

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ им.
ГУМАРБЕКА ДАУКЕЕВА»
Кафедра IT - инжиниринг

«ДОПУЩЕН К ЗАЩИТЕ»

Зав. Кафедрой, PhD, доцент Досжанова А.А

_____ «__» _____ 2020

ДИПЛОМНЫЙ ПРОЕКТ

На тему: Разработка информационной системы учета продаж автомобильных запчастей.

Специальность 5В060200 – «Информатика»

Выполнил: Титов А.С. Группа: ИНФ-16-2

Научный руководитель: к.т.н., доцент Тусупова Б.Б.

Консультанты:

по экономической части: к.э.н., профессор Габелашвили К.Р

(учёная степень, звание, Ф.И.О.)

_____ «__» _____ 2020 г.

по безопасности жизнедеятельности: ассистент Тыщенко Е.М

(учёная степень, звание, Ф.И.О.)

_____ «__» _____ 2020 г.

по программному обеспечению: ст.преп. Майкотов М.Н

(учёная степень, звание, Ф.И.О.)

_____ «__» _____ 2020 г.

Нормоконтролер: ст.преп. Абсатарова Б.Р

(учёная степень, звание, Ф.И.О.)

_____ «__» _____ 2020 г.

Рецензент: _____

(учёная степень, звание, Ф.И.О.)

_____ «__» _____ 2020 г.

Алматы 2020

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
Некоммерческое акционерное общество
«АЛМАТИНСКИЙ УНИВЕРСИТЕТ ЭНЕРГЕТИКИ И СВЯЗИ им.
ГУМАРБЕКА ДАУКЕЕВА»

Институт систем управления и информационных технологий

Кафедра IT-инжиниринг

Специальность 5В060200 – «Информатика»

ЗАДАНИЕ

на выполнение дипломной работы

Студенту Титову Александру Сергеевичу

Тема работы: Разработка информационной системы учета продаж автомобильных запчастей.

Утверждена приказом по университету № 147 от «11» ноября 2019 г.

Срок сдачи законченной работы «01» июня 2020 г.

Исходные данные к работе (требуемые параметры результатов исследования (проектирования) и исходные данные объекта):

Перечень вопросов, подлежащих разработке в дипломной работе, или краткое содержание дипломной работы: анализ существующих автоматизированных информационных систем, проектирование информационной системы, разработка программного обеспечения системы.

Перечень графического материала (с точным указанием обязательных чертежей): физическая модель базы данных стр.18, схема базы данных стр.18, диаграмма последовательности работы базы данных стр.18.

Основная рекомендуемая литература:

1 Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова, О.П. Ляпина, А.В. Гусева. - М.: Academia, 2017. - 416 с;

2 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с;

3 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с.

Консультации по работе с указанием относящихся к ним разделов работы

Раздел	Консультант	Сроки	Подпись
Экономическая часть	Габелашвили К. Р.		
Безопасности жизнедеятельности	Тыщенко Е.М.		
Программная часть	Майкотов М.Н.		
Нормоконтролер	Абсатарова Б.Р.		

График
подготовки дипломной работы

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Исследование предметной области	13.01.2020- 13.02.2020	
Проектирование ориентированной информационной системы	13.01.2020- 20.03.2020	
Разработка программного продукта	21.02.2020- 30.04.2020	

Дата выдачи задания «13» января 2020 г.

Заведующий кафедрой _____ А.А. Досжанова

Научный руководитель работы _____ Б.Б. Тусупова

Задание принял к исполнению студент _____ А.С. Титов

АҢДАТПА

Автоқосалқы бөлшектерді сату бойынша кәсіпорын үшін ақпараттық жүйені әзірлеу-бұл қойылған міндеттердің басты шешімі. Ақпараттық жүйе автосервистің жұмыс процесін жеңілдетуге мүмкіндік береді. Жобада тапсырыстарға, сақтауға, импорттауға және экспортқа үлкен көңіл бөлінген. Тапсырыстарды импорттау MySQL деректер қорының көмегімен жүргізіледі және осы жобада маңызды рөл атқарады. Қосымшадағы барлық процесс деректер қорын пайдалану арқылы өтеді. Қосымша тапсырыс жасауға және оны деректер базасында сақтауға мүмкіндік береді.

АННОТАЦИЯ

Разработка информационной системы для предприятия по продаже автозапчастей – это главное решение поставленной задачи. Информационная система позволяет упростить процесс работы автосервиса. Огромное внимание в проекте уделено заказам, хранению, импорту и экспорту. Импорт заказов ведется с помощью базы данных MySQL и играет важную роль в этом проекте. Весь процесс в приложении протекает с использованием базы данных. Приложение дает возможность совершить заказ и хранить ее в базе данных.

THE SUMMARY

The development of an information system for an enterprise selling auto parts is the main solution to the problem. The information system allows you to simplify the process of car service. A lot of attention in the project is paid to orders, storage, import and export. Orders are imported using the MySQL database and play an important role in this project. The entire process in the application occurs using the database. The application allows you to complete an order and save it in the database.

Содержание

Введение	8
1 Теоретическое обоснование разработки информационной системы	9
1.1 Анализ предметной области	9
1.2 Существующие проблемы систем и их проблемы	9
1.3 Решение проблем.....	15
1.4 Постановка задачи.....	17
2 Описание и обоснование выбора языка программирования и СУБД	19
2.1 Проектирование информационной системы	19
2.2 Анализ современных языков программирования	22
2.3 Выбор языка программирования С#.....	22
2.4 Особенность использования СУБД.....	23
Преимущества SQL Server:.....	23
2.5 Описание структур таблиц базы данных.....	24
3 Разработка информационной системы	25
3.1 Программная реализация	25
3.2 Системные требования.....	34
3.3 Интерфейс.....	35
4 Обоснование эффективности внедрения проекта	46
4.1 Технико-экономическое обоснование дипломных работ, связанных с разработкой программного продукта (ПП).....	46
4.2 Трудоемкость разработки ПП.....	46
4.3 Расчет затрат на разработку ПП.....	47
4.4 Определение договорной цены ПП.....	50
4.5 Расчет результатов от создания и использования ИС.....	50
4.6 Расчет основных показателей экономической эффективности	52
5 Безопасность жизнедеятельности	54
5.1 Определение категории тяжести труда через интегрально-бальную оценку.....	54
5.2 Расчет.....	56
Заключение	60
Список литературы	61
Приложение А	62
Приложение Б.....	63

Введение

Производство вычислительных машин дает возможность автоматизировать обработку данных в разных сферах деятельности человека. Большой шаг для развития обработки данных повлияло появление персональных компьютеров.

Информационные системы являются распространенным классом систем обработки данных.

Разные виды деятельности основывается на информации о свойствах состояния и поведения той части реального мира, с которой связана эта деятельность.

Основной задачей информационной системы является:

- импорт заказов;
- хранение заказов;
- экспорт заказов;

Облегчение работы персонала.

Информационная система позволяет повысить эффективность и качество работы предприятия по продаже автозапчастей.

Главными задачами информационной системы является:

- облегчить работу персонала;
- ускорить процесс работы.

У предприятий есть необходимость систематически улучшать свою экономическую работу для снижения себестоимости.

В первом разделе был анализ предметной области. В этом разделе необходимо рассмотреть существующие системы и найти их проблемы, что в дальнейшем решить в программной реализации.

Второй раздел посвящен выбранному языку программирования. В проекте был выбран язык C# с подключенной MySQL базой данных.

В третьем разделе была разработана и описана программная реализация. Главной задачей раздела было описать работы программы детально через код и интерфейс.

В четвертом разделе приведено обоснование экономической эффективности внедряемой программы на предприятие.

Пятый раздел посвящен вопросам охраны труда и безопасности жизнедеятельности.

1 Теоретическое обоснование разработки информационной системы

1.1 Анализ предметной области

ТОО «Райымбек Курылыс» один из крупнейших компаний в Казахстане занимающаяся автозапчастями, смазочными материалами и другими расходными материалами для всех видов техники с 1997 года. На сегодняшний день компания является официальным дистрибьютором таких мировых брендов:

- Valvoline;
- Fleetguard.

Основным принципом компании является самое высокое качество продукции, для своих клиентов и партнеров, именно поэтому компания сотрудничает исключительно с конвейерными поставщиками.

Более чем за 20 лет работы компания смогли заработать безупречную репутацию среди наших партнеров, которые высоко оценили качество и надежность поставляемых нами товаров, благодаря которым смогли сократить затраты на содержание и обслуживание техники.

Для крупных предприятий у компании есть особенные условия работы. Компания за свой счет предоставляет тестирование наших продуктов, моторного масла, антифриза и фильтров на технике заказчика, чтобы заказчик смог оценить качество и надежность продуктов перед их использованием:

- тест моторных масел;
- тест антифриза;
- тест на чистоту качества топлива.

1.2 Существующие проблемы систем и их проблемы

Предприятие использует 1С.

«1С» является российской компанией, которая сосредоточена на дистрибуции, поддержке и разработке компьютерных программ и баз данных делового и домашнего назначения.

Предприятие является обществом с ограниченной ответственностью. Основатель и директор компании — Борис Нуралиев.

Предприятие себе так назвала из-за поиска в собственной программе за 1 секунду.

У данной программы есть множество минусов:

- слабая поддержка;
- неготовность системы для сдачи регламентированной отчетности;
- неработающие стандартные механизмы;

- уязвимость к редактированию документов задним числом.
- отсутствие тестов типовой конфигурации;
- логическая недоработка переходов на систему;
- плохая адаптация конфигураций не только ERP, но и остальных КА2, УТ. Бухгалтерия и УНФ вроде пашут нормально;
- фантомный механизм разнесения дополнительных расходов на себестоимость товаров;
- проблема с правами доступа;
- неадаптированные модули ERP для Казахстана;

Бухгалтерия от 1С является очень прожорливым продуктом, который расходует приличное количество системных ресурсов. Для комфортной работы в программе необходимо умножать все показатели на 2.

Собственно, системные требования бухгалтерии 1С такие:

- операционная система Windows XP Service Pack 2 и выше, Windows Server 2003 и выше, Ubuntu 12.04 LTS и выше, Alt Linux СПТ 6.0 и выше;
- процессор Intel Pentium/Celeron 1800 МГц и выше;
- оперативная память 1 Гбайт и выше;
- жесткий диск (при установке используется около 300 Мбайт);
- SVGA-дисплей.

Интерфейс 1С очень сложен для первого освоения новичком. Придется потратить много времени для освоения программы или обучения сотрудников предприятия. Есть необходимость долгого привыкания к самой программе.

Программа является очень требовательной к исполнению. Нет возможности свободного использования программы. Есть большая вероятность получения ошибки на каждом этапе работы с программой. Необходимо соблюдать четкий механизм с четким перечнем процедур.

Из-за больших проблем в более дешевых версиях, предприятие создает искусственные проблемы, которые решены в более дорогих версиях. Отсюда следует, что приходится приобретать профессиональную версию, которая стоит огромных средств (таблица 1).

Таблица 1.1

Наименование	Валюта	рекомендованная цена	цена для дилера	цена для партнера	ставка НДС
2900002163513	1С:Бухгалтерия 8 для Казахстана. Базовая версия. Электронная поставка	KZT	21 600	10 800	9 720
4601546045423	1С:Бухгалтерия 8 для Казахстана. Базовая версия *	KZT	21 600	10 800	9 720
460154609071	1С:Бухгалтерия 8 для Казахстана.	KZT	1 800	1 080	900

Наименование	Валюта	рекомендованная цена	цена для дилера	цена для пост. партнера	ставка НДС
3	Учебная версия. 2-е издание *				

Продолжение таблицы 1.1

460154607774 5	1С:Предприятие 8.2. Версия для обучения программированию для Казахстана *	KZT	5 400	3 510	3 240
290000216350 6	1С:Бухгалтерия 8 для Казахстана. Электронная поставка	KZT	64 800	32 400	29 160
460154609581 7	1С:Бухгалтерия 8 для Казахстана	KZT	64 800	32 400	29 160
460154604541 6	1С:Бухгалтерия 8 для Казахстана (USB)	KZT	76 320	38 160	34 344
460154609582 4	1С:Бухгалтерия 8 для Казахстана. Комплект на 5 пользователей	KZT	151 200	75 600	68 040
460154604543 0	1С:Бухгалтерия 8 для Казахстана. Комплект на 5 пользователей (USB)	KZT	162 720	81 360	73 215
290000079687 4	1С:Бухгалтерия 8 для Казахстана OEM (25 шт.) *	KZT		63 480	54 360
290000216359 9	1С:Предприятие 8. Бухгалтерский учет для государственных учреждений Казахстана. Электронная поставка	KZT	77 760	38 880	34 992
460154610461 8	1С:Предприятие 8. Бухгалтерский учет для государственных учреждений Казахстана	KZT	77 760	38 880	34 992
460154604436 5	1С:Предприятие 8. Бухгалтерский учет для государственных учреждений Казахстана (USB)	KZT	89 280	44 640	40 176
460154610974 3	1С:Предприятие 8. Бухгалтерский учет для государственных учреждений Казахстана. Комплект на 5 пользователей	KZT	165 600	82 800	74 520
290000216358 2	1С:Предприятие 8. Бухгалтерский учет для государственных предприятий Казахстана. Электронная поставка	KZT	64 800	32 400	29 160
460154610462 5	1С:Предприятие 8. Бухгалтерский учет для государственных предприятий Казахстана	KZT	64 800	32 400	29 160
460154607822 3	1С:Предприятие 8. Бухгалтерский учет для государственных	KZT	76 320	38 160	34 344

	предприятий Казахстана (USB)				
2900002163612	1С:Предприятие 8. Свод отчетов для Казахстана. Электронная поставка	KZT	540 000	270 000	243 000
4601546104649	1С:Предприятие 8. Свод отчетов для Казахстана	KZT	540 000	270 000	243 000
4601546097828	1С:Предприятие 8. Свод отчетов для Казахстана (USB)	KZT	549 600	274 800	247 320
2900002163537	1С:Предприятие 8. Управление торговлей для Казахстана. Базовая версия. Электронная поставка	KZT	30 240	15 120	13 608

Продолжение таблицы 1.1

4601546051561	1С:Предприятие 8. Управление торговлей для Казахстана. Базовая версия. *	KZT	30 240	15 120	13 608
2900002163520	1С:Предприятие 8. Управление торговлей для Казахстана. Электронная поставка	KZT	90 720	45 360	40 824
4601546095848	1С:Предприятие 8. Управление торговлей для Казахстана	KZT	90 720	45 360	40 824
4601546045447	1С:Предприятие 8. Управление торговлей для Казахстана (USB)	KZT	102 240	51 120	46 008
2900002163551	1С:Предприятие 8. Розница для Казахстана. Базовая версия. Электронная поставка	KZT	21 600	10 800	9 720
4601546129871	1С:Предприятие 8. Розница для Казахстана. Базовая версия	KZT	21 600	10 800	9 720
2900002163544	1С:Предприятие 8. Розница для Казахстана. Электронная поставка	KZT	64 800	32 400	29 160
4601546095862	1С:Предприятие 8. Розница для Казахстана	KZT	64 800	32 400	29 160
4601546066657	1С:Предприятие 8. Розница для КЗ USB	KZT	76 320	38 160	34 344
2900002163575	1С:Предприятие.8. Управление нашей фирмой для Казахстана. Базовая версия. Электронная поставка	KZT	30 240	15 120	13 608
4601546130099	1С:Предприятие 8. Управление нашей фирмой для Казахстана. Базовая версия *	KZT	30 240	15 120	13 608
2900002163568	1С:Предприятие 8. Управление нашей фирмой для Казахстана. Электронная поставка	KZT	90 720	45 360	40 824
460154608721	1С:Предприятие 8. Управление	KZT	90 720	45 360	40 824

8	нашей фирмой для Казахстана				
2900002163629	1С:Предприятие 8. Зарплата и Управление Персоналом для Казахстана. Электронная поставка	KZT	90 720	45 360	40 824
4601546095831	1С:Предприятие 8. Зарплата и Управление персоналом для Казахстана	KZT	90 720	45 360	40 824
4601546045454	1С:Предприятие 8. Зарплата и Управление персоналом для Казахстана (USB)	KZT	102 240	51 120	46 008
4601546104663	1С:Предприятие 8. Зарплата и Управление Персоналом КОРП для КЗ	KZT	504 000	252 000	226 800
2900002163605	1С:Предприятие 8. Зарплата и кадры для государственных организаций Казахстана. Электронная поставка	KZT	90 720	45 360	40 824

Продолжение таблицы 1.1

4601546104632	1С:Предприятие 8. Зарплата и кадры для государственных организаций КЗ	KZT	90 720	45 360	40 824
4601546078230	1С:Предприятие 8. Зарплата и кадры для государственных организаций КЗ	KZT	102 240	51 120	46 008
4601546095855	1С:Предприятие 8. Комплект прикладных решений на 5 пользователей для Казахстана	KZT	237 600	118 800	106 920
4601546045461	1С:Предприятие 8. Комплект прикладных решений на 5 пользователей для Казахстана (USB)	KZT	249 120	124 560	112 104
4601546127969	1С:Комплексная автоматизация 8 для Казахстана. Редакция 2 *	KZT	420 000	210 000	189 000
4601546127976	1С:Комплексная автоматизация 8 для Казахстана. Редакция 2 (USB) *	KZT	504 000	252 000	226 800
4601546127983	1С:Комплексная автоматизация 8 для Казахстана на 10 пользователей + клиент-сервер. Редакция 2 *	KZT	750 000	375 000	337 500
4601546127990	1С:Комплексная автоматизация 8 для Казахстана на 10 пользователей + клиент-сервер. Редакция 2 (USB) *	KZT	900 000	450 000	405 000
2900001756587	Конфигурация "Модуль "Бюджетирование" для 1С:КА 8 для Казахстана	KZT	350 000	175 000	157 500
460154612933	1С:Предприятие 8. Управление	KZT	5 940 000	2 970	2 673 000

8	холдингом для Казахстана *			000	
290000175455 2	1С:Предприятие 8. Управление холдингом для Казахстана. NFR *	KZT			100 000
290000216393 3	1С:Предприятие 8. ERP Управление предприятием 2 для Казахстана. Электронная поставка *	KZT	2 500 000	1 250 000	1 125 000
460154612700 6	1С:Предприятие 8. ERP Управление предприятием 2 для Казахстана *	KZT	2 500 000	1 250 000	1 125 000
460154613522 3	1С:ERP Управление предприятием 2 для Казахстана. Корпоративная поставка *	KZT	16 000 000	8 000 000	7 200 000
460154609589 3	1С:Предприятие 8. Управление производственным предприятием для Казахстана	KZT	792 000	396 000	356 400
460154604146 3	1С:Предприятие 8. Управление производственным предприятием для Казахстана (USB)	KZT	810 000	405 000	364 500
460154609590 9	1С:Предприятие 8. Управление производственным предприятием для Казахстана для 10 пользователей + клиент-сервер	KZT	1 080 000	540 000	486 000

Продолжение таблицы 1.1

460154604147 0	1С:Предприятие 8. Управление производственным предприятием для Казахстана для 10 пользователей + клиент-сервер (USB)	KZT	1 098 000	549 000	494 100
460154612080 9	1С:Предприятие 8. Документооборот КОПП для Казахстана	KZT	960 000	480 000	432 000
290000069590 0	1С:Предприятие 8 для Казахстана. NFR для 1 пользователя	KZT			17 100
290000069591 7	1С:Предприятие 8 для Казахстана. NFR для 10 пользователей + клиент-сервер	KZT			56 700
290000099583 3	1С:Предприятие 8. NFR Управление производственным предприятием для Казахстана	KZT			17 100
290000120464 4	1С:Предприятие 8. Дополнение к NFR для Казахстана. Гос.	KZT			3 240
290000173116 4	1С:Предприятие 8. ERP Управление предприятием 2 для Казахстана. NFR *	KZT			33 000
460154613255 0	1С:Предприятие 8. Комплект для обучения в высших и средних учебных заведениях Казахстана *	KZT		72 000	64 800

460154605550 7	1С:Предприятие 8. Комплект специалиста по разработке и внедрению для Казахстана *	KZT	17 820		17 100
-------------------	---	-----	--------	--	--------

1.3 Решение проблем

Важное преимущество создания собственной информационной системы является ее полная адаптивность к деятельности предприятия. Есть множество готовых систем контроля, но они являются слишком универсальными. Из-за всего этого, система не может предложить необходимые средства для решения ваших проблем. В готовых системах довольно много лишних функций, в которых нет необходимости.

Готовые системы не смогут помочь, когда вы хотите решить конкретную задачу. Собственные информационные системы создаются конкретно под требования предприятия, которые учитывают специфику деятельности и являются полностью адаптивными. Кроме того, в будущем систему можно оптимизировать - дополнять и изменять под возникающие потребности.

Главные преимущества проектирования информационной системы:

- проектирование системы контроля позволяет присоединить разрозненную информацию в одну систему, в которой есть возможность анализировать данные, изменять в отчеты и быстро реагировать на отклонения на рынке;
- система, которая создается под заказ, включает в себя цели, необходимые заказчику, и нет необходимости переплачивать за ненужный функционал;
- корректно произведенная информационная система дает возможность оптимизировать процессы и быстро уменьшить временные, а также финансовые расходы;
- есть возможность использовать информационные системы со смартфонов, ноутбуков и планшетов;
- информация, вносимая в системы, хранится на сервере, поэтому есть возможность получения с любой точки мира и устройства.

В данный момент предприятиям вести бизнес без использования своей информационной системы является трудной задачей. Информационная система является организационно - упорядоченной взаимосвязанной совокупностью средств, а также методов информационных технологий, которые используются для хранения, обработки и выдачи информации для достижения поставленной цели. В данный момент информационная система используется на персональном компьютере в качестве главного технического средства переработки информации.

Информационные системы дают возможность сбора, хранения, обработку, поиск, а также выдачу информации, которая необходима в процессе принятия решений задач из любой области. ИС дают возможность

анализировать проблемы и производить новые продукты. Однако стоит отметить, что создание информационной системы не имеет значение, когда не будет учтена позиция человека, для которого создана информация в системе.

Можно выделить следующие преимущества от внедрения информационных систем на предприятия:

- улучшение качества;
- увеличение эффективности;
- улучшение процесса принятия решения;
- улучшение коммуникации;
- улучшение использования знаний.

ИС дает возможность улучшить товары или услуги, при этом, которая повышает качество, а также уменьшает издержки, при этом добавляя желательные свойства. ИС главным образом используется для быстрого обслуживания, а также для быстрого поиска эксперта, который имеет опыт и знания, необходимые в определенном случае.

ИС дает возможность увеличить эффективность и производительность. К примеру, в организации ИС может управлять оборудованием и мгновенно извещать оператора при возникновении неполадок. В результате есть возможность получить продукцию более высокого качества с меньшим количеством брака.

ИС дает возможность использовать своевременную и надежную информацию, которая позволяет модернизировать процесс принятия решений. Собранная информация о продажах оптовым поставщиком, дает возможность своевременно обнаружить спад в продаже отдельных товаров, которая дает возможность выяснить причины и принять меры.

ИС дает возможность модернизировать коммуникации. Персонал может использовать компьютеры для входа в сеть своего предприятия, когда вы в командировках, дома или просто в другом здании, чтобы отправлять и получать сообщения, просматривать файлы данных предприятия, исследовать проблемы, готовить презентации.

ИС дает возможность модернизировать использование знаний. К примеру, консалтинговые фирмы дает возможность обеспечивать возможность своим клиентам планировать налоги, которые использует экспертные системы по налогообложению, где сконцентрированы знания лучших экспертов фирмы.

Любому предприятию необходимо по возможности иметь несколько локальных ИС для разных предназначений, которые могут взаимодействовать между собой и, которые могут поддерживают контрольные решения на всех уровнях.

Между локальными ИС образуются связи разного характера и назначения. В первом случае, определенные системы имеют возможность иметь связь с огромным количеством работающих в предприятии систем, а

также иметь выход во внешнюю среду, во втором случае, имеют возможность иметь связь только с одной или несколькими подобными.

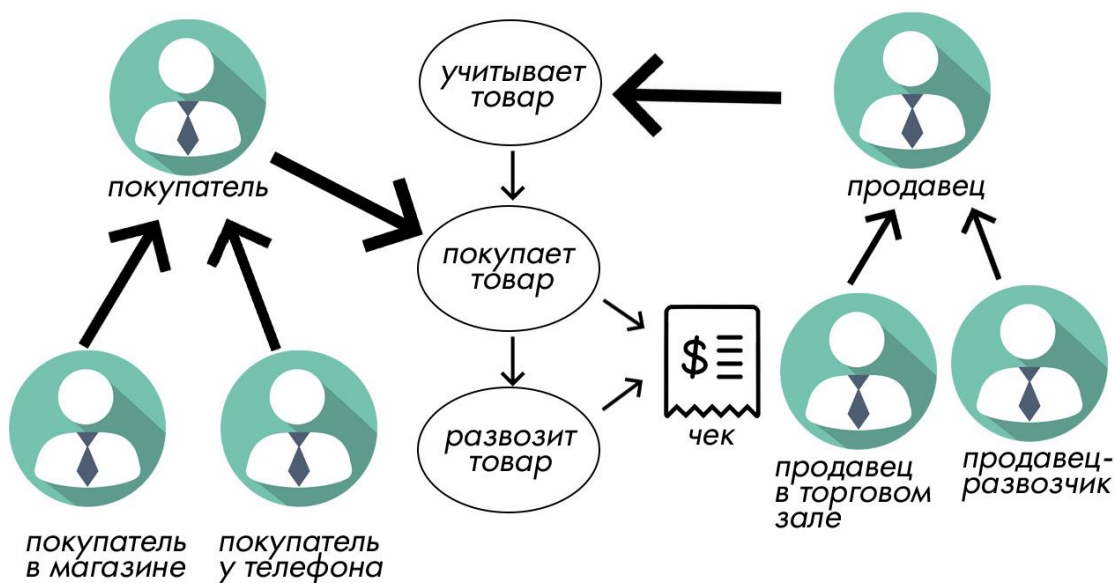


Рисунок 1.1 - Управленческие решения на разных уровнях

Подведя итог, мы имеем, что использование информационной системы на предприятии во многом демонстрирует будущее развитие. Информационные системы дают возможность улучшить и рационализировать систему контроля быстрой деятельности предприятия, а также обеспечивают возможность контроля взаимоотношениями с клиентами и дилерами, контролировать процесс продаж. В итоге мы имеем, что внедрения на предприятии производит сокращение операционных издержек, получение дополнительных доходов вследствие увеличения оборота и роста инвестиционной привлекательности компании.

1.4 Постановка задачи

В данном проекте были выдвинуты задачи разработать информационную систему управления учета заказов, пользователей, а заказ услуг через приложение.

Программа должна облегчать работу организации и полностью ее автоматизировать, а также ускорить весь процесс принятия решений.

Программа имеет обширный функционал. Данная программа очень полезна для предприятия и будет использована полностью.

До разработки информационной системы были поставлены следующие задачи:

- сбор исходных материалов;
- эскизный проект;

- проектирование информационной системы;
- технический проект;
- разработка информационной системы;
- рабочий проект;
- внедрение.

2 Описание и обоснование выбора языка программирования и СУБД

2.1 Проектирование информационной системы

На рисунке 2.1 продемонстрирована диаграмма с возможностями для пользователя

Информационная система позволяет использовать следующие функции пользователя:

- добавление автозапчастей (необходимо заполнить ячейки в самой информационной системе);
- заказ автозапчастей;
- учет автозапчастей (имеется возможность учета автозапчастей);
- просмотр заказов (просмотр заказов в самой программе);
- управление пользователями (имеется возможность контролировать пользователей).

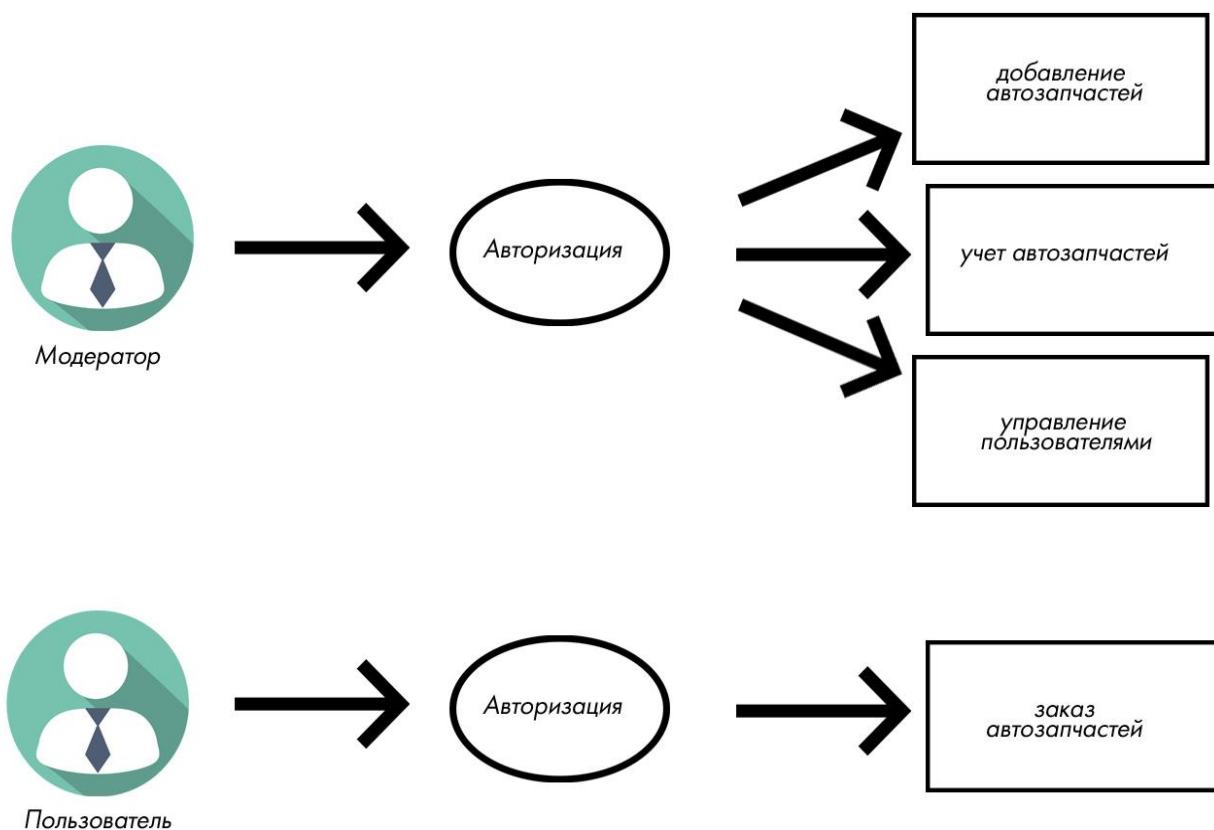


Рисунок 2.1 – Диаграмма возможностей для пользователя и модератора

На рисунках 2.2 и 2.3 представлена физическая модель и схема базы данных. Рисунок содержит детали физической модели базы данных.

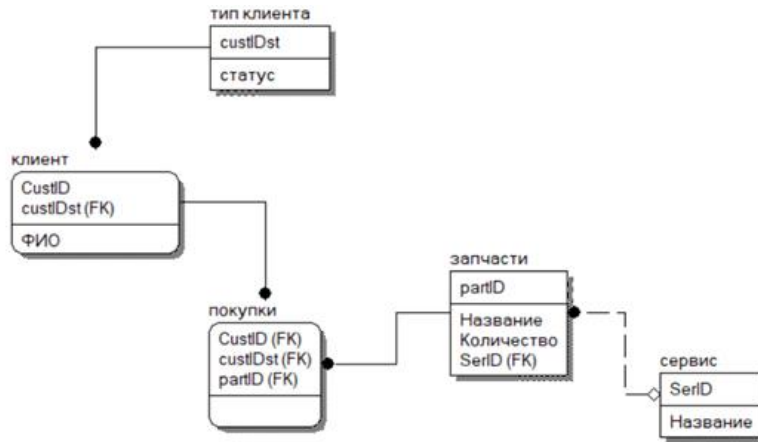


Рисунок 2.2 – Физическая модель базы данных



Рисунок 2.3 – Схема базы данных

На рисунке 2.4 представлена диаграмма последовательности работы информационной системы.

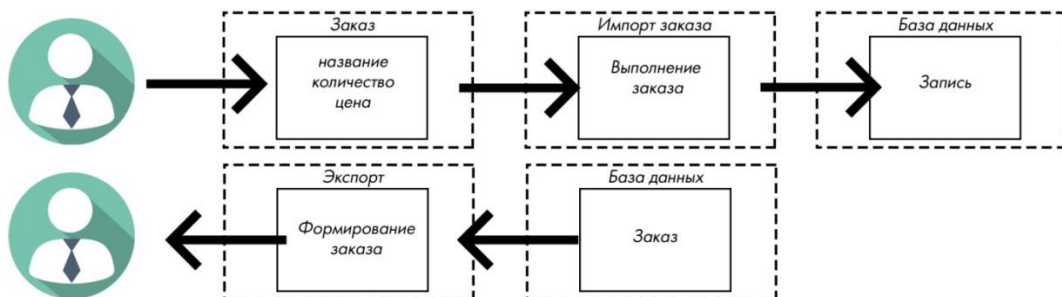


Рисунок 2.4 – Диаграмма последовательности

Ниже приведена блок-схема работы данной программы:

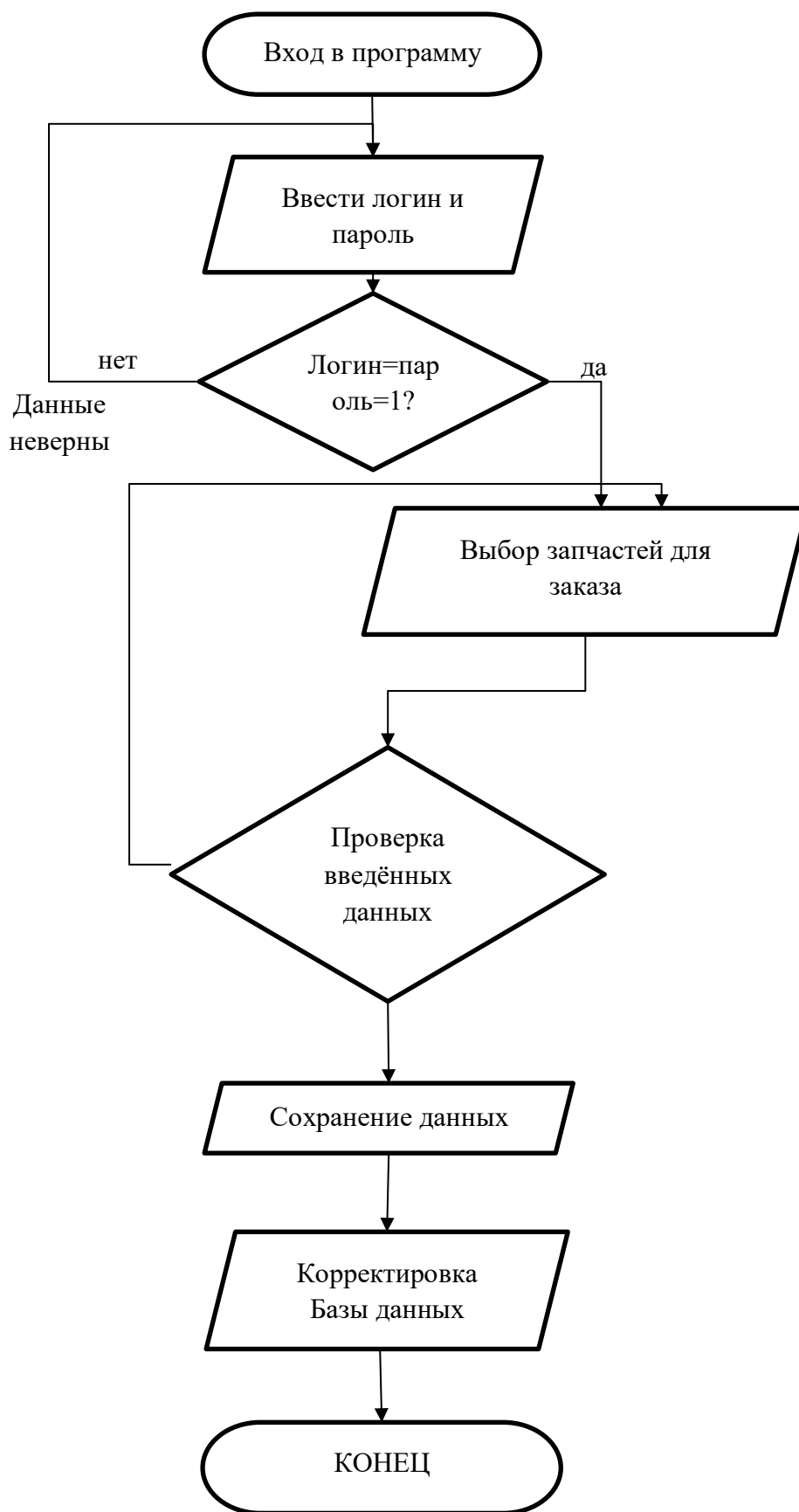


Рисунок 2.5

2.2 Анализ современных языков программирования

Интегрированная среда разработки является комплексное средство, которая включает в себя необходимое для программиста, чтобы создать программное обеспечение.

Существует набор компонентов, которые должны присутствовать в интегрированных средах разработки:

- компилятор;
- редактор исходного кода программ;
- отладчик.

В данный момент интегрированные среды разработки дают возможность программистам использовать больше возможностей. К примеру, большинство IDE являются визуальными. IDE дают возможность создавать интерфейс программы точно в таком же виде, в котором он будет представлен потом пользователю. IDE, не являющиеся визуальными, заставляют программиста вручную писать код, который будет строить будущий интерфейс.

Имеется возможность писать программы на платформах, на которых работает сама IDE. Среда разработки делятся на кросс-платформенные или платформенно. К примеру, кросс-платформенной средой разработки является Eclipse, а платформенно-зависимой - Delphi.

Рассмотрим плюсы и минусы разных языков программирования.

C++, C плюсы:

- возможность полного контроля;
- полный контроль над памятью;
- вы лучше понимаете все ошибки во время компиляции.

C++, C минусы:

- требуется заранее оптимизировать;
- бедная библиотека.

JAVA, C# плюсы:

- можно не управлять памятью;
- богатая библиотека;
- компилируется в байт-код.

JAVA, C# минусы:

- сложность выявления ошибки.

2.3 Выбор языка программирования C#

C# создан в период с 1993 по 2001 годы группой разработчиков Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота. Также была выпущена новая платформа .NET. В этой платформе соединялись сразу несколько языков программирования. Главным нововведением платформы была технология ASP.NET. Это давало возможность быстрого создания веб-приложения, которые

взаимодействовали с базой данных. Для данной платформы был разработан С#. Стоит отметить, что платформа была разработана под этим языком. С# является языком близким по синтаксису с С. Данный язык ближе к С++ и Java. С# с самого истока был предназначен для веб-разработки. Объектно-ориентированный подход дает возможность создавать с помощью С# большие, но в то же время универсальные, масштабируемые и расширяемые приложения.

С# поддерживает множество используемых функций:

- инкапсуляция,
- наследование,
- полиморфизм,
- перегрузка операторов,
- статическая типизация.

Язык всё ещё активно развивается. Относительно других языков С# является молодым языком

У языка С# выделяют много преимуществ:

- поддержка подавляющего большинства продуктов Microsoft;
- бесплатность ряда;
- типы данных имеют фиксированный размер;
- автоматическая “сборка мусора”;
- большое количество синтаксисов;
- низкий порог вхождения.

Инструментарий С# дает возможность решать широкий круг целей. Язык является достаточно мощным и универсальным. На нём часто разрабатывают разные приложения под операционную систему Windows.

Список возможностей разработки не имеет ограничений, из-за широчайшего набора инструментов и средств. Хотя всё это можно реализовать при помощи других языков, но некоторые из них узкоспециализированные, а в некоторых придётся использовать дополнительные инструменты сторонних разработчиков. В С# решить широкий круг задач возможно быстрее, проще и с меньшими затратами времени и ресурсов.

2.4 Особенность использования СУБД

Базы данных является хранилищем, которая содержит огромные массивы данных. Имеется возможность отправления запросов, получение информации при помощи СУБД. В данном проекте используется СУБД, которая была разработана компанией Microsoft под названием SQL Server. Данная программа доступна в разных версиях, которые имеют свою особенность. Замена этой СУБД – система MySQL с открытым программным кодом.

Преимущества SQL Server:

- СУБД увеличивается, поэтому работать с ней на разных платформах;

- объем страниц до 8 кб, отсюда следует, что данные экспортируются быстро;
- рутинные административные задачи автоматизированы;
- реализован поиск по фразам, тексту, словам, можно создавать ключевые индексы;
- в SQL Server имеет возможность репликации через интернет, предусмотрена синхронизация;
- в систему внедрен сервер интерактивного анализа для принятия решений, создания корпоративных отчетов;
- есть возможность формирования запросов на английском языке;
- СУБД имеет поддержку работы с Access, MS Excel.

2.5 Описание структур таблиц базы данных

Структура базы данных состоит из пяти соединенных разными видами связи. В основном используется связь идентифицируемая и неидентифицируемая.

В данной структуре имеются таблицы:

- тип клиента;
- клиент;
- покупки;
- запчасти;
- сервис.

Таблица “тип клиента” связана с таблицей “клиент” связью идентифицируемой и в таблице “клиент” присутствует первичный ключ из таблицы “тип клиента” – custIDst.

Таблицы “клиент” и “запчасти” связаны с таблицей “покупки” связью идентифицируемой и в таблице “покупки” присутствуют первичные ключи из таблицы “клиента” и “запчасти” – custIDst, CustID, partID.

Таблица “запчасти” связана с таблицей “сервис” связью неидентифицируемой.

Таблица “клиент” представляет собой данные ФИО клиента.

Таблица “покупки” представляет собой данные заказов клиента.

Таблица “запчасти” представляют собой данные запчастей, которые имеются в наличии.

Таблица “сервис” представляет собой данные сервиса, которые продают данные запчасти.

3 Разработка информационной системы

3.1 Программная реализация

Область применения: Проект будет выпущен на платформу Windows и будет использоваться в компании.

В данном проекте использовалось множество библиотек языка программирования С#, так как этот язык программирования использовался при написании программы.

Назначение программы: Обеспечение взаимодействия между клиентами автосервиса и базой данных автосервиса, а также полное администрирование этой базы данных.

Для оформления заказа в приложении необходимо нажать на кнопку «Заказать автозапчасти». После её нажатия откроется меню заказа (рисунок 3.1).

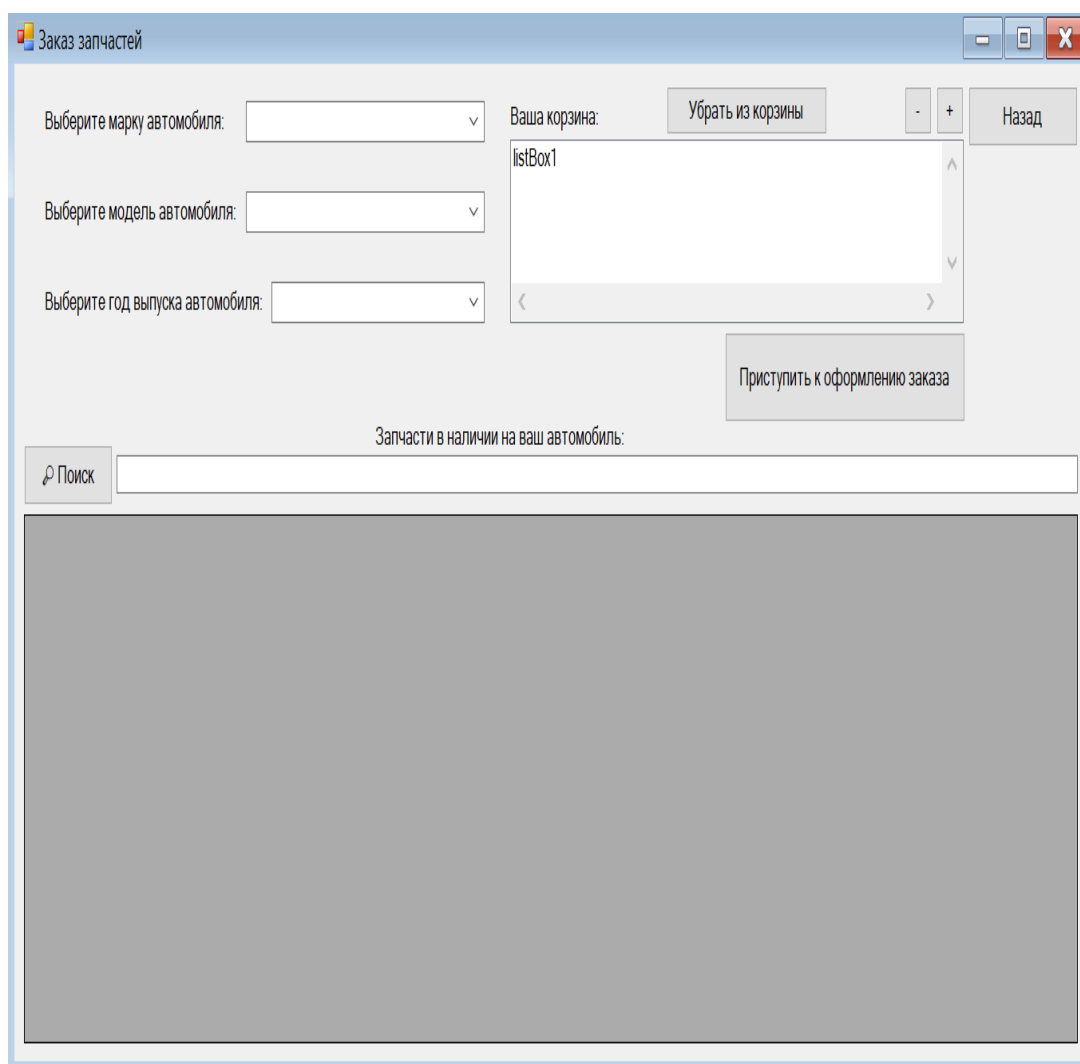


Рисунок 3.1 – Интерфейс программы

При открытии данного меню выполняется следующий код:

```
private void MakeOrder_Load(object sender, EventArgs e)
{
    SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapchasti.mdf;Integrated Security=True");
    string sql = "SELECT * FROM SpareParts";
    SqlCommand sCom = new SqlCommand(sql, connection);
    connection.Open();

    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            if (!markBox.Items.Contains(read["Марка_автомобиля"].ToString().Replace(" ", "")))
            {
                markBox.Items.Add(read["Марка_автомобиля"].ToString().Replace(" ", ""));
            }
        }
    }
    connection.Close();
}
```

Рисунок 3.2 – MakeOrder

С помощью данного кода программа берёт из таблицы автозапчастей данные всех марок автомобилей, содержащихся в таблице, а затем помещает их в первый comboBox (выпадающее меню) на форме.

После того, как клиент выбрал марку автомобиля – активируется следующее выпадающее меню, в которое из той же таблицы добавляются данные о моделях автомобиля выбранной марки:

```
private void markBox_SelectedIndexChanged(object sender, EventArgs e)
{
    modelBox.Enabled = true;
    modelBox.Items.Clear();
    modelBox.Text = "";
    yearBox.Items.Clear();
    yearBox.Text = "";
    SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapchasti.mdf;Integrated Security=True");
    string sql = "SELECT * FROM SpareParts WHERE Марка_автомобиля = " + markBox.Text + "";
    SqlCommand sCom = new SqlCommand(sql, connection);
    connection.Open();

    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            if (!modelBox.Items.Contains(read["Модель_автомобиля"].ToString().Replace(" ", "")))
            {
                modelBox.Items.Add(read["Модель_автомобиля"].ToString().Replace(" ", ""));
            }
        }
    }
    connection.Close();
}
```

Рисунок 3.3 – markBox

После выбора модели - открывается доступ к следующему выпадающему меню, в которое из таблицы автозапчастей добавляются данные о годах выпуска выбранного автомобиля:

```
private void modelBox_SelectedIndexChanged(object sender, EventArgs e)
{
    yearBox.Enabled = true;
    yearBox.Items.Clear();
    yearBox.Text = "";
    SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapchasti.mdf;Integrated Security=True");
    string sql = "SELECT * FROM SpareParts WHERE Модель_автомобиля = " + modelBox.Text + "";
    SqlCommand sCom = new SqlCommand(sql, connection);
    connection.Open();

    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            if (!yearBox.Items.Contains(read["Год_автомобиля"].ToString().Replace(" ", "")))
            {
                yearBox.Items.Add(read["Год_автомобиля"].ToString().Replace(" ", ""));
            }
        }
    }
    connection.Close();
}
```

Рисунок 3.4 – modelBox

Когда все выпадающие меню выбраны, открывается доступ к корзине и данные, на основе выбранных пунктов, выбираются из таблицы автозапчастей и добавляются в dataGridView (таблицу) снизу формы:

```
private void yearBox_SelectedIndexChanged(object sender, EventArgs e)
{
    button2.Enabled = true;
    loadData();
}

private void loadData()
{
    if (dataGridView1.RowCount <= 0)
    {
        dataGridView1.Columns.Add(new DataGridViewCheckBoxColumn());
        dataGridView1.Columns.Add(chk);
        chk.ValueType = typeof(bool);
        chk.HeaderText = "В корзину";
        chk.Name = "chk";
    }

    string connectionString = "Data Source=(localdb)\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapchasti.mdf;Integrated Security=True";
    string sql = "SELECT Назва, Тип_запчасти, Цена, Количество, Идентификационный_номер, Дата_добавления, Поставщик, Принимающий, Марка_автомобиля, Модель_автомобиля, Год";
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    SqlCommand cmd = new SqlCommand(sql, connection);
    SqlDataAdapter sda = new SqlDataAdapter(cmd);
    SqlCommandBuilder sdb = new SqlCommandBuilder(sda);
    DataSet ds = new DataSet();
    sda.Fill(ds, "SpareParts");
    DataTable dt = ds.Tables["SpareParts"];
    connection.Close();
    dataGridView1.DataSource = ds.Tables["SpareParts"];
    dataGridView1.ReadOnly = true;
    dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
    dataGridView1.Columns[12].AutoSizeMode = DataGridViewAutoSizeColumnMode.ColumnHeader;
    for (int i = 0; i < dataGridView1.Columns.Count; i++)
    {
        if (dataGridView1.Columns[i] is DataGridViewImageColumn)
        {
            ((DataGridViewImageColumn) dataGridView1.Columns[i]).ImageLayout = DataGridViewImageCellLayout.Zoom;
        }
    }

    int KolVo = 1;
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        for (int j = 0; j < listBox1.Items.Count; j++)
        {
            if (listBox1.Items[j].ToString().Contains(dataGridView1.Rows[i].Cells[5].Value.ToString().Trim()))
            {
                KolVo = Convert.ToInt32(listBox1.Items[j].ToString().Substring(listBox1.Items[j].ToString().LastIndexOf("-")).Remove(0, 2));
            }
            if (listBox1.Items.Contains(dataGridView1.Rows[i].Cells[5].Value.ToString().Trim() + " " + dataGridView1.Rows[i].Cells[1].Value.ToString().Trim() + " на ")
            {
                DataGridViewCheckBoxCell checkCell = (DataGridViewCheckBoxCell) dataGridView1.Rows[i].Cells[0];
                checkCell.Value = true;
            }
            else
            {
                DataGridViewCheckBoxCell checkCell = (DataGridViewCheckBoxCell) dataGridView1.Rows[i].Cells[0];
                checkCell.Value = false;
            }
        }
    }
}

```

Рисунок 3.5 – YearBox

В каждой строке полученной таблицы с автозапчастями в наличии, находится checkBox (поле для галочки). При нажатии, выбранные запчасти добавляется в корзину в количестве одной штуки:

```
private void dataGridView1_CellValueChanged(object sender, DataGridViewCellEventArgs e)
{
    int KolVo = 1;
    if (dataGridView1.RowCount > 0 && e.ColumnIndex == 0)
    {
        for (int i = 0; i < listBox1.Items.Count; i++)
        {
            if (listBox1.Items[i].ToString().Contains(dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString().Trim()))
            {
                KolVo = Convert.ToInt32(listBox1.Items[i].ToString().Substring(listBox1.Items[i].ToString().LastIndexOf("-")).Remove(0, 2));
            }
        }
        string row = dataGridView1.Rows[e.RowIndex].Cells[5].Value.ToString().Trim() + " " + dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString().Trim() + " на ";
        DataGridViewCheckBoxCell checkCell = (DataGridViewCheckBoxCell) dataGridView1.Rows[e.RowIndex].Cells[0];
        if ((Boolean) checkCell.Value == true)
        {
            if (!listBox1.Items.Contains(row))
            {
                listBox1.Items.Add(row);
            }
        }
        else
        {
            if (listBox1.Items.Contains(row))
            {
                listBox1.Items.Remove(row);
            }
        }
    }
    dataGridView1.ReadOnly = true;
    //dataGridView1.Invalidate();
}

```

Рисунок 3.6 – заполнение таблицы

Количество покупаемых автозапчастей можно редактировать через меню корзины (кнопками «+» и «-»), выделив товар и нажимая кнопки. Количество товара не может превышать доступное количество на складе (программа предупредит пользователя, если это так), и не может быть меньше одного:

```
private void button3_Click(object sender, EventArgs e)
{
    if (listBox1.SelectedItem != null)
    {
        string withoutKolVo = listBox1.SelectedItem.ToString().Remove(listBox1.SelectedItem.ToString().LastIndexOf(listBox1.SelectedItem.ToString()));
        int KolVo = Convert.ToInt32(listBox1.SelectedItem.ToString().Substring(listBox1.SelectedItem.ToString().LastIndexOf("-")).Remove(0, 2));
        int index = listBox1.SelectedIndex;

        int maxKolVo = 0;
        SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Za");
        string sql = "SELECT Количество FROM SpareParts WHERE Идентификационный_номер = " + listBox1.SelectedItem.ToString().Split(' ')[0] + " ";
        SqlCommand sCom = new SqlCommand(sql, connection);
        connection.Open();

        using (SqlDataReader read = sCom.ExecuteReader())
        {
            while (read.Read())
            {
                maxKolVo = Convert.ToInt32(read["Количество"].ToString());
            }
        }
        connection.Close();

        int nextKolVo = KolVo + 1;
        if (nextKolVo <= maxKolVo)
        {
            listBox1.Items.RemoveAt(index);
            listBox1.Items.Insert(index, withoutKolVo + nextKolVo);
            listBox1.SelectedItem = listBox1.Items[index];
        }
        else
        {
            MessageBox.Show("Вы не можете заказать больше запчастей, чем есть на складе!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}

private void button4_Click(object sender, EventArgs e)
{
    if (listBox1.SelectedItem != null)
    {
        string withoutKolVo = listBox1.SelectedItem.ToString().Remove(listBox1.SelectedItem.ToString().LastIndexOf(listBox1.SelectedItem.ToString()));
        int KolVo = Convert.ToInt32(listBox1.SelectedItem.ToString().Substring(listBox1.SelectedItem.ToString().LastIndexOf("-")).Remove(0, 2));
        int index = listBox1.SelectedIndex;

        int nextKolVo = KolVo - 1;
        if (nextKolVo > 0)
        {
            listBox1.Items.RemoveAt(index);
            listBox1.Items.Insert(index, withoutKolVo + nextKolVo);
            listBox1.SelectedItem = listBox1.Items[index];
        }
        else
        {
            MessageBox.Show("Вы не можете заказать менее одной запчасти!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
        }
    }
}
}
```

Рисунок 3.7 – Редактирование корзины

Если же товар в корзину был добавлен случайно – то можно удалить, выделив в корзине и нажав на кнопку «Убрать из корзины»:

```
private void button5_Click(object sender, EventArgs e)
{
    if (listBox1.SelectedItem != null)
    {
        int index = listBox1.SelectedIndex;
        listBox1.Items.RemoveAt(index);
    }
    else
    {
        MessageBox.Show("Выделите элемент!", "Подсказка!", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
}
```

Рисунок 3.8 – Удаление из корзины

На форме заказа также присутствует функция поиска по товарам в таблице:

```
private void button7_Click(object sender, EventArgs e)
{
    for (int i = 0; i < dataGridView1.RowCount; i++)
    {
        dataGridView1.Rows[i].Selected = false;
        for (int j = 0; j < dataGridView1.ColumnCount; j++)
            if (dataGridView1.Rows[i].Cells[j].Value != null)
                if (dataGridView1.Rows[i].Cells[j].Value.ToString().Contains(textBox1.Text))
                {
                    dataGridView1.Rows[i].Selected = true;
                    break;
                }
    }
}
```

Рисунок 3.9 – Поиск

После добавления всех нужных запчастей в корзину и выбора их количества следует нажать кнопку «Приступить к оформлению заказа»:

```
private void button2_Click(object sender, EventArgs e)
{
    if(listBox1.Items.Count > 0)
    {
        zakaz.Clear();
        for (int i = 0; i < listBox1.Items.Count; i++)
        {
            zakaz.Add(listBox1.Items[i].ToString());
        }

        LastOrder LastOrder = new LastOrder();
        LastOrder.ShowDialog();
        LoadData();
    }
    else
    {
        MessageBox.Show("В вашей корзине ничего нет!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}
```

Рисунок 3.10 – Оформление заказа

Откроется форма оформления заказа, в которой будет продублирован список товаров из корзины для перепроверки:

```
public LastOrder()
{
    this.StartPosition = FormStartPosition.CenterScreen;
    InitializeComponent();

    for(int i = 0; i < MakeOrder.zakaz.Count; i++)
    {
        listBox1.Items.Add(MakeOrder.zakaz[i]);

        int defaultPrice = 0;
        int KolVo = Convert.ToInt32(listBox1.Items[i].ToString().Substring(listBox1.Items[i].ToString().LastIndexOf("-").Remove(0, 2)));

        SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Data\\SpareParts.mdf");
        string sql = "SELECT Цена FROM SpareParts WHERE Идентификационный_номер = " + listBox1.Items[i].ToString().Split(' ')[0] + " ";
        SqlCommand sCom = new SqlCommand(sql, connection);
        connection.Open();

        using (SqlDataReader read = sCom.ExecuteReader())
        {
            while (read.Read())
            {
                defaultPrice = Convert.ToInt32(read["Цена"].ToString());
            }
        }

        connection.Close();

        Sum += defaultPrice * KolVo;
    }
    listBox1.Items.Add("Общая сумма = " + Sum + " тенге.");
}
```

Рисунок 3.11 – LastOrder

Пользователь должен ввести все требуемые данные в поля формы оформления заказа, иначе программа выведет ошибку:

```
if (!String.IsNullOrWhiteSpace(textBox1.Text) && !String.IsNullOrWhiteSpace(textBox2.Text) && !String.IsNullOrWhiteSpace(textBox3.Text))
{
    // ...
}
else
{
    MessageBox.Show("Не все данные заполнены верно!", "Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

Рисунок 3.12 – Условия для успешной работы программы

После нажатия кнопки «Оформить», программа добавляет введенные пользователем данные в базу данных в таблицу с клиентами:

```
//Заполнение таблицы покупателей
string order = "";
for (int i = 0; i < listBox1.Items.Count; i++)
{
    if (i != listBox1.Items.Count - 1)
    {
        order += listBox1.Items[i].ToString() + "\n";
    }
    else
    {
        order += listBox1.Items[i].ToString();
    }
}

SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\MSS
string sql = "SELECT * FROM Clients";
SqlCommand sCom = new SqlCommand(sql, connection);
int max = 0;
connection.Open();
using (SqlDataReader read = sCom.ExecuteReader())
{
    while (read.Read())
    {
        if (int.Parse(read["Id"].ToString()) > max)
        {
            max = int.Parse(read["Id"].ToString());
        }
    }
}
max++;
connection.Close();

string connectionString = "Data Source=(LocalDB)\MSSQLLocalDB;AttachDbF
connection = new SqlConnection(connectionString);
sCommand = new SqlCommand();
sCommand.Connection = connection;
connection.Open();
sCommand.CommandType = CommandType.Text;
sCommand.CommandText = "INSERT INTO Clients(Id,Имя,Фамилия,Отчество,Элек
sCommand.Parameters.AddWithValue("@Id", max);
sCommand.Parameters.AddWithValue("@Имя", textBox1.Text);
sCommand.Parameters.AddWithValue("@Фамилия", textBox2.Text);
sCommand.Parameters.AddWithValue("@Отчество", textBox3.Text);
sCommand.Parameters.AddWithValue("@Электронная_почта", textBox4.Text);
sCommand.Parameters.AddWithValue("@Телефон", textBox5.Text);
sCommand.Parameters.AddWithValue("@Адрес", textBox6.Text);
sCommand.Parameters.AddWithValue("@Заказ", order);
sCommand.ExecuteNonQuery();
connection.Close();
```

Рисунок 3.13 – Работа с базой данных

Программа также удаляет из таблицы автозапчастей количество купленных запчастей. Если количество купленных запчастей было равно максимальному количеству этой автозапчасти на складе, то программа полностью удалит эту автозапчасть из базы данных:

```
//Удаление запчастей
for (int i = 0; i < listBox1.Items.Count - 1; i++)
{
    int kolVo = Convert.ToInt32(listBox1.Items[i].ToString().Substring(listBox1.Items[i].ToString().LastIndexOf("-")).Remove(0, 2));
    int maxKolVo = 0;
    connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapcha
    sql = "SELECT Количество FROM SpareParts WHERE Идентификационный_номер = '" + listBox1.Items[i].ToString().Split(' ')[0] + "'";
    sCom = new SqlCommand(sql, connection);
    connection.Open();
    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            maxKolVo = Convert.ToInt32(read["Количество"].ToString());
        }
    }
    connection.Close();

    if (kolVo < maxKolVo)
    {
        int newKolVo = maxKolVo - kolVo;
        connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapchasti.mdf;
        connection = new SqlConnection(connectionString);
        sCommand = new SqlCommand();
        sCommand.Connection = connection;
        connection.Open();
        sCommand.CommandType = CommandType.Text;
        sCommand.CommandText = "UPDATE SpareParts SET Количество = @Количество WHERE Идентификационный_номер = '" + listBox1.Items[i]
        sCommand.Parameters.AddWithValue("@Количество", newKolVo);
        sCommand.ExecuteNonQuery();
        connection.Close();
    }
    else
    {
        connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Za
        sql = "DELETE FROM SpareParts WHERE Идентификационный_номер = '" + listBox1.Items[i].ToString().Split(' ')[0] + "'";
        sCom = new SqlCommand(sql, connection);
        connection.Open();
        sCom.ExecuteNonQuery();
        connection.Close();
    }
}

this.Close();
```

Рисунок 3.14 – Удаление лишнего значения

Данные о клиентах и их заказах сохраняются в разделе программы «Учёт автозапчастей».

При открытии формы учёта – программа заносит в таблицу внизу формы данные обо всех клиентах и их заказах:

```
public OrderList()
{
    this.StartPosition = FormStartPosition.CenterScreen;
    InitializeComponent();
    comboBox1.Items.Clear();

    LoadData();

    SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Data\\OrderList.mdf");
    string sql = "SELECT * FROM Clients";
    SqlCommand sCom = new SqlCommand(sql, connection);
    connection.Open();

    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            comboBox1.Items.Add(read["Id"].ToString() + ": " + read["Фамилия"].ToString().Replace(" ", ""))
        }
    }

    connection.Close();
}

Ссылка: 3
private void LoadData()
{
    string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Data\\OrderList.mdf";
    string sql = "SELECT * FROM Clients";
    SqlConnection connection = new SqlConnection(connectionString);
    connection.Open();
    SqlCommand sCommand = new SqlCommand(sql, connection);
    SqlDataAdapter sAdapter = new SqlDataAdapter(sCommand);
    SqlCommandBuilder sBuilder = new SqlCommandBuilder(sAdapter);
    DataSet sDs = new DataSet();
    sAdapter.Fill(sDs, "Clients");
    sTable = sDs.Tables["Clients"];
    connection.Close();
    dataGridView1.DataSource = sDs.Tables["Clients"];
    dataGridView1.ReadOnly = true;
    dataGridView1.SelectionMode = DataGridViewSelectionMode.FullRowSelect;
}
```

Рисунок 3.15 – Внесение данных в таблицу

Также доступен выбор клиента по ФИО из выпадающего меню. После выбора клиента все его данные выводятся на форму, а его заказ показывается в списке справа от формы:

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    listBox1.Items.Clear();
    SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Data\\OrderList.mdf");
    string sql = "SELECT * FROM Clients WHERE Id = " + comboBox1.Text.Split(':')[0];
    SqlCommand sCom = new SqlCommand(sql, connection);
    connection.Open();

    using (SqlDataReader read = sCom.ExecuteReader())
    {
        while (read.Read())
        {
            clientMail = read["Электронная почта"].ToString().Replace(" ", "");
            clientPhone = read["Телефон"].ToString().Replace(" ", "");
            clientAddress = read["Адрес"].ToString().Replace(" ", "");
            listBox1.Items.AddRange(read["Заказ"].ToString().Split('\n'));
        }
    }

    connection.Close();

    emailLabel.Text = clientMail;
    phoneLabel.Text = clientPhone;
    addressLabel.Text = clientAddress;
}
```

Рисунок 3.16 – Вывод заказа

При необходимости можно удалить клиента и заказ, для этого нужно выделить клиента в таблице снизу формы и нажать на кнопку «Удалить выбранную запись». После этого клиент будет удалён из базы данных программы:

```
private void button3_Click(object sender, EventArgs e)
{
    if (selectedID == 0)
    {
        MessageBox.Show("Выделите запись!", "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    else
    {
        if (MessageBox.Show("Вы действительно хотите удалить выбранную запись?", "Удаление", MessageBoxButtons.YesNo) == DialogResult.Yes)
        {
            dataGridView1.Rows.RemoveAt(dataGridView1.SelectedRows[0].Index);
            sAdapter.Update(sTable);

            comboBox1.Items.Clear();

            SqlConnection connection = new SqlConnection("Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=");
            string sql = "SELECT * FROM Clients";
            SqlCommand sCom = new SqlCommand(sql, connection);
            connection.Open();

            using (SqlDataReader read = sCom.ExecuteReader())
            {
                while (read.Read())
                {
                    comboBox1.Items.Add(read["Id"].ToString() + ": " + read["Фамилия"].ToString().Replace(" ", ""));
                }
            }
            connection.Close();

            if (comboBox1.Items.Count > 0)
            {
                comboBox1.SelectedIndex = 0;
            }
        }
    }
}
```

Рисунок 3.17 – Удаление заказа и клиента

3.2 Системные требования

Рекомендуемые системные требования:

- операционная система: Windows 7/8/10 - 64-bit;
- процессор (CPU): Intel Celeron (Desktop);
- оперативная память (RAM): 1 ГБ (или больше);
- свободное место на жёстком диске: ~300 МВ.

Рекомендуемые системные требования:

- операционная система: Windows 7/8/10 - 64-bit;
- процессор (CPU): Intel Core i3 (Desktop);
- оперативная память (RAM): 2 ГБ (или больше);
- свободное место на жёстком диске: ~300 МВ.

3.3 Интерфейс

Данная программа разработана в среде Microsoft Visual Studio на языке программирования C#. В программе присутствует много отличительных особенностей. Рассмотрим их:

Форма входа:

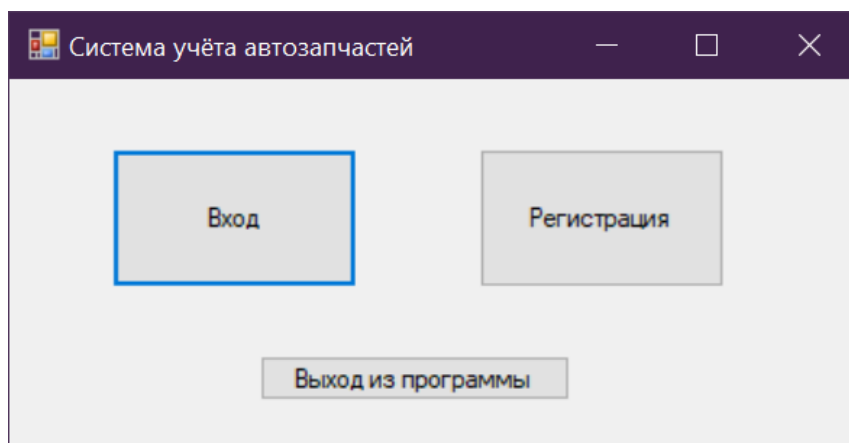


Рисунок 3.18 – Система учета автозапчастей

После открытия программы пользователю предлагается на выбор 2 варианта – Войти в программу, используя свой, заранее зарегистрированный логин и пароль, или зарегистрироваться, создав новый аккаунт (с правами обычного пользователя).

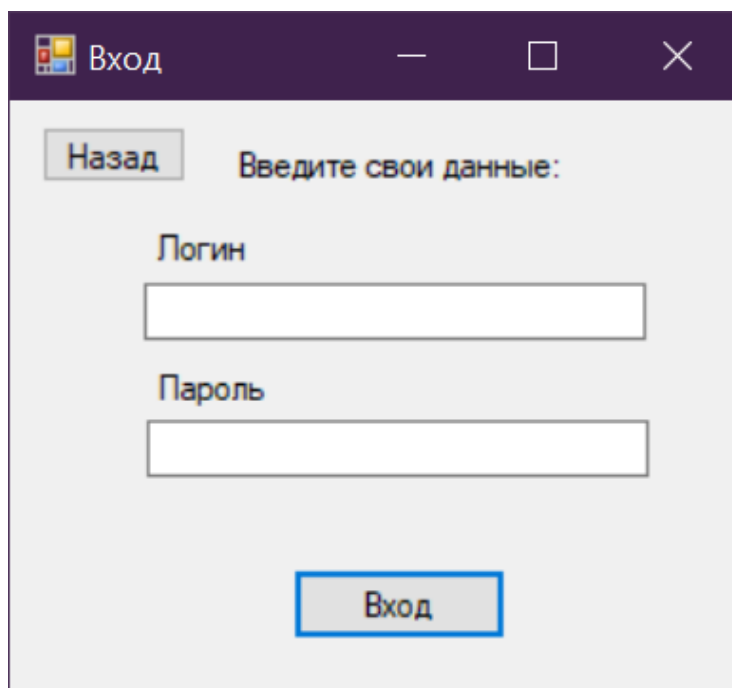


Рисунок 3.19 – Окно входа

После ввода данных, если не соответствуют данным в базе данных – на экран будет выведена ошибка, гласящая, что «Логин или пароль введены неверно». Если же данные для входа введены правильно, то пользователь перейдёт к главному окну программы.

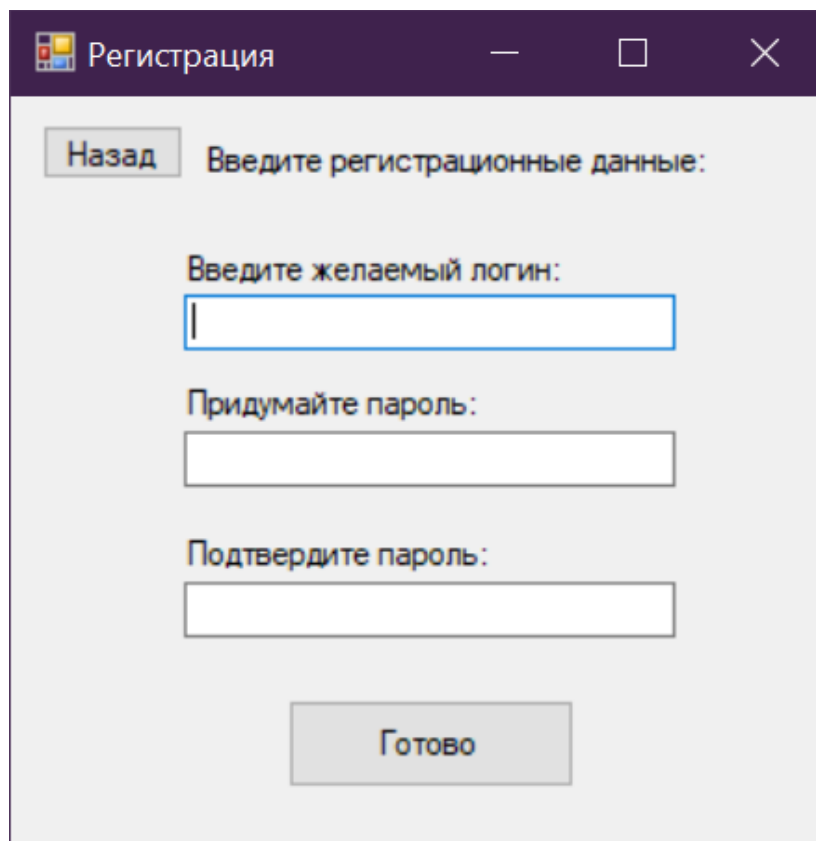


Рисунок 3.20 – Окно регистрации

После ввода и подтверждения регистрационных данных в программе создаётся новый пользователь со стандартными правами доступа (только заказы). Изменять права пользователей может только администратор системы.

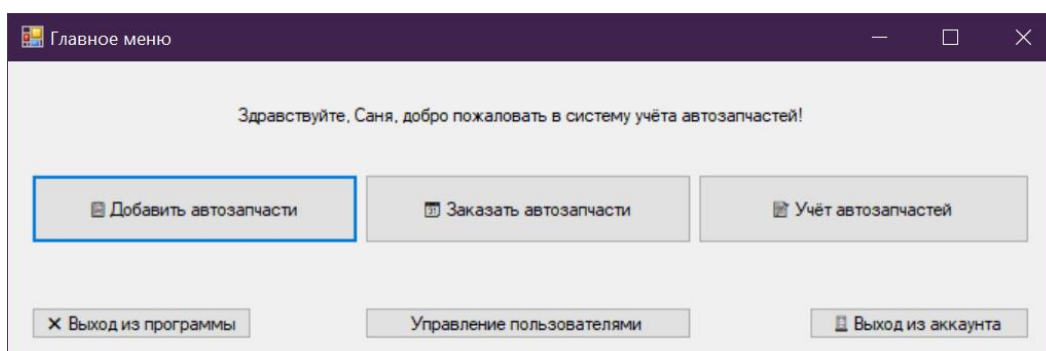


Рисунок 3.21 – Главное окно программы

На главном окне находятся основные элементы взаимодействия пользователя с программой.

При нажатии кнопки «Добавить автозапчасти» открывается форма добавления и изменения автозапчастей, в которой администратор может добавить новую автозапчасть в базу данных, отредактировать старые, добавить к ним фотографии или удалить их.

ID	Название	Тип_запчасти	Количество	Идентификационный_номер	Дата_доб
1	Шины	Зимние шины	12	14881337	12 Январь
2	Диски	Железные диски	5	23212	14 Декабр
3	Руль	Рулевое колесо	8	4561327	12 Сентяб
4	Басы	Аудиосистема	18	6374589	23 Август
*					

Рисунок 3.22 – Форма добавления

При нажатии на главном окне кнопки «Заказать автозапчасти» открывается форма заказа автозапчастей, в которой клиент может выбрать нужную ему автозапчасть, указав марку, модель и год выпуска автомобиля. Программа выведет автозапчасти, находящиеся в наличии именно на введённую марку автомобиля. Далее, поставив галочку напротив нужной запчасти, пользователь добавляет её себе в корзину в количестве одной штуки. В самой корзине клиент может отредактировать количество товара (но не больше, чем имеется на складе, и не меньше, чем 1). После добавления в корзину всех нужных товаров и указания их количества, пользователь нажимает на кнопку «Приступить к оформлению заказа». Которая, в свою очередь, просит клиента ввести ФИО, электронную почту, телефон и адрес, а также проверить его заказ.

Оформление заказа

Введите имя:

Введите фамилию:

Введите отчество:

Электронная почта:

Телефон:

Адрес:

Проверьте ваш заказ:

23212 Диски на Toyota Supra 1992-го года выпуск

Оформить

Рисунок 3.23 – Форма оформления заказа

Если все данные на форме заполнены, то клиенту остаётся просто нажать на кнопку оформить. После этого его ФИО добавится в список клиентов автосервиса в разделе учёта автозапчастей, а также указанное количество автозапчастей в заказе клиента удалится из списка в базе данных.

Если количество заказанных запчастей было равно количеству этих же запчастей в наличии, то данная автозапчасть полностью удалится из базы данных программы, пока она не прибудет на склад и не будет добавлена в базу данных снова.

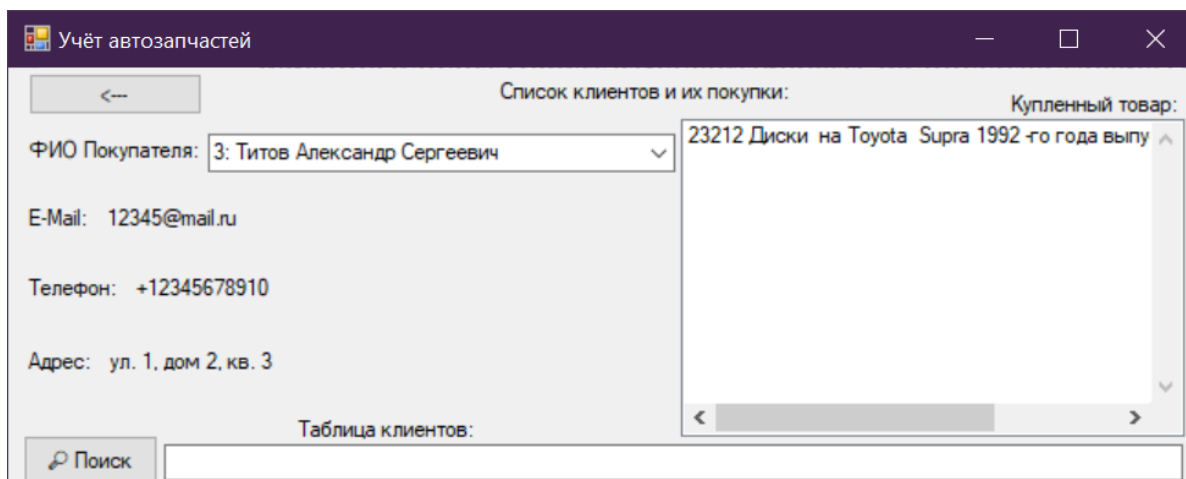


Рисунок 3.24 – Форма учёта автозапчастей

На данной форме отображаются все заказы от выбранного ФИО клиента, а также располагается удобный поиск по ним.

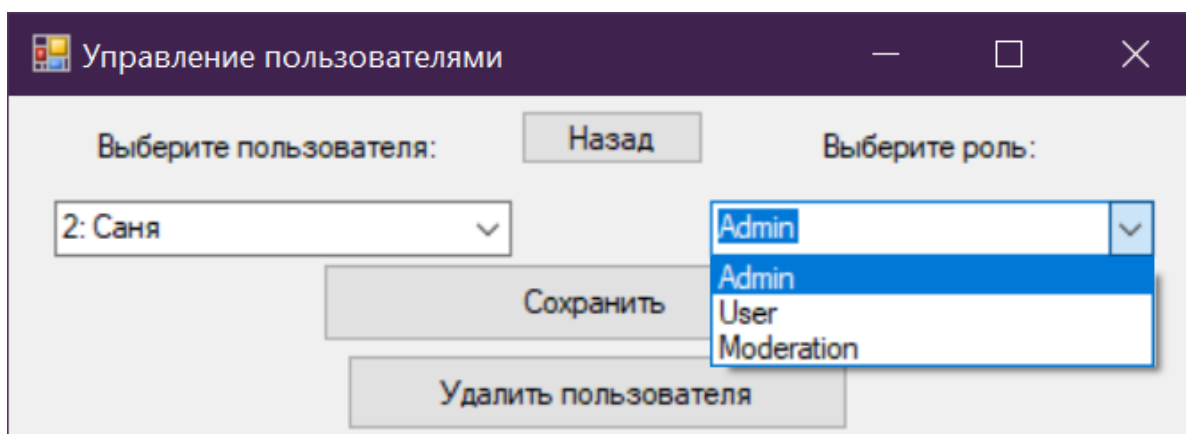


Рисунок 3.25 – Форма управления аккаунтами в программе

Данная форма доступна только Администратору, и позволяет изменять роли пользователей в программе и удалять пользователей.

Разграничение по ролям в системе выглядит следующим образом.

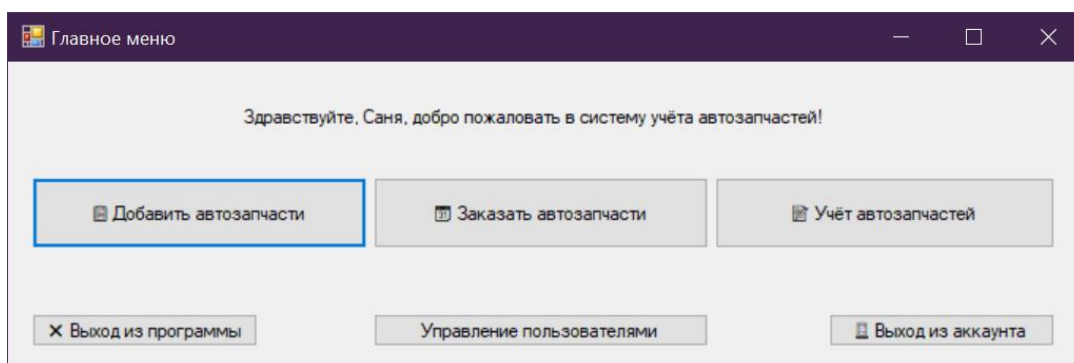


Рисунок 3.26 – Главное окно на роли администратора

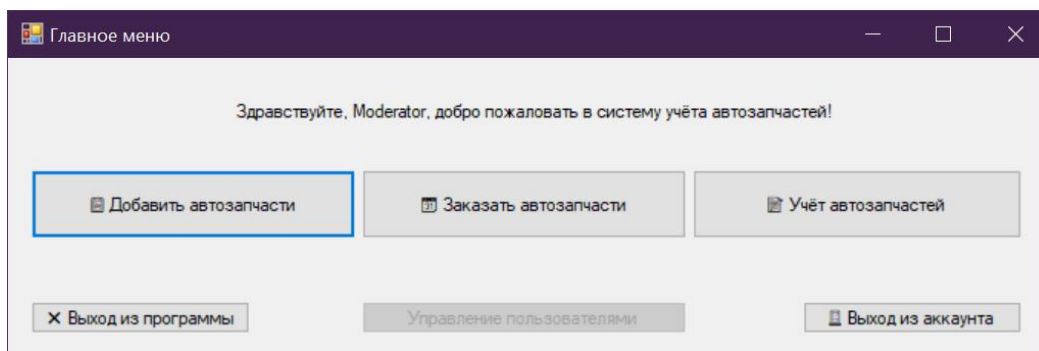


Рисунок 3.27 – Главное окно на роли Модератора

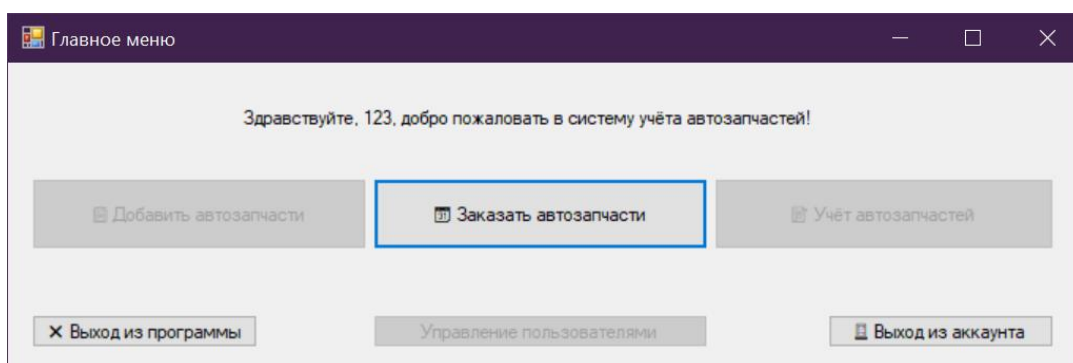


Рисунок 3.28 – Главное окно на роли Пользователя

Как видно из вышеуказанных скриншотов, Администраторам доступны все функции программы, Модераторам – Добавление, Учёт, но не управление аккаунтами, а обычным Пользователям доступен только заказ автозапчастей.

Данное разграничение прав – необходимость в данной системе, т.к. клиент, в целях безопасности, не должен управлять другими пользователями программы и добавлять данные в базу.

Подключение базы данных SQL сети

Для начала Устанавливаем SQL Management Studio, добавив в установщике галочку на компонент PolyBase. Заходим через эту программу в свой сервер, жмём правой кнопкой мыши по «Базы данных» - «Присоединить».

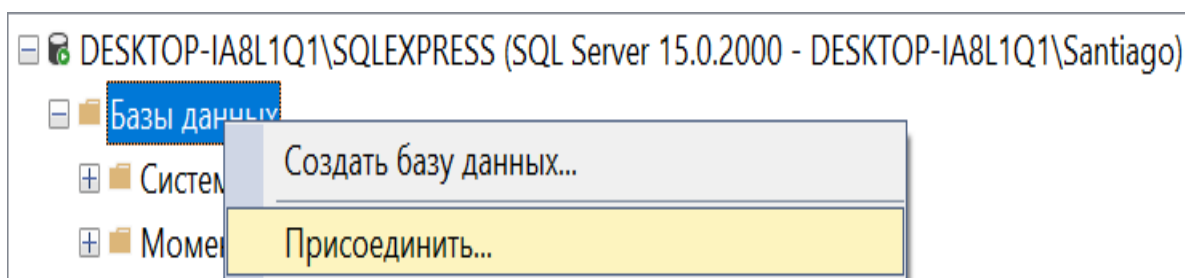


Рисунок 3.29 – Установка SQL Management Studio

Кнопка «Добавить» - Указываем путь к файлу БД .mdf и жмём «ОК». Затем добавленная БД появится во вкладке «Базы данных».

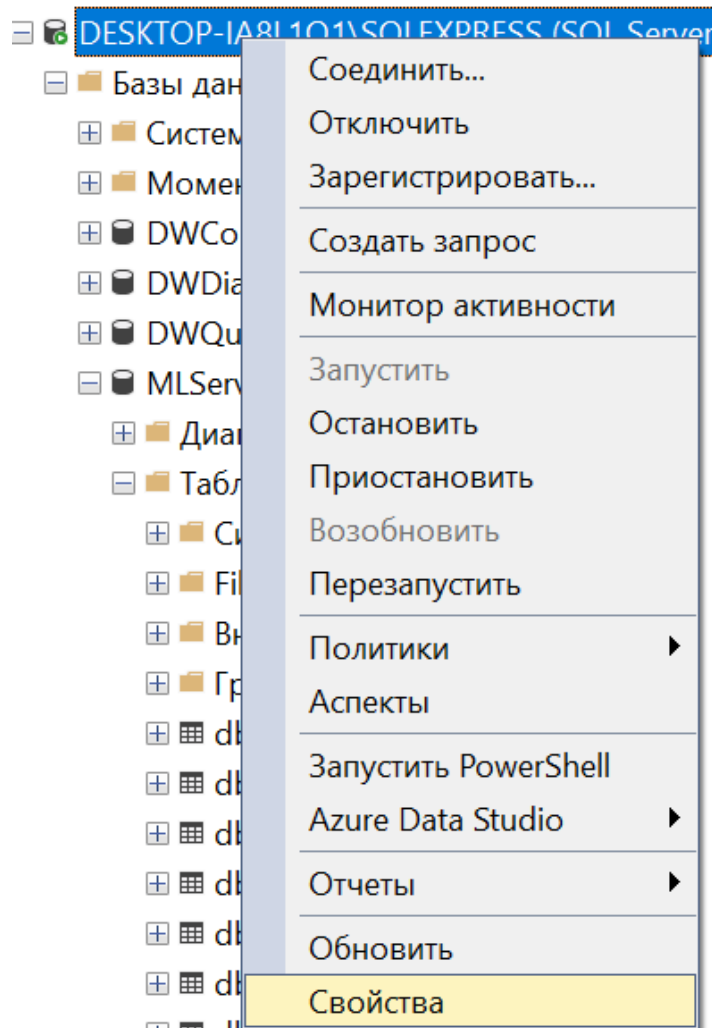


Рисунок 2.30 – Заходим в свойства сервера

Безопасность – Ставим галочку на «Проверка подлинности SQL Server и Windows» - Жмём ОК.

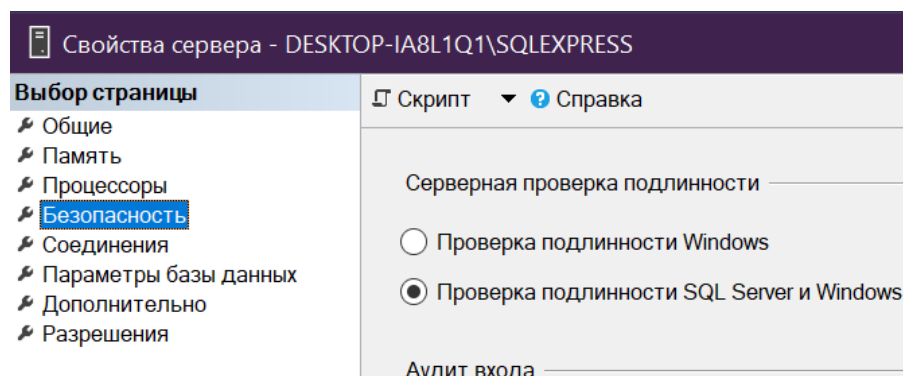


Рисунок 3.31 – Свойства сервера

Во вкладке «Безопасность» - Жмём правой кнопкой по вкладке «Имена для входа» - «Создать имя для входа...», ставим галочку на «Проверка подлинности SQL Server», вводим имя, пароль.

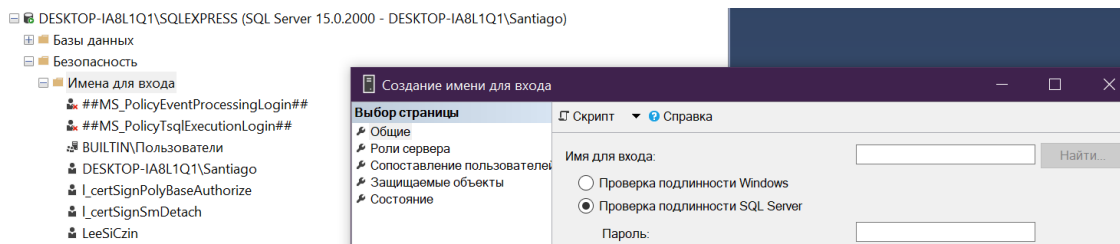


Рисунок 3.32 – Создание имени

Здесь же переходим на вкладку «Роли сервера» - ставим галочки напротив «public» и «sysadmin». На вкладке «Сопоставление пользователей» ставим галку напротив своей добавленной БД, и снизу галочки напротив пунктов «public» и «db_owner». На вкладке «Защищаемые объекты» ставим галку снизу напротив пункта «Соединение SQL».

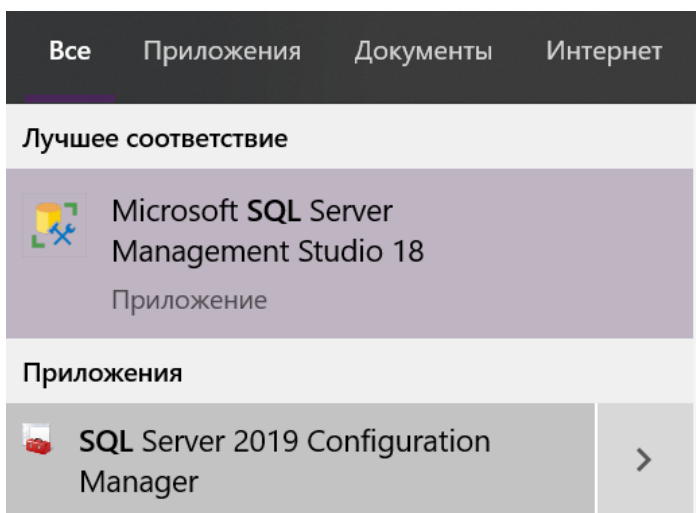


Рисунок 3.33 – Поиск в пуске SQL Server Configuration Manager

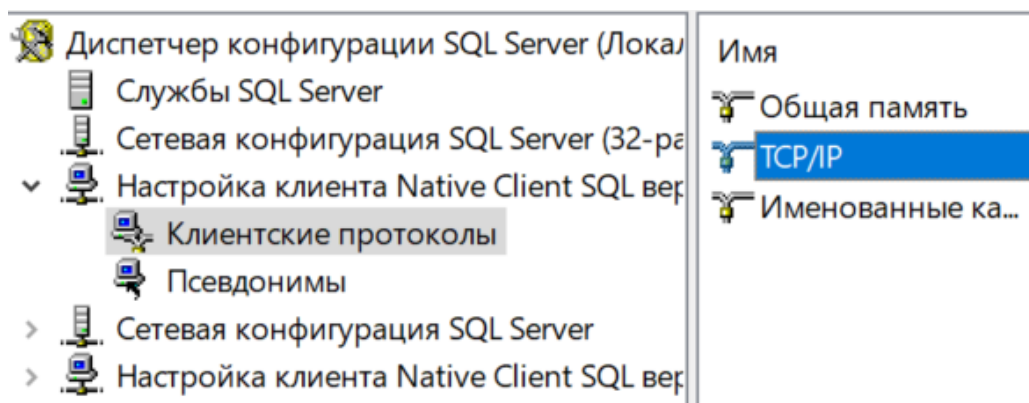


Рисунок 3.34 – Настройка клиента – Клиентские протоколы – TCP/IP

Свойства – Выставить параметр «Включено» на «Да» и параметр «Порт по умолчанию» на 1433.

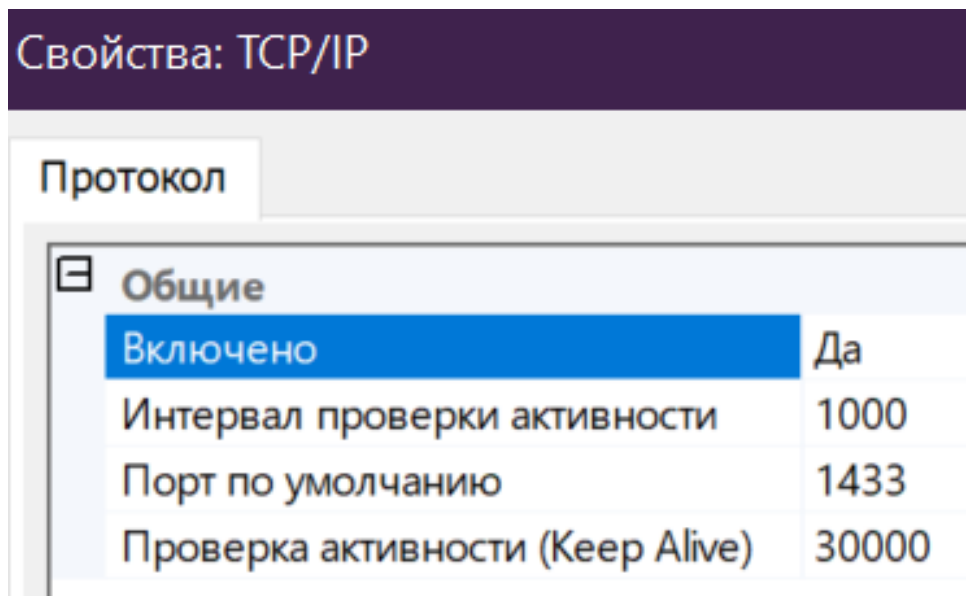


Рисунок 3.35 – Свойства

Убедиться, что в остальных параметрах порт выставлен на 1433.

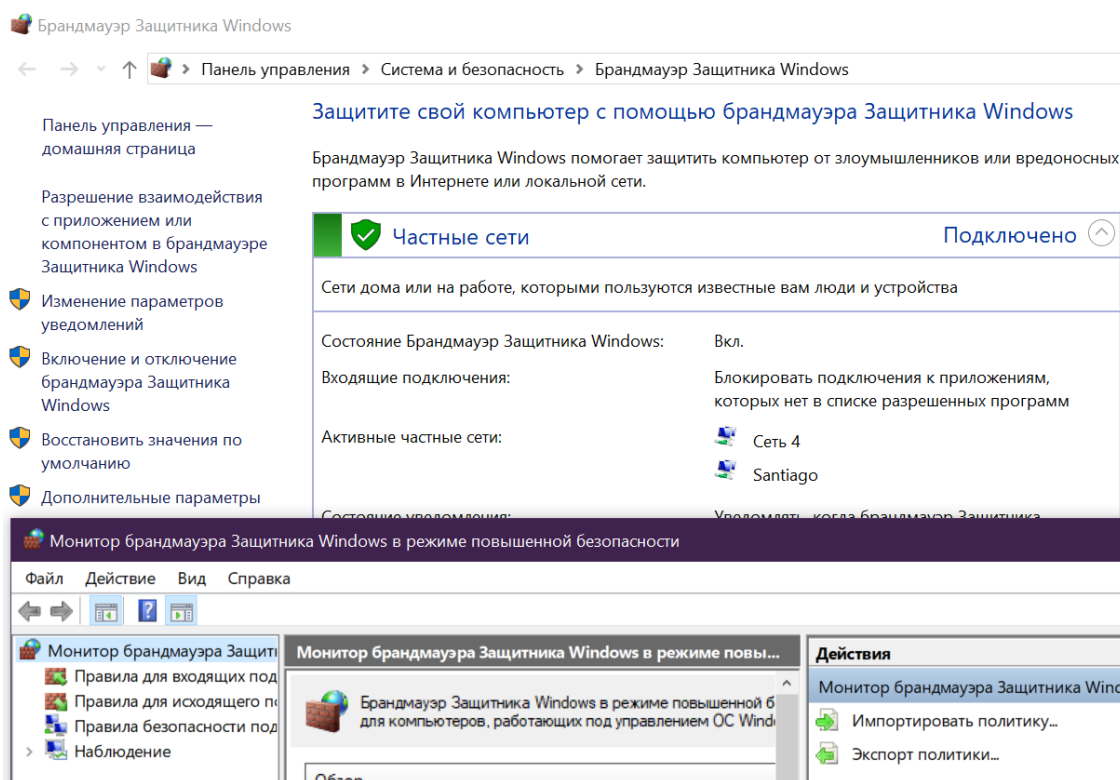


Рисунок 3.36 – Брандмауэр Windows – Дополнительные параметры:

Открываем вкладку «Правила для входящих подключений» - «Создать правило» - «Для порта» - «Далее» - «Протокол TCP» - Вводим порт 1433 – «Далее» - «Разрешить подключение» - Вводим любое имя (можно 1433).

Далее такой же процесс проводим для вкладки «Правила для исходящего подключения». Создаём такое же правило для TCP порта 1433.

Вводим команду ipconfig, затем смотрим в параметрах своего интернет-подключения строку IPv4:

```
Адаптер Ethernet Ethernet:

DNS-суффикс подключения . . . . . : Dlink
Локальный IPv6-адрес канала . . . . : fe80::a007:ada8:f972:bb7d%18
IPv4-адрес . . . . . : 192.168.0.62
Маска подсети . . . . . : 255.255.255.0
Основной шлюз . . . . . : fe80::5ad5:6eff:fede:145e%18
                               192.168.0.1
```

Рисунок 3.37 – Командная строка

Как видно на рисунке – IPv4 адрес здесь будет 192.168.0.62 (у вас будет другой).

Заходим в параметры роутера (в браузере набираем 192.168.0.1), (последний ip на скриншоте, т.е. ip вашего роутера). Заходим в роутер под администратором. Логин и пароль администратора почти на всех роутерах равен: Логин – admin, Пароль - admin, либо Логин – admin, Пароль – пустота.

Если не удаётся войти с этими данными в роутер – то обычно они написаны на обратной стороне роутера. Либо спросите их у поставщика связи.

Далее заходим в параметры Перенаправления портов (Port Forwarding), у каждого роутера вход в данные параметры происходит по разному. В этих параметрах портов – создаём новое свойство:



Рисунок 3.38 – Перенаправления портов

Называем данное свойство любым именем, Протокол указываем TCP, WAN-Порт указываем 1433, LAN-порт указываем 1433, а Destination IP указываем тот, который увидели в Командной строке IPv4 (На рисунке он был 192.168.0.62), у вас он будет другим.

В коде программы меняем все строки подключения с локальных БД, на:

```
«Data Source=IP_Вашего_Компьютера,1433;  
Initial Catalog=Название_вашей_базы_данных_на_сервере;  
User ID=Ваше_имя_пользователя_созданного_на_SQL_сервере;  
Password=Пароль_этого_пользователя»
```

IP Компьютера можно найти, к примеру, на сайте <https://www.2ip.ru>

```
string connectionString = "Data Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=" + Environment.CurrentDirectory + "\\Zapchasti.mdf;Integrated Security=True";
```

Рисунок 3.39 – Строки подключения в коде

4 Обоснование эффективности внедрения проекта

4.1 Технико-экономическое обоснование дипломных работ, связанных с разработкой программного продукта (ПП)

Технико - экономическое обоснование разработки должно содержать:

- определение трудоемкости разработки ПП;
- расчет затрат на разработку ПП;
- определение возможной цены разработанного ПП;
- оценку социально - экономических результатов функционирования

ПП.

4.2 Трудоемкость разработки ПП

Трудоемкость есть экономическая характеристика, которая показывает затраты рабочего времени на производство единицы продукции или на выполнение конкретной технологической операции. Создание ПП занимает примерно 112 часов для одной продукции.

Таблица 4.1 - Распределение работ по этапам и видам и оценка их трудоемкости

Этап разработки ПП	Вид работы на данном этапе	Трудоемкость разработки ПП, чел.× ч.
Техническое задание	Перед началом разработки нужно получить техническое задание и составить план по разработке: интерфейс, функции и прочее.	12
Разработка	Данный этап подразумевает создание самого ПП. Здесь происходит реализация всех идей из этапа планирования.	60
Тестирование	После создания ПП, нужно обязательно его протестировать для выявления ошибок в работе или его улучшения.	12
ИТОГО трудоемкость выполнения дипломной работы		84

4.3 Расчет затрат на разработку ПП

Определение затрат на разработку ПП производится путем составления соответствующей сметы, которая включает следующие статьи:

- 1) Материальные затраты
- 2) Затраты на оплату труда
- 3) Социальный налог
- 4) Амортизация основных фондов
- 5) Прочие затраты

Когда происходит процесс создания ПП, требуется обеспечить персонал материальными ресурсами для комфортной работы.

Таблица 4.2 - Затраты на материальные ресурсы

Наименование материального ресурса	Количество израсходованного материала	Цена за единицу, тг	Сумма, тг
Visual Studio 2019	1	600000	600000
Lenovo IdeaPad	1	200000	200000
Windows 10 Pro	1	75000	75000
Microsoft SQL	1	400000	400000
Бумага	73	5	365
Ручка	5	150	750
Карандаш	5	100	500
Ластик	2	50	100
Корректор	2	150	300
Линейка	1	300	300
ИТОГО затраты на материальные ресурсы			1277315

Общая сумма затрат на материальные ресурсы (ЗМ) определяется по формуле:

$$Z_c = 600000 + 200000 + 75000 + 400000 + 365 + 1300 + 260 + 750 + 500 + 100 + 300 + 300 = 1277315 \quad (4.1)$$

Таблица 4.3 - Затраты на электроэнергию

Наимен. оборудов.	Паспортная мощность, кВт	Коэф. использ. мощности	Время работы оборудования, ч	Цена элек/энер, тг/кВт×ч	Кол-во	Сумма, тг
Lenovo IdeaPad	0,55	0,9	84	19,17	1	1180,87
Лампочка	0,08	0,9			3	515,29
ИТОГО затраты на электроэнергию						1696,16

Общая сумма затрат на электроэнергию (ЗЭ) рассчитывается по формуле:

$$Z_э = (0,55 + 0,08) * 84 * 19,17 = 1696,16 \quad (4.2)$$

При разработке ПП требуется сотрудники имеющие знание в данной сфере и оплата их труда.

Затраты на оплату труда рассчитываются по форме, приведенной в таблице 1.4.

Таблица 4.4 - Затраты на оплату труда

Категория работника	Количество	Квалификация	Трудоемкость разработки ПП, чел.×ч	Часовая ставка, тг/ч	Сумма, тг
Разработчик	1	Бакалавриат	84	1309,52	110000
ИТОГО затраты на оплату труда					110000

Общая сумма затрат на оплату труда (ЗТР) определяется по формуле:

Таблица 4.5 - Социальный налог:

СО(Социальные отчисления)	3,5	(ЗП - ОПВ)*3,5%	3 465,5
ВОСМСЮ (Отчисления на ВОСМСЮ)	2,0	ЗП*2%	2 200,0
СН (Социальный налог)	9,5	(ЗП-ОПВ-ВОСМС)*9,5%-СО	5 730,5
Всего уплаченные налоги			11 396,0

$$НР = ЗП \cdot \frac{Н_{нр}}{100} = 110\,000 \cdot 0,7 = 77\,000 \text{ тенге} \quad (4.3)$$

Амортизационные отчисления — это денежные средства, предназначенные для возмещения износа предметов, относящихся к основным средствам предприятия

Таблица 4.6 - Амортизация основных фондов (ОФ)

Наименование оборудования и ПО	Стоимость оборудования и ПО, тг	Годовая норма амортизации, %	Время работы оборудования и ПО для разработки ПП, ч	Сумма, тг
Lenovo IdeaPad	200000	14,29	16	2381,67
Windows 10 Pro	75000	33,33	16	2083,125
Visual Studio	600000	33,33	16	16665
ИТОГО амортизация основных фондов				21192,795

Общая сумма амортизационных отчислений определяется по формуле:

Применение основных фондов колеблется от 3 до 10 лет. Оборудование используется в течение 7 лет, а программное обеспечение – 3 года. Применяя формулу, заполним Таблицу 1.5 для отображения амортизации основных фондов.

$$H_{A1} = 100/7 = 14,29\%.$$

$$H_{A3} = 100/3 = 33,33\%.$$

$$Z_{ам} = \frac{200000 * 0,1429 * 16}{1 * 12 * 16} = 2381,67 \text{ тг}$$

$$Z_{ам} = \frac{75000 * 0,3333 * 16}{1 * 12 * 16} = 2083,125 \text{ тг}$$

$$Z_{ам} = \frac{600000 * 0,3333 * 16}{1 * 12 * 16} = 16665 \text{ тг}$$

Также стоит учитывать прочие расходы:

- аренда помещения - 75000;
- коммунальные услуги – 5492,94.

Таблица 4.7 - Смета затрат на разработку ПП

Статьи затрат	Сумма, тг
1) Материальные затраты, в том числе:	
- материалы	1277315
- электроэнергия	1696,16
2) Затраты на оплату труда.	110000
3) Отчисления на социальные нужды.	11 396,0
4) Амортизация основных фондов.	21192,795
5) Прочие затраты.	80 492,94
ИТОГО по смете	1 502 092,895

4.4 Определение договорной цены ПП

Величина возможной (договорной) цены ПП должна устанавливаться с учетом эффективности, качества и сроков ее выполнения на уровне, отвечающем экономическим интересам заказчика (потребителя) и исполнителя.

Договорная цена (Ц_д) для прикладных ПП рассчитывается по формуле:

$$Ц_{д} = З_{\text{нир}} * \left(1 + \frac{P}{100}\right) = 1502092,89 * \left(1 + \frac{30}{100}\right) = 1952720,76 \quad (4.4)$$

З_{нир} - затраты на разработку ПП (из таблицы 1.6), тг;

P - средний уровень рентабельности ПП. %

Далее определяется цена реализации с учетом налога на добавленную стоимость (НДС), ставка НДС устанавливается законодательно Налоговым Кодексом РК. На 2013 год ставка НДС установлена в размере 12%.

Цена реализации с учетом НДС рассчитывается по формуле:

$$Ц_{р} = Ц_{д} + Ц_{д} * \text{ндс} = 1952720,76 + 1952720,76 * 0,12 = 2187047,25 \quad (4.5)$$

4.5 Расчет результатов от создания и использования ИС

Информационная система для управления проектированием предназначена в главную очередь для контроля работы проектной группы. Она упрощает процесс принятия решений по управлению, а также способна повысить эффективность и скорость работы сотрудников, за счет уменьшения времени.

Для оценки экономии от использования разрабатываемой ИС необходимо сравнить эксплуатационные расходы с ее применением и без.

Статьи затрат при применении ИС включают в себя:

- заработная плата специалиста, осуществляющего поддержку и сопровождение системы;
- износ оборудования;
- накладные расходы.

Данная система не предполагает расхода каких-либо материалов..

В расходные материалы входит бумага, картридж.

Пачка бумаги стоит – 1500 тг.

Картридж – 23000.

В месяц уходит 5 пачек бумаги и один картридж.

$$P = ((1500 \cdot 5) + (23000)) \cdot 12 = 366\,000 \text{ тенге}$$

В распоряжении предприятия находятся один персональный компьютер (стоимость одного ~ 400 000 тенге), принтер (120 000 тенге) и прочая компьютерная периферия (~50 000 тенге); итого затрат на оборудование – 570 000 тенге.

Износ оборудования рассчитывается исходя из 25 % амортизационных отчислений за год.

$$A = (400\,000 + 120\,000 + 50\,000) \cdot 0,25 = 142\,500 \text{ тенге}$$

Данной работой занимаются 2 сотрудника, нанятые для контроля бумажного оборота методических трудов и прочих документов.

Каждый сотрудник получает – 125 000 тенге

ЗП = 3 000 000 тенге

ОПВ = 3 000 000 · 0,1 = 300 000

СО = (3 000 000 – 300 000) · 0,035 = 94 500

ВОСМС = 3 000 000 · 0,02 = 60 000

СН = (3 000 000 - 300 000 – 60 000) · 0,095 – 94 500 = 156 300

ОТ = СО + СН + ВОСМС = 75 600 + 125 600 + 48 000 = 310 800 тенге

Общие накладные расходы составят:

$$НР = ЗП \cdot \frac{Н_{НР}}{100} = 3\,000\,000 \cdot 0,7 = 2\,100\,000 \text{ тенге} \quad (4.6)$$

Статьи затрат без применения ИС включают в себя:

- заработная плата сотрудников;
- износ используемого ими оборудования;
- расход материалов (к примеру, канцелярия – бумага, картридж и т.д.)
- накладные расходы.

Таблица 4.8 – Годовые эксплуатационные затраты

Статьи	Без применения ИС (1 сотрудник – куратор проекта)	С применением ИС (1 сотрудник – программист, сопровождающий систему)
Годовая заработная плата	3 000 000	1 320 000
Социальные отчисления и налоги	310 800	136 752
Расходуемые материалы	366 000	0
Амортизационные отчисления	142 500	21192
Накладные расходы	2 100 000	924 000
Всего	5 919 300	2 401 944

Ожидаемая условно-годовая экономия определяется по формуле:

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 + \sum \mathcal{E}_i \quad (4.7)$$

где $\mathcal{E}_{\text{уг}}$ - величина экономии, тенге;

C_1 и C_2 – показатели текущих затрат по базовому и внедряемому вариантам, тенге;

$\sum \mathcal{E}_i$ – ожидаемый дополнительный эффект от различных факторов, тенге.

$$\mathcal{E}_{\text{уг}} = C_1 - C_2 = 3\,517\,356 \text{ тенге}$$

4.6 Расчет основных показателей экономической эффективности

Так как разработанная информационная система несет более социальный эффект, чем экономический, целесообразно оценивать его эффективность за счет экономии в сравнении с предыдущим периодом работы без использования ИС.

Величина ожидаемого годового экономического эффекта от внедрения ИС рассчитывается по формуле:

$$\mathcal{E}_{\text{г}} = \mathcal{E}_{\text{уг}} - K \cdot E_{\text{н}} \quad (4.7)$$

где $\mathcal{E}_{\text{г}}$ - ожидаемый годовой экономический эффект, тенге;

$\mathcal{E}_{\text{уг}}$ — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения, тенге;

$E_{\text{н}}$ - нормативный коэффициент экономической эффективности капитальных вложений.

Нормативный коэффициент экономической эффективности капитальных вложений определяется по формуле:

$$E_H = \frac{1}{T_H} \quad (4.7)$$

где T_H — нормативный срок окупаемости капитальных вложений, лет.

Нормативный срок окупаемости капитальных вложений. принимается исходя из срока морального старения - технических средств и проектных решений ИС ($T_H=1,2,3...n$), для программных продуктов срок окупаемости принимаем равным 4 года. 1546006

$$E_H = \frac{1}{4} = 0,25$$

$$\mathcal{E}_r = 3\,517\,356 - 1\,546\,006 \cdot 0,25 = 3\,888\,366 \text{ тенге}$$

Расчетный коэффициент экономической эффективности капитальных вложений составляет:

$$E_p = \frac{\mathcal{E}_{уг}}{K}$$

где E_p - расчетный коэффициент экономической эффективности капитальных вложений;

$\mathcal{E}_{уг}$ — ожидаемая условно-годовая экономия, тенге;

K — капитальные вложения на создание системы, тенге.

$$E_p = \frac{3\,517\,356}{2\,187\,047} = 1,61$$

Расчетный срок окупаемости капитальных вложений составляет:

$$T_p = \frac{1}{E_p} \quad (4.8)$$

где E_p - коэффициент экономической эффективности капитальных вложений.

$$T_p = \frac{1}{1,61} = 0,6 \text{ года} \approx 7,2 \text{ месяцев}$$

Таблица 4.9 – Показатели сравнительной экономической эффективности от внедрения программного продукта

Наименование показателей	Значение
Условная годовая экономия затрат, тенге	3 517 356
Коэффициент эконом. эффек. капитальных вложений (E_p)	1,61
Срок окупаемости капитальных вложений (T_p)	0,6

5 Безопасность жизнедеятельности

5.1 Определение категории тяжести труда через интегрально-бальную оценку

Условия труда является совокупностью факторов производственной среды, оказывающих влияние на здоровье и работоспособность человека в процессе труда.

Факторы, составляющие условия труда, делятся на 4 группы:

- санитарно-гигиенические факторы (влажность, температура воздуха, освещенность, наличие шума и вибраций, электромагнитных излучений);
- психофизиологические факторы (физическая нагрузка, нервное напряжение, темп работы, монотонность работы);
- эстетические факторы (цветовым оформлением рабочих мест и помещений);
- социально - психологические факторы (сплоченностью коллектива).

Условия труда оказывают существенное влияние на работоспособность человека и его здоровье.

Для предотвращения или уменьшения негативного воздействия вредных производственных факторов, уменьшения работоспособности, предупреждения появления профзаболеваний необходимо планировать и проводить мероприятия по улучшению условий труда, для этого следует производить анализ условий труда с определением категории тяжести работ.

Можно использовать категорию тяжести работы, чтобы оценить влияния вредных факторов на здоровье и работоспособность, учитывающую суммарное воздействие всех факторов производственной среды.

Тяжесть работы представляет из себя функциональное напряжение организма работников. Проблема может возникать под влиянием как физической (рабочая поза, масса перемещаемых грузов и т.д.), так и нервно-эмоциональной нагрузки (монотонность, число объектов наблюдения ит.д.), а так же под воздействием внешних условий производственной среды (освещенность, превышение уровня шумов и вибрации и т.д.).

Из-за воздействия вредных производственных факторов в процессе труда возможно формирование одного из трех функциональных состояний организма: нормальное, пограничное (между нормой и патологией) и патологическое.

По существующей классификации условия труда можно разделить на 6 категорий тяжести работ.

К первой категории тяжести следует отнести те виды работ, которые могут выполняться при оптимальных условиях внешней производственной среды и при оптимальной величине физической, умственной и нервно-эмоциональной нагрузки. В этом случае рабочая нагрузка будет соответствовать физиологическими возможностями организма человека и его способностям.

Ко второй категории тяжести следует отнести те работы, в результате выполнения которых уровни вредных и опасных производственных факторов не превышают оптимальных или предельно допустимых значений. При этом работоспособность не ухудшается, отклонений в состоянии здоровья, которые могут быть связаны с профессиональной деятельностью, не наблюдается. Возможные функциональные изменения исчезают во время отдыха.

К третьей категории тяжести следует отнести работы, выполняемые в условиях, при которых у практически здоровых людей возникают реакции, характерные пограничному состоянию организма. Наблюдается некоторое снижение трудовых показателей (производительности труда). Применение рациональных режимов труда и отдыха достаточно быстро может устранить данные негативные последствия.

К четвертой категории тяжести следует отнести работы, в результате выполнения которых, в организме может формироваться достаточно глубокое пограничное состояние даже у практически здоровых людей. Большинство физиологических показателей при этом ухудшается (замедляются реакции), особенно в конце смены или рабочей недели. Могут появиться характерные производственно обусловленные состояния.

К пятой категории тяжести следует отнести работы, при выполнении которых в конце смены и (или) рабочей недели формируются реакции, которые характерны для патологического состояния организма у практически здоровых людей и которые, как правило, исчезают у большинства работников после полноценного отдыха. Но у некоторых лиц изменения могут привести к производственно обусловленным и профессиональным заболеваниям.

К шестой категории тяжести следует отнести работы, в результате выполнении которых, явно будут проявляться признаки патологического состояния в организме человека. Эти работы выполняются в особо вредных (критических) условиях труда. При этом патологические реакции могут развиваться достаточно быстро, могут иметь необратимый характер и часто сопровождаются тяжелыми нарушениями функций жизненно важных органов и систем.

Условия труда оказывают непосредственное влияние на состояние организма, которое характеризуется определенными реакциями. Для оценки негативного воздействия на человека внешних условий необходимо определять категорию тяжести работ.

При проведении количественного анализа тяжести труда должны учитываться санитарно-гигиенические и психофизиологические факторы производственной среды, характеризующие условия труда.

К санитарно-гигиеническим факторам производственной среды в соответствии с ГОСТ следует отнести:

- микроклимат в рабочей зоне;
- наличие и концентрацию вредных веществ различных классов

опасности;

- наличие и концентрацию производственной пыли;
- виброакустические факторы и ультразвук;
- интенсивность теплового излучения;
- электромагнитные излучения различных диапазонов частот;
- ионизирующие излучения (рентгеновское, гамма-, α - β -излучения);
- биологические факторы.

К психофизиологическим факторам соответствии с ГОСТ следует отнести:

- физическую, динамическую и статическую нагрузки;
- рабочую позу и перемещения в пространстве;
- сменность, продолжительность непрерывной работы в течение суток;
- разряд зрительных работ;
- число важных объектов наблюдения;
- темп работы, монотонность работы;
- объем получаемой и перерабатываемой информации;
- режим труда и отдыха;
- нервно-эмоциональная нагрузка;
- интеллектуальная нагрузка.

5.2 Расчет оценки рабочей среды

Решение:

По исходным данным и таблицам приложения 1 проводим выставление баллов каждому фактору рабочей среды и показателю до и после проведения мероприятий по оздоровлению условий труда. При оценке необходимо корректировать значение балла в зависимости от времени воздействия. Результаты оценки представляем в виде таблицы (таблица 5.1).

Таблица 5.1 - Балльная оценка факторов рабочей среды и условий труда

Фактор рабочей среды и условия труда	Значение показателя	До проведения мероприятий	После проведения мероприятий
Температура воздуха на РМ в теплый период года, С	30/20	4	1
Превышение допустимого уровня звука, дБа	55/30	1	1
РМ стационарное, поза свободная, масса переносимых грузов до 5 кг	--	1	1

Продолжение таблицы 5.1

Работа в утреннюю смену. Продолжительность непрерывной работы в течение суток, часов	8/6	1	1
Длительность сосредоточенного наблюдения, % от продолжительности рабочей смены	50/25	2	1
Обоснованный режим труда и отдыха без применения музыки и гимнастики	--	4	2
Нервно-эмоциональная нагрузка возникает в результате простых действий по индивидуальному плану	--	1	1

После оценки в баллах факторов и показателей необходимо рассчитать интегральную оценку тяжести труда до и после проведения мероприятий:

а) до проведения мероприятий по улучшению условий труда:

$$U_1 = \left(4 + \frac{4 + 1 + 1 + 1 + 2 + 4 + 1}{6} \times \frac{6 - 4}{6} \right) * 10 = 47,78$$

Определяем, что данные условия труда относятся к четвертой категории тяжести труда

К четвертой категории тяжести следует отнести работы, в результате выполнения которых, в организме может формироваться достаточно глубокое пограничное состояние даже у практически здоровых людей. Большинство физиологических показателей при этом ухудшается (замедляются реакции), особенно в конце смены или рабочей недели. Могут появиться характерные производственно обусловленные состояния.

б) после проведения мероприятий по улучшению условий труда.

Так как после проведения мероприятий изменилось время воздействия факторов рабочей среды и условий труда, необходимо пересчитать оценку факторов.

Принимаем продолжительность смены равной 480 мин.

В нашем случае после проведения мероприятий изменилось температура, поэтому бальную оценку необходимо провести с учетом данного изменения:

$$X_{кор1} = 1 \times \frac{420}{480} = 0,9$$

и при изменении длительного сосредоточенного наблюдения:

$$X_{\text{кор2}} = 1 \times \frac{240}{480} = 0,5$$

Интегральная балльная оценка после проведения мероприятий с учетом коррекции будет равна:

$$U_2 = \left(2 + \frac{0,9 + 1 + 1 + 1 + 0,5 + 2 + 1}{6} \times \frac{6 - 2}{6} \right) * 10 = 28,22$$

Определяем, что данные условия труда относятся ко второй категории тяжести труда. В таких условиях возникают реакции, характерные начальной стадии пограничного состояния организма.

Ко второй категории тяжести следует отнести те работы, в результате выполнения которых уровни вредных и опасных производственных факторов не превышают оптимальных или предельно допустимых значений. При этом работоспособность не ухудшается, отклонений в состоянии здоровья, которые могут быть связаны с профессиональной деятельностью, не наблюдается. Возможные функциональные изменения исчезают во время отдыха.

Определим рост травматизма до проведения мероприятий по улучшению условий труда:

$$Y_1 = \frac{1}{1,3 - 0,0185 \times 47,78} = 2,4$$

После проведения мероприятий (изменение температуры воздуха рабочей среды, уменьшение уровня шума и времени воздействия на оператора и т.д.) категория тяжести труда снизится до третьей ($U_2=28,22$), что будет соответствовать росту травматизма в 1,69 раза по сравнению с рациональными условиями труда:

$$Y_1 = \frac{1}{1,3 - 0,0185 \times 28,22} = 1,3$$

При проведении мероприятий по улучшению условий труда, категория тяжести изменилась с четвертой до второй. Как отмечалось выше, тяжесть труда негативно влияет на степень утомления, а значит и работоспособность человек. Для исследования динамики изменения работоспособности и производительности необходимо рассчитать значения показателей утомления и работоспособности:

- а) до проведения комплекса мероприятий:
 - показатель утомления по формуле (3):

$$Y_1 = \frac{47,78 - 15,56}{0,64} = 50$$

- уровень работоспособности по формуле (4):

$$R_1 = 100 - 50 = 50$$

б) после проведения комплекса мероприятий:

- показатель утомления:

$$Y_2 = \frac{28,22 - 15,56}{0,64} = 20$$

- уровень работоспособности:

$$R_2 = 100 - 20 = 80$$

5) Изменение производительности труда (прирост производительности труда) за счет изменения работоспособности по формуле (5) составит:

$$\Pi_{\text{пт}} = \left(\frac{80}{50} - 1 \right) \times 100 \times 0,2 = 12$$

Заключение

В результате дипломного проектирования была выполнена информационная система, которая позволяет заказать автозапчасти у предприятия, учет автозапчастей, а также управлять пользователями.

Главное отличие этой программы от других программ, которые были ранее использованы:

- есть возможность поддержания связи с клиентом;
- программа адаптирована под нужды предприятия;
- возможность хранения заказов;
- автоматизация рабочего процесса.

Система, которая ранее использовалась предприятием, не была адаптирована под работу и давала, в большинстве случаев, сбои в работе. Предприятие отказалось в одно время от его использования из-за потери информации.

В проекте все было реализовано для эффективной работы предприятия. Программа даст огромное преимущество по сравнению с прошлыми ее аналогами и уменьшит издержки предприятия.

Список литературы

- 1 Перлова, О.Н. Проектирование и разработка информационных систем: Учебник / О.Н. Перлова, О.П. Ляпина, А.В. Гусева. - М.: Academia, 2017. - 416 с;
- 2 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с;
- 3 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с;
- 4 Альфред, В. Ахо Компиляторы. Принципы, технологии и инструментарий / Альфред В. Ахо и др. - М.: Вильямс, 2015. - 266 с;
- 5 Бишоп, Дж. C# в кратком изложении / Дж. Бишоп, Н. Хорспул. - М.: Бином. Лаборатория знаний, 2013. - 472 с;
- 6 Вагнер, Билл C# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2013. - 320 с;
- 7 Зиборов, В.В. Visual C# 2012 на примерах / В.В. Зиборов. - М.: БХВ-Петербург, 2013. - 480 с;
- 8 Зиборов, Виктор Visual C# 2010 на примерах / Виктор Зиборов. - М.: "БХВ-Петербург", 2011. - 432 с;
- 9 Ишкова, Э. А. Самоучитель C#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2013. - 496 с;
- 10 Касаткин, А. И. Профессиональное программирование на языке си. Управление ресурсами/А.И. Касаткин. - М.: Высшая школа, 2012. - 432 с;
- 11 Лотка, Рокфорд C# и CSLA .NET Framework. Разработка бизнес-объектов / Рокфорд Лотка. - М.: Вильямс, 2010. - 816 с;
- 12 Мак-Дональд, Мэтью Silverlight 5 с примерами на C# для профессионалов / Мэтью Мак-Дональд. - М.: Вильямс, 2013. - 848 с;
- 13 Марченко, А. Л. Основы программирования на C# 2.0 / А.Л. Марченко. - М.: Интернет-университет информационных технологий, Бином. Лаборатория знаний, 2011. - 552 с;
- 14 Подбельский, В. В. Язык C#. Базовый курс / В.В. Подбельский. - М.: Финансы и статистика, Инфра-М, 2011. - 384 с;
- 15 Прайс, Джейсон Visual C# 2.0. Полное руководство / Джейсон Прайс, Майк Гандэрлой. - М.: Век +, Корона-Век, Энтроп, 2010. - 736 с;
- 16 Рихтер, Джеффри CLR via C#. Программирование на платформе Microsoft .NET Framework 4.0 на языке C# / Джеффри Рихтер. - М.: Питер, 2013. - 928 с;
- 17 Смоленцев, Н. К. MATLAB. Программирование на Visual C#, Borland JBuilder, VBA (+ CD-ROM) / Н.К. Смоленцев. - М.: ДМК Пресс, 2011. - 456 с;
- 18 Троелсен, Эндрю Язык программирования C# 5.0 и платформа .NET 4.5 / Эндрю Троелсен. - М.: Вильямс, 2015. - 486 с.

Приложение А

Техническое задание

1 Общие сведения

1.1 Наименование системы

Полное наименование системы:

Информационная система учета продаж автомобильных запчастей .

1.2 Сроки начала и окончания работ

Дата начала: 15.01.2020

Дата окончания: 31.04.2020

2 Назначение и цели создания системы

2.1 Назначение системы

Информационная система предназначена для учета автозапчастей, добавление автозапчастей, управление пользователями и автоматизации процессов работы персонала.

3 Рекомендации к разработке программы

Информационная система может быть разработана на языке С# с использованием Microsoft SQL Server.

4 Требование к внешнему виду системы

Обязательным условием является простота и понятливость интерфейса.

5 Технические требования

Информационная система не требует мощных персональных компьютеров и может работать на более простом железе.

6 Экономические требования

- возможная (договорная) цена продукта составила 2 187 047 тг;
- стоимость разработки продукта составила 1 502 092 тг.

Приложение Б (ЛИСТИНГ ПРОГРАММЫ)

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.SqlClient;
using System.Diagnostics;
using System.Drawing;
using System.IO;
using System.Windows.Forms;
using System.Linq;
using System.Management;
using Microsoft.Office.Interop.Word;
using SautinSoft.Document;
using SautinSoft.Document.Drawing;
using System.Collections;

namespace WindowsFormsApp1
{
    public partial class Form1 : Form
    {
        List<string> category = new List<string>();
        List<int> countsPages = new List<int>();

        SqlConnection sqlcon = new SqlConnection("Data
Source=(LocalDB)\\MSSQLLocalDB;AttachDbFilename=D:\\Univer\\1Diplom\\
WindowsFormsApp1\\WindowsFormsApp1\\WindowsFormsApp1\\WindowsFor
msApp1\\Database1.mdf;");
        SqlCommand sqlcmd = new SqlCommand();
        SqlDataAdapter da = new SqlDataAdapter();
        System.Data.DataTable dt = new System.Data.DataTable();
        public string path = "";
        public static string date = "";
        public int selectedIndex = 0;
        public int selectedID = 0;
        public bool IndexIsChanged = false;
        string formatForRedakt = "";
        string nameForRedakt = "";
        string RedaktID = "";

        ManagementEventWatcher stopWatch = new ManagementEventWatcher(new
WqlEventQuery("SELECT * FROM Win32_ProcessStopTrace"));
    }
}
```

Продолжение приложения Б

```
ManagementEventWatcher startWatch = new ManagementEventWatcher(new  
WqlEventQuery("SELECT * FROM Win32_ProcessStartTrace"));
```

```
public Form1()  
{  
    InitializeComponent();  
    stopWatch.EventArrived += stopWatch_EventArrived;  
    startWatch.EventArrived += startWatch_EventArrived;  
}
```

```
//СЛЕЖЕНИЕ ЗА ЗАКРЫТЫМИ ПРИЛОЖЕНИЯМИ НАЧАЛО  
void stopWatch_EventArrived(object sender, EventArgs e)  
{  
    if (e.NewEvent.Properties["ProcessName"].Value.ToString() ==  
"WINWORD.EXE" || e.NewEvent.Properties["ProcessName"].Value.ToString()  
== "EXCEL.EXE" || e.NewEvent.Properties["ProcessName"].Value.ToString() ==  
"notepad.exe")  
    {  
        if(e.NewEvent.Properties["ProcessID"].Value.ToString() == RedaktID)  
        {  
            byte[] FileBytes = null;  
            try  
            {  
                FileStream FS = new FileStream("c:\\temp\\RedaktFile" +  
formatForRedakt.Replace(" ", ""), System.IO.FileMode.Open,  
System.IO.FileAccess.Read);  
                BinaryReader BR = new BinaryReader(FS);  
                long allbytes = new FileInfo("c:\\temp\\RedaktFile" +  
formatForRedakt.Replace(" ", "")).Length;  
                FileBytes = BR.ReadBytes((Int32)allbytes);  
  
                FS.Close();  
                FS.Dispose();  
                BR.Close();  
            }  
            catch (Exception ex)  
            {  
                MessageBox.Show("Ошибка при чтении файла. " +  
ex.ToString());  
            }  
        }  
    }  
}
```