

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

09.03.03 Прикладная информатика
(код и наименование направления подготовки, специальности)

Бизнес-информатика
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Разработка автоматизированной системы для службы «Техническая поддержка» (на примере ИП Ибрагимов С.К.)

Студент

Ю.А. Степанова
(И.О. Фамилия)

(личная подпись)

Руководитель

Н.Н. Рогова
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы «Разработка автоматизированной системы для службы «Техническая поддержка» (на примере ИП Ибрагимов С.К.)».

Ключевые слова: информационная система, программное обеспечение, база данных, моделирование.

Данная выпускная квалификационная работа посвящена разработке автоматизированной информационной системы для службы «Техническая поддержка».

Работа состоит из трех глав. В первой главе приводится исследование предметной области – технического отдела. А также сформулированы основные требования к разрабатываемому программному обеспечению, цель разработки системы, определены необходимые функциональные возможности и особенности работы. Разработаны модели бизнес-процессов использования информационной системы в образовательных целях и редактирования информации «Как должно быть» с учетом применения в нем разрабатываемого ПО.

Вторая глава содержит описание практической части, а именно: логическое и физическое моделирование предметной области, описание технологического обеспечения задачи, демонстрация контрольного примера.

Третья глава служит для оценки и обоснования экономической эффективности проекта. На базе представленных расчетов можно говорить о том, что проект экономически эффективен.

Результатом бакалаврской работы является автоматизированная система для службы технической поддержки.

В выпускной квалификационной работе содержатся 61 страница с приложениями, 28 рисунков, 10 таблиц, 45 литературных источников.

Оглавление

Введение.....	4
Глава 1 Функциональное моделирование информационной системы для службы «Техническая поддержка».....	6
1.1 Техничко-экономическая характеристика деятельности ИП Ибрагимов К.С.....	6
1.2 Концептуальное моделирование системы для службы «Техническая поддержка»	8
1.3 Анализ существующих разработок и обоснование выбора технологии проектирования	12
1.4 Постановка задачи на разработку автоматизированной информационной системы	14
Глава 2 Разработка и реализация проектных решений	17
2.1 Логическое моделирование автоматизированной системы «Техническая поддержка».....	17
2.2 Физическое моделирование автоматизированной системы «Техническая поддержка».....	22
2.3 Технологическое обеспечение задачи	28
2.4 Контрольный пример реализации проекта и его описание	30
Глава 3 Оценка и обоснование экономической эффективности проекта	42
3.1 Выбор и обоснование методики расчета экономической эффективности.....	42
3.2 Расчет показателей экономической эффективности проекта.....	42
Заключение	51
Список используемой литературы	52
Приложение А – Программный код.....	56

Введение

Для успешного функционирования отдела технической поддержки его руководитель должен грамотно планировать свою работу.

Большинство современных компаний предпочитает использовать коммерческие автоматизированные системы технической поддержки, которые требуют достаточно высоких затрат. Небольшие компании не способны позволить себе подобных решений, поэтому они вынуждены использовать более упрощенные аналоги [28].

Все вышесказанное объясняет актуальность выбранной темы – использование автоматизированных систем технической поддержки в настоящее время не только позволяет экономить время, затрачиваемое на различные рабочие процессы, но и повышает показатель эффективности труда.

Таким образом, внедрение автоматизированной системы технической поддержки приведет к сокращению времени сотрудников, затрачиваемому на рутинные операции, следовательно, повысится продуктивность реальной деятельности.

Данная выпускная квалификационная работа посвящена разработке автоматизированной информационной системы для службы «Техническая поддержка».

Объектом исследования данной работы является деятельность службы «Техническая поддержка».

Предметом исследования является автоматизация деятельности сотрудников службы «Техническая поддержка».

Целью данной работы является разработка информационной системы для службы «Техническая поддержка».

Для достижения поставленной цели необходимо решить следующие задачи:

- выделить основные бизнес-процессы службы технической поддержки и провести их анализ;
- сформулировать постановку задачи на разработку ИС с описанием требований;
- провести проектирование информационной системы;
- выбрать средства реализации информационной системы;
- реализовать информационную систему;
- подсчитать экономическую эффективность от внедряемого проекта.

Результатом выпускной квалификационной работы будет являться автоматизированная система для службы «Техническая поддержка».

Структурно работа состоит из введения, трех разделов основной части, заключения, списка использованной литературы и приложений.

Первая глава аналитическая, вторая – практическая, третья - экономическая.

Глава 1 Функциональное моделирование информационной системы для службы «Техническая поддержка»

1.1 Техничко-экономическая характеристика деятельности ИП Ибрагимов К.С.

1.1.1 Характеристика деятельности ИП Ибрагимов С.К.

Основной деятельностью рассматриваемого предприятия, «ИП Ибрагимов С.К.», является розничная торговля автомобильными принадлежностями. На рисунке 1 представлена организационная структура ИП Ибрагимов С.К.



Рисунок 1 - Организационная структура ИП Ибрагимов С.К.

Работа над проектом велась в ИТ-отделе предприятия, основными функциями которого являются:

- обеспечение правильной эксплуатации и бесперебойной работы всех сотрудников и клиентов компании;
- обеспечение технического обслуживания компьютеров, вычислительной техники, серверного оборудования и программного обеспечения;

- планирование приобретения и распределения средств организационной и вычислительной техники, а также программного обеспечения;
- внедрение и сопровождение программных продуктов;
- установка на серверы, персональные компьютеры и рабочие станции, пользовательские и сетевые программы;
- осуществление контроля за использованием сетевых ресурсов, выявление ошибок, которые пользователи допускают в сети.

1.1.2 Сущность задачи автоматизации

Задача управления службой технической поддержки является очень важной для всех информационных систем. В задачи руководителя данной службы входит:

- управление инцидентами и запросами на обслуживание;
- управление существующими и новыми проблемами;
- управление каталогом услуг;
- контроль за учетом рабочего времени;
- участие и управление IT и бизнес-проектами;
- полнофункциональная замена сотрудников технической поддержки при необходимости;
- управление взаимоотношениями с внешними поставщиками услуг;
- координация специалистов технической поддержки;
- разработка отчетности по уровню предоставляемых услуг;
- составление планов развития подчиненных [36].

Автоматизация процесса работы службы «Техническая поддержка» в ИП Ибрагимов С.К. позволит оптимизировать существующую деятельность соответствующего отдела, а именно: вести единую картотеку ремонта, формировать различную отчетность, что позволит сократить не только время

выполнения основных технологических операций, но и случаи ошибок, возникающих под воздействием человеческого фактора. Кроме того, внедрение автоматизированной системы позволит сократить время поиска информации, за счет чего повысится производительность отдела.

Актуальность рассматриваемой задачи очевидна. Использование современных информационных технологий позволит ускорить процесс формирования печатных документов и сводной отчетной документации с использованием единой информационной базы, за счет чего появится возможность мгновенного доступа к требуемым данным, расположенным в информационной системе, анализ которых позволит принимать управленческие решения для оптимизации деятельности и своевременной закупки запасных деталей [32].

1.2 Концептуальное моделирование системы для службы «Техническая поддержка»

1.2.1 Выбор методологии/технологии моделирования бизнес-процессов

В настоящее время существует несколько различных методологий моделирования бизнес-процессов.

«Система ARIS является комплексом средств для анализа и моделирования деятельности предприятия и разработки систем автоматизации. Состоящая из множества модулей, она позволяет осветить различные аспекты деятельности организации. Для описания бизнес-процессов в виде последовательности действий применяется модуль ARIS eEPC» [10].

«Методология IDEF представляет собой семейство методов, применяемых для процесса моделирования. Эта технология активно используется, начиная с конца 1980-х годов» [10].

«IDEF0 – представляет собой графический язык, позволяющий сформировать модель системы, в которой описывается функции и структура системы, подчиненность объектов системы, а также потоки информации и материальных объектов, связывающие эти объекты» [10].

«IDEF3 – позволяет документировать технологические процессы. В IDEF3 описывается сценарий и последовательность действий для каждого процесса» [10].

Их сравнительная характеристика представлена в таблице 1.

Таблица 1 - Методологии моделирования бизнес-процессов

Критерий сравнения	ARIS	IDEF0	IDEF3
Принцип построения	Временная последовательность выполнения процедур	Принцип доминирования	Временная последовательность выполнения процедур
Описание процедуры	Объект диаграммы	Объект диаграммы	Объект диаграммы
Входящий документ	Отдельный объект	Стрелки	-
Исходящий документ	Отдельный объект	Стрелки	-
Исполнитель	Отдельный объект	Стрелки	-
Контроль выполнения	-	Стрелки	-
Обратная связь	-	Стрелки	-

Очевидно, что методология IDEF0 предоставляет наиболее широкие возможности моделирования бизнес-процессов, что и является ее ключевым преимуществом.

1.2.2 Моделирование бизнес-процессов

В настоящее время деятельность службы технической поддержки основана на индивидуальном взаимодействии с клиентами, поступающими на вход данного процесса (см. рисунок 2).



Рисунок 2 - Контекстная диаграмма бизнес-процесса «Обработка заявок»

Клиенты создают заявки в службу технической поддержки, сотрудники соответствующего отдела их обрабатывают и оформляют отчеты о проделанной работе. Это представлено на декомпозированной диаграмме (рисунок 3).

Диаграмма «Как есть» отражает несовершенство имеющейся системы – сотрудники используют разрозненные файлы и хранилища данных, что не является оптимальным решением задачи и требует большого количества времени для обработки.

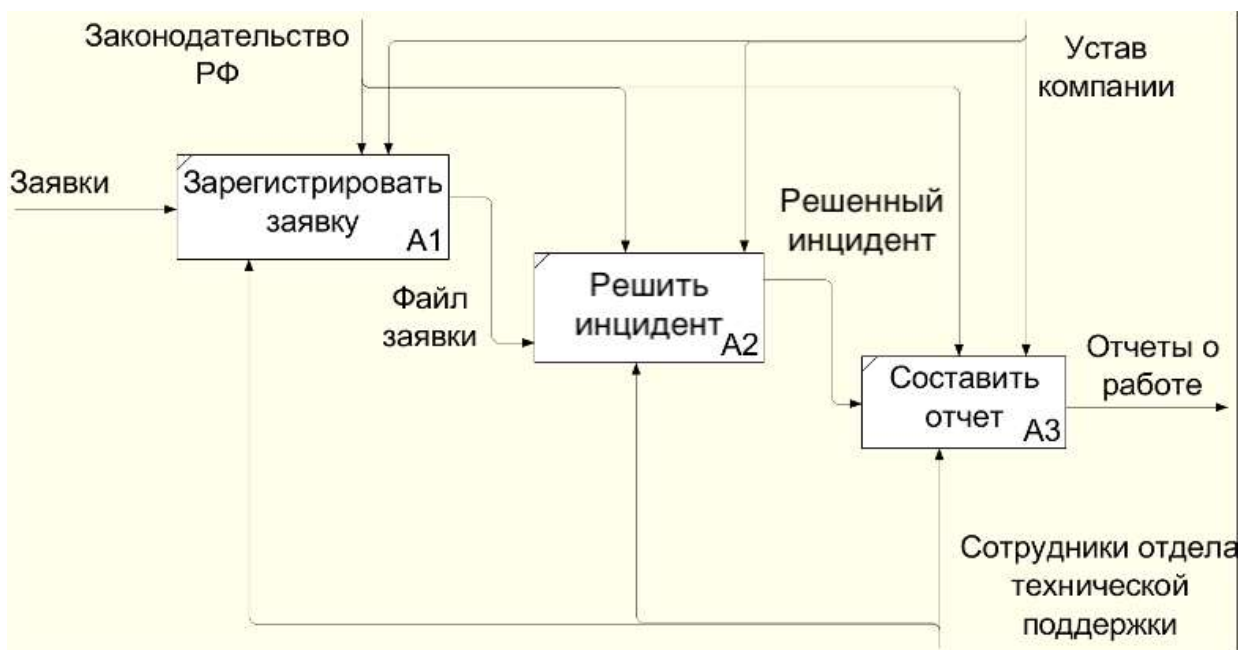


Рисунок 3 – Декомпозиция бизнес-процесса «Обработка заявок»

Для понимания как можно усовершенствовать существующий бизнес-процесс представлена модель «Как должно быть» (рисунки 4-5).



Рисунок 4 – Диаграмма бизнес-процесса «Как должно быть»

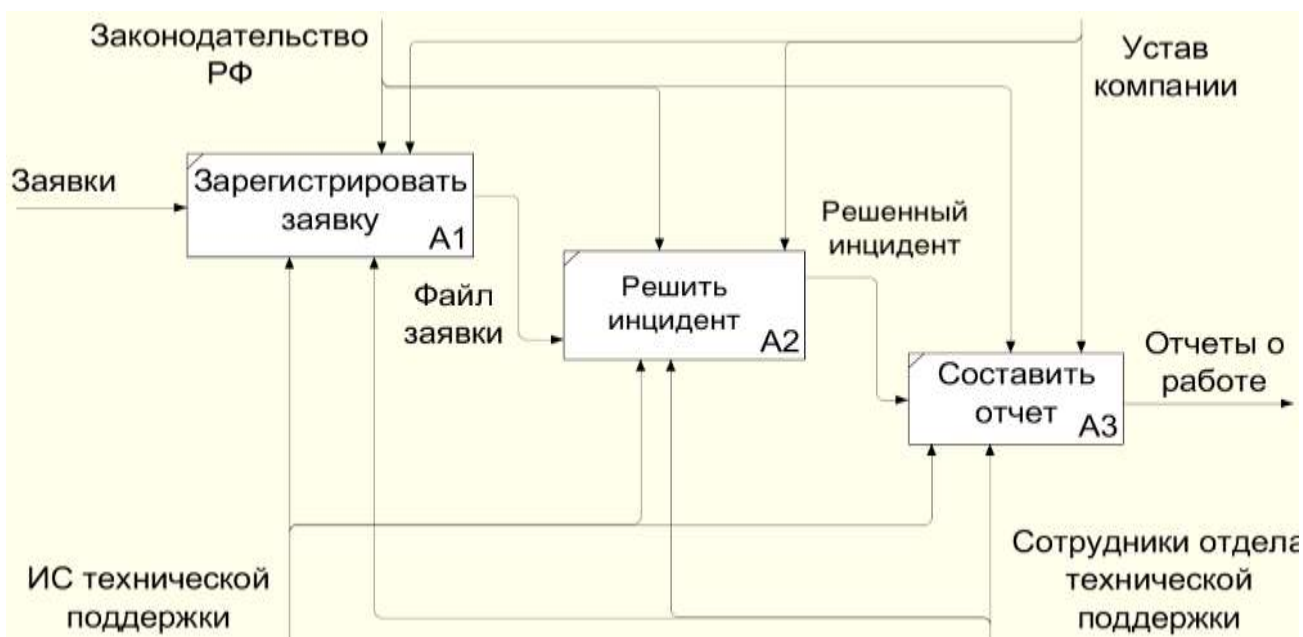


Рисунок 5 - Декомпозиция диаграммы «Как должно быть»

Отличительной чертой данной модели является наличие информационной системы, на которую перекладываются задачи регистрации заявок и формирования отчетов, за счет чего сотрудники отдела технической поддержки будут заниматься непосредственно обработкой заявок.

Таким образом, в процессе работы службы технической поддержки появится взаимодействие с новой АИС, в результате чего повысится эффективность ее работы.

1.3 Анализ существующих разработок и обоснование выбора технологии проектирования

При решении вопроса покупки любого программного обеспечения необходимо сравнивать следующие характеристики:

- предоставляемые функциональные возможности;

- стоимость лицензии.

Именно эти два пункта являются основой дальнейшего анализа.

В современном мире существует множество специализированных решений, выполняющих функции системы Service Desk. Одной из компаний, занимающихся подготовкой и реализацией данных решений, является Hewlett-Packard (HP).

Решение компании HP носит название «OpenView Service Desk» - это самостоятельное решение, предназначенное не только для автоматизации служб технической поддержки, но и для внедрения процессов управления ИТ-услугами.

Инициатор заявки при этом заполняет следующие поля:

- планируемые затраты;
- ожидаемая дата завершения работы;
- максимально возможное время на выполнение.

Еще один популярный программный продукт, используемый в службах технической поддержки - JIRA Service Desk – онлайн-сервис для управления заявками, поддержкой и автоматизации сервисной службы с понятным самообслуживанием. Данное решение наиболее популярно в большинстве крупных корпораций по всему миру. Диаграмма работы с инцидентами в JIRA Service Desk представлена на рисунке 6.

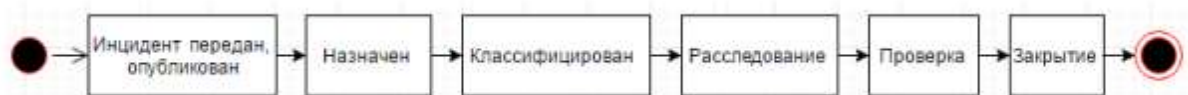


Рисунок 6 – Диаграмма, описывающая процесс ответа на инцидент в JIRA Service Desk

Сравнительный анализ программных продуктов представлен в таблице 1.

Таблица 2 - Сравнительная характеристика существующих программных продуктов

Категории сравнения	HP OpenView Service Desk	JIRA Service Desk
Интеграция с другими сервисами	+	+
Сложность внедрения	Легко	Сложно
Удобство интерфейса	Интуитивно-понятный	Интуитивно-понятный
Возможность самостоятельной настройки	+	+
Стоимость лицензии	До 45\$/шт.	До 20\$/шт.

Сравнение данных разработок подтверждает факт необходимости использования собственной АИС, реализующей только нужный функционал.

1.4 Постановка задачи на разработку автоматизированной информационной системы

Согласно методологии FURPS+ при постановке задачи необходимо учесть ряд факторов:

- Функциональные требования.

Разрабатываемая система должна предоставлять разграничение прав доступа – различные группы пользователей должны обладать различными правами (руководитель, техник, пользователь).

Пользователям необходимо предусмотреть возможность создавать заявки, техникам добавить возможность обрабатывать заявки.

Для руководителя требуется добавить возможность доступа ко всей информации и составлять отчеты.

- Требования к удобству работы.

В результате разработки автоматизированной системы изменится процесс обработки входной информации – все существующие данные переместятся в базу данных, где будут храниться и обрабатываться в едином формате.

Стоит отметить, что процесс переноса данных потребует дополнительных временных затрат, однако, в дальнейшем весь ввод исходной информации автоматизируется посредством специальных форм системы.

- Требования к производительности.

Разрабатываемая АИС должна работать под управлением ОС семейства Microsoft Windows с минимальным объемом оперативной памяти 512 Mb.

- Требования к простоте поддержки.

Исходный код разрабатываемой АИС должен быть прокомментирован, легко читаем и расширяем с целью возможных дальнейших доработок.

Процесс разработки автоматизированной информационной системы состоит из ряда этапов:

- изучение деятельности ИП Ибрагимов С.К.;
- формулирование задачи;
- проведение анализа существующих систем, позволяющих решить данную задачу;
- проведение анализа существующих средств разработки;
- определение ресурсов;
- формулирование списка входных и выходных данных;
- проектирование информационной модели;
- создание базы данных;
- создание приложения для работы с базой данных;
- обоснование экономической эффективности проекта.

Таким образом, была описана постановка задачи, которая должна быть выполнена в рамках работы над ВКР. Слабым местом проекта можно назвать процесс переноса имеющихся данных в будущую базу данных.

Выводы по главе 1

В данной главе была исследована предметная область – технический отдел. С учетом созданных концептуальных моделей и проведенного анализа были сформулированы основные требования к разрабатываемому ПО, цель разработки системы, а также определены необходимые функциональные возможности и особенности работы.

Созданы модели бизнес-процессов использования информационной системы в образовательных целях и редактирования информации «Как должно быть» с учетом применения в нем разрабатываемого ПО.

Глава 2 Разработка и реализация проектных решений

2.1 Логическое моделирование автоматизированной системы «Техническая поддержка»

2.1.1 Логическая модель и ее описание

Взаимодействие сущностей автоматизированной системы отображают логические модели. Они строятся исходя из модели «Как должно быть».

Графическая нотация стандарта UML включает в себя широкий набор диаграмм, позволяющих описать требуемые аспекты разрабатываемой системы. Набор диаграмм, используемых при разработке конкретного программного продукта, определяется разработчиком, в зависимости от требований к проекту и необходимого уровня полноты описания. Использование диаграмм нотации UML при разработке является общепринятым решением, поскольку они обеспечивают достаточно подробное описание требуемых спецификаций при сохранении простоты перевода моделей в программный код. В данной работе для общего описания функционала продукта и существующих ролей используется диаграмма вариантов использования (рисунок 7) [4].

На рисунке 7 изображена диаграмма вариантов использования будущей системы.

Из данной диаграммы видно, что в системе предполагается использование трех ролей с различными функциями:

- администратор - управление списком пользователей, ведение справочника категорий технических средств, ведение реестра технических средств;

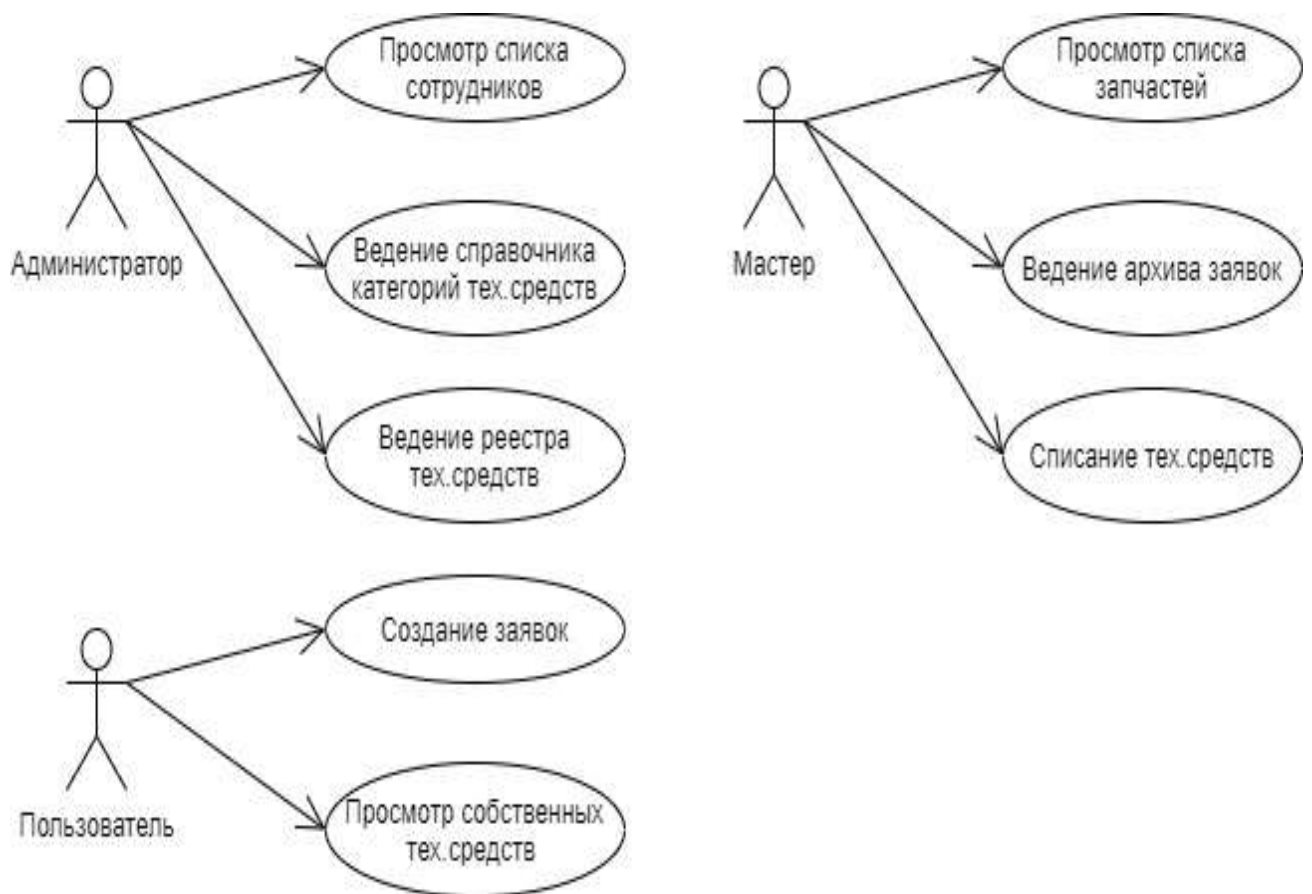


Рисунок 7 - Диаграмма вариантов использования

- мастер - работа со списком запчастей, ведение архива заявок, реализация списания технических средств;
- пользователь - создание заявки, просмотр списка закрепленных за ним технических средств.

Основу нормативно-справочной информации составляют данные ИП Ибрагимов С.К. - данные о сотрудниках, их оборудовании, необходимых запчастях.

2.1.4 Характеристика базы данных

Схемы таблиц спроектированной базы данных отражены в таблицах 3-7.

Таблица 3 – «Категории»

Идентификатор	Тип данных	Комментарий
c_id	int not null auto_increment	Идентификатор
c_name	varchar(50) not null	Категория

Таблица 4 – «Сотрудники»

Идентификатор	Тип данных	Комментарий
e_id	int not null auto_increment	Идентификатор
e_furname	varchar(50) not null	Фамилия
e_firstname	varchar(50) not null	Имя
e_secondname	varchar(50) not null	Отчество
e_post	varchar(50) not null	Должность
e_phone	varchar(12) not null	Контактный телефон
e_login	varchar(10) not null	Логин
e_password	varchar(10) not null	Пароль
e_access	int not null	Уровень доступа

Таблица 5 – «Оборудование»

Идентификатор	Тип данных	Комментарий
e_id1	int not null auto_increment	Уникальный идентификатор
e_c_id	int not null	Идентификатор категории
e_id	int not null	Идентификатор устройства
e_name	varchar(50) not null	Название
e_techdescr	varchar(250)	Описание
e_status	varchar(20) not null	Статус
e_issdate	date not null	Дата ввода в эксплуатацию

Таблица 6 – «Заявки»

Идентификатор	Тип данных	Комментарий
o_id1	int not null auto_increment	Уникальный идентификатор
e_id1	int not null	Идентификатор оборудования
o_create	date not null	Дата заведения заявки
o_complete	date	Дата закрытия заявки
o_description	varchar(256) not null	Описание проблемы
o_note	varchar(256)	Комментарий

Таблица 7 – «Запчасти»

Идентификатор	Тип данных	Комментарий
t_id	int not null auto_increment	Уникальный идентификатор
t_o_id	int not null	Идентификатор оборудования
t_name	varchar(256) not null	Наименование
t_description	varchar(256)	Описание
t_price	float(8, 2) not null	Стоимость
t_count	int not null	Количество

Для создания связи между таблицами использовались следующие команды:

- таблицы «employee» и «equipment» - alter table equipment add constraint FK_employee_equipment foreign key (e_id) references employee (e_id) on delete restrict on update restrict;
- таблицы «category» и «equipment» - alter table equipment add constraint FK_equipment_category foreign key (e_c_id) references category (c_id) on delete restrict on update restrict;

- таблицы «orders» и «equipment» - alter table orders add constraint FK_equipment_order foreign key (e_id1) references equipment (e_id1) on delete restrict on update restrict;
 - таблицы «orders» и «tools» - alter table tools add constraint FK_order_tool foreign key (t_o_id) references orders (o_id) on delete restrict on update restrict.
- Схема полученной базы данных представлена на рисунке 8.

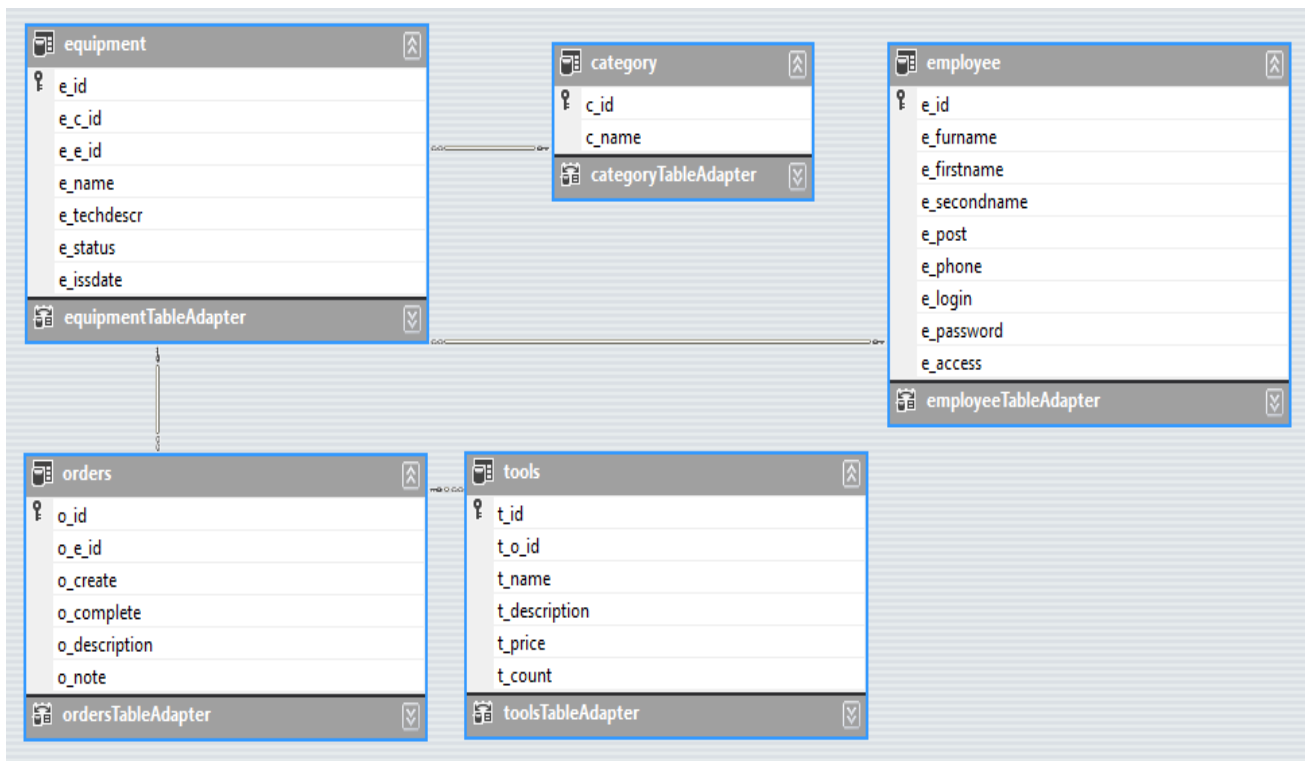


Рисунок 8 – Схема данных

Результатная информация разрабатываемой системы представляется в виде отчетов. К данной информации относятся:

- данные о заявках;
- данные о запчастях.

Таким образом, была представлена структура данных разрабатываемой информационной системы. Показано, что информационная база должна содержать таблицы с данными о сотрудниках, оборудовании и запчастях.

2.2 Физическое моделирование автоматизированной системы «Техническая поддержка»

2.2.1 Выбор архитектуры АИС

Существует несколько видов архитектуры АИС:

- файл-серверная – использует один компьютер для реализации функций обработки данных и диалогов [3];
- клиент-серверная – построена на принципе разделения компонентов приложения и их размещения там, где они будут функционировать наиболее эффективно. В таких архитектурах сервер баз данных реализуется отдельно от клиентских компьютеров [16];
- многоуровневая – системы с многоуровневой архитектурой состоят из трех уровней:
 - «клиентские приложения (нижний уровень)» [29];
 - «сервер приложений, на котором выполняется прикладная логика (средний уровень)» [29];
 - «удаленный сервер для файловых операций (верхний уровень)» [29];
- технологии интернет/интранет – системы, применяемые в браузерных приложениях.

В рамках выполнения данной работы используется файл-серверная архитектура.

2.2.2 Функциональная схема проекта

Для определения основных функций ПО, на рисунке 9 представим функциональную схему проекта.



Рисунок 9 - Функциональная схема проекта

Более подробный перечень функций для каждой группы пользователей описан в разделе 2.4.

2.2.3 Структурная схема проекта

На рисунке 10 представлена структурная схема классов проекта, которая демонстрирует общую структуру иерархии классов системы.

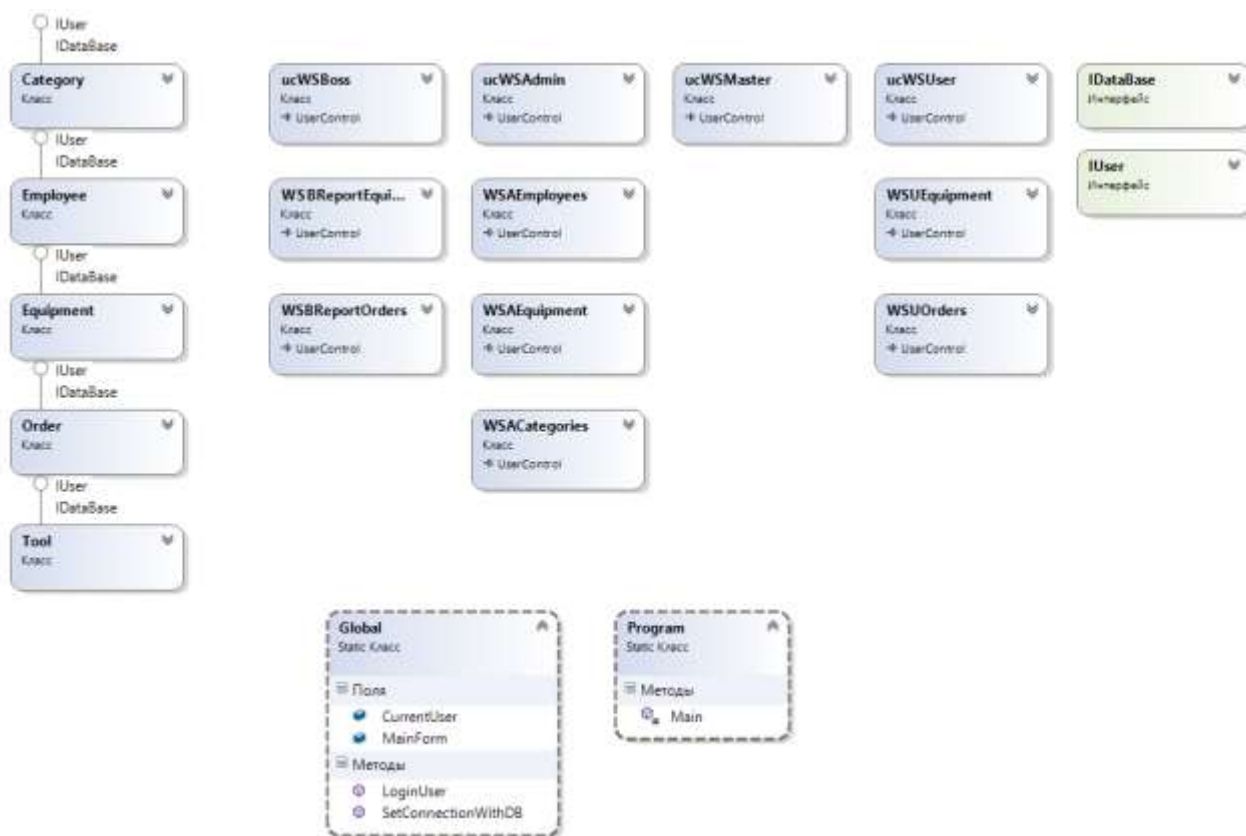


Рисунок 10 - Структурная схема проекта

Описание данной схемы представлено в п. 2.2.4 Описание программных модулей.

2.2.4 Описание программных модулей

Для работы с базой данных был создан интерфейс IDataBase, определяющий следующие методы:

- Initialize – инициализация объекта;
- InsertRow – вставка записи в БД;
- UpdateRow – обновление записи в БД;
- DeleteRow – удаление записи из БД.

Для работы с пользователями был создан интерфейс IUser, определяющий следующие методы:

- CreateNew – добавление пользователя;
- EditData – редактирование данных о пользователе;
- Delete – удаление данных о пользователе.

Для единообразия диалоговых окон созданы специальные классы:

- DLG_CATEGORY – ввод и редактирование данных о категории;
- DLG_COMPLETE – ввод комментария к завершению заявки;
- DLG_EMPLOYEE – ввод и редактирование данных сотрудника;
- DLG_EQUIPMENT – ввод и редактирование данных технического средства;
- DLG_ORDER – ввод и редактирование данных заявки;
- DLG_TOOL – ввод и редактирование данных компонента.

Сущности, реализованные в виде классов:

- Category – класс представления и манипулирования данными категорий технических средств. Атрибуты данного класса:
 - id – идентификатор;
 - name – наименование;
- Employee – класс представления и манипулирования данными сотрудника.

Атрибуты данного класса:

- id – идентификатор;
- furname – фамилия;
- firstname – имя;
- secondname – отчество;
- post – должность;
- phone – номер телефона;
- login – логин пользователя;
- password – пароль;
- access – уровень привилегий;

- Equipment – класс представления и манипулирования данными технических средств. Атрибуты данного класса:
 - id – идентификатор;
 - name – наименование;
 - techdescr – техническое описание;
 - status – статус;
 - issdate – дата ввода в эксплуатацию;
 - owner – ответственный владелец;
 - category – категория;
- Order – класс представления и манипулирования данными заявки. Атрибуты данного класса:
 - id – идентификатор;
 - descr – описание;
 - note – комментарии;
 - createdate – дата создания;
 - completedate – дата завершения;
 - tech – техническое средство;
- Tool – класс представления и манипулирования данными компонента для ремонта. Атрибуты данного класса:
 - id – идентификатор;
 - descr – описание;
 - name – наименование;
 - price – стоимость;
 - count – количество;
 - order – заявка;
 - default_order_id – номер заявки по умолчанию.

Для разграничения прав доступа выделено четыре группы пользователей, для каждой из которых спроектирована своя графическая подсистема:

- подсистема администратора;
- подсистема работы руководителя;
- подсистема мастера;
- подсистема сотрудника.

Также стоит обратить внимание на глобальный класс системы – Global, отвечающему за реализацию авторизации. Поля данного класса:

- Entity.Employee CurrentUser – текущий пользователь системы;
- F_MAIN MainForm – ссылка на главную форму для конкретного пользователя.

Методы класса:

- SetConnectionWithDB – попытка установить подключение к базе данных с проверкой введенных параметров;
- LoginUser – попытка авторизации пользователя с проверкой введенных данных и, в случае успеха, отрисовкой формы в соответствии с правами пользователя.

2.2.5 Схема взаимосвязи программных модулей и информационных файлов

На рисунке 11 изображена схема программных элементов.

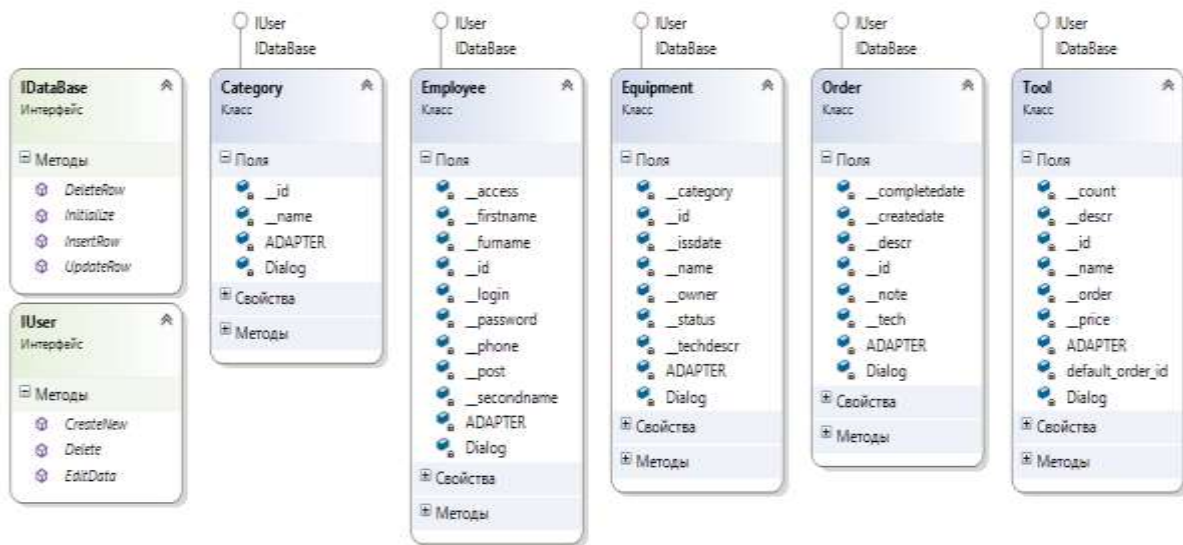


Рисунок 11 – Основные программные элементы

Данная схема отражает основные программные элементы и их поля.

Таким образом, был описан процесс физического моделирования предметной области.

2.3 Технологическое обеспечение задачи

2.3.1 Организация технологии сбора, передачи, обработки и выдачи информации

После запуска приложения выводится графическая форма, реализующая следующие возможности:

- установить соединение с базой данных;
- авторизоваться;
- выйти из программы.

Взаимодействие пользователя с информационной системой реализовано в диалоговом и событийном режимах. Диалог предоставляет пользователю возможность выбора одной из нескольких альтернатив [10].

Событие представляет собой процесс, вызванный пользовательскими действиями. Частным случаем являются программные события – получение определенным полем фокуса редактирования или потеря фокуса ввода.

Работа с АИС реализована в интерактивном режиме. «Файлы оперативной информации представляют собой исходные ресурсы и подгружаются по запросу пользователя» [17].

2.3.2 Схема технологического процесса сбора, передачи, обработки и выдачи информации

На рисунке 12 изображена схема технологического обеспечения.

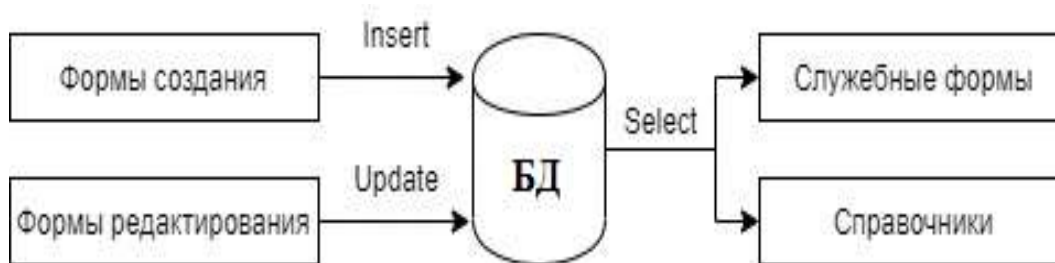


Рисунок 12 – Схема технологического процесса сбора, передачи, обработки и выдачи информации

Сбор информации происходит путем ввода данных на графических формах. Данные, введенные пользователем, сперва обрабатываются программным путем, а затем перенаправляются в БД в виде SQL-запросов.

Выдача информации происходит путем использования служебных форм и форм справочников [28].

Таким образом, было описано технологическое обеспечение задачи, а именно – процессы сбора, передачи, обработки и выдачи информации.

2.4 Контрольный пример реализации проекта и его описание

Данная система позволяет клиентам авторизоваться и отправить заявку в службу технической поддержки.

После запуска программного обеспечения выводится главное окно программы, которое предназначено для отображения всей информации системы. Окно разделено на несколько областей (рисунок 13).

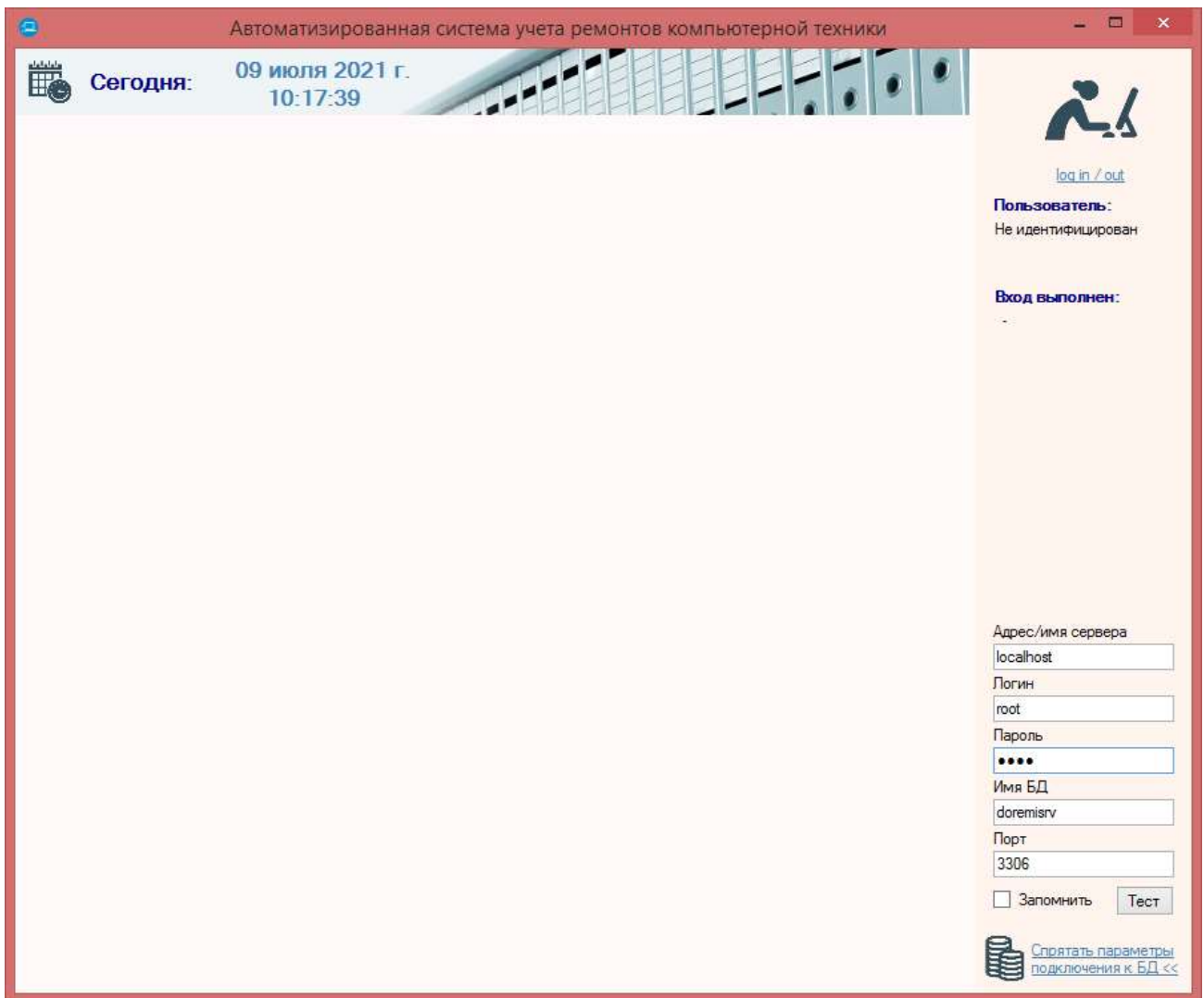


Рисунок 13 – Главное окно программы

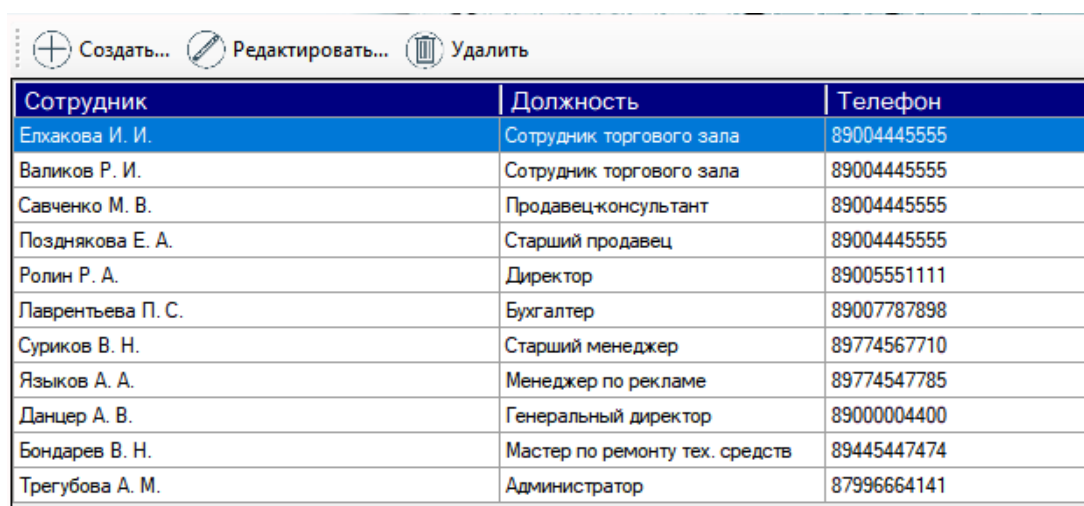
Правая часть окна приложения имеет пользовательскую панель, определяющую доступ к системе. На форме представлены следующие элементы:

- отображения данных текущего пользователя;
- авторизации в системе;
- настройки подключения к серверу базы данных: должны быть указаны адрес сервера, логин и пароль подключения к серверу, имя базы данных и порт; также имеется кнопка тестирования подключения к БД с введенными настройками; чтобы не вводить настройки подключения каждый раз, предусмотрена опция их записи в файл – server.dat, который находится в директории исполняемого файла приложения.

В системе существует 4 типа пользователя.

Подсистема администратора предназначена для работы администратора, в рамках которой осуществляется управление вспомогательными справочниками.

Для учета пользователей компьютерной техники администратор ведет учетную таблицу сотрудников. На рисунке 14 приведен пример учетной таблицы пользователей системы.



Сотрудник	Должность	Телефон
Елхакова И. И.	Сотрудник торгового зала	89004445555
Валиков Р. И.	Сотрудник торгового зала	89004445555
Савченко М. В.	Продавец-консультант	89004445555
Позднякова Е. А.	Старший продавец	89004445555
Ролин Р. А.	Директор	89005551111
Лаврентьева П. С.	Бухгалтер	89007787898
Суриков В. Н.	Старший менеджер	89774567710
Языков А. А.	Менеджер по рекламе	89774547785
Данцер А. В.	Генеральный директор	89000004400
Бондарев В. Н.	Мастер по ремонту тех. средств	89445447474
Трегубова А. М.	Администратор	87996664141

Рисунок 14 – Таблица учета пользователей

Для управления данными таблицы предусмотрено меню команд, расположенное вверху таблицы.

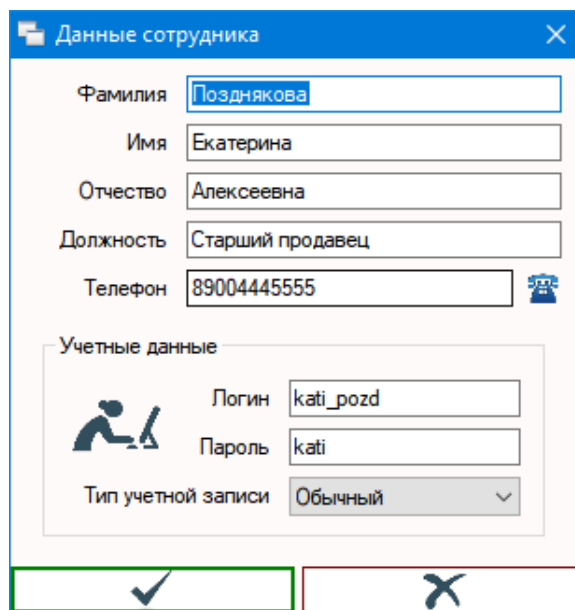
На рисунке 15 представлено диалоговое окно, предназначенное для ввода и редактирования данных пользователей. Диалог ввода данных предусматривает проверку введенных пользователем значений на полноту заполнения и корректность введенных данных.

С целью ведения учета категорий компьютерной техники администратор использует учетную таблицу категорий. На рисунке 16 приведен пример учетной таблицы категорий компьютерной техники.

Для управления данными таблицы предусмотрено меню команд, расположенное вверху таблицы.

В отдельном окне можно вводить и редактировать категорию компьютерной техники.

Диалог ввода данных предусматривает проверку введенных пользователем значений на полноту заполнения и корректность введенных данных.



The image shows a Windows-style dialog box titled "Данные сотрудника" (Employee Data). It contains several text input fields for personal and contact information, and a section for account details. At the bottom, there are two buttons: a green checkmark and a red 'X'.

Фамилия	<input type="text" value="Позднякова"/>
Имя	<input type="text" value="Екатерина"/>
Отчество	<input type="text" value="Алексеевна"/>
Должность	<input type="text" value="Старший продавец"/>
Телефон	<input type="text" value="89004445555"/>
Учетные данные	
Логин	<input type="text" value="kati_pozd"/>
Пароль	<input type="text" value="kati"/>
Тип учетной записи	<input type="text" value="Обычный"/>

Рисунок 15 – Диалог ввода данных пользователя

Наименование
Компьютер
МФУ
Сканер
Принтер
Кассовый аппарат

Рисунок 16 – Таблица учета категорий компьютерной техники

Для учета компьютерной техники администратор ведет учетную таблицу оборудования. На рисунке 17 приведен пример учетной таблицы компьютерной техники.

Тип	Описание	Ответственный владелец	Статус
Компьютер	Компьютер сотрудника торг. зала 1	Елхакова И. И.	списано
Компьютер	Компьютер сотрудника торг. зала 1	Елхакова И. И.	в работе
Компьютер	Компьютер сотрудника торг. зала 2	Валиков Р. И.	в работе
Компьютер	Компьютер продавца-консультанта	Савченко М. В.	в работе
Компьютер	Компьютер старшего продавца-консультанта	Позднякова Е. А.	в ремонте
Компьютер	Компьютер старшего продавца-консультанта	Позднякова Е. А.	в работе
Компьютер	Компьютер менеджеров	Позднякова Е. А.	списано
Компьютер	Компьютер директора	Ролин Р. А.	в работе
Компьютер	Компьютер бухгалтера	Лаврентьева П. С.	в работе
Компьютер	Компьютер старшего менеджера	Суриков В. Н.	в работе
Компьютер	Компьютер менеджера по рекламе	Языков А. А.	в работе
Компьютер	Компьютер генерального директора	Данцер А. В.	в работе
Компьютер	Компьютер мастера по ремонту технических средств	Бондарев В. Н.	в работе

Рисунок 17 – Таблица учета компьютерной техники

Управление данными таблицы предусмотрено в меню команд, которое расположено вверху таблицы.

С целью ввода и редактирования данных компьютерной техники используется соответствующий диалог (пример см. на рисунке 18).

Рисунок 18 – Диалог ввода данных компьютерной техники

Диалог ввода данных предусматривает проверку введенных пользователем значений на полноту заполнения и корректность введенных данных.

Представим состав действий в меню обычного пользователя: пользователь может получать актуальную информацию о технике, за которую он несет ответственность, а также о созданных им заявках на ремонт.

На рисунке 19 приведен пример таблицы технических средств пользователя – таблица доступна только для просмотра.

Тип	Наименование	Описание	статус
Компьютер	Компьютер старшего продавца-консультанта	Windows 7, HDD 500GB, CPU Core i2, RAM 4Gb	в ремонте
Компьютер	Компьютер старшего продавца-консультанта	Windows 10, HDD 1000GB, CPU Core i5, RAM 8Gb	в работе
Компьютер	Компьютер менеджеров	Windows 2000, 512 MB ОЗУ, 160 GB hdd	списано
Сканер	Сканер менеджеров	HP Laser Jet 11	в работе
Принтер	Принтер №1 торгового зала	Epson	в работе
Принтер	Принтер №2 торгового зала	HP	в ремонте

Рисунок 19 – Таблица просмотра подчиненной техники пользователя

Для учета заявок на ремонт компьютерной техники пользователя предусмотрена таблица заявок, созданных пользователем. Пример таблицы приведен на рисунке 20. Из таблицы пользователь может узнать статус своей заявки, а также увидеть комментарии мастера.

№	Создано	Завершено	Описание	Комментарий мастера
0001	06.05.2021	08.05.2021	Мерцает экран	Заменена видеокарта
0005	15.05.2021	15.05.2021	Сильно тормозит - невозможно работать	Списание ввиду сильного устаревания оборудования.
0006	15.05.2021	15.05.2021	Пропадает связь с компьютером	Заменен шнур USB (взят из запасов)
0002	12.05.2021		Плохо печатает	
0003	15.05.2021	15.05.2021	Перетерся кабель подключения в сеть - заменить	Новый кабель

Рисунок 20 – Таблица учета заявок пользователя

Управление данными таблицы предусмотрено в меню команд, которое расположено вверху таблицы.

Заявки, у которых поле «Завершено» не пусто, считаются выполненными. Такие заявки доступны только для просмотра.

Для ввода и редактирования данных заявки на ремонт техники используется соответствующий диалог (рисунок 21).

Диалог ввода данных предусматривает проверку введенных пользователем значений на полноту заполнения и корректность введенных данных.

При создании заявки на ремонт пользователь выбирает единицу техники, которая подлежит ремонту. После создания заявки выбранная техника автоматически помечается системой статусом «в ремонте».

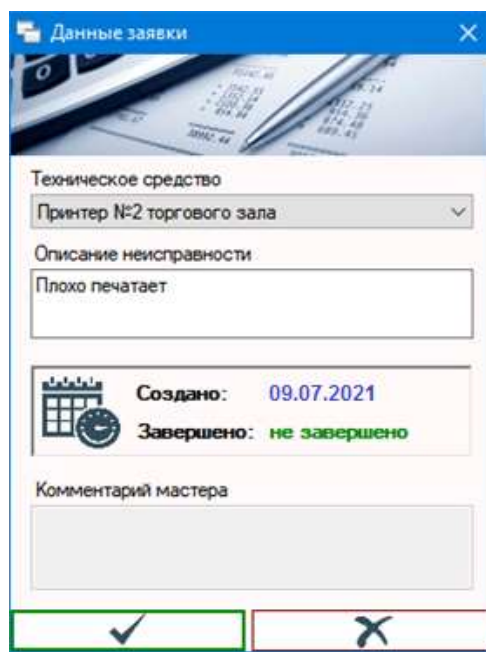


Рисунок 21 – Диалог создания заявки на ремонт

Подсистема мастера содержит все необходимые инструменты, чтобы оперативно реагировать и вести учет заявок пользователей на ремонт техники. Меню мастера позволяет:

- указать запчасти и комплектующие, необходимые и примененные при выполнении ремонта;
- показать текущие (актуальные) заявки пользователей на ремонт;
- показать архивные (выполненные) заявки пользователей на ремонт;
- завершить выбранную заявку на ремонт, пометив ее соответствующим статусом, информируя пользователя о выполнении заявки (соответствующая заявке единица техники также автоматически помечается статусом «в работе»);
- завершить выбранную заявку на ремонт со списанием техники, пометив ее соответствующим статусом, информируя пользователя о выполнении

заявки (соответствующая заявке единица техники также автоматически помечается статусом «в работе»);

- завершить выбранную заявку на ремонт со списанием техники, пометив ее соответствующим статусом, информируя пользователя о выполнении заявки (соответствующая заявке единица техники также автоматически помечается статусом «списано»);
- сформировать документ «карточка заявки».

Все актуальные или архивные заявки на ремонт техники отображаются на рабочем месте мастера в виде соответствующей учетной таблицы (рисунок 22), в которой мастер может выполнять доступные в меню команды над выбранной заявкой. Выбранной в данный момент считается заявка, в позиции которой находится курсор таблицы (соответствует выделенной строке).

Сегодня: 17 мая 2021 г. 0:28:31

№ заявки	Инв. №	Создано	Завершено	Тип	Описание
0001	0004	06.05.2021	10.05.2021	Компьютер	Компьютер старшего продавца консультанта
0004	0001	15.05.2021	15.05.2021	Компьютер	Компьютер сотрудника торг. зала 1
0005	0019	15.05.2021	15.05.2021	Компьютер	Компьютер менеджеров
0006	0018	15.05.2021	15.05.2021	Сканер	Сканер менеджеров
0002	0014	12.05.2021		Принтер	Принтер №2 торгового зала
0003	0013	15.05.2021	15.05.2021	Принтер	Принтер №1 торгового зала

log in / out

Пользователь:
Бондарев Владимир Николаевич

Ввод выполнен:
17 мая 2021 г. 0:25

Адрес/имя сервера
192.168.0.6

Логин
root

Пароль

Имя БД
dome1agv

Порт
3306

Запомнить

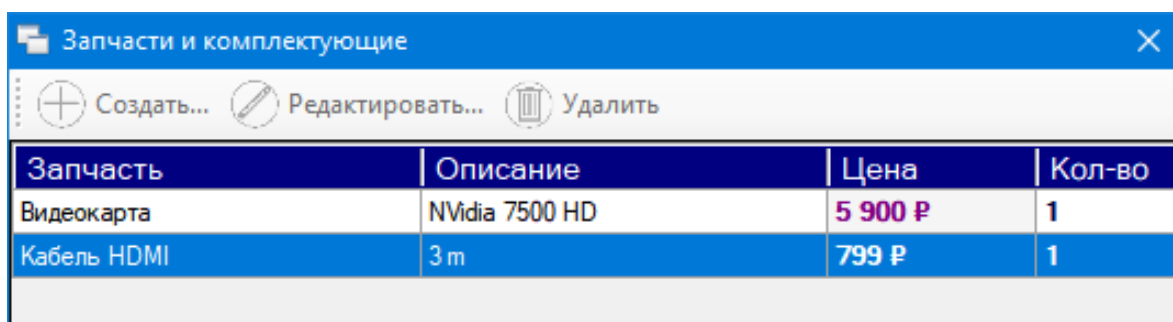
Справка, параметры подключения к БД <<

Рисунок 22 – Таблица учета заявок пользователей в подсистеме мастера

При условии, когда мастер инициирует завершение выполнения заявки по окончании ремонта или при списании техники (соответственно, команды «Завершить» и «Списать» из меню), система предлагает ввести краткий комментарий мастера к заявке, в которой может быть указан состав выполненных работ, рекомендации или другая информация на выбор мастера. Комментарий не обязателен (поле можно оставить пустым), однако для завершения заявки данный диалог должен быть подтвержден кнопкой применения.

По любой заявке может быть необходимым приобретение дополнительных запчастей и комплектующих, которые пойдут на замену неисправным. Для этого в системе есть возможность учета купленных компонентов по каждой заявке – соответствует самому первому пункту меню мастера. Все купленные для выполнения заявки компоненты представляются в виде учетной таблицы (рисунок 23). Для управления данными таблицы предусмотрено меню команд, расположенное вверху таблицы. Команды не доступны, если заявка была завершена, поскольку завершенные заявки доступны только для просмотра.

Для ввода и редактирования данных компонентов для заявки на ремонт техники используется соответствующий диалог.



Запчасть	Описание	Цена	Кол-во
Видеокарта	NVidia 7500 HD	5 900 Р	1
Кабель HDMI	3 м	799 Р	1

Рисунок 23 – Таблица учета комплектующих для ремонта

Диалог ввода данных предусматривает проверку введенных пользователем значений на полноту заполнения и корректность введенных данных.

Последней командой, доступной в меню мастера, является генерация отчетного документа по заявке – карточки заявки на ремонт. В данном документе указываются все необходимые сведения, касающиеся заявки, включая состав закупленных компонентов с подсчетом общей стоимости ремонта. Пример сгенерированной системой карточки заявки приведен на рисунке 24.

Карточка заявки №0001

Сохранить... Печать... Предпросмотр... Закрыть

Карточка заявки на ремонт № 00001 от 6 мая 2021 г.

1. Создал: Позднякова Е. А.
2. Тип: Компьютер: Компьютер старшего продавца-консультанта; инв.№: 0004
3. Описание неисправности: Мерцает экран
4. Комментарий мастера: Заменена видеокарта
5. Выполнено: 06.05.2021
6. Закупленные запчасти и комплектующие:

№п/п (1)	Компонент (2)	Стоимость (3)	Количество (4)	Сумма (5)
1	Видеокарта NVidia 7500 HD	5 900 Р	1	5 900 Р
2	Кабель HDMI 3 м	799 Р	1	799 Р

7. Итого:
Общее количество запчастей и комплектующих: 2
на сумму: 6699,00 руб.

Нач. техн. отд. / _____ /

Рисунок 24 – Форма карточки заявки

Руководитель нуждается в оперативном и быстром получении сводных данных по состоянию и эффективности работы технического отдела. Для этого

системное меню руководителя включает пункты, предоставляющие возможности доступа к формированию информации в виде сводных отчетов:

- по состоянию технических ресурсов Предприятия – позволяет обладать общей картиной о количестве и состоянии компьютерной техники пользователей;
- по выполненным ремонтам мастерами технического отдела – позволяет оценить работы выполненные, техническим отделом.

На рисунке 25 приведен пример формы сводных данных по состоянию технических ресурсов.

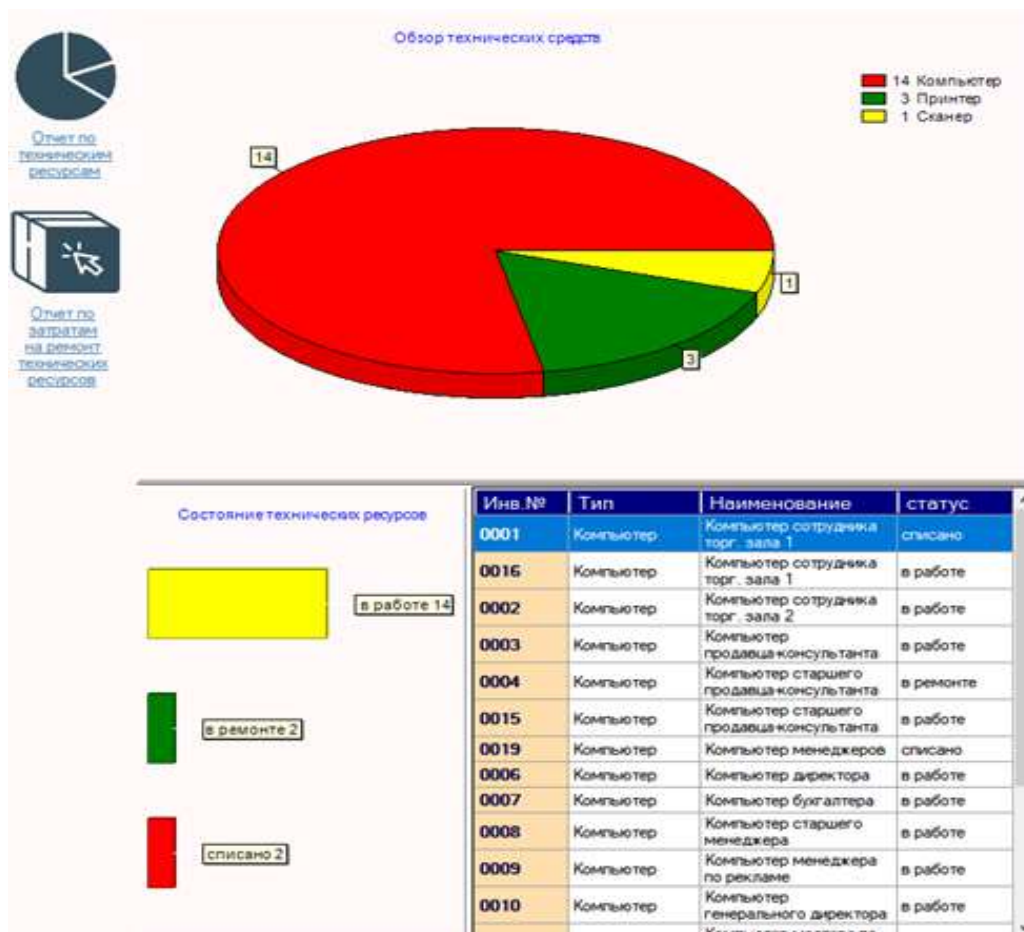


Рисунок 25 – Форма отчета по техническим ресурсам

Эти данные включают сводную таблицу, в которую сведены все технические ресурсы Предприятия с указанием их типов, назначения, инвентарного номера и текущего технического состояния, и элементы инфографики:

- диаграмму секторов, отражающую общее количество компьютерной техники на Предприятии по их категориям;
- диаграмму, отражающую общее количество компьютерной техники на Предприятии по их техническому состоянию.

В качестве итогов также подсчитываются общие затраты на все выполненные заявки.

Выводы по главе 2

В рамках второй главы проведено описание практической части, а именно: логическое и физическое моделирование предметной области, представление технологического обеспечения задачи, демонстрация контрольного примера.

Глава 3 Оценка и обоснование экономической эффективности проекта

3.1 Выбор и обоснование методики расчета экономической эффективности

«Показатель экономической эффективности применяется с целью оценки необходимости его внедрения. Базой исчисления экономической эффективности является сопоставление существующего метода обработки данных (базовый вариант) и внедряемого (проектный вариант)» [11].

Основой сравнения принято считать ручное выполнение работ по составлению и анализу отчетности в рамках работы службы технической поддержки до внедрения автоматизированной системы.

«Достижимый эффект вычисляется за счет сопоставления экономии от использования информации с затратами на ее получение» [11].

3.2 Расчет показателей экономической эффективности проекта

«Затраты на разработку проекта Q_p вычисляются по формуле:

$$Q_p = \sum t_i \quad (1)$$

где, t_i – i -й этап проекта» [11].

В таблице 8 представлены основные этапы проекта.

Таблица 8 - Этапы проекта

Этап	№ работы	Содержание работы	Трудоемкость	
			(чел-дни)	(чел-час)
Изучение деятельности ИП Ибрагимов С.К.	1	«Анализ бизнес-процессов	3	24
	2	Анализ используемого программного обеспечения	7	56
Проектирование и разработка БД	1	Сбор данных о сущностях	3	24
	2	Проектирование модели БД» [11]	4	32

Продолжение таблицы 8

Этап	№ работы	Содержание работы	Трудоемкость	
			(чел-дни)	(чел-час)
Проектирование и разработка программного продукта	1	«Сбор исходных требований»	3	24
	2	Анализ существующих инструментов для разработки	1	8
	3	Проектирование системы	7	56
	4	Разработка программы	21	168
Тестирование	1	Подготовка тестов	3	24
	2	Проведение тестирования	7	56
	3	Анализ полученных результатов	2	16
	4	Внесение изменений в программный код (при необходимости)» [11]	7	56

«Для расчета чел-часов, при стандартном рабочем дне, необходимо умножить количество чел-дней на рабочее время: $Q_p = 68$ чел-дней, либо 544 чел-часов» [11].

«Количество исполнителей, необходимых для реализации проекта, определяется формулой:

$$N = \frac{Q_p}{F}, \quad (2)$$

где Q_p – затраты труда» [11],

F – «фонд рабочего времени, вычисляемый по формуле:

$$F = T \cdot F_M, \quad (3)$$

где T – время выполнения проекта в месяцах» [11],

F_M – «фонд времени в текущем месяце, вычисляемый по формуле:

$$F_M = \frac{t_p \cdot (D_K - D_B - D_{\Pi})}{12} \quad (4)$$

где t_p – продолжительность рабочего дня,

D_K – количество дней в году,

D_B – количество выходных дней,

D_{II} – количество праздничных дней» [11].

Получаем, $F_M = \frac{8(365-118)}{12} \approx 165$.

Тогда, $F = 2 * 165 = 330$. Здесь 2 – количество месяцев, затраченных на реализацию проекта.

Число исполнителей $N = \frac{544}{330} \approx 1$.

Сетевая модель проекта.

На рисунке 26 изображен график сетевой модели.

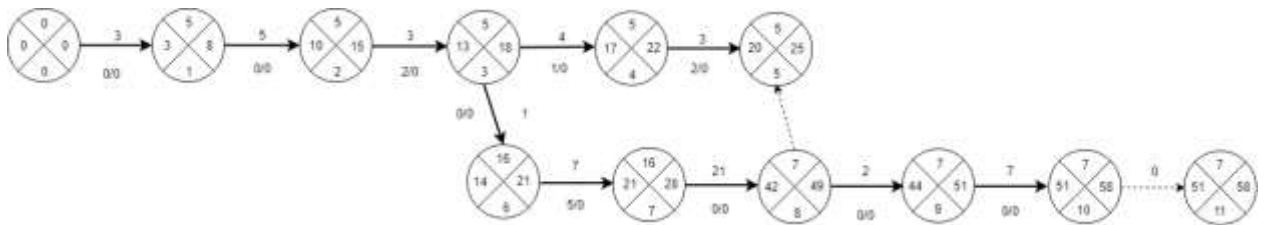


Рисунок 26 - Сетевая модель

Критический путь модели проходит через вершины: 0-1-2-3-4-5-6-7-8-9-10.

«Данные о работах, необходимых для построения сетевой модели и времени, затрачиваемом на выполнение указанных работ» [11], представлены в таблице 9.

Таблица 9 - Работы проекта

№	Событие	Код работы	Работа	t, чел-дни
00	«Начало проекта	0-1	Анализ бизнес-процессов	3
11	Бизнес-процессы проанализированы	1-2	Анализ используемого программного обеспечения	7
22	Используемое программное обеспечение проанализирована	2-3	Сбор данных о сущностях	3
33	Информация о сущностях собрана	3-4	Сбор исходных требований	4
44	Требования собраны	4-5	Проектирование модели БД	3

Продолжение таблицы 9

№	Событие	Код работы	Работа	t, чел-дни
55	Модель базы данных спроектирована	3-6	Выбор инструментов для разработки» [11]	1
66	«Инструменты для разработки системы определены	6-7	Проектирование клиентской части системы	7
77	Проект клиентской части составлен	7-8	Написание программного кода	21
88	Написание программного кода завершено	3-9	Подготовка тестов для системы	3
99	Получены результаты тестирования системы	8-9	Анализ результатов тестирования	2
110	Проведен анализ результатов тестирования	9-10	Внесение изменений в программный код (при необходимости)	7
11	Изменения в код внесены (при необходимости)	10-11	Ожидание внесения изменений» [11]	0

Ранние сроки наступления событий:

$$T_0^p = 0 \quad (5)$$

$$T_1^p = 0 + 3 = 3 \quad (6)$$

$$T_2^p = 3 + 7 = 10 \quad (7)$$

$$T_3^p = 10 + 3 = 13 \quad (8)$$

$$T_4^p = 13 + 4 = 17 \quad (9)$$

$$T_5^p = 17 + 3 = 20 \quad (10)$$

$$T_6^p = 13 + 1 = 14 \quad (11)$$

$$T_7^p = 14 + 7 = 21 \quad (12)$$

$$T_8^p = 21 + 21 = 42 \quad (13)$$

$$T_9^p = 42 + 2 = 44 \quad (14)$$

$$T_{10}^p = 44 + 7 = 51 \quad (15)$$

$$T_{11}^p = 51 + 0 = 51 \quad (16)$$

Поздние сроки наступления событий:

$$T_{11}^{\Pi} = 58 \quad (17)$$

$$T_{10}^{\Pi} = 58 \quad (18)$$

$$T_9^{\Pi} = 58 - 7 = 51 \quad (19)$$

$$T_8^{\Pi} = 51 - 2 = 49 \quad (20)$$

$$T_7^{\Pi} = 49 - 21 = 28 \quad (21)$$

$$T_6^{\Pi} = 28 - 7 = 21 \quad (22)$$

$$T_5^{\Pi} = 25 \quad (23)$$

$$T_4^{\Pi} = 25 - 3 = 22 \quad (24)$$

$$T_3^{\Pi} = 22 - 4 = 18 \quad (25)$$

$$T_2^{\Pi} = 18 - 3 = 15 \quad (26)$$

$$T_1^{\Pi} = 15 - 7 = 8 \quad (27)$$

$$T_0^{\Pi} = 0 \quad (28)$$

На рисунке 28 представлена диаграмма Ганта, отображающая график работы.

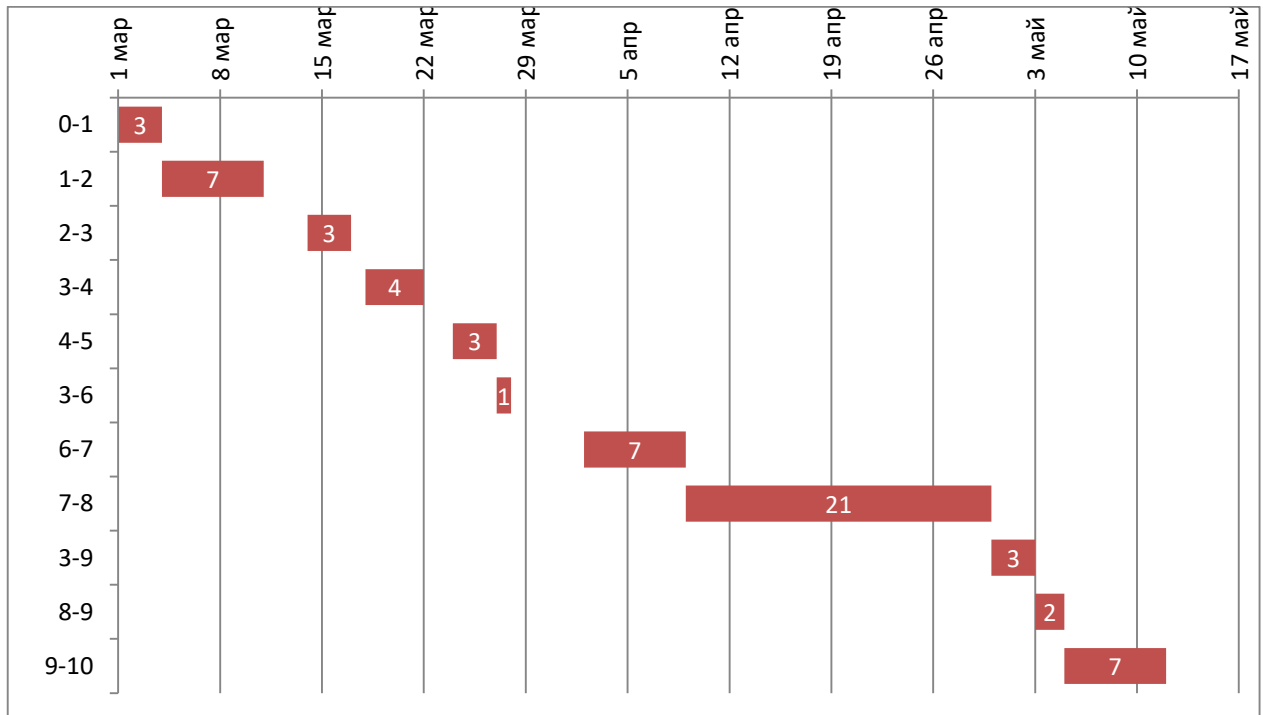


Рисунок 27 - Диаграмма Ганта

«Обозначения для работ взяты из таблицы 8.

Для вычисления затрат на выполнение проекта используется формула» [11]:

$$K = C_{\text{ЗАРП}} + C_{\text{ОБ}} + C_{\text{ОРГ}} + C_{\text{НАКЛ}} \quad (29)$$

«где, K – общие затраты,

$C_{\text{ЗАРП}}$ – заработная плата исполнителей,

$C_{\text{ОБ}}$ – затраты на оборудование,

$C_{\text{ОРГ}}$ – затраты на организацию рабочих мест,

$C_{\text{НАКЛ}}$ – накладные расходы» [11].

$C_{\text{ЗАРП}}$ определяется формулой:

$$C_{\text{ЗАРП}} = C_{\text{З.ОСН}} + C_{\text{З.ДОП}} + C_{\text{З.ОТЧ}} \quad (30)$$

«где, $C_{\text{З.ОСН}}$ и $C_{\text{З.ДОП}}$ – основная и дополнительная заработная плата соответственно,

$C_{\text{З.ОТЧ}}$ – отчисления с заработной платы» [11].

$C_{\text{З.ОСН}}$ вычисляется по формуле:

$$C_{\text{З.ОСН}} = T_{\text{ЗАН}} \cdot O_{\text{ДН}} \quad (31)$$

«где $T_{\text{ЗАН}}$ – количество отработанных дней исполнителя;

$O_{\text{ДН}}$ – дневной оклад» [11]:

$$O_{\text{ДН}} = \frac{O_{\text{МЕС}} \cdot 8}{F_{\text{М}}} \quad (32)$$

«где $O_{\text{МЕС}}$ – месячный оклад,

$F_{\text{М}}$ – фонд рабочего времени» [11].

$$O_{\text{МЕС}} = 18000 \cdot 1,13 = 20340 \text{ р.} \quad (33)$$

Получаем, $O_{\text{ДН}} = \frac{20340 \cdot 8}{165} \approx 986,2 \text{ р.}$

$$C_{\text{З.ОСН}} = 986,2 \cdot 68 = 67060,36 \text{ р.} \quad (34)$$

$$C_{\text{З.ДОП}} = 0,2 \cdot 67060,36 = 13412,07 \text{ р.} \quad (35)$$

$$C_{\text{З.ОТЧ}} = (67060,36 + 13412,07) \cdot 0,3 = 24141,73 \text{ р.} \quad (36)$$

где 0,3 – размер единого социального налога в 2021 году.

$$C_{\text{ЗАРП}} = 67060,36 + 13412,07 + 24141,73 = 104614,2 \text{ р.} \quad (37)$$

$$C_{\text{ОБ}} = 6000 \text{ р.} \quad (38)$$

Затраты на организацию рабочих мест $C_{\text{ОРГ}}$ определяются формулой:

$$C_{\text{ОРГ}} = \frac{C_{\text{КВМ}}}{12} S \frac{T_{\text{АР}} \cdot 8}{F_M} \quad (40)$$

где $C_{\text{КВМ}}$ – стоимость аренды квадратного метра площади за год,

S – площадь рабочего помещения,

$T_{\text{АР}}$ – срок аренды.

$$C_{\text{ОРГ}} = 1041 \cdot 40 \cdot \frac{68 \cdot 8}{165} = 137285,8 \text{ р} \quad (41)$$

Величина накладных расходов вычисляется по формуле:

$$C_{\text{НАКЛ}} = 0.6 \cdot C_{\text{З.ОСН}} \quad (42)$$

$$C_{\text{НАКЛ}} = 0.6 \cdot 67060,36 = 40236,22 \text{ р.}$$

Суммарные затраты на выполнение проекта равны:

$$K = 67060,36 + 6000 + 137285,8 + 40236,22 = 250582,4 \text{ р.} \quad (43)$$

«Затраты на внедрение разработанного ПО вычисляются по формуле» [11]:

$$C_{\text{ВН}} = C_{\text{ВН.ЗАРП}} + C_{\text{ВН.ОБ}} + C_{\text{ВН.ОРГ}} + C_{\text{ВН.НАКЛ}} \quad (44)$$

«где $C_{\text{ВН.ЗАРП}}$ – заработная плата исполнителей, участвующих во внедрении,

$C_{\text{ВН.ОБ}}$ – затраты на необходимое оборудование,

$C_{\text{ВН.ОРГ}}$ – затраты на обеспечение рабочих мест сотрудников, работающих с внедренным программным обеспечением,

$C_{\text{ВН.НАКЛ}}$ – накладные расходы при внедрении» [11].

Так как для использования внедряемого ПО не предполагается закупка нового оборудования и модернизация существующих рабочих мест, параметры $C_{\text{ВН.ОБ}}$ и $C_{\text{ВН.ОРГ}}$ приравниваются нулю. Таким образом, необходимо вычислить $C_{\text{ВН.ЗАРП}}$ и $C_{\text{ВН.НАКЛ}}$:

$$\begin{aligned} C_{\text{ВН.ЗАРП}} &= 986,2 \cdot 22 + 0,2 \cdot 986,2 \cdot 22 + \\ &+ (986,2 \cdot 22 + 0,2 \cdot 986,2 \cdot 22) \cdot 0,3 = 33845,76 \text{ р.} \end{aligned} \quad (45)$$

где 22 дня – время внедрения проекта.

$$C_{\text{ВН.НАКЛ}} = 986,2 \cdot 22 \cdot 0,6 = 13017,6 \text{ р.} \quad (46)$$

Тогда затраты на внедрение равны:

$$K_{\text{ВН}} = 33845,76 + 0 + 0 + 13017,6 = 46863,36 \text{ р.} \quad (47)$$

Общие затраты включают в себя затраты на разработку и внедрение:

$$K_{\text{ОБ}} = K_{\text{ВН}} + K$$

$$K_{\text{ОБ}} = 46863,36 + 250582,4 = 297445,8 \text{ р.} \quad (48)$$

«Для определения стоимости ПО применяется формула» [11]:

$$K_{\text{ПО}} = (\Delta K + K_{\text{ВН}}) \cdot (1 + D_{\text{ПРИБ}}) \quad (49)$$

«где ΔK – стоимость одной копии программы,

$K_{\text{ВН}}$ – затраты на внедрение» [11],

$D_{\text{ПРИБ}}$ – «процент прибыли от продажи, определяемый по формуле» [11]:

$$D_{\text{ПРИБ}} = \left(\frac{K_{\text{ПР}}}{\Delta K + K_{\text{ВН}}} - 1 \right) \cdot 100\% \quad (50)$$

$$D_{\text{ПРИБ}} = 15\%.$$

«Для вычисления прибыли, получаемой от продажи каждой установки программного продукта, используется следующая формула» [11]:

$$C_{\text{ПРИБ}} = K_{\text{ПР}} \cdot D_{\text{ПРИБ}} \cdot (1 - N_{\text{НДС}}) \quad (51)$$

где $N_{\text{НДС}}$ – величина налога на добавочную стоимость.

$$C_{\text{ПРИБ}} = 9000 \text{ р.} \quad (52)$$

Фрагмент таблицы расходов и прибыли по проекту приведен в таблице 9.

Таблица 10 - Фрагмент таблицы расходов и прибыли по проекту

Период расчета	Разработка (проект)				Прибыль	
	Частичная стоимость	Сальдо начальное по кредиту	Погашение кредита	Сальдо конечное по кредиту	Расчетная прибыль	Чистая прибыль
1-4. 200х	88906,12	-320062,03	88906,12	-231155,9	13779,38	11023,51
5-8. 200х	88906,12	-231155,9	88906,12	-142249,79	13779,38	11023,51

Продолжение таблицы 10

Период расчета	Разработка (проект)				Прибыль	
	Частичная стоимость	Сальдо начальное по кредиту	Погашение кредита	Сальдо конечное по кредиту	Расчетная прибыль	Чистая прибыль
9-12. 200х	88906,12	-142249,79	88906,12	-53343,67	13779,38	11023,51
1-3. 200х	88906,12	-53343,67	55006,12	0	49341,83	39473,47
4-6. 200х	88906,12	0	0	0	102685,5	82148,4

Кредит, упоминаемый в таблице, взят на 3 месяца, под 25% годовых.

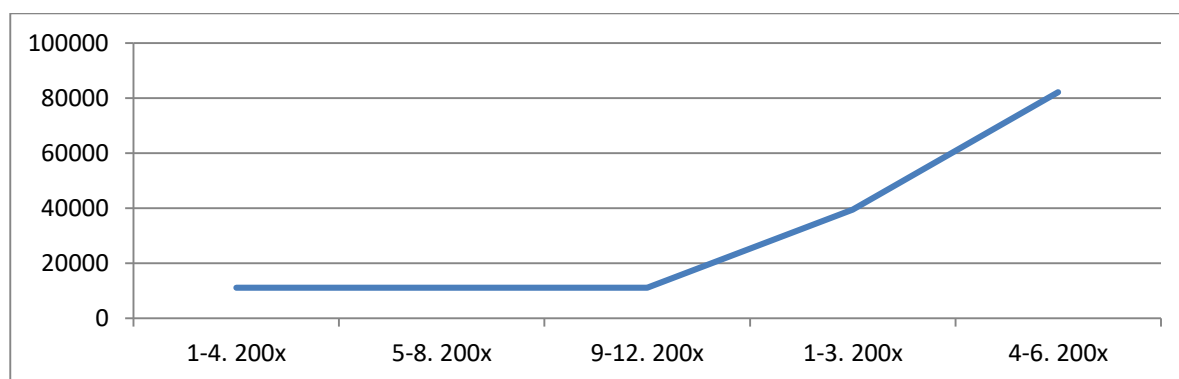


Рисунок 28 - Диаграмма изменения чистой прибыли

В данной главе приведены расчеты экономической эффективности разрабатываемого проекта. Срок выполнения проекта составляет 68 дней, суммарные затраты на проект составляют 297445,8 р.

В результате проведенных расчетов можно сделать вывод о том, что проектируемая система является эффективным решением задачи автоматизации.

Выводы по главе 3

Данная глава служит для оценки и обоснования экономической эффективности проекта. На базе представленных расчетов можно говорить о том, что проект экономически эффективен.

Заключение

В рамках выполнения данной выпускной квалификационной работы была разработана автоматизированная система для службы «Техническая поддержка».

В процессе анализа предметной области была проведена технико-экономическая характеристика деятельности ИП Ибрагимов К.С., выполнено концептуальное моделирование системы для службы «Техническая поддержка», созданы модели бизнес-процессов «Как есть» и «Как должно быть» с учетом применения в нем разрабатываемого ПО, а также проведены анализ существующих разработок и обоснование выбора технологии проектирования.

С учетом созданных концептуальных моделей и проведенного анализа были сформулированы основные требования к разрабатываемому ПО, цель разработки системы, а также определены необходимые функциональные возможности и особенности работы.

В ходе проектирования ПО была сформулирована поставка задачи на разработку ИС, выполнено логическое и физическое моделирование информационной системы, описано технологическое обеспечение задачи и исполнен контрольный пример реализации проекта.

В работе проведен расчет экономической эффективности проекта. В результате проведенных расчетов можно сделать вывод о том, что проектируемая система является эффективным решением задачи автоматизации рабочего места руководителя отдела службы технической поддержки.

Возможно дальнейшее расширение функциональных возможностей автоматизированной информационной системы согласно актуальным требованиям предприятия с целью преобразования ПО в полноценную многопользовательскую систему.

Список используемой литературы

1. Абрамян М. Visual C# на примерах / Михаил Абрамян. - М.: БХВ-Петербург, 2016. - 436 с.
2. Аграновский А.В. Технологии разработки веб-приложений на основе PHP и MySQL: учеб. пособие. – СПб.: ГУАП, 2018. – 130 с.
3. Бабанов А.М. Технология разработки программного обеспечения. – Томск: Изд-во ТГУ, 2016. – 157 с.
4. Бельков С.А. Прикладное программирование с использованием языка С-Шарп. - Екатеринбург: Изд-во Урал. ун-та, 2017.— 120 с.
5. Вагнер Б. C# Эффективное программирование / Билл Вагнер. - М.: ЛОРИ, 2017. - 320 с.
6. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2016. – 544 с.
7. Веллинг Л., Томсон Л., Разработка веб-приложений с помощью PHP и MySQL. – М.: Вильямс, Альфа-книга, 2017. – С.768.
8. Гарсиа-Молина Г. Системы баз данных. Полный курс: пер. с англ. / Гектор ГарсиаМолина, Джеффри Д. Ульман, Дженнифер Уидом. — М.: Вильямс, 2019. — 1088 с.
9. Голицына О.Л. Языки программирования. – М.: ИНФРА-М, 2016. – 400 с.
10. Грекул В.И. Проектирование информационных систем / В.И. Грекул, Н.Л. Коровкина, Г.А. Левочкина. - М.: Издательство Юрайт, 2017. - 385 с.
11. Грибов В.Д. Экономика организации (предприятия). - М.: КНОРУС, 2016. - 416 с.
12. Гросс К. C# 2008 и платформа .NET 3.5 Framework / Кристиан Гросс. - М.: Вильямс, 2016. - 480 с.

13. Гукин Д. Программирование на C# для чайников / Д. Гукин. - М.: Диалектика, 2019. - 384 с.
14. Дейт К. Дж. Введение в системы баз данных: пер. с англ. / Крис Дж. Дейт. — 8-е изд. — М.: Вильямс, 2019. — 1328 с.
15. Дейтел П. Как программировать на Visual C# 2012 / П. Дейтел. - М.: Питер, 2016. - 312 с.
16. Дрейер М. C# для школьников. – М.: БИНОМ, 2016. – 128 с.
17. Евдокимов П.В. C# на примерах. – СПб.: Изд-во «Наука и техника», 2016. – 304 с.
18. Ишкова Э.А. Самоучитель C#. Начала программирования / Э.А. Ишкова. - М.: Наука и техника, 2017. - 496 с.
19. Казанский А.А. Программирование на visual C#. - Москва: Юрайт, 2020. - 192 с.
20. Керниган Б.У. Язык программирования C. – М.: Вильямс, 2016. – 288 с.
21. Костерин В.В. Разработка сайтов и web-страниц: учебное пособие / В.В. Костерин, Е.В. Бунова, С.А. Богатенков. – Челябинск: Издательский центр ЮУрГУ, 2016. – 110 с.
22. Куликов С.С. Работа с MySQL, MS SQL Server и Oracle в примерах : практ. пособие. / С. С. Куликов. — Минск: БОФФ, 2016. — 556 с.
23. Культин Н. Основы программирования в Microsoft Visual C# 2010 / Никита Культин. - М.: БХВ-Петербург, 2017. - 389 с.
24. Кумагина Е.А. Модели жизненного цикла и технологии проектирования программного обеспечения / Е.А. Кумагина, Е.А. Неймарк. – Нижний Новгород: Изд-во ННГУ, 2016. – 41 с.
25. Магда Ю.С. NI Measurement Studio. Практика разработки систем измерения и управления на C# / Ю.С. Магда. - М.: ДМК Пресс, 2017. - 567 с.

26. Магда Ю. С. NI Measurement Studio. Практика разработки систем измерения и управления на C# / Ю.С. Магда. - М.: ДМК Пресс, 2016. - 373 с.
27. Мартынов Н.Н. C# для начинающих. – М.: Кудиц-Пресс, 2017. – 270 с.
28. Моргунов Е.П. Язык SQL. Базовый курс: учеб.-практ. пособие / Е. П. Моргунов; под ред. Е. В. Рогова, П. В. Лузанова; Postgres Professional. — М., 2017. — 257 с.
29. Нейгел К. C# 2008 и платформа .NET 3.5 для профессионалов / Кристиан Нейгел и др. - М.: Вильямс, 2016. - 695 с.
30. Ник Р. Visual Studio 2010 для профессионалов / Рендольф Ник. - М.: Диалектика / Вильямс, 2016. - 516 с.
31. Новиков Б. Настройка приложений баз данных / Борис Новиков, Генриетта Домбровская. — СПб.: БХВ-Петербург, 2019. — 240 с.
32. Осипов Н.А. Технологии программирования. Учебное пособие. – СПб: Университет ИТМО, 2016. – 61 с.
33. Подсорин В.А. Экономические методы управления жизненным циклом производственных и социальных систем. – М.: МГУПС (МИИТ), 2016. – 78 с.
34. Пугачев С. Разработка приложений для Windows 8 на языке C# / С. Пугачев, А. Шериев, К. Кичинский. - М.: БХВ-Петербург, 2016. - 416 с.
35. Секлетова Н.Н. Анализ рынка информационных систем и технологий / Н.Н. Секлетова, А.С. Тучкова, О.И. Захарова. – Самара: ПГУТИ, 2018. – 215 с.
36. Селко Д. Стиль программирования Джо Селко на SQL: пер. с англ. / Джо Селко. — М.: Русская редакция; СПб.: Питер, 2020. — 206 с.
37. Соловей Л.В. Программирование на языке C# / Л.В. Соловей, Н.Н. Мирошниченко, Н.Г. Пономарёв. – Х.: НТУ «ХПИ», 2016. – 356 с.
38. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4 / Эндрю Троелсен. - Москва: Огни, 2016. - 238 с.

39. Тюгашев, А.А. Основы программирования [Текст] / А.А. Тюгашев. – СПб.: Университет ИТМО, 2016. – 160 с.
40. Федотова, Е.Л. Информационные технологии и системы: Уч.пос / Е.Л. Федотова. - М.: Форум, 2018. - 149 с.
41. Фримен А. ASP.NET MVC 4 с примерами на С# 5.0 для профессионалов / Адам Фримен. - М.: Вильямс, 2016. - 688 с.
42. Хайруллин, Р.С. Программирование на С# [Текст] / Р.С. Хайруллин. – Казань: Изд-во Казан. гос. архитект.-строит. ун-та, 2017. – 159 с.
43. Чистов, Д.В. Информационные системы в экономике: Учебное пособие / Д.В. Чистов. - М.: Инфра-М, 2019. - 248 с.
44. Шустова Л.И. Базы данных. - Москва: Инфра-М, 2016. - 304 с.
45. Ясенев, В.Н. Информационные системы в экономике (для бакалавров) / В.Н. Ясенев, О.В. Ясенев. - М.: КноРус, 2015. - 352 с.

Приложение А

Код разработки главной формы

Листинг А.1 – Код разработки главной формы класс F_MAIN.cs

```
using System;
using System.IO;
using System.Windows.Forms;

namespace Doremi_Service
{
    /// <summary>
    /// Главная форма системы
    /// </summary>
    public partial class F_MAIN : Form
    {
        /// <summary>
        /// Конструктор формы
        /// </summary>
        public F_MAIN()
        {
            // Создание и инициализация компонентов
            InitializeComponent();
            // Установка параметров
            Global.MainForm = this;
            // Активация таймеров
            tmrTodayDateTimeInfo.Enabled = true;
            // Чтение данных подключения
            GetConnectionSettings();
        }
        // ДОСТУП К ЭЛЕМЕНТАМ ФОРМЫ -----
        /// <summary>
        /// Доступ к главной панели рабочего стола системы
        /// </summary>
        public Panel Workspace { get { return pnlWorkspace; } }
        /// <summary>
        /// Доступ к панели авторизации
        /// </summary>
        public Panel Authorization { get { return pnlAuthorization; } }
        /// <summary>
```


Продолжение Приложения А

```
/// Доступ к панели настроек подключения
/// </summary>
public Panel ConnectionPanel { get { return pnlConnectionSettings; } }
/// <summary>
/// Доступ к метке информации о текущем пользователе
/// </summary>
public Label CurrentUserInfo { get { return lblCurrentUser; } }
/// <summary>
/// Доступ к метке времени авторизации
/// </summary>
public Label LoginTimeStamp { get { return lblLoginTimeStamp; } }
/// <summary>
/// Смена рабочего стола
/// </summary>
/// <param name="__workspace_">Рабочий стол пользователя</param>
public void ChangeWorkspace(UserControl __workspace_)
{
    // Предварительная очистка
    Workspace.Controls.Clear();
    // Помещение рабочего стола
    Workspace.Controls.Add(__workspace_);
    // На всю область
    __workspace_.Dock = DockStyle.Fill;
}
/// -----
/// <summary>
/// Отображение текущей даты / времени
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
/// <param name="e">Параметры события</param>
private void tmrTodayDateTimeInfo_Tick(object sender, EventArgs e)
{
    lblTodayDate.Text = DateTime.Now.ToLongDateString();
    lblTodayTime.Text = DateTime.Now.ToLongTimeString();
}
/// <summary>
/// Показать/скрыть данные подключения к БД
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
```

Продолжение Приложения А

```
/// <param name="е">Параметры события</param>
private void InkConnectionSettings_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    if (pnlConnectionSettings.Visible)
    {
        InkConnectionSettings.Text = "Показать параметры" +
Environment.NewLine + "подключения к БД >>";
        pnlConnectionSettings.Visible = false;
    }
    else
    {
        InkConnectionSettings.Text = "Спрятать параметры" +
Environment.NewLine + "подключения к БД <<";
        pnlConnectionSettings.Visible = true;
    }
}
}
```

```
/// <summary>
/// Получение настроек подключения из файла
/// </summary>
protected void GetConnectionSettings()
{
    // Поток чтения данных их файла
    FileStream fs = null;
    StreamReader sr = null;
    // Попытка прочитывать данные
    try
    {
        fs = new FileStream("server.dat", FileMode.Open);
        sr = new StreamReader(fs);
        // Чтение данных
        txtServerAddress.Text = sr.ReadLine().Split(new char[] { '=' },
StringSplitOptions.RemoveEmptyEntries)[1];
        txtServerLogin.Text = sr.ReadLine().Split(new char[] { '=' },
StringSplitOptions.RemoveEmptyEntries)[1];
        txtServerPassword.Text = sr.ReadLine().Split(new char[] { '=' },
StringSplitOptions.RemoveEmptyEntries)[1];
    }
}
```

Продолжение Приложения А

```
txtDataBaseName.Text = sr.ReadLine().Split(new char[] { '=' },
StringSplitOptions.RemoveEmptyEntries)[1];
    txtServerPort.Text = sr.ReadLine().Split(new char[] { '=' },
StringSplitOptions.RemoveEmptyEntries)[1];
    // Закрыть файл
    sr.Close();
    fs.Close();
}
catch (Exception) { }
}

/// <summary>
/// Сохранение настроек подключения в файл
/// </summary>
protected void SetConnectionSettings()
{
    // Поток записи данных в файл
    FileStream fs = null;
    StreamWriter sw = null;
    // Попытка записать данные
    try
    {
        fs = new FileStream("server.dat", FileMode.Create);
        sw = new StreamWriter(fs);
        // Запись данных
        sw.WriteLine("adress=" + txtServerAddress.Text);
        sw.WriteLine("login=" + txtServerLogin.Text);
        sw.WriteLine("password=" + txtServerPassword.Text);
        sw.WriteLine("db=" + txtDataBaseName.Text);
        sw.WriteLine("port=" + txtServerPort.Text);
        // Закрыть файл
        sw.Close();
        fs.Close();
    }
    catch (Exception) { }
}

/// <summary>
/// Смена / авторизация текущего пользователя
```

Продолжение Приложения А

```
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
/// <param name="e">Параметры события</param>
private void lnkChangeUser_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    // Сбросить текущего пользователя
    Global.CurrentUser = null;
    // Сбросить подсистемы
    Workspace.Controls.Clear();
    // Показать панель авторизации
    pnlAuthorization.Visible = true;
    pnlConnectionSettings.Enabled = true;
    txtLogin.Text = "";
    txtPassword.Text = "";
}
/// <summary>
/// Попытка авторизоваться
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
/// <param name="e">Параметры события</param>
private void btnLogin_Click(object sender, EventArgs e)
{
    // Пробуем соединиться
    bool test_result = Global.SetConnectionWithDB(txtServerAddress.Text,
txtServerLogin.Text,
    txtServerPassword.Text, txtDataBaseName.Text, txtServerPort.Text);
    // Если соединение установлено, то пробуем авторизоваться
    if (test_result)
    {
        // Сохранение данных подключения (если выбрано)
        if (chkRemindData.Checked) SetConnectionSettings();
        // Авторизация
        Global.LoginUser(txtLogin.Text, txtPassword.Text);
    }
}
/// <summary>
/// Тестирование подключения
/// </summary>
/// <param name="sender">Объект, вызвавший событие</param>
```

Продолжение Приложения А

```
/// <param name="e">Параметры события</param>
private void btnTestConnection_Click(object sender, EventArgs e)
{
    // Пробуем соединиться
    bool test_result = Global.SetConnectionWithDB(txtServerAddress.Text,
txtServerLogin.Text,
    txtServerPassword.Text, txtDataBaseName.Text, txtServerPort.Text);
    // Выводим сообщение
    MessageBox.Show(test_result ? "Соединение установлено!" :
"Соединение не установлено!", Application.ProductName,
    MessageBoxButtons.OK, test_result ? MessageBoxIcon.Information :
MessageBoxIcon.Error);
    // Сохранение данных подключения (если выбрано)
    if (chkRemindData.Checked) SetConnectionSettings();
}
}
}
```