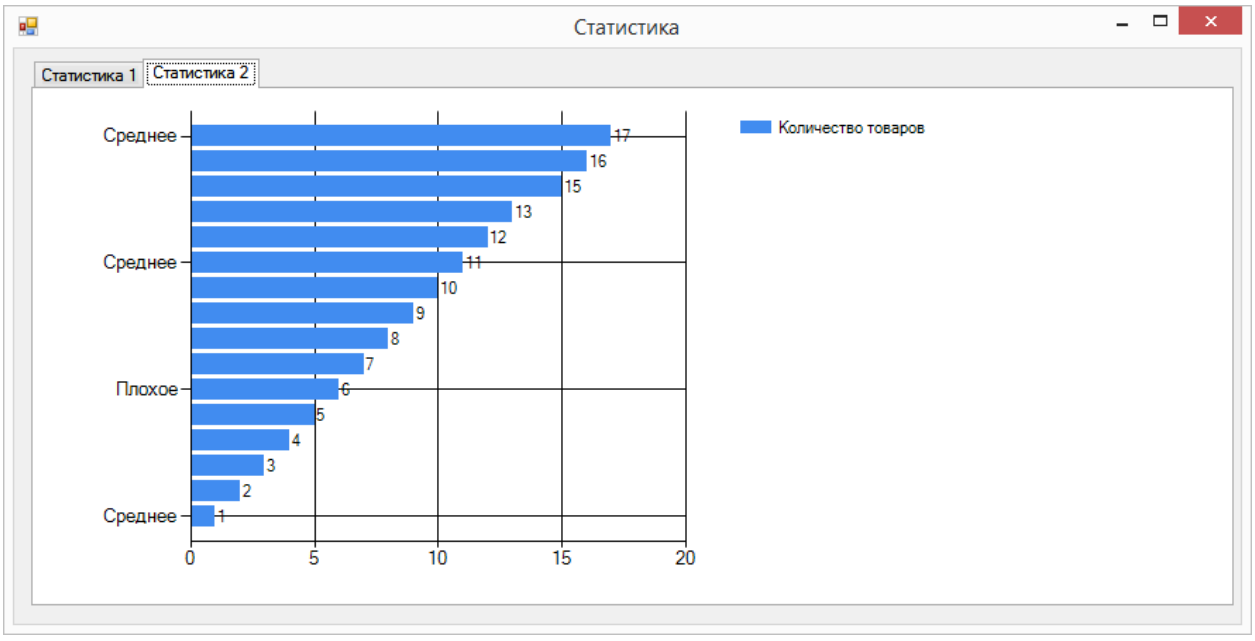


Рисунок 3.36 – Статистика количества товаров, хранящихся в ломбарде

Вернемся в меню «Дополнительно» и рассмотрим справку ломбарда. На рисунке 3.37 показано окно «Дополнительно».

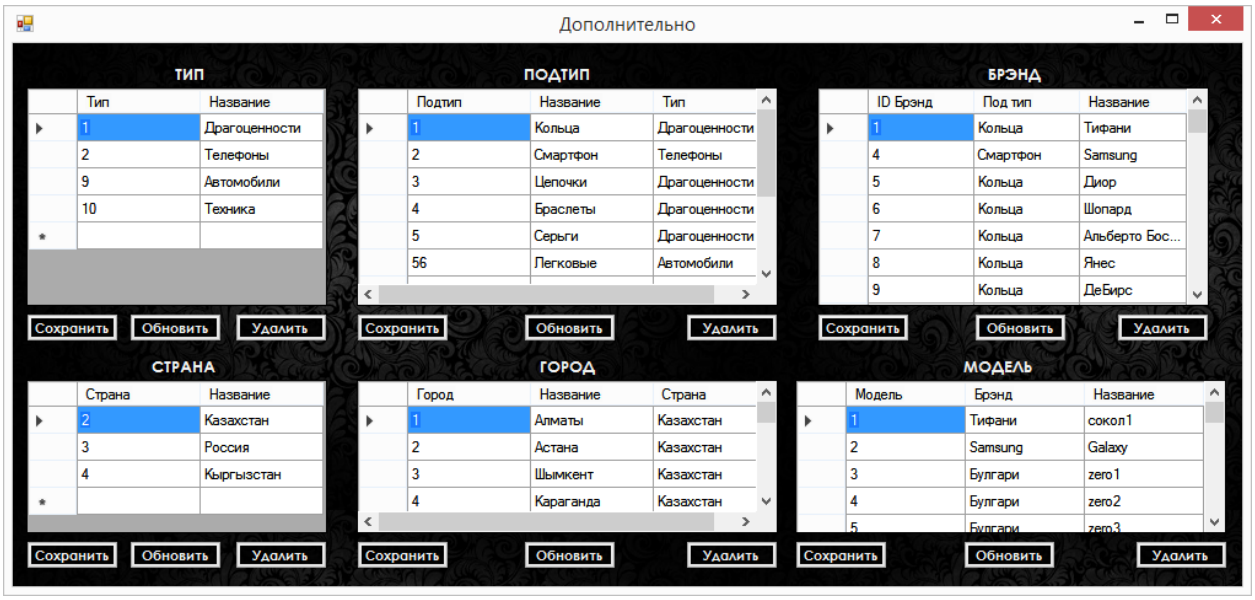


Рисунок 3.37 – Окно «Дополнительно»

Здесь видно справку вещей, которые существуют в таблицах базы данных ломбарда. Например, если опираться на справку вещей то, пусть тип вещи будет – 1 Драгоценности, подтип – 1 Кольца, Брэнд – 1 Тифани, Модель – 1 сокол1.

Например, если опираться на справку стран и городов, в котором проживают наши сотрудники, то, пусть Страна – 2 Казахстан, Город – 1 Алматы.

На рисунке 3.38 показаны реквизиты разработчика.

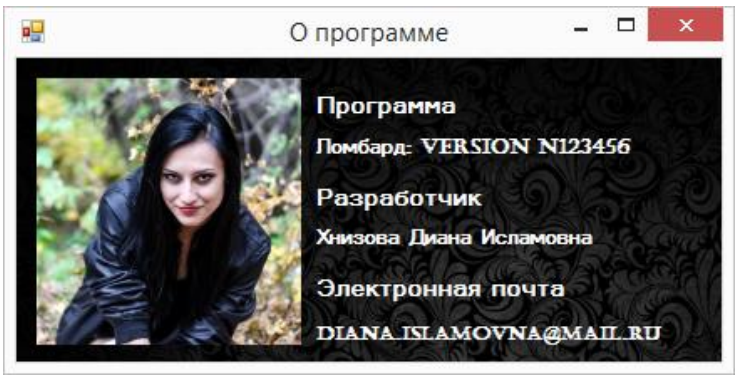


Рисунок 3.38 – Реквизиты разработчика

4 Экономическое обоснование проекта

4.1 Описание работы и обоснование необходимости

Целью данного дипломного проекта является создание БД для ломбарда. Коммерческий и предпринимательский успех сильно зависит от использования информационных систем, которые позволяют автоматизировать бизнес частично или полностью. Также должны обеспечивать огромную скорость передачи данными между компонентами, снижая трудоёмкость рутинной работы, такой как учет и обработка данных. Это всё позволяет оперативно производить обмен между отделами. Ломбард сильно отличается от других предприятий, являясь сложным экономически–социальным объектом, на деятельность которого оказывает влияние целого ряда финансово–экономических, социально–психологических и правовых аспектов. Деятельность фирм иногда не уделяет внимание изменениям, происходящим в схеме ломбардных услуг. Это всё выражается в отсутствии высокого спроса на потребление услуг. Это всё сопровождается финансовыми потерями предприятий, организационно–экономический механизм функционирования ломбардов в нынешних условиях отсутствует, которые учитывает изменения ситуаций в макроэкономике страны, требуемая нормативно–правовая база отсутствует, которая должна регулировать развитие описываемого вида деятельности. Ломбарды традиционно относятся к предприятиям бытового обслуживания населения.

4.2 Трудовые ресурсы, используемые в работе.

В данной работе используется интеллектуальный труд, стоимость затрат которого выше, чем физического труда.

В ломбарде работают следующие сотрудники:

– программист – разработка программного обеспечения;

– дизайнер – разработка интерфейса;

– разработчик базы данных – разработка и администрирование
базы данных.

Количество сотрудников, задействованных в разработке представлено в
таблице 4.1.

Таблица 4.1 – Сотрудники и их заработная плата

Должность	Количество человек	Заработная плата в месяц, тенге
Программист	1	90000
Дизайнер	1	70000
Разработчик базы данных	1	70000
Итого	3	230000

4.2.1 Оборудование, используемое в работе

Оборудование, которое используется при разработке программного обеспечения представлено в таблице 4.2.

Таблица 4.2 – Перечень оборудования для разработки ПО

Наименование изделий	Характеристика	Количество единиц	Цена за единицу, тенге	Общая сумма, тенге
Ноутбук	Intel Core i7–3632QM 8Gb DDR3 Memory Nvidia GeForce 710M HDD 750Gb 15.6" 16:9 HD LED LCD 6–cell Li–ion battery	1	110000	110000
Принтер	HP Deskjet Ink Advantage 1515 All–in–One(B2L57C)	1	20000	20000
Итого:				130000

Цены на оборудование представлены без учета НДС.

4.3.2 Программное обеспечение, используемое в работе

В процессе разработки программного обеспечения и базы данных для ломбарда необходимо было использовать следующее программное обеспечение:

– Microsoft Windows Professional 8.1 – операционная система;

– Visual Studio 2013 – набор инструментов для создания программного обеспечения;

– Microsoft SQL Server – система управления реляционной базы данных (РСУБД), разработанная корпорацией Microsoft.

Программное обеспечение, которое использовалось при разработке программного обеспечения, представлено в таблице 4.3.

Таблица 4.3 – программное обеспечение, необходимое для разработки программного обеспечения и базы данных для ломбарда

Программное обеспечение	Стоимость, тенге
Microsoft Windows Professional 8.1	30000
Visual Studio 2013	15000
Microsoft SQL Server	бесплатно
Итого:	45000

Цены на ПО приведены без учета НДС.

4.3.3 Сроки реализации проекта

Весь процесс разработки, а также сроки реализации программного обеспечения и базы данных для ломбардов состоит из 9 этапов и включает в себя:

– постановка задачи;

– изучение и сбор информации для создания базы данных;

– проектирование и создание базы данных для ломбарда;

– изучение и сбор информации для проектирования и создания программного обеспечения;

– разработка дизайна и интерфейса программного обеспечения;

– разработка и написание программного кода;

– отладка;

– тестирование программного обеспечения;

– оформление отчета.

График реализации и разработки проекта показан в таблице 4.4.

Таблица 4.4 – График разработки и реализации проекта

Перечень работ		Недели от начала работ									
		1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10	11	12
1 этап	Постановка задачи										
2 этап	Изучение и сбор информации для создания базы данных										
3 этап	Проектирование и создание базы данных для ломбарда										
4 этап	Изучение и сбор информации для проектирования и создания программного обеспечения										
5 этап	Разработка дизайна и интерфейса программного обеспечения										
6 этап	Разработка и написание программного кода										

Продолжение таблицы 4.4

1	2	3	4	5	6	7	8	9	10	11	12
7 этап	Отладка										
8 этап	Тестирование программного обеспечения										

9 этап	Оформление отчета										

4.3.4 Расчет стоимости работы по разработке

Для того, чтобы определить использование и затраты времени на разработку программного обеспечения на каждом его этапе, а также трудовые затраты, необходимо рассчитать стоимость работы по разработке программного обеспечения.

Использую формулу, для того, чтобы рассчитать затраты на разработку данного программного обеспечения:

$$С = ФОТ + О_с + А + Э + С_{пр} + Н, \quad (4.1)$$

где ФОТ – фонд оплаты труда, тг;

$О_с$ – социальный налог, тг;

А – амортизационные отчисления, тг;

Э – затраты на электроэнергию, тг;

$С_{пр}$ – прочие расходы, тг;

Н – накладные расходы, тг.

4.3.5 Расчет затрат на оплату труда

Каждому разработчику, который принимает участие в разработке данного программного обеспечения необходимо осуществить оплату труда.

Для того, чтобы рассчитать затраты на оплату труда задействованных разработчиков, будем использовать формулу:

$$ФОТ = З_{осн} + З_{доп}, \quad (4.2)$$

где $З_{осн}$ – основная заработная плата, тг;

$З_{доп}$ – дополнительная заработная плата, тг.

Оплата труда программиста, дизайнера и разработчика базы данных можно считать условной, на договорной основе в размере 90000, 70000 и 70000 тенге.

Так как на различных этапах разработки программного обеспечения разработчики задействованы неравномерно, нам

необходимо рассчитать средний дневной заработок, а затем общий размер заработной платы в зависимости от фактического участия разработчиков.

Средний дневной заработок каждого разработчика рассчитывается по формуле:

$$D = \frac{ЗПм}{Др},$$

(4.3)

где $ЗП_м$ – ежемесячный размер заработной платы, тг;

$Д_р$ – количество рабочих дней в месяце (это 26 дней – шестидневная рабочая неделя).

Рассчитаю по формуле (4.3) дневной заработок программиста:

$$D = \frac{90000}{26} = 3461.5 \text{ тенге/день}$$

Заработная плата за один час вычисляется по формуле:

$$H = \frac{ЗПм}{Др \cdot Чр},$$

(4.4)

где $ЗП_м$ – ежемесячный размер заработной платы, тг;

$Д_р$ – количество рабочих дней в месяце, дней.

$Ч_р$ – количество часов рабочего дня ($Ч_р$ = 8), ч.

Рассчитаю заработную плату за один час по формуле (4.4):

$$H = \frac{90000}{26 \cdot 8} = 432.7 \text{ тенге/час}$$

Теперь необходимо определить длительность цикла в днях по каждому виду работ. Рассчитаем это значение по данной формуле:

$$t_n = \frac{T}{q_n \cdot z \cdot K},$$

(4.5)

где Т – трудоемкость этапа, норма–час;

q_n – количество исполнителей по этапу;

z – Продолжительность рабочего дня, z = 8 часов;

К – Коэффициент выполнения норм времени, К = 1.1.

Округлю полученную величину в большую сторону до целых дней. Для программиста и постановки задачи вычислю по формуле (4.5):

$$t_n = \frac{24}{1 \cdot 8 \cdot 1.1} \approx 3 \text{ дня}$$

Общее количество дней составляет:

$$t = 3+2+11+2+3+14+2+3+3 = 43 \text{ дня} \qquad (4.6)$$

Таким образом, для проведения всех работ необходимо 43 дня. Сводные данные по расчету заработной платы персонала, задействованного в разработке проекта приведены в таблице 4.5.

Таблица 4.5 – Сводные результаты расчета затрат на заработную плату

Наименование этапов	Исполнитель	Трудоемкость, норма–час	Длительность цикла, дни	Заработная плата за час работы, тенге	Сумма заработной платы, тенге
1	2	3	4	5	6
Постановка задачи	Программист	24	3	432.7	10384.8
Изучение и сбор информации для создания базы данных	Разработчик базы данных	14	2	336.5	5384.6

Проектирование и создание базы данных для ломбарда	Разработчик базы данных	96	11	336.5	29615.3
Изучение и сбор информации для проектирования и создания программного обеспечения	Программист	14	2	432.7	6923

Продолжение таблицы 4.5

1	2	3	4	5	6
Разработка дизайна и интерфейса программного обеспечения	Дизайнер	24	3	336.5	8076.9
Разработка и написание программного кода	Программист	120	14	432.7	48461
Отладка	Программист	14	2	432.7	6923
Тестирова	Программи	24	3	432.7	10384.8

ние программного обеспечения	ст				
Оформление отчета	Программист	24	3	432.7	10384.8
Итого:		354	43	3605.7	136538.4

Дополнительная заработная плата составляет 10% от основной заработной платы и вычисляется по формуле:

$$З_{доп} = З_{осн} \cdot 0.1, \quad (4.7)$$

Подставив в формулу (4.7) получу:

$$З_{доп} = 136538.4 \cdot 0.1 = 13653.84 \text{ тенге}$$

Таким образом, затраты на оплату труда согласно произведенным расчетам и в соответствии с формулой (4.2) составит:

$$ФОТ = 136538.4 + 13653.84 = 150192.24 \text{ тенге}$$

4.3.6 Расчет затрат по социальному налогу

Каждый социальный налог составляет 11% от дохода работника согласно (ст. 358 п. 1 НК РК) и рассчитывается по формуле:

$$О_с = (ФОТ – ПО) \cdot 11\%, \quad (4.8)$$

где ПО – пенсионные отчисления, тг.

Пенсионные отчисления составляют 10% от ФОТ и социальным налогом не облагаются, вычисляются отчисления по формуле:

$$ПО = ФОТ \cdot 10\% \quad (4.9)$$

Подставлю значения в формулу (4.9):

$$ПО = 150192.24 \cdot 0.1 = 15019.224 \text{ тенге}$$

Таким образом, в соответствии с произведенными расчетами и согласно формуле (4.8) размер отчислений на социальные нужды составит:

$$O_c = (150192.24 – 15019.224) \cdot 0.11 = 14869.1 \text{ тенге}$$

4.3.7 Расчет амортизационных отчислений

Амортизационные отчисления рассчитываются по формуле

$$A_i = \frac{H_A \cdot C_{пер} \cdot N}{100 \cdot n}, \quad (4.10)$$

где H_A – норма амортизации, тг;

$C_{ПЕР}$ – первоначальная стоимость оборудования, тг;

N – Количество дней на выполнение работ, дней;

n – Количество рабочих дней в году, дней.

Норма амортизации на компьютерную технику составляет 40% от стоимости всего оборудования, на программное обеспечение – 15%. Таким образом, амортизационный отчисления по используемому оборудованию, в соответствии с формулой (4.10) составят:

$$A_i = \frac{40 \cdot 110000 \cdot 43}{100 \cdot 365} = 5183.56 \text{ тенге}$$

Таким образом, суммарные затраты на амортизацию составляют:

$$A_i = 5183.56 + 1413.7 + 795.21 = 7392.47 \text{ тенге} \quad (4.11)$$

Сводные результаты расчета амортизационных отчислений представлены в таблице 4.6.

Таблица 4.6 – Сводные данные по расчету затрат на амортизацию

Наименование оборудования	Количество	Норма амортизации, %	Сумма амортизации, тенге
Ноутбук	1	40	5183.56
Принтер	1	40	1413.7
Программное обеспечение	3	15	795.21
Итого:			7392.47

4.3.8 Расчет затрат на электроэнергию

В процессе разработки программного обеспечения и базы данных необходимо использовать электрооборудование. Для того, чтобы узнать затраты на электроэнергию существует формула (4.12). Затраты на электроэнергию для производственных нужд включают в себя расходы электроэнергии на оборудование и дополнительные нужды. И рассчитываются по формуле:

$$Э = З_{эл.эн.об.} + З_{доп.}, \quad (4.12)$$

где $З_{эл.эн.об.}$ – затраты на электроэнергию для оборудования, тг;

$З_{доп.}$ – затраты на электроэнергию для дополнительных нужд, тг.

Расходы электроэнергии для оборудования рассчитываются по формуле:

$$З_{эл.эн.об.} = W \cdot T \cdot S \cdot К_{исп}, \quad (4.13)$$

где W – потребляемая мощность, Вт;

T – время работы, часы;

S – тариф (1кВт = 16.02 тенге);

$К_{исп}$ – коэффициент использования ($К_{исп}$ = 0,9).

Для ноутбука расходы составят по формуле (4.13):

$$З_{эл.эн.об.(ноутбук)} = 0.7 \cdot 354 \cdot 16.02 \cdot 0.9 \cdot 2 = 7145.6 \text{ тенге}$$

Общая сумма затрат на электроэнергию основного оборудования согласно формуле (4.12) составляет:

$$З_{эл.эн.об.} = 7146.6 + 415.24 = 7561.84 \text{ тенге}$$

Результаты расчета затрат на электроэнергию показаны в таблице 4.7.

Таблица 4.7 – Сводные данные о затратах на электроэнергию

Наименование приборов	Количество	Потребляемая мощность, Вт	Число рабочих дней	Коэффициент использования	Время работы оборудования, часы	Сумма затрат, тенге
Ноутбук	2	0.7	60	0.9	354	7145.6
Принтер	1	0.9	4	0.9	32	415.24
Итого:						7561.84

4.3.9 Расчет накладных и прочих расходов

Прочие расходы включают в себя:

1) расходы на Интернет на 60 дней (2месяца). Стоимость интернета за один месяц составляет 4700 тенге. Получаем:

$$Расх = 4700 \cdot 2 = 9400 \text{ тенге} \quad (4.14)$$

2) расходы на канцелярские товары:
- бумага формата А4 – 935 тенге;
- упаковка ручек – 250 тенге;
- 2 карандаша – 40 тенге;
- стиральная резинка – 45 тенге;
- линейка – 60 тенге;
- стикеры – 250 тенге.

Общая стоимость канцелярских товаров:

$$Расх_{общ} = 935 + 250 + 40 + 45 + 60 + 250 = 1580 \text{ тенге} \quad (4.15)$$

Прочие расходы составляют:

$$Спр = 9400 + 1580 = 10980 \text{ тенге} \quad (4.16)$$

Накладные расходы составляют 30% от всех затрат и рассчитываются по формуле:

$$Н = (ФОТ + О_с + А + Э + С_{ПР}) \cdot 0.3 \quad (4.17)$$

Подставлю значения в формулу (4.17):

Н = (150192.24 + 14869.1 + 7392.47 + 7561.84 + 10980) · 0.3 = 57298.7 тенге

4.3.10 Расчет стоимости по всем статьям затрат и определение структуры затрат

В соответствии с формулой (4.1) суммарные затраты по разработке моего проекта составляют:

С = 150192.24 + 14869.1 + 7392.47 + 7561.84 + 10980 + 57298.7 =
= 248294.35 тенге

Смета затрат по разработке программного обеспечения и базы данных для ломбарда, а также структура расходов представлены в таблице 4.8 и на рисунке 4.1.

Таблица 4.8 – Суммарные данные по стоимости разработки проекта

Наименование статьи затрат	Сумма, тенге
Фонд оплаты труда	150192.24
Социальный налог	14869.1
Амортизация	7392.47
Затраты на электроэнергию	7561.84
Прочие расходы	10980
Накладные расходы	57298.7
Итого:	248294.35

Рисунок 4.1 – Структура затрат по разработке программного обеспечения и базы данных для ломбарда

4.3.11 Цена интеллектуального труда

Цена реализации проекта складывается из стоимости и чистого дохода:

$$Ц = С + П, \tag{4.18}$$

где С – стоимость продукта, тг;

П – чистый доход, тг.

При определении первоначальной цены следует задаться желаемым уровнем рентабельности (здесь 20%) реализации программных продуктов:

$$Ц_П = С \cdot (1 + \frac{Р}{100}), \tag{4.19}$$

где Р – рентабельность, равная 20%.

Поставив в формулу (4.19) получу:

$$Ц_П = 248294.35 \cdot (1 + \frac{20}{100}) = 297953.22 \text{ тенге}$$

Цена реализации проекта рассчитывается по формуле:

$$Ц_Р = Ц_П + Ц_П \cdot НДС, \quad (4.20)$$

где НДС – налог на добавленную стоимость по ставке 12%.

В соответствии с формулой (4.20) цена реализации проекта составит:

$$Ц_Р = 297953.22 + 297953.22 \cdot 0.12 = 333707.61 \text{ тенге}$$

Окончательная цена на интеллектуальный труд будет варьироваться, и изменяться в зависимости от требований нужд.

Вывод

Цена услуги реализации данного проекта на рынке составила 248294.35 тенге. Разработка ПО и базы данных для ломбардов является трудом интеллектуальным, основная долю затрат составляют затраты на оплату труда 150192.24 тенге (60%). Цена реализации продукта равна 333707.61 тенге. Программный продукт «Разработка аннуитетно кредитной, информационной, административной системы для ломбардов» относится к дорогим продуктам.

Данная программа позволяет автоматизировать процесс учета товаров, находящихся в заем у ломбарда, уменьшая при этом вероятность потери данных, т.к. данные не хранятся в не в электронном виде. Также программа позволяет хранить множество копий данных на разных носителях, что обеспечивает высокую сохранность. По сравнению с архивами, хранящимися в бумажном виде, программа обеспечивает более гибкую работу с данными, что уменьшает количество бумажной работы, при этом уменьшаются затраты на покупку бумаги и тонера принтера. Благодаря снижению количества использованной бумаги и тонера благоприятно влияет на окружающую среду, т. к. тонер тяжело утилизируется и негативно влияет на окружающую среду.

5 Безопасность жизнедеятельности

5.1 Анализ условий труда обслуживающего

Целью данного дипломного проекта является написание программного продукта и базы данных для ломбардов, а также предусмотреть безопасность работы на ПК.

В список опасных и вредных факторов при работе за компьютером входят:

1) высокая температура поверхностей ПК;
2) токсичные вещества;
3) высокая или низкая влажность воздуха;
4) недостаток естественного света;
5) невысокая искусственная освещенность рабочей поверхности;
6) повышенная яркость света;
7) высокий уровень шума на рабочем месте;
8) пониженная контрастность;
9) высокая напряженность магнитного поля;
10) нервно–эмоциональные перегрузки;
11) монотонность трудового процесса;
12) высокий уровень статического электричества.

Что касается здоровья сотрудников, то выделю несколько рисков, которым сопровождается влияние на организм человека компьютера, обуславливается наличием электромагнитного излучения, вызывающего:

1) проблемы зрения;
2) проблемы с мышцами и суставами;
3) стресс;
4) депрессия и нервные расстройства, обуславливающие влиянием компьютера на психику человека, малоподвижный образ жизни;
5) часы сверхурочные (более 9 часов в сутки);
6) работа в ночное время, вследствие чего нарушается выработка гормона мелатонина.

Физически вредные и опасные факторы. К таким факторам можно отнести: повышенный уровень электромагнитного, рентгеновского, ультрафиолетового и инфракрасного излучения; повышенный уровень статического электричества и запыленности воздуха рабочей зоны; повышенное содержание положительных

аэронов и пониженное содержание отрицательных аэроионов в воздухе рабочей зоны; повышенный уровень блескости и облупленности; неравномерность распределения яркости в поле зрения; повышенная яркость светового изображения; высокое напряжение в сети, при похождении которого сквозь человека причинит ранения.

Химически вредные и опасные факторы. К ним относятся следующее: высокая концентрация в воздухе аммиака, двузонно, окиси углерода, формальдегида м фенола.

Психофизические вредные и опасные факторы. Психофизиологические вредные и опасные факторы: напряжение зрения и внимания; интеллектуальные, эмоциональные и длительная постоянная нагрузка; монотонность выполняемой работы; огромный объем информации, обрабатываемый человеком; плохая организация рабочего места.

Обычными ощущениями, испытываемые к концу рабочего дня операторы ПЭВМ, являются: переутомление глаз, головная боль, боль в шеи, руках и спине, низкая концентрация внимания.

Магнитное поле. Работающий компьютер создаёт магнитное поле вокруг себя, обладающее способностью воздействовать на наш организм. Из-за влияния его на наши клетки и ткани, происходит нарушение деятельности организма и снижается активность мозга. Это проявляется в качестве боли в голове, высокой утомляемости, ухудшается самочувствие. Электромагнитное поле оказывает тепловое воздействие на организм, вследствие чего повышается температура, нагреваются ткани и органы. Больше всего подвержены воздействию печень, поджелудочная железа, мочевой пузырь, желудок. Впоследствии это может стать причиной язвы, внутреннего кровотечения и перфорации.

Шум. На рабочем месте сотрудников источниками являются разговаривающие люди, внешний шум и отчасти – компьютер, принтер. Они издают небольшой шум, поэтому в помещении необходимо использовать звукопоглощение, согласно (СНиП–II–12–77 «Защита от шума») допустимый уровень шума при работе компьютера не должен превышать 50 дБ. Для выбранного помещения выбрано звукопоглощающие облицовка, состоящая из матов, из супертонкого стекловолокна с оболочкой из стеклоткани, которую нужно разместить на потолке и верхних частях стен.

Максимальное звукопоглощение будет достигнуто при облицовке не менее 60% общей площади ограждающих поверхностей помещения.

Электростатическое поле, вредные вещества в воздухе. Компьютер во время своей работы образует вокруг себя электростатическое поле, демонизирующее окружающую среду. В воздух испускаются вредные вещества во время нагревания при нагревании платы. Из-за этого воздух в помещении становится сухим, плохо ионизированным, со специфическим запахом и в общем "тяжёлым" для дыхания. Следовательно, такой воздух вреден для организма и приведёт к аллергическим заболеваниям, сердечным болезням и дыхательные органов.

Технический персонал состоит из четырех сотрудников: главный технический специалист, менеджер, бухгалтер, администратор работающие непосредственно на ПК. Технический специалист работает 3 раза в неделю по 6 часов. Менеджер и администратор работают каждый день по 6 часов с ПК производя работу на БД Ломбард. Бухгалтер работает 3 раза в неделю по 3 часа.

Работа сотрудников связана компьютерами, соответственно с вредным дополнительным воздействием целой группы факторов, что существенно снижает производительность их труда. К таким факторам отнесу:

1) неправильную освещенность;
2) эргономические нарушения к требованиям рабочего места;
3) наличие напряженности.

Согласно ГОСТ 12.1.005–88 ССБТ «Оптимальные и допустимые нормы микроклимата, в зависимости от категории работ», производимая работа в помещение относится к работе лёгкой тяжести (1а).

В помещениях с ПК необходимо строго соблюдать следующие условия микроклимата. В холодный период года:

1) температура в помещении:
   - оптимальная температура 24 °С, необходимая температура 26 °С;
   - требуемая влажность 55%, допустимая влажность 70%;
   - допустимая скорость воздуха 0.1 м/с;
2) температура в серверной:

– оптимальная температура 15 °С , необходимая температура 20 °С;
– относительная влажность 50%, допустимая влажность 65%;
– допустимая скорость воздуха 0.1 м/с;

Тёплый период года:

1) Температура в помещении:
– оптимальная температура 24 °С, необходимая температура 25 °С;
– относительная влажность 53%, допустимая влажность 65%;
– допустимая скорость воздуха 0.1 м/с;

2) Температура в серверной:
– оптимальная температура 15 °С, необходимая температура 20 °С;
– относительная влажность 40%, допустимая влажность 55%;
– допустимая скорость воздуха 0.05 м/с;

Помещение имеет 1 комнату. Размер помещения, где работает менеджер и бухгалтер: длина 4 м, ширина 5 м, высота потолка 3 м. Имеется окно, двойное, размеры 2х2 м. Искусственное освещение – два светодиодных светильника с двумя лампами каждый. Желательное количество рабочих мест – 2. Выделен стол размером 2х1.5 м каждому работнику и офисное кресло. Помещение располагается в здание на 1–м этаже.

План помещения выбранного для размещения оборудования и технического персонала изображен на рисунке 5.1.

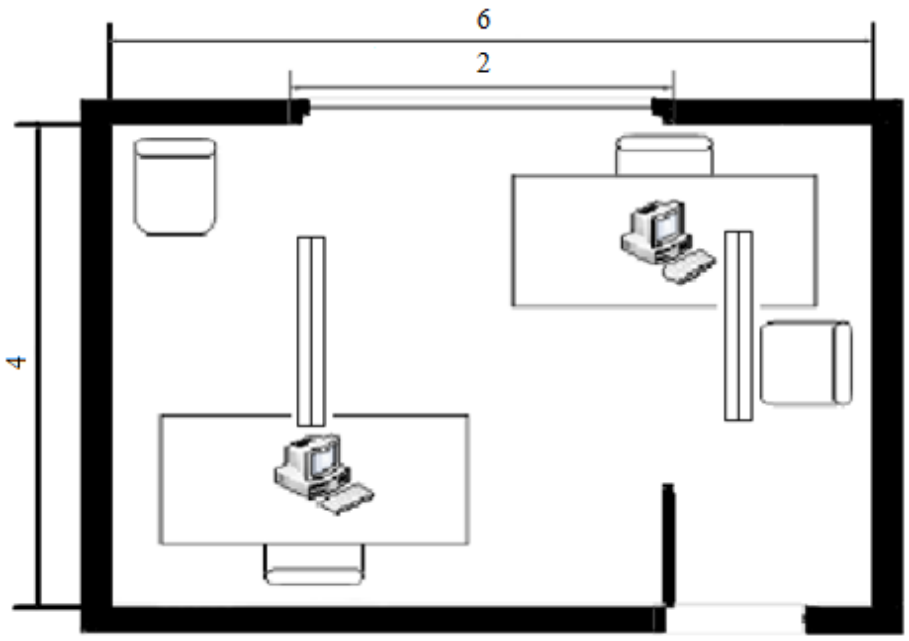


Рисунок 5.1 – План рабочего места

Проанализировав условия труда, рассмотрим более подробно следующие моменты.

5.2 Расчетная часть.

5.2.1 Расчет системы кондиционирования.

Кондиционирование улучшает микроклимат в помещении и должно выполняться согласно нормам [2]. Норма пространства помещения на человека требуется 6 м^2. В офисе задумывается присутствие воздуховодов, которые располагаются за подвесными потолками. Воздух неспешно подается и выводится из офиса с использованием вентиляционных отверстий в стенах или через особые воздуховоды, установленные на потолке. Кондиционер также установлен, который осуществляет освежение воздуха. Для расчета примем следующие размеры рабочего помещения

– длина a = 6 м;
– ширина b = 4 м;
– высота h = 3 м;
– окно = 2x2 м.

Соответственно объем помещения равен:

$$V_{пом} = a \cdot b \cdot h = 72 \text{ м}^3 \quad (5.1)$$

Необходимый для обмена объем воздуха $V_{вент}$ определим исходя из уравнения теплового баланса.

$$V_{вент} \cdot C\ (t_{уход} - t_{приход}) \cdot Y = Q \cdot 3600, \qquad (5.2)$$

где $V_{вент}$ – объем воздуха, необходимый для обмена, $м^3$;

$Q_{избыт}$ – избыточная теплота, Вт;

C – удельная теплопроводность воздуха, C=1000 Дж/кг · К;

Y – плотность воздуха, Y=1.2 мг/$см^3$;

$t_{приход}$ – 20 °C.

Температура уходящего воздуха определяется по формуле:

$$t_{уход} = t_{р.м.} + (h - 2) \cdot t, \qquad (5.3)$$

где t = 1–5 °C – превышение t на 1м высоты помещения, °C;

$t_{р.м.}$ = 23 °C – температура на рабочем месте, °C;

h = 3 м – высота помещения, м.

Температура уходящего воздуха согласно формуле (5.3) равна:

$$t_{уход} = 23 + (\ 4 - 2\ ) \cdot 3 = 29\ ^oC$$

Найду избыток тепла от оборудования:

$$Q_{избыт} = Q_{изб.1} + Q_{изб.2} + Q_{изб.3}, \qquad (5.4)$$

где $Q_{изб.}$ – избыток тепла от электрооборудования и освещения, Дж.

Избыток тепла рассчитывается по формуле:

$$Q_{изб.1} = E \cdot P, \qquad (5.5)$$

где E – коэффициент потерь электроэнергии на теплоотвод, E=0.55;

P – мощность, Вт.

Т.к. в помещении 5 ламп по 60 Вт, следовательно, мощность их равна:

$$P = 60\ Вт \cdot 5 = 300\ Вт \qquad (5.6)$$

Таким образом, избыток тепла от освещения равно согласно формуле (5.5):

$$Q_{изб.1} = 0.55 \cdot 300 = 165 \text{ Вт}$$

Найду теплопоступление от солнечной радиации:

$$Q_{изб.2} = m \cdot S \cdot k \cdot Q_c, \quad (5.7)$$

где m – число окон;

S – площадь окна, $м^2$;

K – коэффициент, учитывающий остекление. Для двойного остекления k = 0.6;

$Q_c = 200$ $Вт/м^2$ – теплопоступление от окон.

Подставив значения в формулу (5.7) получу:

$$Q_{изб.2} = 4 \cdot 1 \cdot 0.6 \cdot 200 \text{ Вт}/м^2 = 480 \text{ Вт}$$

Найду тепловыделение людей:

$$Q_{изб.3} = n \cdot q\ , \quad (5.8)$$

где q = 80 Вт/чел;

n – число людей, например, n = 2.

Подставлю значения в формулу (5.8):

$$Q_{изб.3} = 2 \cdot 80 = 160 \text{ Вт}$$

Рассчитаю общее тепловыделение по формуле (5.4):

$$Q_{избыт} = 165 + 480 + 160 = 805 \text{ Вт}$$

Из уравнения теплового баланса, формула (5.2), следует:

$$V_{вент} = \frac{3600 \cdot 805}{1000 \cdot 1.2 \cdot (29 - 20)} = 268.3 \text{ } м^3 = 268 \text{ } м^3$$

Местоположение кондиционера в офисном помещении надо продумать тщательно. На потолок подвесной можно произвести

монтаж кондиционера и через воздуховоды направлять воздух. Благодаря чему температура и воздух будут распределены равномерно. Снижение затрат на оборудования можно достичь благодаря объединением с вентиляционной системой. Друг от друга блоки располагаются на интервале до 20 м.

В моём случае установка канального кондиционера не возможно, поэтому выберу монтаж сплит–системы. Сплит–системы – это комплекс устройств, состоящая из агрегатов двух типов – внутреннего и наружного блока. Монтаж первого блока производится в самом помещении, а вне помещения производится инсталляция наружного блока. Благодаря сплит–системе можно добиться невысокого шума, сплит-система является компактной и имеет большой функционал. На рисунке 5.2 показано расположение сплит–системы.

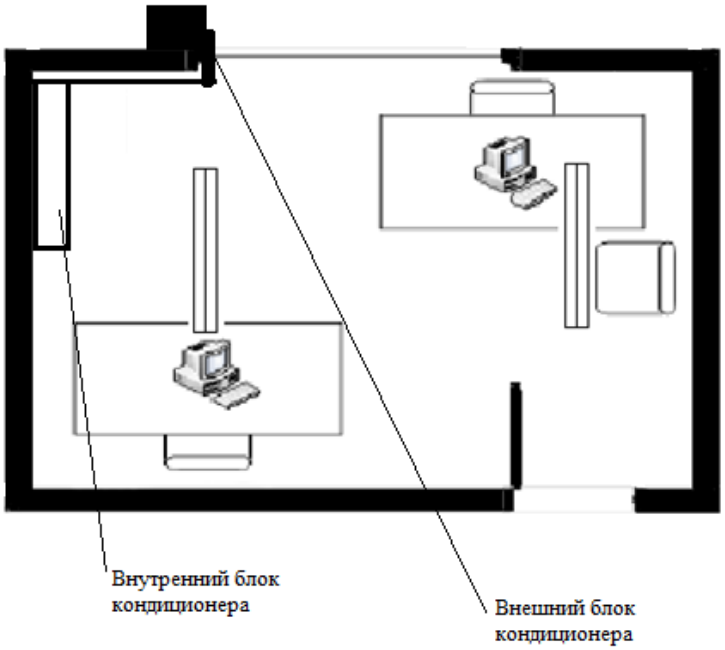


Рисунок 5.2 – Расположение кондиционера в помещении

5.2.2 Расчет искусственного освещения

Определю расчетную высоту подвеса:

$$h_{расч} = H - (h_{пов} + h_{свес}) = 4 - (0.8 + 0.1) = 3.1 \text{ м} \quad (5.9)$$

Найдем расстояние между светильниками:

$$L_A = \lambda \cdot h_p \quad (5.10)$$

Расстояние между светильниками по формуле (5.10):

$$L_A = 0.916 \cdot 2.15 = 1.969 \text{ м}$$

Для определения количества светильников определим световой поток, падающий на поверхность по формуле:

$$F = \frac{E \cdot K \cdot S \cdot Z}{\mu}, \quad (5.11)$$

где F – рассчитываемый световой поток, лм;

E – нормированная минимальная мощность (E=300 лк), лк;

S – площадь, освещаемого помещения (S=24 м^2), м^2;

Z – отношение средней освещенности к минимальной (Z=1.1);

К – коэффициент запаса (К=1.2);

μ – коэффициент использования, зависит от характеристик светильника, размеров помещения, значения коэффициентов P_c, $P_п$ (P_c=30%, $P_п$=50%).

Значение определим по таблице коэффициентов использования светового потока. Для этого вычислим индекс помещения:

$$i = \frac{S}{h_p \cdot (a + b)} = \frac{24}{3.1 \cdot (6 + 4)} = 0.77 \quad (5.12)$$

Зная индекс помещения, определим коэффициент использования светового потока: μ = 45%.

Подставим все значения в формулу (5.11) для определения светового потока F:

$$F = \frac{300 \cdot 1.2 \cdot 24 \cdot 1.1}{0.45} = 21120 \text{ лм}$$

Для освещения выбираем светодиодные лампы типа Е27, световой поток которых F = 2500 лм.

Рассчитаем необходимое количество ламп по формуле:

$$N = \frac{F}{F_Л}, \quad (5.13)$$

где N – определяемое количество ламп;

F – световой поток (F= 21120 лм), лм;

$F_Л$ – световой поток лампы ($F_Л$=2500 лм), лм.

Подставлю значения в формулу (5.13):

$$N = \frac{21120}{2500} = 9$$

Для обеспечения необходимой освещенности помещения с параметрами 6х4х4 необходимо установить 9 штук светодиодных ламп типа Е27.

Заключение

За время создания программного обеспечения и базы данных, а также написания дипломной работы была досконально изучена предметная область проекта.

Реализована задача автоматизации составления и просмотра данных ломбарда. Была разработана концептуальная модель базы данных. Для разработанной базы данных в ходе выполнения проекта было создано клиентское приложение, которое обеспечивает отображение наиболее существенных пользовательских запросов. Данные программы имеют удобный интерфейс, который позволяет пользователю получить нужную ему информацию. Также были решены способы работы с отчетами и документами для сотрудников ломбарда.

В ходе разработки программы были смоделированы UML диаграммы, для более удобного и понятно объяснения работы программы.

База данных содержит триггеры, пакеты, функции и хранимые процедуры, которые описаны выше, это позволяет ускорить процесс обработки информации.

Данная программа позволяет автоматизировать процесс учета товаров, находящихся в займе у ломбарда, уменьшая при этом вероятность потери данных, так как данные не хранятся в бумажном виде. Также программа позволяет хранить несколько копий данных на разных носителях, что обеспечивает высокую сохранность данных. По сравнению с архивами, хранящимися в бумажном виде, программа обеспечивает более гибкую работу с данными, что уменьшает количество бумажной работы, при этом уменьшаются затраты на закупку бумаги и тонера для принтера. Благодаря снижению количества использованной бумаги и тонера благоприятно влияет на окружающую среду, так как тонер тяжело утилизируется и негативно влияет на окружающую среду.

В среднем цена услуг на реализацию данного проекта на рынке составила 248294.35 тенге. Разработка ПО и базы данных для ломбардов является интеллектуальным трудом, основную долю в общей сумме затрат составляют затраты на оплату труда 150192.24 тенге (60%). Цена реализации продукта составляет 333707.61 тенге.

Был произведен анализ условий труда в офисе, который соответствует ГОСТ–ам, СанПиН–ам и СНиП–ам. Также, были произведены расчеты системы кондиционирования, искусственного освещения.

Список литературы

1) http://www.unesco.kz/ci/projects/omrc/pechbillalmatr/pubister10.htm –Экономическая ситуация в Казахстане и перспективы развития. М. Торгаев.

2) https://msdn.microsoft.com/ru–ru/library/z1zx9t92.aspx – введение в язык C# и .NET Framework.

3) https://www.microsoft.com/ru–ru/server–cloud/products/sql–server/overview.aspx – About Sql Server 2012.

4) https://ru.wikipedia.org/wiki/Microsoft_Visual_Studio – Статья о Visual Studio.

5) Г.Шилдт. Самоучитель C#. – СПб.: БХВ–Петербург, 2007.– 688 с.

6) CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4–е изд. Рихтер Д.

7) Язык программирования C#. Классика Computers Science. 4–е изд. Хейлсберг А., Торгерсен М., Вилтамут С., Голд П.

8) Хансен Г., Хансен Д. Базы данных: разработка и управление. – М.: ЗАО «Издательство БИНОМ», 1999.

9) Питер Роб, Карлос Коронел Системы баз данных: проектирование, реализация и управление, 5–е издание, – BHV Санкт–Петербург, 2004. – 1040 с.

10) Кренке Д. Теория и практика построения баз данных. Изд.9 – Питер, 2005.

11) MICROSOFT SQL SERVER 2005. Реализация и обслуживание. Учебный курс Microsoft/ Пер. с английского – М. «Русская редакция», Спб.: «Питер», 2007. – 768 стр.

12) Мамаев Е. MS SQL Server 2000. Проектирование и реализация баз данных. Сертификационный экзамен. – BHV, СПб. 2004г, 416 с.

13) Бекешева А.И. Методические указания к выполнению экономической части дипломной работы для бакалавров – Алматы: АУЭС, 2013 – 24 с.

14) Плю Р., Стефенс Р., Райан К. Освой самостоятельно SQL за 24 часа. – М.: Издательский дом «Вильямс», 2000.

15) СНиП РК 2.04–05–2002. Естественное и искусственное освещение. Общие требования. – Алматы, 2002.

16) Сивков В.П., Смирнов С.Г., Козьяков А.Ф. и др. Сборник типовых расчетов по курсу «Охрана труда». – М.: МВТУ, 1979. – 79с.
17) ГОСТ 12.1.005–88 ОБЩИЕ САНИТАРНО–ГИГИЕНИЧЕСКИЕ ТРЕБОВАНИЯ К ВОЗДУХУ РАБОЧЕЙ ЗОНЫ.
18) ГОСТ 30494–96 ПАРАМЕТРЫ МИКРОКЛИМАТА В ПОМЕЩЕНИЯХ.

Приложение А

Листинг программы

```
using asd;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Runtime.Serialization;
using System.Runtime.Serialization.Formatters.Binary;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.OleDb;
using MySql.Data.SqlClient;
using System.Data.Sql;
using System.Data.SqlClient;

namespace asd
{

    class Class1
    {
        DataGridView datagridView4;//список сотрудников
        DataGridView datagridView1;//список клиентов
        DataGridView dataGridView2;//список договоров
        DataGridView datagridView3;//список заложенных вещей
        DataGridView datagridView5;//авто
        DataGridView datagridView6;//искусство
        DataGridView datagridView7;//кухонные
        DataGridView datagridView8;//техника
        DataGridView datagridView9;//ювелирные
        DataGridView datagridView15;//зп

        //для сотрудников
        SqlConnection conn = new SqlConnection();
```

```
SqlDataAdapter adapter = new SqlDataAdapter();
SqlCommand sc = new SqlCommand();
DataSet d = new DataSet();
BindingSource b = new BindingSource();
```

```
        //для клиентов
        SqlConnection conn1;
        SqlDataAdapter adapter1;
        DataSet d1 = new DataSet();
        BindingSource b1 = new BindingSource();

        //для договоров
        SqlConnection conn2;
        SqlDataAdapter adapter2;
        DataSet d2 = new DataSet();
        BindingSource b2 = new BindingSource();

        //для вещей
        SqlConnection conn3;
        SqlDataAdapter adapter3;
        DataSet d3 = new DataSet();
        BindingSource b3 = new BindingSource();

        //для зп
        SqlConnection conn9;
        SqlDataAdapter adapter9;
        DataSet d9 = new DataSet();
        BindingSource b9 = new BindingSource();

        //для сотрудников
        public void read_MySql_employee(ref DataGridView dgv)
        {
           datagridView4 = dgv;
           try
           {
              dgv.Columns.Clear();
              conn = new SqlConnection("DATA SOURCE = localhost;
Initial catalog = lombard; Integrated Security = True;");
              conn.Open();
              adapter = new SqlDataAdapter("select · from employee",
conn);
```

```
            adapter.Fill(d, "employee");
            conn.Close();
            b.DataSource = d.Tables["employee"];
            dgv.DataSource = b;
```

Продолжение приложения А

```
 }          catch (Exception exp) { MessageBox.Show(exp.ToString()); }
      }
      public void save_MySql_employee(ref DataGridView dgv)
      {
         datagridView4 = dgv;
         try
         {
            SqlDataAdapter da = new SqlDataAdapter();

            da.SelectCommand = new SqlCommand("SELECT
surname,num_employee,name,patronymic,dob,phone,adress,IIN,num_id
entity_card,issued,date_of_issue,id_position,login,pass FROM
`lombard`.`employee`", conn);
            da.UpdateCommand = new SqlCommand("UPDATE
employee SET surname=@surname, name=@name,
patronymic=@patronymic, dob=@dob, phone=@phone,
adress=@adress, IIN=@IIN, num_identity_card=@num_identity_card,
issued=@issued, date_of_issue=@date_of_issue,
id_position=@id_position, login=@login, pass=md5(@pass)  WHERE
num_employee=@num_employee");
            da.UpdateCommand.Parameters.Add("@surname",
SqlDbType.VarChar, 20, "surname");
            da.UpdateCommand.Parameters.Add("@num_employee",
SqlDbType.Int, 10, "num_employee");
            da.UpdateCommand.Parameters.Add("@name",
SqlDbType.VarChar, 20, "name");
            da.UpdateCommand.Parameters.Add("@patronymic",
SqlDbType.VarChar, 20, "patronymic");
            da.UpdateCommand.Parameters.Add("@dob",
SqlDbType.Date, 10, "dob");
```

```
            da.UpdateCommand.Parameters.Add("@phone",
SqlDbType.BigInt, 100, "phone");
            da.UpdateCommand.Parameters.Add("@adress",
SqlDbType.VarChar, 20, "adress");
            da.UpdateCommand.Parameters.Add("@IIN",
SqlDbType.BigInt, 100, "IIN");
            da.UpdateCommand.Parameters.Add("@num_identity_card",
SqlDbType.Int, 10, "num_identity_card");
            da.UpdateCommand.Parameters.Add("@issued",
SqlDbType.VarChar, 20, "issued");
            da.UpdateCommand.Parameters.Add("@date_of_issue",
SqlDbType.Date, 10, "date_of_issue");
```

Продолжение приложения А

```
da.UpdateCommand.Parameters.Add("@id_position", SqlDbType.Int,
10, "id_position");
            da.UpdateCommand.Parameters.Add("@login",
SqlDbType.VarChar, 20, "login");
            da.UpdateCommand.Parameters.Add("@pass",
SqlDbType.VarChar, 255, "pass");
            da.UpdateCommand.Connection = conn;

            da.InsertCommand = new SqlCommand("INSERT INTO
employee(surname,num_employee,name,patronymic,dob,phone,adress,I
IN,num_identity_card,issued,date_of_issue,id_position,login,pass)" +
                "VALUES
(@surname,@num_employee,@name,@patronymic,@dob,@phone,@a
dress,@IIN,@num_identity_card,@issued,@date_of_issue,@id_positio
n,@login,HASHBYTES( 'md5', @pass ))", conn);
            da.InsertCommand.Parameters.Add("@surname",
SqlDbType.VarChar, 20, "surname");
            da.InsertCommand.Parameters.Add("@num_employee",
SqlDbType.Int, 10, "num_employee");
            da.InsertCommand.Parameters.Add("@name",
SqlDbType.VarChar, 20, "name");
            da.InsertCommand.Parameters.Add("@patronymic",
SqlDbType.VarChar, 20, "patronymic");
```

```
            da.InsertCommand.Parameters.Add("@dob",
SqlDbType.Date, 10, "dob");
            da.InsertCommand.Parameters.Add("@phone",
SqlDbType.BigInt, 100, "phone");
            da.InsertCommand.Parameters.Add("@adress",
SqlDbType.VarChar, 20, "adress");
            da.InsertCommand.Parameters.Add("@IIN",
SqlDbType.BigInt, 100, "IIN");
            da.InsertCommand.Parameters.Add("@num_identity_card",
SqlDbType.Int, 10, "num_identity_card");
            da.InsertCommand.Parameters.Add("@issued",
SqlDbType.VarChar, 20, "issued");
            da.InsertCommand.Parameters.Add("@date_of_issue",
SqlDbType.Date, 10, "date_of_issue");
            da.InsertCommand.Parameters.Add("@id_position",
SqlDbType.Int, 10, "id_position");
            da.InsertCommand.Parameters.Add("@login",
SqlDbType.VarChar, 20, "login");
```

Продолжение приложения А

```
  da.InsertCommand.Parameters.Add("@pass", SqlDbType.VarChar,
255, "pass");
            da.InsertCommand.Connection = conn;

            conn.Open();
            da.Update(d, "employee");
            conn.Close();

        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
        finally { MessageBox.Show("Saved"); }
    }

    //для Клиентов
    public void read_MySql_clients(ref DataGridView dgv)
    {
        datagridView1 = dgv;
        try
```

```
        {

            dgv.Columns.Clear();
            conn1 = new SqlConnection("DATA SOURCE = localhost;
Initial catalog = lombard; Integrated Security = True;");
            conn1.Open();
            adapter1 = new SqlDataAdapter("select · from clients",
conn1);
            adapter1.Fill(d1, "clients");
            conn1.Close();
            b1.DataSource = d1.Tables["clients"];
            dgv.DataSource = b1;

        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
    }
    public void save_MySql_clients(ref DataGridView dgv)
    {
        datagridView1 = dgv;
        try
        {
            SqlDataAdapter da1 = new SqlDataAdapter();

            da1.SelectCommand = new SqlCommand("SELECT
num_client, surname, name, patronymic, dob,phone, adress, IIN,
num_identity_card, issued, date_of_issue FROM `lombard`.`clients`",
conn1);
```

Продолжение приложения А

```
da1.UpdateCommand = new SqlCommand("UPDATE clients SET
surname=@surname, name=@name, patronymic=@patronymic,
dob=@dob,phone=@phone, adress=@adress, IIN=@IIN,
num_identity_card=@num_identity_card,issued=@issued,date_of_issue
=@date_of_issue WHERE num_client=@num_client");
            da1.UpdateCommand.Parameters.Add("@num_client",
SqlDbType.Int, 10, "num_client");
            da1.UpdateCommand.Parameters.Add("@surname",
SqlDbType.VarChar, 20, "surname");
```

```
            da1.UpdateCommand.Parameters.Add("@name",
SqlDbType.VarChar, 20, "name");
            da1.UpdateCommand.Parameters.Add("@patronymic",
SqlDbType.VarChar, 20, "patronymic");
            da1.UpdateCommand.Parameters.Add("@dob",
SqlDbType.Date, 10, "dob");
            da1.UpdateCommand.Parameters.Add("@phone",
SqlDbType.BigInt, 100, "phone");
            da1.UpdateCommand.Parameters.Add("@adress",
SqlDbType.VarChar, 20, "adress");
            da1.UpdateCommand.Parameters.Add("@IIN",
SqlDbType.BigInt, 100, "IIN");

da1.UpdateCommand.Parameters.Add("@num_identity_card",
SqlDbType.Int, 10, "num_identity_card");
            da1.UpdateCommand.Parameters.Add("@issued",
SqlDbType.VarChar, 20, "issued");
            da1.UpdateCommand.Parameters.Add("@date_of_issue",
SqlDbType.Date, 10, "date_of_issue");
            da1.UpdateCommand.Connection = conn1;

            da1.InsertCommand = new SqlCommand("INSERT INTO
clients(num_client, surname, name, patronymic, dob,phone, adress, IIN,
num_identity_card, issued, date_of_issue)" +
                " VALUES (@num_client, @surname, @name,
@patronymic, @dob, @phone, @adress, @IIN, @num_identity_card,
@issued, @date_of_issue)", conn1);
            da1.InsertCommand.Parameters.Add("@num_client",
SqlDbType.Int, 10, "num_client");
            da1.InsertCommand.Parameters.Add("@surname",
SqlDbType.VarChar, 20, "surname");
            da1.InsertCommand.Parameters.Add("@name",
SqlDbType.VarChar, 20, "name");
```

Продолжение приложения А

```
 da1.InsertCommand.Parameters.Add("@patronymic",
SqlDbType.VarChar, 20, "patronymic");
```

```
            da1.InsertCommand.Parameters.Add("@dob",
SqlDbType.Date, 10, "dob");
            da1.InsertCommand.Parameters.Add("@phone",
SqlDbType.BigInt, 100, "phone");
            da1.InsertCommand.Parameters.Add("@adress",
SqlDbType.VarChar, 20, "adress");
            da1.InsertCommand.Parameters.Add("@IIN",
SqlDbType.BigInt, 100, "IIN");
            da1.InsertCommand.Parameters.Add("@num_identity_card",
SqlDbType.Int, 10, "num_identity_card");
            da1.InsertCommand.Parameters.Add("@issued",
SqlDbType.VarChar, 20, "issued");
            da1.InsertCommand.Parameters.Add("@date_of_issue",
SqlDbType.Date, 10, "date_of_issue");
            da1.InsertCommand.Connection = conn1;

            conn1.Open();
            da1.Update(d1, "clients");
            conn1.Close();

        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
        finally { MessageBox.Show("Saved"); }
    }

    //для договоров
    public void read_MySql_contracts(ref DataGridView dgv)
    {
        dataGridView2 = dgv;
        try
        {
            dgv.Columns.Clear();
            conn2 = new SqlConnection("DATA SOURCE = localhost;
Initial catalog = lombard; Integrated Security = True;");
            conn2.Open();
            adapter2 = new SqlDataAdapter("select · from contract",
conn2);
            adapter2.Fill(d2, "contract");
            conn2.Close();
```

```
            b2.DataSource = d2.Tables["contract"];
            dgv.DataSource = b2;
```


```
        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
        finally { MessageBox.Show("Read"); }
    }
    public void save_MySql_contracts(ref DataGridView dgv)
    {
        dataGridView2 = dgv;
        try
        {
            SqlDataAdapter da2 = new SqlDataAdapter();

            da2.SelectCommand = new SqlCommand("SELECT
num_contract, date_of_completion, time,
closing_period_of_the_contract, cont_state, num_client, thing_id,
num_employee FROM `lombard`.`contract`", conn2);
            da2.UpdateCommand = new SqlCommand("UPDATE
contract SET date_of_completion=@date_of_completion, time=@time,
closing_period_of_the_contract=@closing_period_of_the_contract,
cont_state=@cont_state, num_client=@num_client,
thing_id=@thing_id, num_employee=@num_employee WHERE
num_contract=@num_contract");
            da2.DeleteCommand = new SqlCommand("DELETE contract
SET date_of_completion=@date_of_completion, time=@time,
closing_period_of_the_contract=@closing_period_of_the_contract,
cont_state=@cont_state, num_client=@num_client,
thing_id=@thing_id, num_employee=@num_employee WHERE
num_contract=@num_contract");
            da2.UpdateCommand.Parameters.Add("@num_contract",
SqlDbType.Int, 10, "num_contract");

da2.UpdateCommand.Parameters.Add("@date_of_completion",
SqlDbType.Date, 10, "date_of_completion");
            da2.UpdateCommand.Parameters.Add("@time",
SqlDbType.Date, 10, "time");
```

```
da2.UpdateCommand.Parameters.Add("@closing_period_of_the_contra
ct", SqlDbType.Date, 10, "closing_period_of_the_contract");
            da2.UpdateCommand.Parameters.Add("@cont_state",
SqlDbType.VarChar, 20, "cont_state");
            da2.UpdateCommand.Parameters.Add("@num_client",
SqlDbType.Int, 10, "num_client");
            da2.UpdateCommand.Parameters.Add("@thing_id",
SqlDbType.Int, 10, "thing_id");
            da2.UpdateCommand.Parameters.Add("@num_employee",
SqlDbType.Int, 10, "num_employee");
            da2.UpdateCommand.Connection = conn2;
```

Продолжение приложения А

```
da2.DeleteCommand.Connection = conn2;
            da2.InsertCommand = new SqlCommand("INSERT INTO
contract VALUES (num_contract, date_of_completion, time,
closing_period_of_the_contract, cont_state, num_client, thing_id,
num_employee)", conn2);
            da2.InsertCommand.Parameters.Add("@num_contract",
SqlDbType.Int, 10, "num_contract");
            da2.InsertCommand.Parameters.Add("@date_of_completion",
SqlDbType.Date, 10, "date_of_completion");
            da2.InsertCommand.Parameters.Add("@time",
SqlDbType.Date, 10, "time");

da2.InsertCommand.Parameters.Add("@closing_period_of_the_contract
", SqlDbType.Date, 10, "closing_period_of_the_contract");
            da2.InsertCommand.Parameters.Add("@cont_state",
SqlDbType.VarChar, 20, "cont_state");
            da2.InsertCommand.Parameters.Add("@num_client",
SqlDbType.Int, 10, "num_client");
            da2.InsertCommand.Parameters.Add("@thing_id",
SqlDbType.Int, 10, "thing_id");
            da2.InsertCommand.Parameters.Add("@num_employee",
SqlDbType.Int, 10, "num_employee");
            da2.InsertCommand.Connection = conn2;

            conn2.Open();
```

```
            da2.Update(d2, "contract");
            conn2.Close();

        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
        finally { MessageBox.Show("Saved"); }
    }

    //для вещей
    public void read_MySql_things(ref DataGridView dgv)
    {
        datagridView3 = dgv;
        try
        {
            dgv.Columns.Clear();
            conn3 = new SqlConnection("DATA SOURCE = localhost;
Initial catalog = lombard; Integrated Security = True;");
```

Продолжение приложения А

```
  conn3.Open();
            adapter3 = new SqlDataAdapter("select · from things", conn3);
            adapter3.Fill(d3, "things");
            conn3.Close();
            b3.DataSource = d3.Tables["things"];
            dgv.DataSource = b3;

        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
        finally { MessageBox.Show("Read"); }
    }
    public void save_MySql_things(ref DataGridView dgv)
    {
        datagridView3 = dgv;
        try
        {
            SqlDataAdapter da3 = new SqlDataAdapter();

            da3.SelectCommand = new SqlCommand("SELECT thing_id,
assessed_value, state, price, color,
```

```
comments,id_podtype,id_type,id_mod,id_brand FROM
`lombard`.`things`", conn3);
            da3.UpdateCommand = new SqlCommand("UPDATE laid
assessed_value=@assessed_value, state=@state, price=@price,
color=@color, comments=@comments, id_podtype=@id_podtype,
id_type=@id_type, id_mod=@id_mod, id_brand=@id_brand WHERE
thing_id=@thing_id");
            da3.DeleteCommand = new SqlCommand("DELETE laid
assessed_value=@assessed_value, state=@state, price=@price,
color=@color, comments=@comments, id_podtype=@id_podtype,
id_type=@id_type, id_mod=@id_mod, id_brand=@id_brand WHERE
thing_id=@thing_id");
            da3.UpdateCommand.Parameters.Add("@thing_id",
SqlDbType.Int, 10, "thing_id");
            da3.UpdateCommand.Parameters.Add("@assessed_value",
SqlDbType.VarChar, 20, "assessed_value");
            da3.UpdateCommand.Parameters.Add("@state",
SqlDbType.VarChar, 20, "state");
            da3.UpdateCommand.Parameters.Add("@price",
SqlDbType.Int, 10, "price");
            da3.UpdateCommand.Parameters.Add("@color",
SqlDbType.VarChar, 20, "color");
            da3.UpdateCommand.Parameters.Add("@comments",
SqlDbType.VarChar, 20, "comments");
```

Продолжение приложения А

```
  da3.UpdateCommand.Parameters.Add("@id_podtype", SqlDbType.Int,
10, "id_podtype");
            da3.UpdateCommand.Parameters.Add("@id_type",
SqlDbType.Int, 10, "id_type");
            da3.UpdateCommand.Parameters.Add("@id_mod",
SqlDbType.Int, 10, "id_mod");
            da3.UpdateCommand.Parameters.Add("@id_brand",
SqlDbType.Int, 10, "id_brand");
            da3.UpdateCommand.Connection = conn3;
            da3.DeleteCommand.Connection = conn3;
```

```
            da3.InsertCommand = new SqlCommand("INSERT INTO
things VALUES
(@thing_id,@assessed_value,@state,@price,@color,@comments,@id_
podtype,@id_type,@id_mod,@id_brand)", conn3);
            da3.InsertCommand.Parameters.Add("@thing_id",
SqlDbType.Int, 10, "thing_id");
            da3.InsertCommand.Parameters.Add("@assessed_value",
SqlDbType.VarChar, 20, "assessed_value");
            da3.InsertCommand.Parameters.Add("@state",
SqlDbType.VarChar, 20, "state");
            da3.InsertCommand.Parameters.Add("@price",
SqlDbType.Int, 10, "price");
            da3.InsertCommand.Parameters.Add("@color",
SqlDbType.VarChar, 20, "color");
            da3.InsertCommand.Parameters.Add("@comments",
SqlDbType.VarChar, 20, "comments");
            da3.InsertCommand.Parameters.Add("@id_podtype",
SqlDbType.Int, 10, "id_podtype");
            da3.InsertCommand.Parameters.Add("@id_type",
SqlDbType.Int, 10, "id_type");
            da3.InsertCommand.Parameters.Add("@id_mod",
SqlDbType.Int, 10, "id_mod");
            da3.InsertCommand.Parameters.Add("@id_brand",
SqlDbType.Int, 10, "id_brand");
            da3.InsertCommand.Connection = conn3;

            conn3.Open();
            da3.Update(d3, "things");
            conn3.Close();

        }
```

Продолжение приложения А

```
 catch (Exception exp) { MessageBox.Show(exp.ToString()); }
          finally { MessageBox.Show("Saved"); }
      }

      //для зп
      public void read_MySql_zp(ref DataGridView dgv)
```

```
        {
            datagridView15 = dgv;
            try
            {
                dgv.Columns.Clear();
                conn9 = new SqlConnection("DATA SOURCE = localhost;
Initial catalog = lombard; Integrated Security = True;");
                conn9.Open();
                adapter9 = new SqlDataAdapter("select · from kassa", conn9);
                adapter9.Fill(d9, "kassa");
                conn9.Close();
                b9.DataSource = d9.Tables["kassa"];
                dgv.DataSource = b9;

            }
            catch (Exception exp) { MessageBox.Show(exp.ToString()); }
            finally { MessageBox.Show("Read"); }
        }
        public void save_MySql_zp(ref DataGridView dgv)
        {
            datagridView15 = dgv;
            try
            {
                SqlDataAdapter da9 = new SqlDataAdapter();

                da9.SelectCommand = new SqlCommand("SELECT
id_zp,proc_zp,num_employee,num_contract FROM `lombard`.`kassa`",
conn9);
                da9.UpdateCommand = new SqlCommand("UPDATE kassa
proc_zp=@proc_zp, num_employee=@num_employee,
num_contract=@num_contract WHERE id_zp=@id_zp");
                da9.DeleteCommand = new SqlCommand("DELETE kassa
proc_zp=@proc_zp, num_employee=@num_employee,
num_contract=@num_contract WHERE id_zp=@id_zp");
                da9.UpdateCommand.Parameters.Add("@id_zp",
SqlDbType.Int, 10, "id_zp");
```

Продолжение приложения А

```
 da9.UpdateCommand.Parameters.Add("@proc_zp",
SqlDbType.VarChar, 20, "proc_zp");
            da9.UpdateCommand.Parameters.Add("@num_employee",
SqlDbType.Int, 10, "num_employee");
            da9.UpdateCommand.Parameters.Add("@num_contract",
SqlDbType.Int, 10, "num_contract");
            da9.UpdateCommand.Connection = conn9;
            da9.DeleteCommand.Connection = conn9;

            da9.InsertCommand = new SqlCommand("INSERT INTO
kassa VALUES
(@id_zp,@proc_zp,@num_employee,@num_contract)", conn9);
            da9.InsertCommand.Parameters.Add("@id_zp",
SqlDbType.Int, 10, "id_zp");
            da9.InsertCommand.Parameters.Add("@proc_zp",
SqlDbType.VarChar, 20, "proc_zp");
            da9.InsertCommand.Parameters.Add("@num_employee",
SqlDbType.Int, 10, "num_employee");
            da9.InsertCommand.Parameters.Add("@num_contract",
SqlDbType.Int, 10, "num_contract");
            da9.InsertCommand.Connection = conn9;

            conn9.Open();
            da9.Update(d9, "kassa");
            conn9.Close();

        }
        catch (Exception exp) { MessageBox.Show(exp.ToString()); }
        finally { MessageBox.Show("Saved"); }
    }
  };
}
```

Приложение Б

Запросы базы данных

Описание для таблицы "clients"

```
CREATE TABLE "clients" (
  "num_client"       NUMBER(38, 0) NOT NULL,
  "surname"          VARCHAR2(20 BYTE),
  "name"             VARCHAR2(20 BYTE),
  "patronymic"       VARCHAR2(20 BYTE),
  "dob"              DATE          NOT NULL,
  "phone"            NUMBER(38, 0) NOT NULL,
  "adress"           VARCHAR2(20 BYTE),
  IIN                NUMBER(38, 0) NOT NULL,
  "num_identity_card" NUMBER,
  "issued"           VARCHAR2(20 BYTE),
  "date_of_issue"    DATE          NOT NULL,
  CONSTRAINT "PK_clients" PRIMARY KEY ("num_client") USING
INDEX TABLESPACE USERS STORAGE (INITIAL 64 K
    NEXT 1 M
    MAXEXTENTS UNLIMITED)
)
TABLESPACE USERS
STORAGE (INITIAL 64 K
        NEXT 1 M
        MAXEXTENTS UNLIMITED)
LOGGING;
```

Описание для таблицы "employee"

```
CREATE TABLE "employee" (
  "surname"          VARCHAR2(20 BYTE),
  "num_employee"     NUMBER(38, 0)    NOT NULL,
  "name"             VARCHAR2(20 BYTE),
  "patronymic"       VARCHAR2(20 BYTE),
  "dob"              DATE             NOT NULL,
  "phone"            NUMBER(38, 0)    NOT NULL,
  "adress"           VARCHAR2(20 BYTE),
  IIN                NUMBER(38, 0)    NOT NULL,
```

```
  "num_identity_card" NUMBER(38, 0),
  "issued"           VARCHAR2(20 BYTE),
  "date_of_issue"    DATE             NOT NULL,
  "id_position"      NUMBER(38, 0),
  "login"            VARCHAR2(10 BYTE) NOT NULL,
```

Продолжение приложения Б

```
"pass"             VARCHAR2(255 BYTE),
  "id_city"          NUMBER(38, 0),
  "id_country"       NUMBER(38, 0),
  CONSTRAINT "PK_employee" PRIMARY KEY ("num_employee")
USING INDEX TABLESPACE USERS STORAGE (INITIAL 64 K
   NEXT 1 M
   MAXEXTENTS UNLIMITED),
  FOREIGN KEY ("id_city")
  REFERENCES "city" ("id_city"),
  FOREIGN KEY ("id_country")
  REFERENCES "country" ("id_country"),
  FOREIGN KEY ("id_position")
  REFERENCES "position" ("id_position")
)
TABLESPACE USERS
STORAGE (INITIAL 64 K
        NEXT 1 M
        MAXEXTENTS UNLIMITED)
LOGGING;
```

Описание для таблицы "models"

```
CREATE TABLE "models" (
  "id_mod"     NUMBER(38, 0) NOT NULL,
  "mod_name"   VARCHAR2(20 BYTE),
  "year"       DATE,
  "id_brand"   NUMBER(38, 0),
  "id_podtype" NUMBER(38, 0),
  CONSTRAINT "PK_models" PRIMARY KEY ("id_mod") USING
INDEX TABLESPACE USERS STORAGE (INITIAL 64 K
NEXT 1 M
MAXEXTENTS UNLIMITED),
```

```
  FOREIGN KEY ("id_brand")
  REFERENCES "brand" ("id_brand"),
  FOREIGN KEY ("id_podtype")
  REFERENCES "pod_type" ("id_podtype")
)
TABLESPACE USERS
STORAGE (INITIAL 64 K
        NEXT 1 M
        MAXEXTENTS UNLIMITED)
LOGGING;
```

Продолжение приложения Б

Вывод данных для таблицы "clients"

```
INSERT INTO "clients" VALUES
(1, 'Eenoia', 'Eeno', 'Eenoia?e', '06–MAR–94', 87021547832, 'oaoeeia,12', 121447, 14777, 'i? ?e', '06–MAR–94');
INSERT INTO "clients" VALUES
(2, 'Eanaiiaa', 'Ee?a', 'Iaeoee', '18–JUN–88', 87052785461, 'aaeoo?nuiiaa,87', 124785, 155555, 'i? ?e', '02–MAR–03');
INSERT INTO "clients" VALUES
(3, '?oaaiia', 'Eae?', 'Eae?iae?', '05–MAR–88', 87052757341, 'o?oica,78', 45788888, 15544, 'i? ?e', '05–MAR–06');
INSERT INTO "clients" VALUES
(6, 'as', 'as', 'as', '12–SEP–91', 21, 'c', 1, 1, 'c', '15–MAR–08');
INSERT INTO "clients" VALUES
(8, 'Iao?iaa', 'Iao?eaia', 'Aeeiiaia', '12–APL–89', 87015623987, 'naeia,16', 5789, 2233333, 'i? ?e', '10–MAR–06');
INSERT INTO "clients" VALUES
(9, 'Aaeeiaa', 'Aoeuie?a', 'Oaeeiiaia', '13–SEP–81', 87025467896, 'oaeyieia,38', 2125, 23632, 'i? ?e', '12–NOV–07');
INSERT INTO "clients" VALUES
(7, 'cxxz', 'zxcxz', 'zxcxz', '11–SEP–01', 1542, 'sad', 65234156, 56231, 'sadzx', '11–SEP–16');
```

Вывод данных для таблицы "employee"

```
INSERT INTO "employee" VALUES
```

('Iae?iaiiaa', 1, 'Aeaioa', 'Oaeeiiaia', '06–MAR–94', 87027541875, 'aaecaeiaa,12', 123132, 14578, 'iaa ?e', '13–JUN–11', 1, 'asd ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('Oieciaa', 2, 'Aeaia', 'Eneaiiaia', '06–APL–94', 87052754341, 'naeia,38', 123456, 111111, 'i? ?e', '06–APL–06', 1, 'dia ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('Eaeeaaa', 3, '?aoca', 'Aoiao?aiiaia', '06–MAR–94', 87021547892, 'aaay,32', 124578, 222222, 'i? ?e', '06–MAR–05', 2, 'rau ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('Ecaaeeaa', 4, 'Ecei', 'Eceiiae?', '02–JUN–78', 87024578963, 'aaaa?eia,80', 333333, 333321, 'i? ?e', '02–JUN–90', 3, 'izi ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES

Продолжение приложения Б

('Einaiiaa', 5, 'Aeaa', 'Ea?eiiaia', '02–JUN–90', 87052457896, '?icuaaeeaaa,12', 45899, 44444, 'iaa ?e', '02–JUN–11', 5, 'aida ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('Iaaiaoia', 6, 'Niaaoe', 'Ooeaoiae?', '02–JUN–88', 87075514872, 'naeia,12', 4725626, 5578888, 'i? ?e', '02–JAN–90', 4, 'smagi ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('?a?eania', 7, 'Aaaaiee', 'Iao?iae?', '02–JUN–90', 87024567891, 'oaeyieia,18', 147852, 123456, 'i? ?e', '02–JUN–15', 5, 'evg ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('Aaaoeeia', 8, '?aooai', 'Ee?eeiaia', '02–NOV–90', 87025469871, 'aae?aiiaa,21', 457892, 12211111, 'i? ?e', '02–JUN–90', 5, 'raushan ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('Iai', 9, 'Iaenei', 'Aaaaiuaae?', '08–SEP–75', 87052456789, 'iaaie,87', 457896, 122448, 'i? ?e', '09–MAR–11', 4, 'maks ', ' ,?b¬Y[–K–#Kp', NULL, NULL);

INSERT INTO "employee" VALUES
('Aiai?ieiaa', 10, 'Aaaia', 'Ai?eniaia', '02–MAR–65', 87024578912, 'i?aaaa,15', 789456, 555558, 'iaa ?e', '14–AUG–10', 2, 'adema ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('?noiia', 11, '??ee', 'Ea?eiiae?', '08–SEP–75', 87012457896, 'aeu–oa?aae,1', 4578921, 1114542, 'i? ?e', '02–NOV–90', 3, 'uriy ', ' ,?b¬Y[–K–#Kp', NULL, NULL);
INSERT INTO "employee" VALUES
('?icuaaeeaa', 12, 'A?cai', 'A?caiiae?', '02–JUN–88', 87052457896, 'aaay,14', 124578, 258741, 'i? ?e', '02–JUN–15', 3, 'arza ', ' ,?b¬Y[–K–#Kp', NULL, NULL);

Вывод данных для таблицы "models"

INSERT INTO "models" VALUES
(1, 'Emachine ET1850', '02–JUN–88', 9, 10);
INSERT INTO "models" VALUES
(2, 'AMC605', '02–JUN–88', 9, 7);
INSERT INTO "models" VALUES
(3, 'Aspire TC–605', '02–MAR–15', 9, 10);
INSERT INTO "models" VALUES
(4, 'AM1930', '15–MAR–01', 9, 7);
INSERT INTO "models" VALUES

Продолжение приложения Б

(5, 'imedia L4880', '02–NOV–90', 9, 7);
INSERT INTO "models" VALUES
(6, 'Pavilion p6–241', '02–NOV–00', 14, 10);
INSERT INTO "models" VALUES
(7, '110–112er', '02–MAR–15', 14, 9);
INSERT INTO "models" VALUES
(8, 'ZCG56BGW', '10–FEB–15', 7, 1);
INSERT INTO "models" VALUES
(9, 'GC–B 379 SVQA', '03–DEC–10', 3, 1);
INSERT INTO "models" VALUES
(10, 'NR–B 591 BR–C4', '04–JUN–06', 8, 3);
INSERT INTO "models" VALUES
(11, 'NR–B651BR–N4', '17–SEP–11', 8, 3);

```
INSERT INTO "models" VALUES
(12, 'DMC–FS28EE–S ', '02–JAN–14', 8, 21);
INSERT INTO "models" VALUES
(13, 'DMC–FZ48EE–K', '10–NOV–13', 8, 21);
INSERT INTO "models" VALUES
(14, 'DSLR–A 390L KIT18–55', '13–NOV–03', 23, 22);
INSERT INTO "models" VALUES
(15, 'SLT–A55VL', '02–JAN–14', 23, 22);
INSERT INTO "models" VALUES
(16, 'SLT–A56VL', '02–JAN–14', 23, 22);
INSERT INTO "models" VALUES
(17, 'SLT–A55HL', '04–JUN–06' , 23, 22);
INSERT INTO "models" VALUES
(18, 'T100TA–DK005H 10,1', '04–JUN–06' , 11, 9);
INSERT INTO "models" VALUES
(19, 'T100TA–DK 10,1', '04–JUN–06', 11, 9);
INSERT INTO "models" VALUES
(20, '20PHH4109/60', '04–JUN–06', 20, 5);
INSERT INTO "models" VALUES
(21, '20P109/60', '02–JAN–14', 20, 5);
INSERT INTO "models" VALUES
(22, 'UE32H4000AK', '02–JAN–14', 5, 27);
INSERT INTO "models" VALUES
(23, 'UE32H5000AK', '02–JAN–14', 5, 27);
INSERT INTO "models" VALUES
(24, 'UE32H6000AK', '02–JAN–14', 5, 27);
COMMIT;
INSERT INTO "models" VALUES
(25, '50PZ 850', '02–JAN–14', 3, 28);
```

Продолжение приложения Б

```
INSERT INTO "models" VALUES
(26, '50PZ 855', '02–JAN–14', 3, 28);
INSERT INTO "models" VALUES
(27, 'PE51H4500', '02–JAN–14', 5, 28);
INSERT INTO "models" VALUES
(28, 'PE51H4501', '02–JAN–14', 5, 28);
INSERT INTO "models" VALUES
(29, 'PE51H4511', '02–JAN–14', 5, 28);
```

```
INSERT INTO "models" VALUES
(30, 'PE51H4502', '02–JAN–14', 5, 28);
INSERT INTO "models" VALUES
(31, 'NX mini kit 9 mm', '02–JAN–14', 5, 21);
INSERT INTO "models" VALUES
(32, 'EV–NX1000', '02–JAN–14', 5, 23);
INSERT INTO "models" VALUES
(33, 'EV–NX1001', '02–JAN–14', 5, 23);
```

Приложение В

Функции, процедуры и триггеры

Процедура вывода контракта, суммы его и кем был подписан, а также клиента, на которого он оформлен.

```
create or replace procedure contract (num number)
is
cursor asd(num1 number)
is
select c."num_contract",c."sum",e."surname" as "empsur",cl."surname" from
"contract" c
inner join "employee" e
on c."num_employee"=e."num_employee"
inner join "clients" cl
on c."num_client"=cl."num_client"
where c."num_employee"=num1;
begin
for emp_1 in asd(num)
loop
   DBMS_OUTPUT.PUT_LINE (lpad(emp_1."num_contract",20,' ')||' cost '||emp_1."sum"||' client '||emp_1."surname"|| ' emp '||emp_1."empsur" );
end loop;
end;
```

Функция считает полную стоимость прибыли клиента от заказа

```
create or replace function allcost (num number) return number
is
vip number;
begin
select sum("sum"·0.01·"%vypl") into vip from "contract" where "num_client"=num;
return vip;
end;
```

Процедура считает полную стоимость прибыли всех сотрудников от заказа

```
create or replace procedure allcost2
is
cursor cur
is
select sum("sum"·0.01·'%vypl") "ad","num_employee"
from "contract"
group by "num_employee";
begin
```



```
for ssd in cur
loop
DBMS_OUTPUT.PUT_LINE (ssd."num_employee"||' summ
'||ssd."ad");
end loop;
end;

Функция считает прибыль от продажи вещи
create or replace function pribil (num number) return number
is
vip number;
begin
select c."sum"–t."price" into vip from "contract" c
inner join "things" t
on t."thing_id"=c."thing_id"
where "num_contract"=num;
return vip;
end;
```

Триггер, запрещающий повторять ИИН

```
create or replace TRIGGER mol
before  insert or update of "IIN" ON "clients"
for each  row
DECLARE
clientsn number;
BEGIN
select count(·) into clientsn from "clients" s where s."IIN"=:new."IIN";
IF (clientsn=1) then
RAISE_APPLICATION_ERROR(–20323,'Oshibka pri vvode IIN. Nelyazya povtiryt');
end if;
end;
```