

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

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 информационной системы учета технического обслуживания
компьютерной техники в ООО «Пласт».

Обучающийся

А.В. Садовников

(И.О. Фамилия)

(личная подпись)

Руководитель

к.т.н. Н.В. Хрипунов

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

Тольятти 2024

Аннотация

Выпускная квалификационная работа состоит из 57 страниц, 23 рисунков, 3 таблиц, 20 источников.

Техника, обслуживание, учет, компьютерная техника, рабочее место, информационная система.

Объектом исследования является деятельность компании ООО «Пласт».

Предмет исследования – процесс учета технического обслуживания компьютерной техники.

Целью работы является разработка автоматизированной системы, которая будет вести учет технического обслуживания компьютерной техники в ООО «Пласт».

Достижение цели требует решение таких задач:

- изучение деятельности компании ООО «Пласт»;
- изучение и моделирование бизнес-процессов «как есть» и «как должно быть»;
- логическое проектирование информационной системы;
- реализация информационной системы, которая будет вести учет технического обслуживания компьютерной техники.

Результатом выпускной квалификационной работы будет созданная информационная система для учета технического обслуживания компьютерной техники.

Для выполнения работы были применены технология проектирования с помощью языка UML, водопадная методология проектирования и разработки программного продукта. Для реализации системы были использованы инструменты такие как язык программирования C#, среда разработки Visual Studio 2022, система управления базой данных MS SQL Server 2020.

Первая глава была посвящена изучению деятельности компании ООО «Пласт», включая основные бизнес–процессы, на основании чего был смоделирован и построен процесс, который показывает, как устроен учет технического обслуживания компьютерной техники в компании. Проведенный анализ позволил смоделировать процесс, который показывает, что изменится в процессе после автоматизации.

Во второй главе была построена логическая модель данных, основанная на проведенном концептуальном проектировании. Было проведено проектирование структуры информационной системы и обоснованы требования к аппаратно-программному обеспечению.

В третьей главе осуществляется выбор основных инструментов реализации системы для учета технического обслуживания. Разработана база данных и внесены данные для получения информации при запросе. Проведена реализация информационной системы и написано руководство пользователя.

Оглавление

Введение.....	5
Глава 1 Функциональное моделирование предметной области.....	7
1.1 Характеристика компании ООО «Пласт»	7
1.2 Концептуальное моделирование предметной области	9
1.2.1 Выбор технологии концептуального моделирования предметной области.....	10
1.2.2 Моделирование бизнес-процессов предметной области.....	11
1.2.3 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ» ..	15
1.3 Анализ существующих разработок.....	18
1.4 Постановка задачи на разработку системы	22
Глава 2 Проектирование информационной системы	24
2.1 Модели и методологии проектирования и разработки информационной системы	24
2.2 Логическая модель информационной системы	27
2.3 Проектирование базы данных информационной системы.....	29
2.4 Обоснование вида логической модели	34
2.5 Требования к аппаратно-программному обеспечению системы	35
Глава 3 Проектирование информационной системы	36
3.1 Архитектура информационной системы	36
3.2 Выбор инструментов разработки	36
3.4 Разработка физической модели данных	38
3.5 Разработка автоматизированной информационной системы.....	40
3.6 Руководство пользователя.....	43
Заключение	45
Список используемой литературы и используемых источников.....	46
Приложение А Программный код.....	49

Введение

Компания ООО «Пласт» занимается производством пластмассовых изделий [5]. Произведенную продукцию отправляют заказчикам согласно полученных заявок.

Планированием производства и заказа материалов и сырья занимается отдел планирования компании. Отгрузкой продукции компании занимается склад на основе получения от бухгалтерии подтверждения об оплате заказчика.

Все отделы в компании ООО «Пласт» имеют рабочие места, оснащенные компьютерной техникой и специализированными информационными системами.

Повышение эффективности работы сотрудников всех отделов будет достигнуто, когда компьютерная техника будет исправно работать на протяжении рабочего дня. Для этого необходимо выполнять обслуживание компьютерной техники и вести его учет.

Объект исследования – деятельность компании ООО «Пласт».

Предмет исследования – процесс учета технического обслуживания компьютерной техники.

Цель работы – разработать информационную систему учета технического обслуживания компьютерной техники в ООО «Пласт».

Достижение цели требует решение таких задач:

- изучение деятельности компании ООО «Пласт»,
- изучение и моделирование бизнес-процессов «как есть» и «как должно быть»,
- логическое проектирование информационной системы,
- реализация информационной системы, которая будет вести учет технического обслуживания компьютерной техники.

Результатом выпускной квалификационной работы будет созданная информационная система для учета технического обслуживания компьютерной техники.

В процессе выполнения работы были использованы: водопадная методология, объектно-ориентированный язык программирования C#, среда разработки Visual Studio 2022, система управления базой данных MS SQL Server 2020.

Первая глава была посвящена изучению деятельности компании ООО «Пласт», включая основные бизнес-процессы, на основании чего был смоделирован и построен процесс, который показывает, как устроен учет технического обслуживания компьютерной техники в компании. Проведенный анализ позволил смоделировать процесс, который показывает, что изменится в процессе после автоматизации.

Во второй главе была выбрана методология проектирования информационной системы, построена логическая модель данных, основанная на проведенном концептуальном проектировании. Было проведено проектирование структуры информационной системы и обоснованы требования к аппаратно-программному обеспечению.

В третьей главе осуществляется выбор основных инструментов реализации системы для учета технического обслуживания. Разработана база данных и внесены данные для получения информации при запросе. Проведена реализация информационной системы и написано руководство пользователя.

Глава 1 Функциональное моделирование предметной области

1.1 Характеристика компании ООО «Пласт»

Основные функции компании направлены на производство пластиковых изделий.

Компания представляет собой стабильный динамично развивающийся бизнес, занимающий лидирующие позиции на рынке хозяйственных товаров, посуды и товаров для дома не только в Самаре и Самарской области,

Ассортимент реализуемой продукции включает более 3000 наименований, в том числе посуда, товары для кухни, вазы, фужеры, бокалы, кашпо, товары для быта, пластиковая мебель, одноразовая посуда.

Реализация продукции происходит на основе поступающих заявок от клиентов. Заявки обрабатываются в отделе заказов, где менеджеры отдела проверяют содержимое заявок и передают заказ на склад, откуда заказ отгружается заказчику.

Перед тем как отправить заказ клиенту, выставляется счет на оплату, после чего ожидается оплата и только после подтверждения оплаты заказ отправляется.

Бухгалтерия выполняет все финансовые операции и отчетность перед налоговыми органами.

Отдел планирования на основе полученных заявок и объема произведенных изделий планирует план на следующие год по производству изделий, а также оставляет план для заказа сырья, на основе которого и производятся изделия.

Каждый отдел работает в своей специализированной информационной системе. В отделе бухгалтерии внедрена система 1С: Бухучет и торговля. В отделах планирования и закупок внедрена система 1С: Предприятие. На складе сотрудники пользуются системой 1С: Склад.

Чтобы процессы в компании всегда выполнялись без сбоев, необходимо следить за техническим состоянием компьютерной техники. Данными функциями занимается сервисный отдел компании, где инженеры занимаются ремонтом и обслуживанием компьютерной техники. Чтобы получать актуальные данные по состоянию техники необходимо вести учет обслуживания.

Процесс ведения учета технического обслуживания позволит:

- видеть статус техники;
- вести учет частоты ремонта одной и той же техники;
- вести заказ необходимых запчастей;
- проводить своевременное обслуживание.

Автоматизация учета технического обслуживания компьютерной техники обеспечит прозрачность выполнения плана по обслуживанию техники, а также расходованию запасных частей в отделе сервисного обслуживания.

Рассмотрим организационную структуру компании ООО «Пласт»:

- генеральный директор,
- отдел планирования,
- материальный отдел,
- отдел по работе с клиентами,
- сервисный отдел,
- склад,
- сервисный отдел,
- инженерный отдел,
- отдел сбыта,
- бухгалтерия,
- отдел маркетинга.

Структура управления в организации ООО «Пласт» показана на рисунке

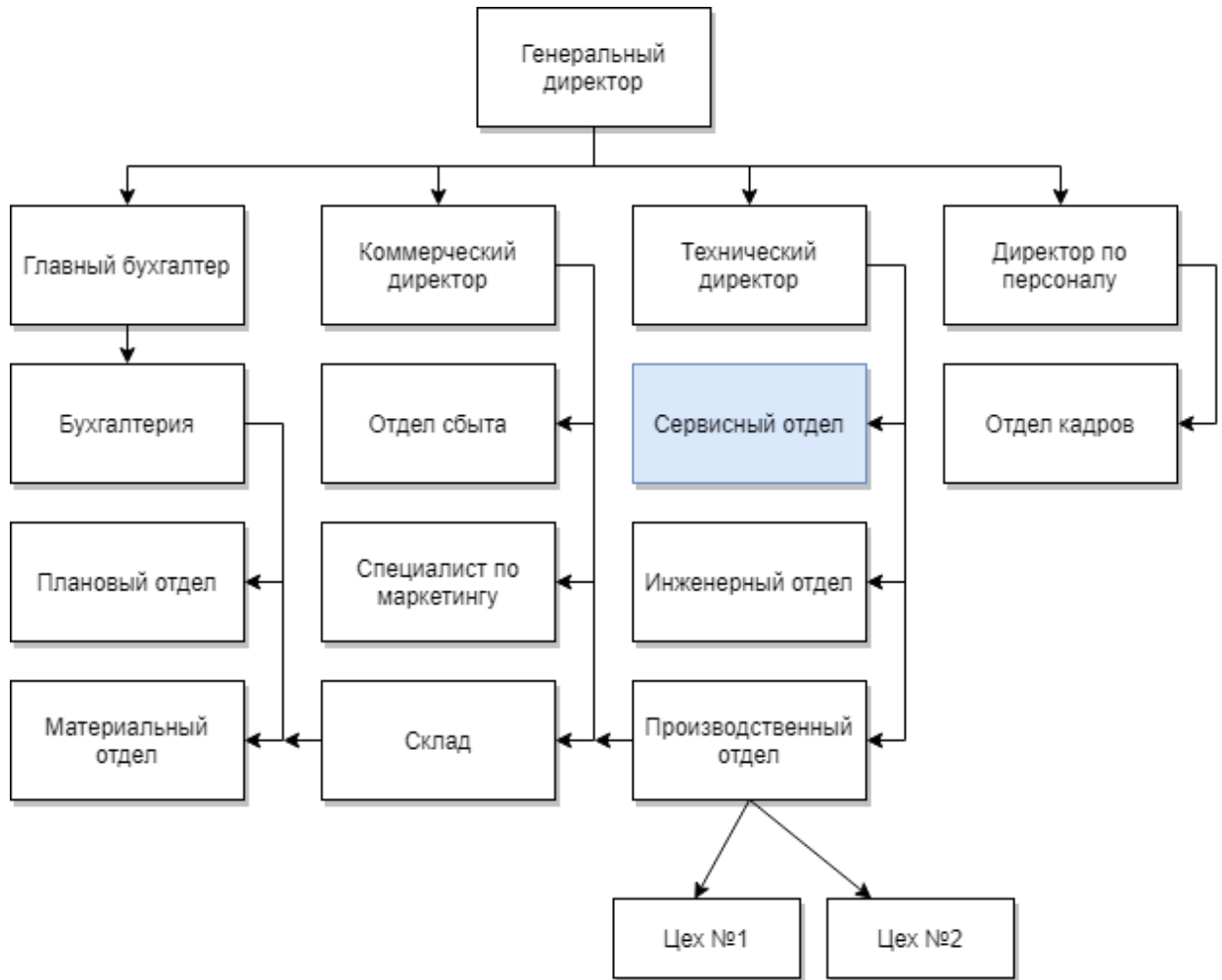


Рисунок 1 – Организационная структура ООО «Пласт»

Рассмотрим деятельность сервисного отдела, который находится в подчинении технического директора и выполняет поручения, которые относятся к сервисному обслуживанию. Сервисное обслуживание подразумевает техническое обслуживание любых технических средств, оргтехники компании.

1.2 Концептуальное моделирование предметной области

1.2.1 Выбор технологии концептуального моделирования предметной области

Моделирование информационной системы начинается с концептуального проектирования, в котором подробно изучаются бизнес-процессы компании и выявляются места, которые требуют автоматизации [9].

Рассмотрим несколько нотаций с помощью которой можно проводить концептуальное проектирование. Подробно будут рассмотрены такие нотации как: BPMN, UML, IDEF0, ERD, DFD.

BPMN - это стандартная нотация для моделирования бизнес-процессов. Она позволяет визуализировать шаги, роли, потоки данных и решения в рамках бизнес-процессов. В случае АРМ логиста интернет-магазина, вы можете использовать BPMN для моделирования процессов обработки заказов, управления инвентаризацией, согласования доставок и т.д. [8].

UML - это универсальная нотация, которая может быть применена для различных аспектов моделирования, включая структурное и функциональное моделирование. В этом случае, UML диаграммы классов, диаграммы последовательности и диаграммы состояний могут быть полезными для описания объектов, действий и жизненного цикла процессов логиста.

IDEF0 предоставляет структурную нотацию для моделирования функциональных аспектов системы. Диаграммы IDEF0 позволяют описать функции, взаимосвязи и порядок выполнения задач, что полезно для описания операций логиста.

ERD — это нотация для моделирования баз данных. Нотация помогает определить связи между таблицами такими как оборудование, заявка, сотрудник.

DFD — представляет собой нотацию, показывающую обмен информацией между объектами системы. Она может быть использована для понимания, как данные перемещаются в системе и какие процессы их

обрабатывают. В контексте логистики интернет-магазина, DFD может помочь представить потоки заказов, статусов доставок и т.д.

Проведем сравнение описанных технологий и отобразим их в таблице 1.

Таблица 1 – Сравнительная таблица

Критерий	Нотация		
	ARIS	IDEF0	UML
Сложность изучения	да	нет	да
Легкий способ построения	нет	да	нет
Подход к проектированию	процессный	функциональный	объектно-ориентированный
Наличие способа декомпозиции	есть	есть	есть

На основе анализа можно сделать вывод что самым простым и доступным является технология моделирования IDEF0, которая показывает процесс как с абстрактной стороны, так и позволяет углубляться в процесс с помощью декомпозиции процесса.

IDEF0 предоставляет удобный способ визуализации структурных аспектов системы, разбивая процессы на функциональные блоки и их взаимодействие [1].

1.2.2 Моделирование бизнес-процессов предметной области

Бизнес-процессы в компании охватывают выполнение многих функций таких как прием заказов, производство изделий из пластмассы, заказ сырья и материалов, отгрузка продукции, техническое обслуживание компьютерной техники. Определим какие функции выполняются в сервисном отделе при техническом обслуживании компьютерной техники:

- приемка заявки;
- обработка заявки;

- выяснение причины неисправности;
- плановый осмотр и обслуживание техники;
- заказ комплектующих;
- выполнение ремонта;
- закрытие заявки.

В сервисном отделе сотрудники выполняют все эти процессы вручную без каких-либо автоматизированных систем. Все поступающие данные на бумажных носителях хранятся в отделе в отдельной папке с заявками на обслуживание. При любом перемещении отдела со всей документацией, к примеру в другое помещение есть риск утери части информации, которая в данный момент храниться только на бумаге.

Проведем проектирование модели процесса «Как есть» в нотации IDEF0.

На рисунке 2 представлена диаграмма.

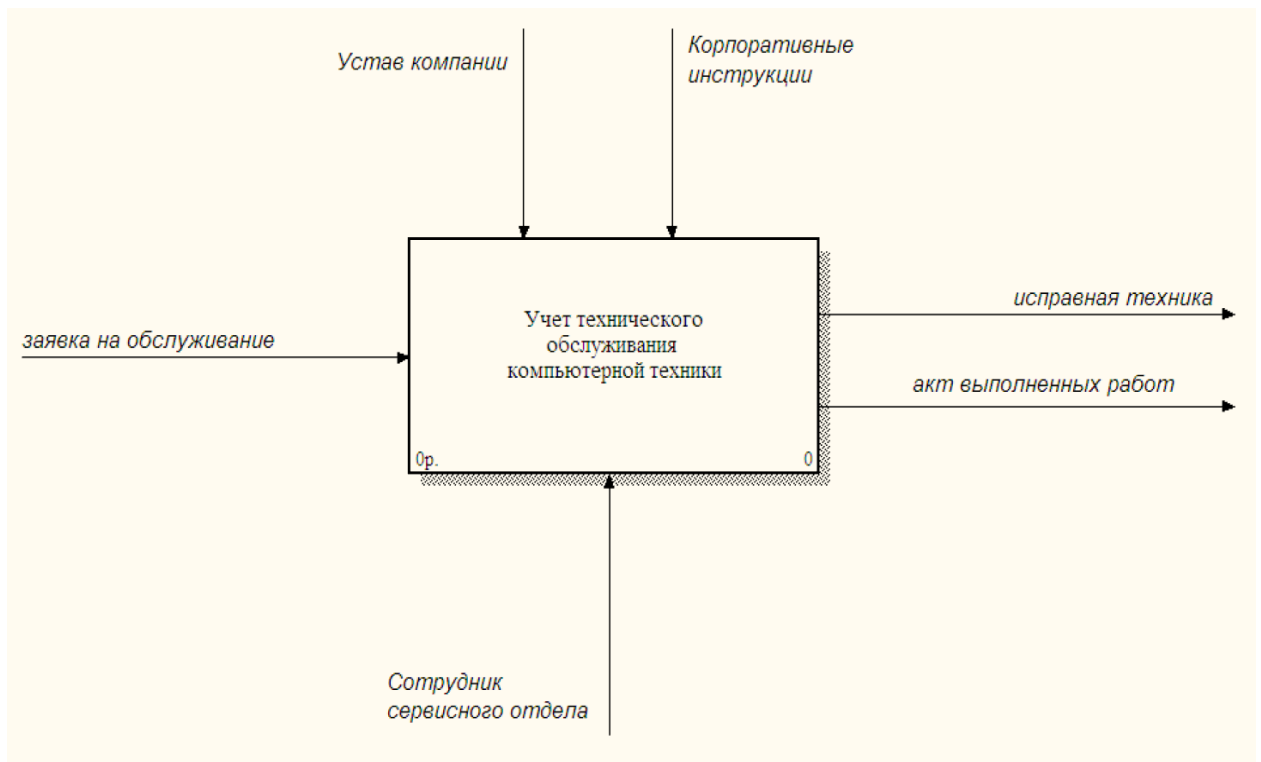


Рисунок 2 – Контекстная модель процесса «Учет технического обслуживания компьютерной техники»

Для полного понимания ведения всех процессов при учете технического обслуживания как есть сейчас есть, проведена декомпозиция процесса [6].

Рисунок 3 показывает декомпозицию модели.

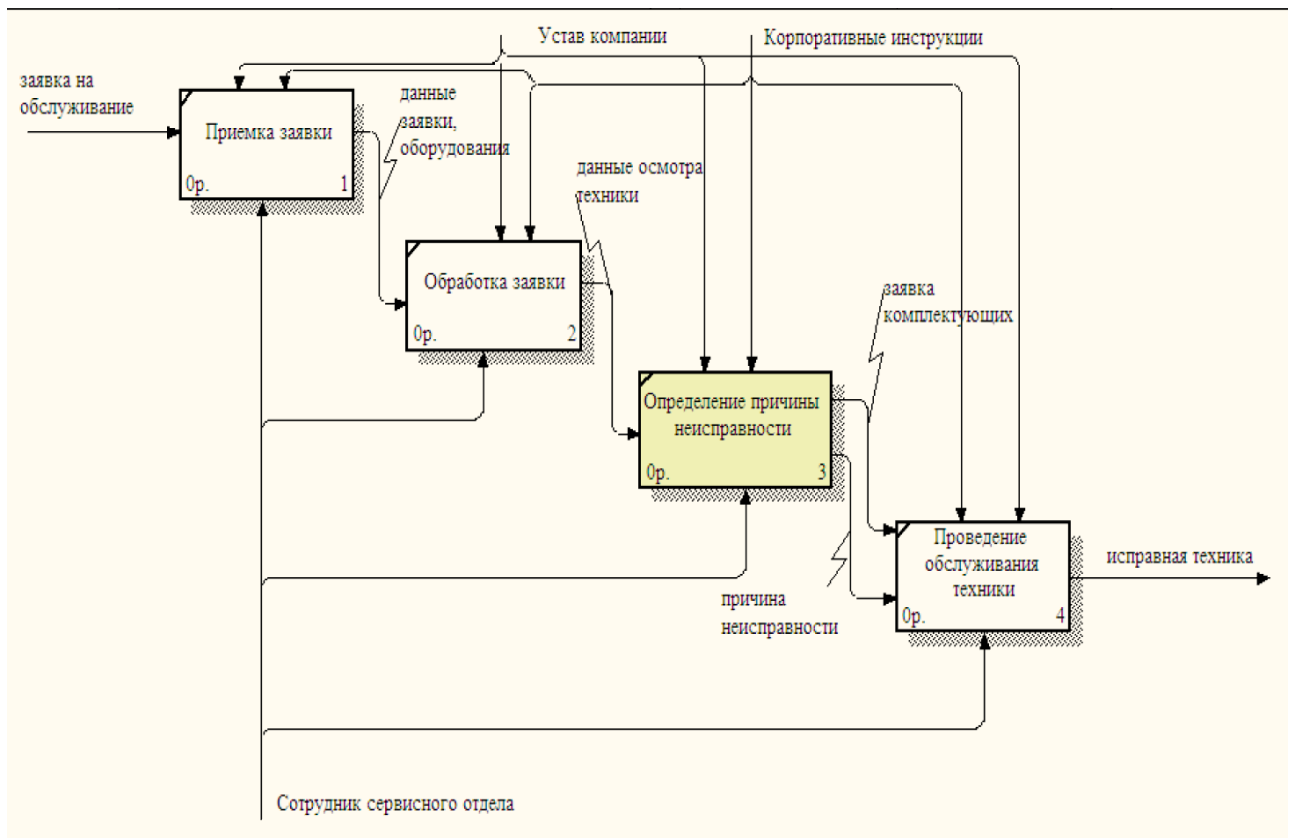


Рисунок 3 – Декомпозиция модели процесс учета технического обслуживания

Сотрудник сервисного отдела получает заявку на обслуживание, обычно это происходит на бумажном носителе, чтобы была возможность фиксировать процесс получения заявки в журнале заявок. Обработка заявки начинается, когда при приемке заявке получена вся информация о оборудовании и сотруднике и отделе откуда получена заявка. Обработка заявка включает первичный осмотр оборудования и внесение информации в заявку. Далее сотрудники определяют неисправность техники и решают нужны ли дополнительные комплектующие для устранения неисправности или нет. После проведения обслуживания техники выдается исправная техника сотруднику, который подавал заявку на обслуживание.

1.2.3 Разработка и анализ модели бизнес-процесса «КАК ЕСТЬ»

Смоделированные модели процесса «Как есть» наглядно показали, как происходит сейчас процесс учета технического обслуживания компьютерной техники.

Заявки на обслуживание поступают на бумажных носителях и фиксируются в журнале заявок, что говорит о низком уровне защиты информации от утери или порчи при перемещениях документации. Еще одним не маловажным пунктом является то, что сотрудник, подававший заявку никак, не отслеживает выполнение работ по его заявке и не известно выполняются ли вообще работы по данной заявке.

Таким образом необходимо автоматизировать эти процессы, чтобы заявки можно было создавать в информационной системе, которая будет работать локально. При создании заявки она должна поступать в сервисный отдел и иметь статус «принята», после того как сотрудник отдела ее принял.

Также необходимо, чтобы сотрудник сервисного отдела имел возможность менять статус заявки на этапе ее выполнения. Сотрудник, подающий заявку, тоже должны иметь возможность видеть статус заявки, которую подали.

Автоматизация данных процессов значительно сократит время подачи и получения заявки, уменьшит количество ошибок и неточностей при определении неисправности, а также позволит вести корректный учет технического обслуживания компьютерной техники в сервисном отделе.

Автоматизация увеличит прозрачность выполнения заявок и сократит вероятность потери данных, так как все данные при поступлении будут сохраняться в базу данных.

Целью создания информационной системы является автоматизация процесса учета технического обслуживания компьютерной техники в компании ООО «Пласт». Информационная система будет выполнять следующие задачи:

- автоматизация обработки заказов;
- управление заявками;
- ведение учета заявок;
- создание заявок;
- отслеживание жизненного цикла заявки.

Автоматизация позволит рационально использовать данные, чтобы вести корректный заказ комплектующих для последующего ремонта техники.

Система позволит сотрудникам сервисного отдела корректно распределять заявки между сотрудниками согласно количеству полученных заявок и загруженности сотрудников на текущих работах.

Спроектируем модели процесса «Как должно быть» после проведения автоматизации.

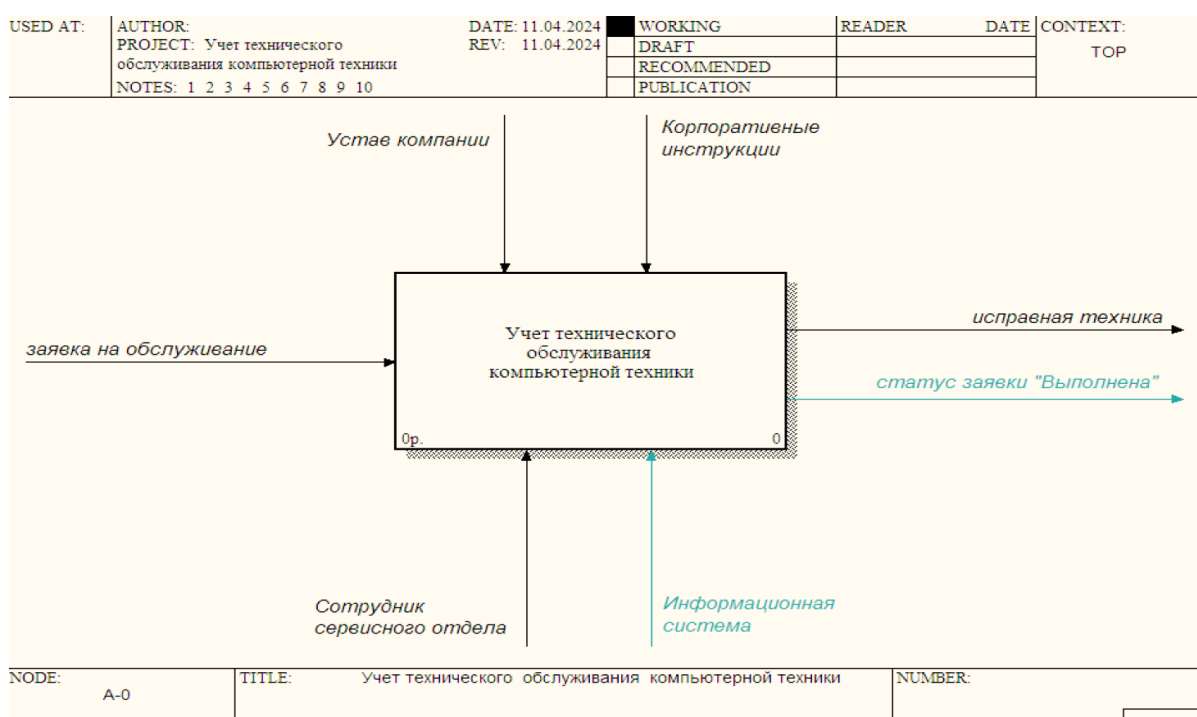


Рисунок 4 – Диаграмма модели «КАК ДОЛЖНО БЫТЬ» первого уровня

Проведем декомпозицию процесса и посмотрим какие именно процессы изменятся после автоматизации. На рисунке 5 показана диаграмма декомпозиции модели «КАК ДОЛЖНО БЫТЬ».

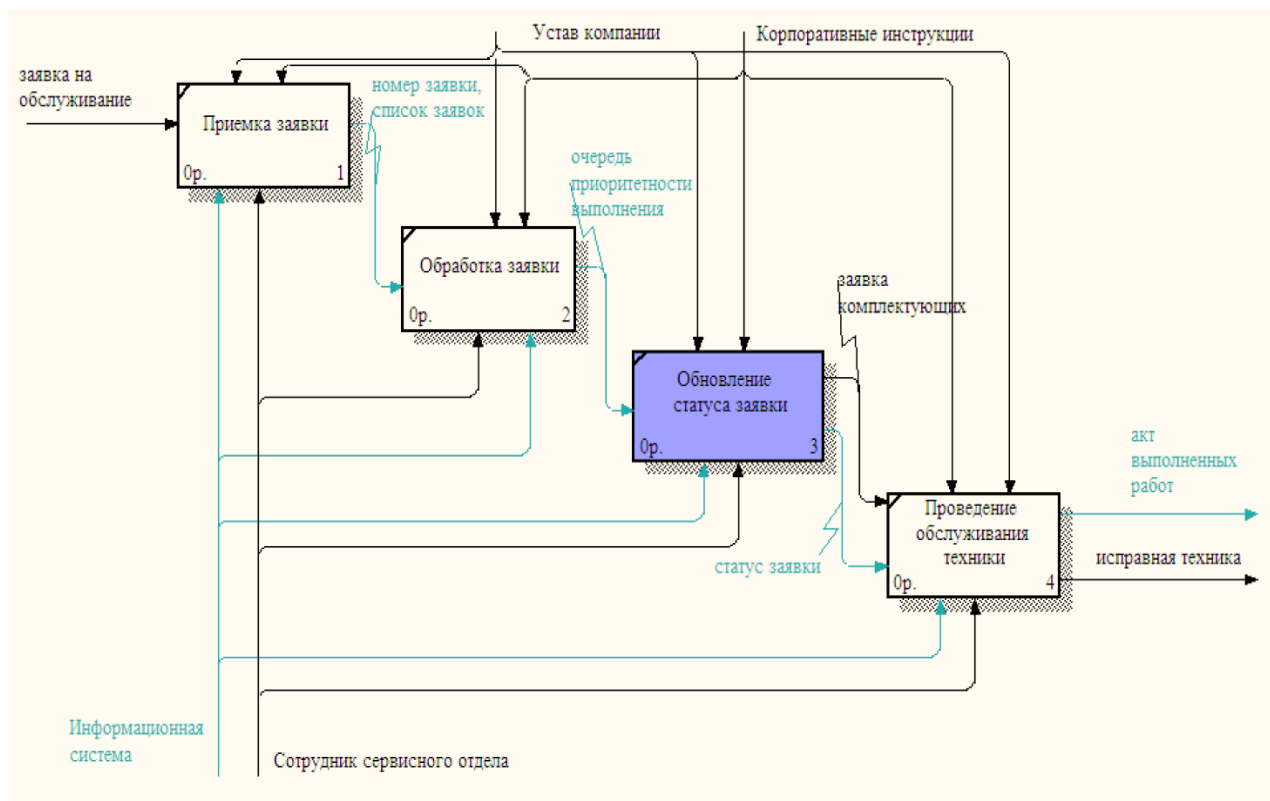


Рисунок 5 – Диаграмма декомпозиции модели «КАК ДОЛЖНО БЫТЬ»

Другим цветом на диаграмме показаны функции и функции, которые изменятся после того, как будет внедрена информационная система в отделе сервисного обслуживания. На рисунке 5 видим, что блок «Определение причины неисправности» изменился на блок «Обновление статуса заявки». Таким образом можно будет увидеть какие заявки находятся в процессе, какие приняты, а какие уже выполнены и закрыты. Также в процессе изменились данные входящие в каждый блок, кроме первого блока, где входящей информацией также осталась заявка на обслуживание.

1.3 Анализ существующих разработок

Рассмотрим наиболее популярные разработки, представленные на рынке, которые выполняют подобные задачи [3].

Сравним три схожие системы, которые являются онлайн системами:

- okdesk,
- ITSM 365,
- helpDeskEddy [16].

Возможности по фиксации инцидентов для пользователей.

Принимает обращения клиентов из всех возможных источников в одной системе. Полученные заявки распределяются по отделам своей компании с помощью настроек прав доступа для сотрудников. Все заявки будут приходить в одну систему и далее регистрироваться и обрабатываться специалистами в едином формате. Руководство всегда сможет увидеть всю историю обращений каждого клиента и выгрузить нужные отчеты для оценки эффективности работы персонала.

Возможность привязки инцидентов к ИТ-услугам, рисунок 6.

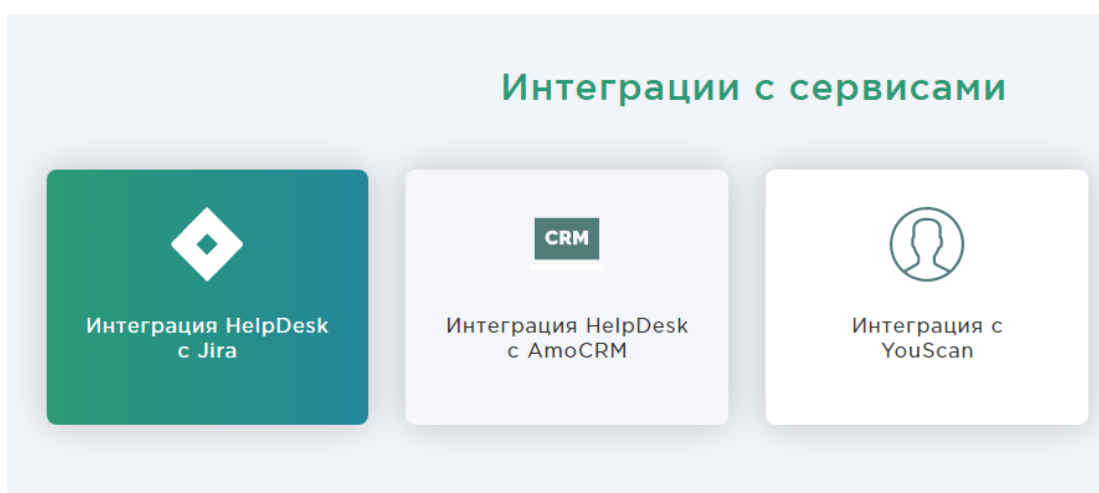


Рисунок 6 – Интеграции с сервисами

Возможности автоматизации в системе позволяют создать полноценного бота, который будет реагировать на сообщения от клиентов и автоматически отправлять в ответ информацию, прописанную в настройках.

Для подключенных в системе каналов Telegram, VK, Facebook и Viber есть возможность настроить отправку кнопок, при нажатии на которые произойдет заданное действие, например переход по ссылке или отправка ответа [14].

Уведомление пользователей о решении инцидентов происходит, используя кнопки обращение клиента в контакт центр, можно сделать более комфортным. Популярные варианты:

- уточнение темы вопроса и информирование;
- запрос на оценку после завершения переписки.

Знания - один из важнейших ресурсов любой компании. Хранить и накапливать их. Это поможет сократить количество заявок от клиентов, т.к. они всегда смогут найти ответ на актуальный вопрос в базе знаний.

Okdesk позволяет управлять всеми заявками в одном месте и дает возможность регистрации заявок самым удобным способом.

Okdesk позволяет не пропустить ни одной регулярной заявки или проведения плановых работ у клиента и с поддерживаемым оборудованием.

Исполнители заявок или менеджеры постоянно на выездах, встречах или работе. Они могут пользоваться мобильным приложением.

Полнофункциональное мобильное приложение с чек-листами позволит специалистам быть всегда на связи, планировать выезды оптимальным образом, совершать все необходимые действия «на лету», ничего не забывать и оставаться в курсе назначенных заявок, рисунок 7.

* Тема
Осмотр систем кондиционирования

Клиент: Булкофф

Тип заявки: Регулярное тех.обслуживание

Приоритет: Высокий

Договор: ТО-12-34

Объект обслуживания: Булочная 146

Плановая дата решения: 2018/05/18 16 : 37

Указать вручную

Повторяющаяся заявка

* Следующее повторение
[] [] 00 : 00

* Повторять через
[] дн. нед. мес.

body

Рисунок 7 – Привязка к IT-услугам

Клиентский портал доступен в любом месте, где есть интернет. 24 часа, 7 дней в неделю. Удобство взаимодействия обусловлено привлечением клиента к важным вопросам и оперативного получения ответов. Оповещение пользователей о решении инцидентов происходит с помощью мобильного приложения. Система позволяет всем контактным лицам самостоятельно регистрировать заявки и отслеживать ход их исполнения [15].

Клиенты сами подтверждают выполнение заявки и оценивают качество работ.

Система позволяет регистрировать заявки удаленно и рассчитать время ее выполнения, рисунок 8.

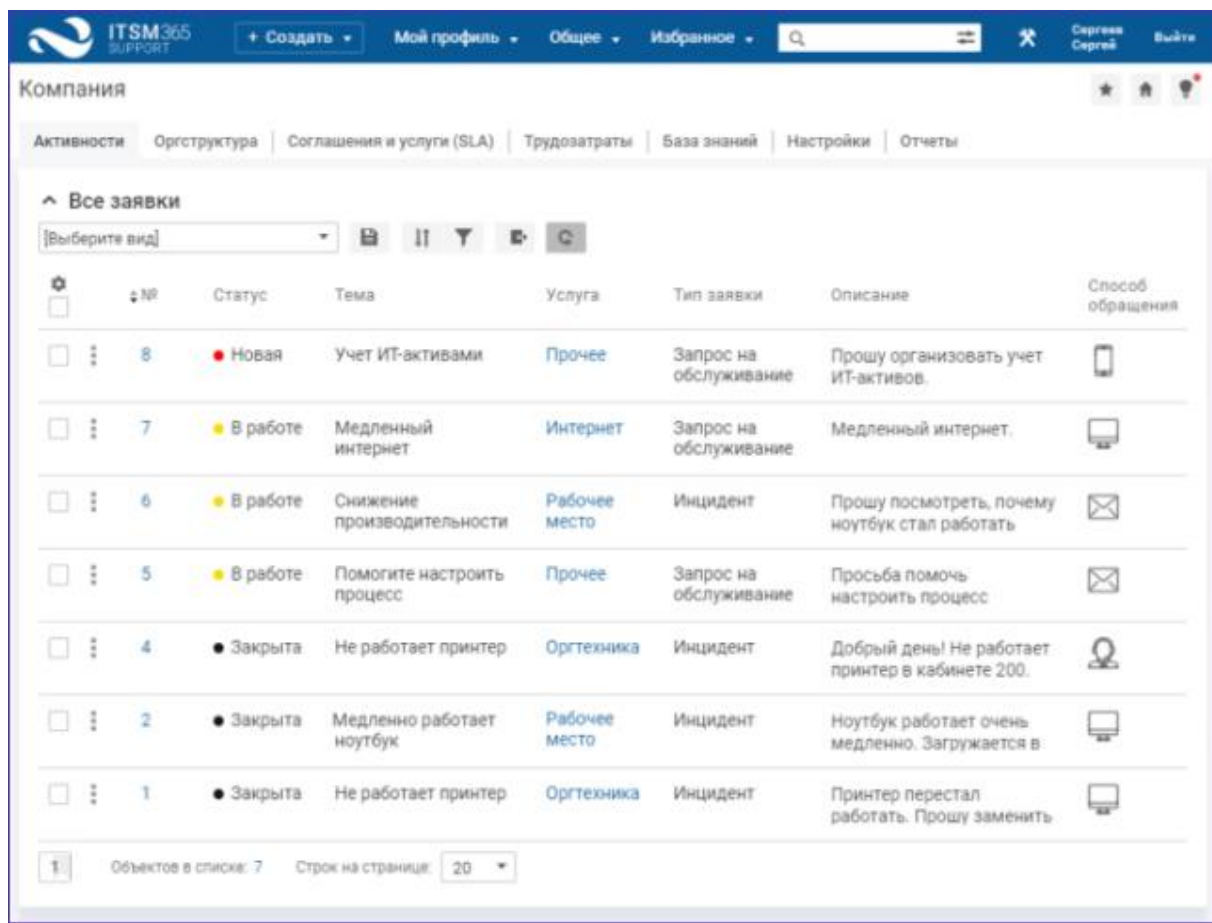


Рисунок 8 – Возможности фиксации инцидентов

Создать заявку, контролировать ее исполнение и статус, общаться со службой поддержки через комментарии к заявке — все это дает доступ к portalу.

В ходе анализа систем были сформированы первоначальные представления о каждой системе из чего следует вывод о наиболее привлекательной как для начинающего пользователя, так и для профессионала является система Okdesk.

Система обладает полнофункциональным мобильным приложением с чек-листами, которое позволяет создавать объекты, вести учёт оборудования, видеть статус работ по заявке, вести переписку с диспетчерами и нужными коллегами, общаться с заказчиком напрямую. Для пользователя эта система удобна базой знаний, которую предлагает разработчик, с помощью которой

можно создавать самостоятельно заявки. Цена на такую систему тоже приемлема, можно выбрать тариф для своих потребностей и потребностей компании.

Однако руководством компании ООО «Пласт» было принято решение о разработке собственной системы, которая будет настроена и создана специально для выполнения нужд сервисного отдела.

1.4 Постановка задачи на разработку системы

После рассмотрения аналогичных программных продуктов, представленных на отечественном рынке, не было найдено наиболее подходящей системы для внедрения в процессы компании. Таким образом появилась необходимость создания информационной системы для автоматизации учета технического обслуживания компьютерной техники.

Информационная система должна позволять:

- принимать заявки в реальном времени;
- позволять создавать заявки;
- возможность редактирования данных заявки;
- возможность сохранять заявки в системе;
- возможность выбирать данные из готовых списков;
- просматривать заявки и ставить их в приоритет;
- разграничение доступа пользователей;
- регистрировать заявку и присваивать ей статус;
- обновлять информацию о заявках.

Нефункциональные требования определяют характеристики, качество и ограничения, которые должны быть учтены при разработке и эксплуатации информационной системы.

Нефункциональные требования:

- отклики системы в течении 5 сек.;

- обозначения на русском или латинском языках;
- масштабируемость;
- гибкость в интеграции с другими системами;
- приятный интерфейс;
- простота в использовании.

Выводы по первой главе.

Первая глава включает изучение деятельности и бизнес-процессов компании ООО «Пласт». Полученные знания позволили провести моделирование процесса учета технического обслуживания компьютерной техники «как есть», провести анализ и выявить места, которые нуждаются в автоматизации. На основе чего была смоделирована модель «как должно быть», где показано, что изменится в процессе после автоматизации. Также в первой главе были определены функциональные и нефункциональные требования к системе.

Глава 2 Проектирование информационной системы

2.1 Модели и методологии проектирования и разработки информационной системы

При проектировании и разработке автоматизированной системы можно использовать различные модели и методологии [4].

Модель разработки Waterfall, также известная как каскадная модель, является одной из самых старых и наиболее традиционных методологий разработки программного обеспечения. Она представляет собой последовательную последовательность этапов, в которой каждый следующий этап начинается после завершения предыдущего. Этот подход подразумевает, что каждая фаза является предпосылкой для следующей, и проект продвигается вниз по «каскаду» этапов, как вода водопада.

Этапы модели Waterfall:

- анализ,
- проектирование,
- реализация,
- тестирование,
- внедрение,
- сопровождение.

Преимущества каскадной модели:

- структурированный и понятный процесс разработки;
- четкое определение требований на начальных этапах;
- хорошо подходит для проектов с небольшими изменениями требований в процессе разработки.

Недостатки каскадной модели:

- малая гибкость в отношении изменений требований;
- отсутствие возможности получения реальной обратной связи от пользователей до фазы внедрения;

- долгий цикл разработки.

Каскадная модель наиболее подходит для проектов с жесткими требованиями, когда они хорошо определены с самого начала и маловероятны изменения в ходе разработки [4].

Agile (Гибкая методология) — это набор подходов к разработке программного обеспечения, которые подчеркивают гибкость, сотрудничество, адаптивность и итеративность. Гибкая методология была создана для более эффективного реагирования на изменения в требованиях клиентов, обеспечения более коротких циклов разработки и создания более функциональных и полезных продуктов.

Основные подходы в Agile:

- Scrum,
- Kanban,
- Extreme Programming (XP),
- Crystal.

Преимущества Agile:

- гибкость в адаптации к изменениям;
- улучшенное сотрудничество и коммуникация внутри команды;
- возможность получения реальной обратной связи от заказчика в более ранних итерациях;
- постепенное создание функционального продукта через короткие циклы разработки.

Недостатки Agile:

- может потребовать большего участия и времени у заказчика;
- не всегда подходит для проектов с жесткими и четко определенными требованиями;
- может быть сложным в управлении при больших и сложных проектах.

Гибкая методология широко применяется для создания программного обеспечения, особенно в ситуациях, когда требования клиента могут изменяться, и важно быстро реагировать на эти изменения [10].

Инкрементальная модель разработки — это подход, который предполагает создание системы путем последовательного добавления функциональности в виде инкрементов или модулей. Каждый инкремент представляет собой небольшую часть функциональности, которая может быть разработана, протестирована и внедрена независимо от остальных компонентов системы [7].

Этот метод разработки позволяет создавать более функциональный продукт поэтапно, начиная с базовых возможностей и постепенно добавляя более сложные и расширенные функции. Каждый инкремент может быть внедрен в работающую систему, что позволяет пользователям начать пользоваться продуктом даже на ранних этапах разработки.

Преимущества инкрементальной модели:

- быстрое начало;
- обратная связь;
- постепенное улучшение;
- легкая адаптация.

Недостатки инкрементальной модели:

- сложность управления;
- возможные проблемы интеграции;
- потребность в дополнительных ресурсах.

Инкрементальная модель особенно полезна для проектов, где необходимо быстро предоставить базовую функциональность пользователям, а затем постепенно расширять возможности системы. Она позволяет сблизиться с реальными потребностями пользователей и быстрее внести изменения в продукт на ранних этапах разработки.

Для разработки информационной системы для внедрения в сервисный отдел для учета технического обслуживания наиболее подходящей методологией является водопадная модель.

Водопадная модель хорошо подходит, когда требования к системе четко определены и не подвержены значительным изменениям. Требования к системе, ее функциональности и процессам определены на раннем этапе.

2.2 Логическая модель информационной системы

С помощью проектирования и логической модели информационной системы, программисты смогут понять, как состоит архитектура системы. Модель строится с помощью разных диаграмм, которые показывают взаимодействие пользователей системы и функции, выполняемые ими в системе.

На диаграмме прецедентов изображены все пользователи, которые будут использовать систему и функциональность системы [2].

На диаграмме представлены два актора (сотрудник сервисного отдела и сотрудник компании), прецеденты — это функции допустимые для пользователей системы, рисунок 9.

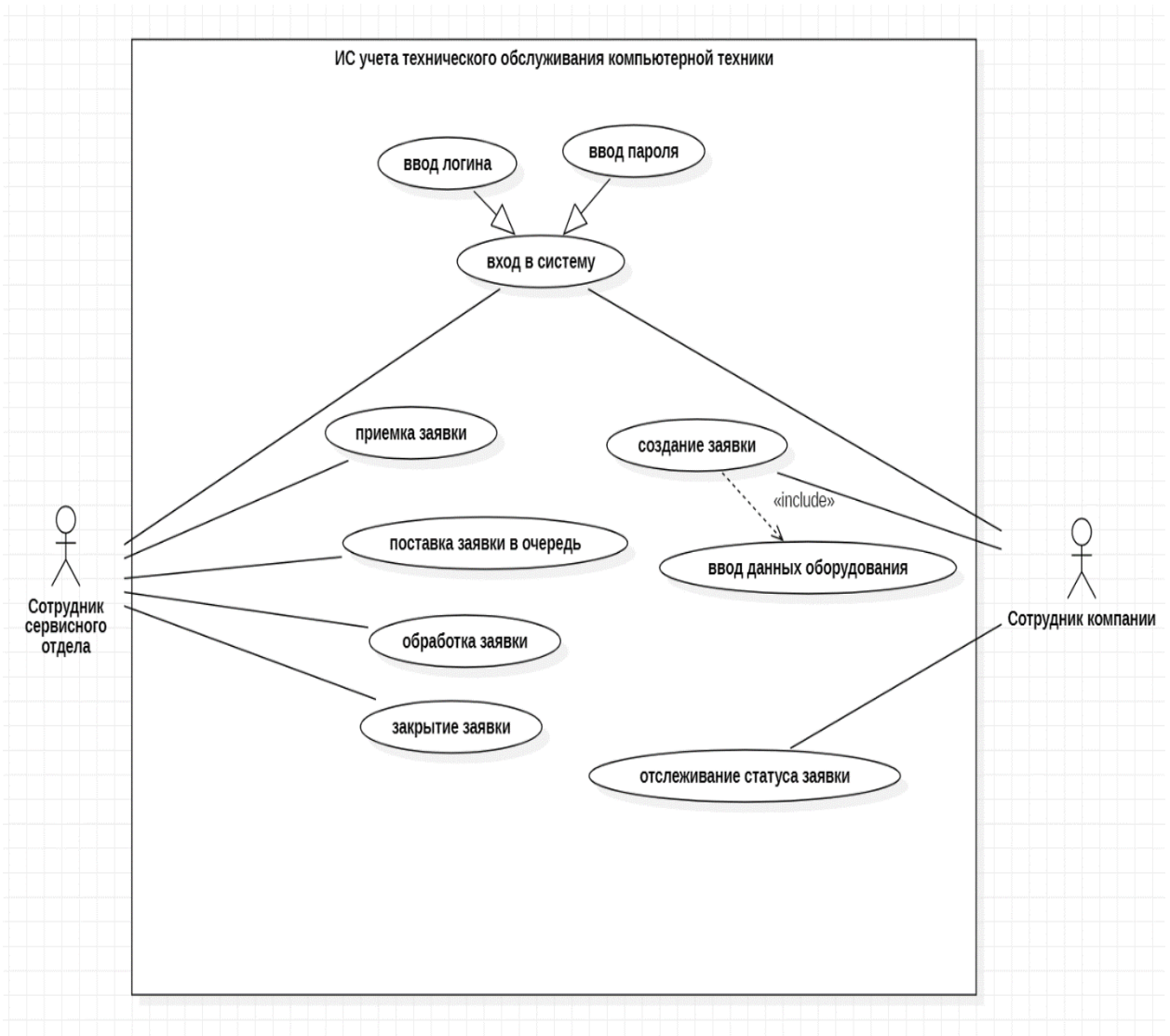


Рисунок 9 – Диаграмма вариантов использования системы

На диаграмме компонентов присутствует система учета технического обслуживания, операционная система Windows 10, веб-сервер, который будет использоваться и система управления базой данных MS SQL Server 2020. Диаграмма представлена на рисунке 10.

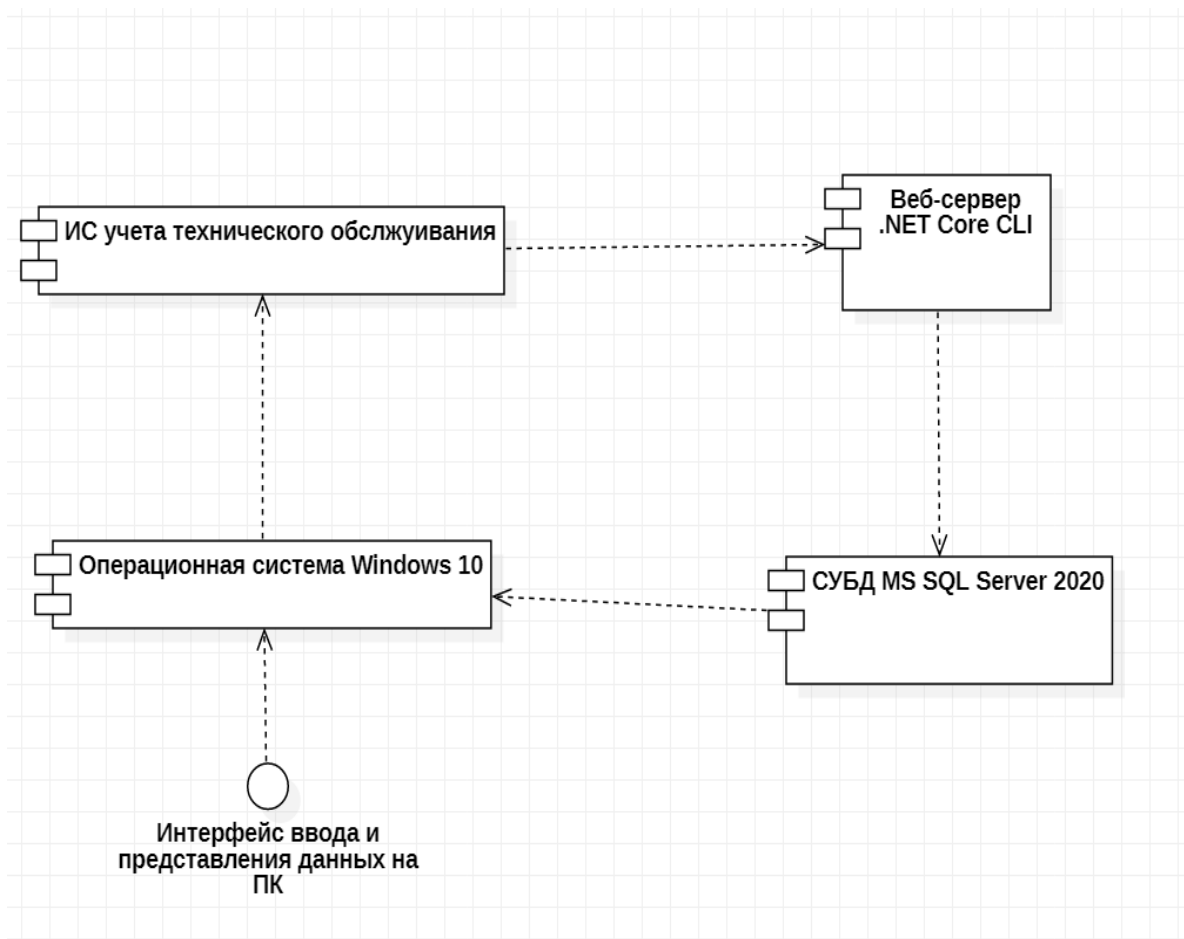


Рисунок 10 – Диаграмма компонентов

Разработанные модели позволяют разработчикам понять логику информационной системы для учета технического обслуживания.

Для реализации системы также потребуется база данных, которая будет содержать все поступающие данные процесса.

2.3 Проектирование базы данных информационной системы

Проведем инфологическое проектирование структуры базы данных.

Сущности:

Оборудование: Инвентарный номер, Название, Модель, Серийный номер, Дата установки.

Заявки: id заявки, Название заявки, id Сотрудника, id отдела, id оборудования, Дата заявки, Текущий статус заявки.

Сотрудники: id сотрудника, id отдела, Фамилия, Имя, Отчество, Должность.

Отделы: id отдела, Название отдела, Номер телефона.

Связи:

Связь между таблицами Оборудование и Заявка один-ко-многим. На одно и то же оборудование может быть составлено несколько заявок в разное время.

Связь между таблицами Заявка и Сотрудник имеет связь многие-к-одному. Один сотрудник отдела может за время составлять множество заявок на технического обслуживание компьютерной техники.

Связь между таблицами Отдел и Сотрудник имеют один-ко-многим. В одном отделе могут работать множество сотрудников.

Диаграмма ER-уровня концептуальной модели со слабыми сущностями представлена на рисунке 11.

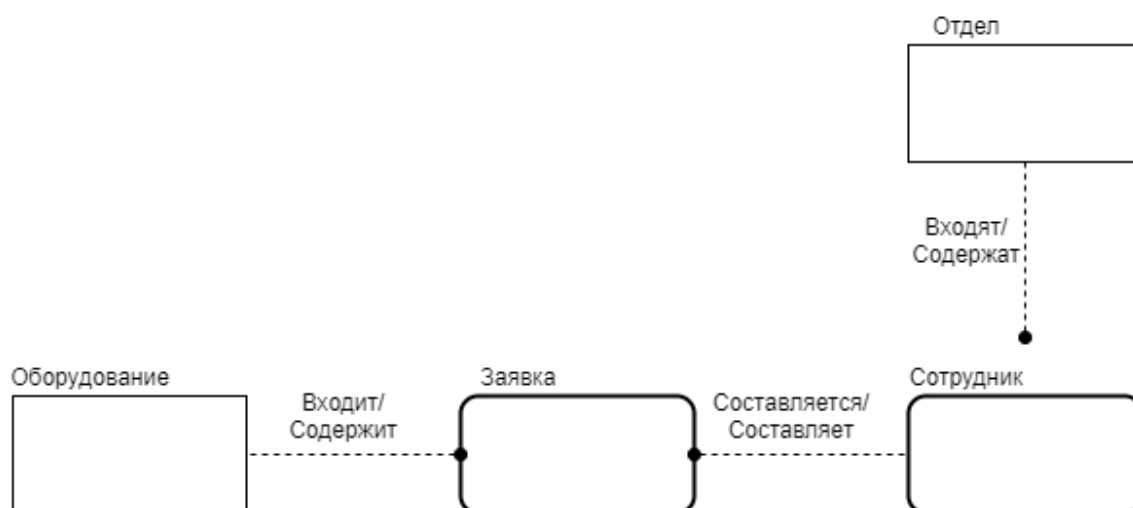


Рисунок 11 – Диаграмма ER-уровня концептуальной модели

Для каждой сущности ставится в соответствие отношение с тем же составом атрибутов.

В сущность Оборудование вводится суррогатный ключ Инвентарный номер, так как может быть оборудование с одинаковыми названиями. Этот же атрибут становится ключевым.

В сущности, Заявки так же вводится суррогатный ключ id заявки, он же и является ключевым атрибутом.

В сущности, Сотрудники вводится суррогатный ключ id сотрудника, так как могут сотрудники с одной фамилией, он же и является ключевым атрибутом.

В сущности, Отделы ключевой атрибут id отдела.

В отношении Оборудование входит в Заявку, добавляется атрибут: Инвентарный номер, отношение 1: М. Из этих же атрибутов состоит первичный ключ, так как это были ключевые атрибуты сущностей.

В отношении Заявка составляется Сотрудником, добавляются атрибуты: id отдела и id сотрудника, отношение М:1. Из этих же атрибутов состоит первичный ключ, так как это были ключевые атрибуты сущностей.

В отношении Сотрудники входят в Отделы, добавляется атрибут: id отдела, отношение М:1. Из этих же атрибутов состоит первичный ключ, так как это были ключевые атрибуты сущностей.

Построим диаграмму FA-уровня, рисунок 12.

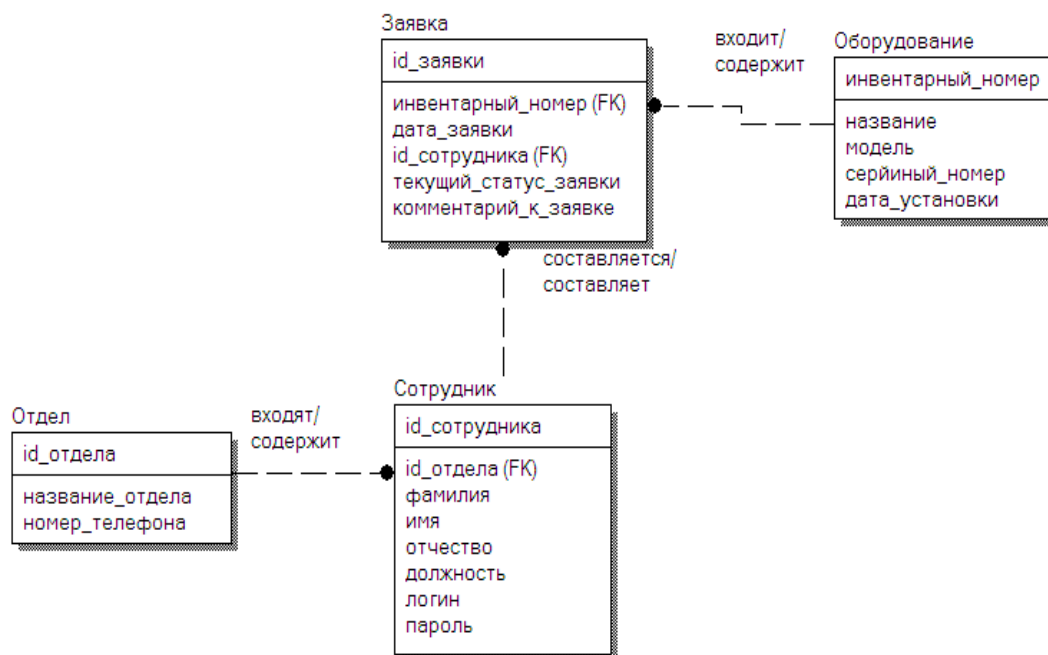


Рисунок 12 – Диаграмма FA-уровня

В таблице 2 показаны атрибуты каждой сущности, которые будут находиться в базе данных.

Таблица 2 –Атрибуты сущностей

Имя	Смысл	Тип	Сущность	Ограничение
id отдела	Идентификационный номер	Числовой	Отдел	Число: 1,2,3...
Название отдела	Название отдела	Текст	Отдел	255 СИМВОЛОВ
Номер телефона	Телефон отдела	Текст	Отдел	255 СИМВОЛОВ
id сотрудника	Идентификационный номер	Числовой	Сотрудник	Число: 1,2,3...
Фамилия	Фамилия сотрудника	Текст	Сотрудник	255 СИМВОЛОВ
Имя	Имя сотрудника	Текст	Сотрудник	255 СИМВОЛОВ

Продолжение таблицы 2

Имя	Смысл	Тип	Сущность	Ограничение
Отчество	Отчество сотрудника	Текст	Сотрудник	255 символов
Должность	Должность сотрудника	Текст	Сотрудник	255 символов
id заявки	Идентификационный номер	Текст	Заявка	Число: 1,2,3...
Название заявки	Название заявки	Текст	Заявка	15 символов
Дата заявки	Дата заявки	Дата	Заявка	Краткий формат даты
Текущий статус заявки	Статус заявки на текущий день	Текст	Заявка	255 символов
Инвентарный номер	Инвентарный номер оборудования	Текст	Оборудование	255 символов
Название	Название оборудования	Текст	Оборудование	255 символов
Модель	Модель оборудования	Текст	Оборудование	255 символов
Серийный номер	Серийный номер оборудования	Текст	Оборудование	255 символов
Дата установки	Дата установки оборудования	Текст	Оборудование	255 символов

Все типы данных присвоены атрибутам, можно переходить к физическому созданию базы данных, которое будет осуществляться в СУБД MS SQL Server.

2.4 Обоснование вида логической модели

Выбранная система управления базой данных работает локально после установки на ПК [20].

Установленная СУБД позволяет создавать таблицы, в которых будет храниться информация для осуществления процессов (рисунок 13).

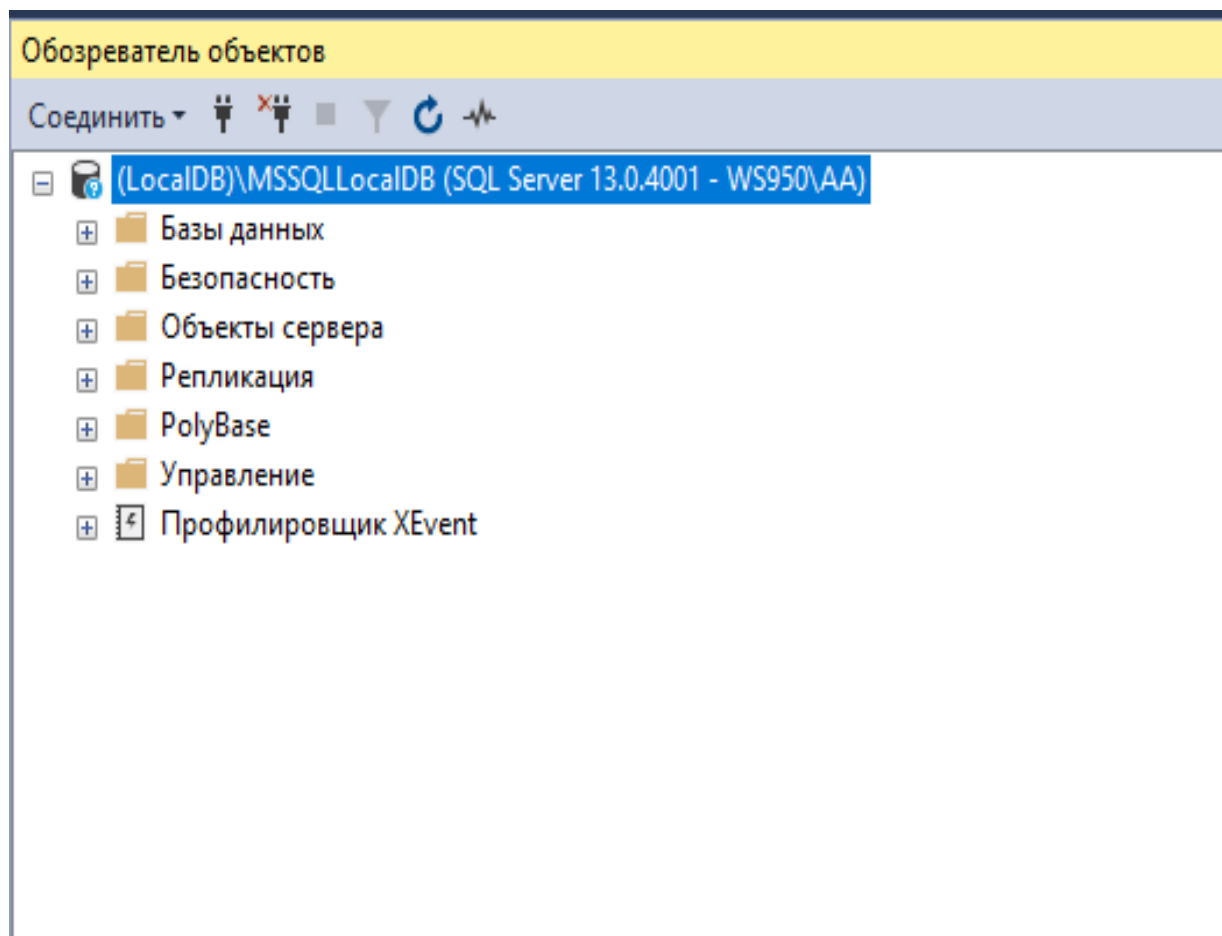


Рисунок 13 – Меню базы данных

База данных должна быть заполнена информацией и содержать только те типы данных, которые были ограничительной мерой для сохранения целостности базы данных.

2.5 Требования к аппаратно-программному обеспечению системы

Информационная система будет содержать базу данных, которая будет располагаться на сервере БД.

Сервер базы данных оснащен хранилищем данных. Хранилище поддерживает функцию резервного копирования.

На сервере базы данных установлено следующее программное обеспечение.

- операционная система - Windows Server;
- сетевые протоколы TCP/IP.

Рабочие места сотрудников сервисного отдела должны иметь персональный компьютер и все сопутствующие устройства.

Выводы по главе 2.

Во второй главе проведено проектирование логической модели информационной системы что включает диаграмму вариантов использования и диаграмму компонентов. Также проведено инфологическое проектирование базы данных, на основе чего построения модели базы данных.

Глава 3 Проектирование информационной системы

3.1 Архитектура информационной системы

Информационная система будет иметь архитектуру клиент-сервер.

Архитектура системы включает:

Пользовательский интерфейс — это часть, с которой логист будет непосредственно взаимодействовать. Интерфейс предоставляет пользователю данные, полученные из базы данных по средствам запросов.

Бизнес-логика и контроллеры - этот компонент обрабатывает логику системы, включая обработку запросов от пользовательского интерфейса, проверку данных, взаимодействие с базой данных и управление бизнес-процессами сервисного отдела, такими как учет технического обслуживания компьютерной техники.

Система управления базами данных (СУБД), хранение данных о заявках, оборудовании, отделах, сотрудниках и других сущностях. Обеспечивает выполнение операций чтения, записи и обновления данных [11].

Веб-сервер, обработка запросов от клиентов (веб-интерфейс, консольное приложение) и передача их на бизнес-логику.

Сервер баз данных, обеспечивает хранение данных, выполнение запросов и обеспечение целостности данных.

3.2 Выбор инструментов разработки

Для хранения поступающей информации в системе будет база данных, получение данных будет осуществляться после запроса пользователя к базе данных через интерфейс пользователя. Выберем основные инструменты для разработки информационной системы.

Для написания системы выберем язык программирования C#.

C# – считается самым популярным языком программирования, он позволяет интегрировать систему с разными инструментами и Фреймворками, разработанными как компанией Microsoft, так и другими компаниями [12],[13].

Microsoft Visual Studio 2022 – представляет собой набор интегрированных сред разработки, он, как и многие был разработан программистами Microsoft как бесплатная и ограниченная по функциям версия несвободной Microsoft Visual Studio [18].

Visual Studio достаточно многофункциональная, позволяет использовать множество качественных плагинов, что позволяет в будущем расширять функциональность приложения и подключить другие языки [19].

Бизнес-логика отделена от кода интерфейса, для написания которого используется декларативное программирование, что позволяет в дальнейшем расширять приложения и улучшает тестируемость.

Для написания интерфейса пользователя была использована платформа Windows Presentation Foundation (WPF), которая позволяет создавать клиентские приложения для настольных систем Windows с привлекательным пользовательским интерфейсом.

Системные требования:

- Intel Intel Pentium II 400 МГц процессор или совместимый;
- операционная система Microsoft Windows 7/8/10;
- ОЗУ не менее 128 МБ, рекомендуется 256 МБ;
- жесткий диск - 750 МБ свободного места для полной установки;
- CD-ROM или флеш-накопитель;
- SVGA монитор (800x600, 256 цветов) или с более высоким разрешением.

Таким образом для разработки системы выбрали все необходимые инструменты.

3.4 Разработка физической модели данных

Разработка физической модели происходит в самой СУБД [17]. Созданные таблицы показаны на рисунке 14.

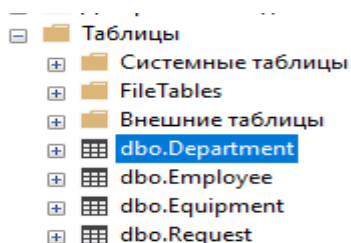


Рисунок 14 – Таблицы

Таблицы наполнены данными с помощью администратора системы который ответственный за ведение информации и ее обновлении, таблицы показаны на рисунке 15 -16.

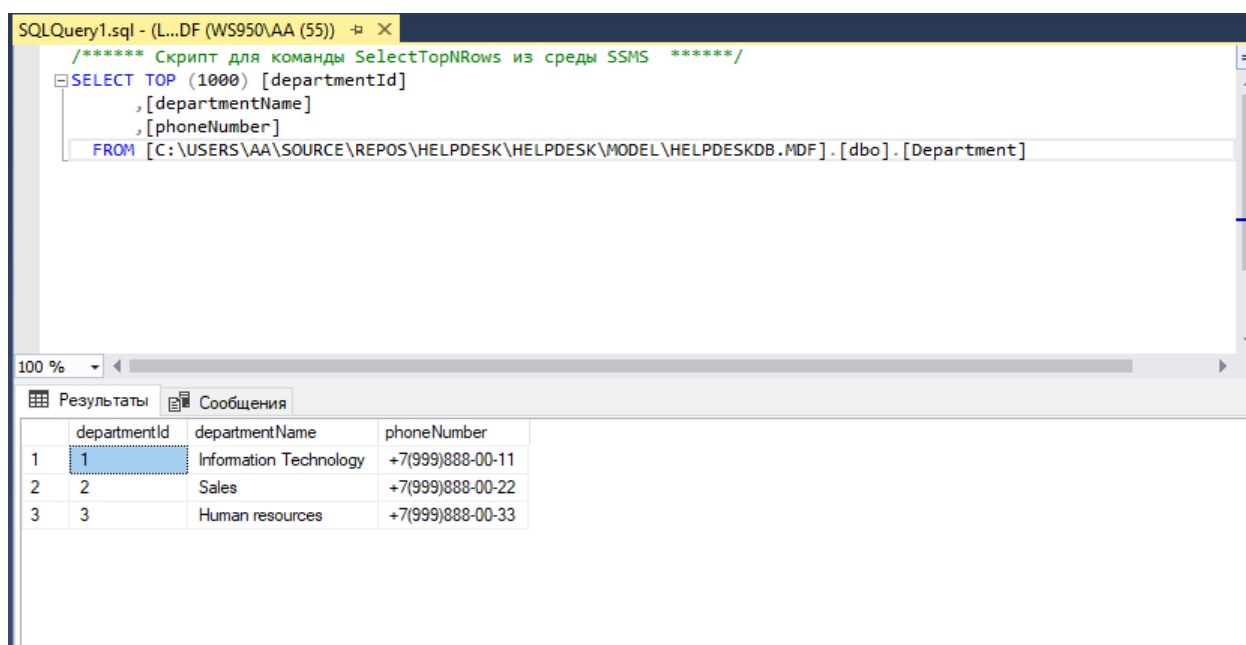


Рисунок 15 – Таблица Department

SQLQuery1.sql - (...P-QGV56DP\AA (70))

```

/***** Скрипт для команды SelectTopNRows из среды SSMS *****/
SELECT TOP (1000) [requestId]
, [requestName]
, [employeeId]
, [equipmentId]
, [requestDate]
, [currentRequestStatus]
FROM [C:\USERS\AA\DESKTOP\HELPDESK\HELPDESK\BIN\DEBUG\MODEL\HELPDESKDB.MDF].[dbo].[Request]

```

100 %

Результаты Сообщения

requestId	requestName	employeeId	equipmentId	requestDate	currentRequestStatus
1	Не работает монитор	5	10	2024-03-25 00:00:00.000	В работе
2	Не включается принтер	3	2	2024-04-01 00:00:00.000	Новая заявка
3	Заменить картридж	7	2	2024-04-01 01:42:55.030	Новая заявка
4	Заменить мышь	7	7	2024-04-01 01:47:10.447	Новая заявка
5	не работает клавиатура	7	8	2024-04-01 01:47:58.333	Новая заявка
6	Почистить монитор	7	1	2024-04-02 01:49:06.510	Новая заявка
7	Не работает мышь	5	7	2024-04-05 01:51:35.077	Новая заявка
8	Не работает	7	1	2024-04-07 11:50:48.290	Новая заявка
9	Не работает	7	4	2024-04-08 13:48:59.060	Новая заявка
10	Не работает	7	3	2024-04-05 22:08:34.813	Новая заявка
11	Не работает	7	5	2024-04-08 22:12:31.737	Новая заявка
12	Не работает	7	6	2024-04-10 22:14:48.590	Новая заявка
13	Не работает	7	2	2024-04-12 22:15:00.043	Новая заявка
14	Не работает	7	6	2024-04-15 22:17:16.587	Новая заявка
15	Не работает	7	4	2024-04-16 22:45:42.193	Новая заявка
16	Не работает	7	4	2024-04-18 09:05:46.130	Новая заявка
17	Сломался монитор	7	3	2024-04-17 15:36:26.020	Новая заявка
18	Необходима чистка клавиатуры	7	10	2024-04-18 15:42:53.763	Новая заявка

Рисунок 16 – Таблица Request

После создания и наполнения таблиц укажем связь между ними, схема базы данных представлена на картинке 17.

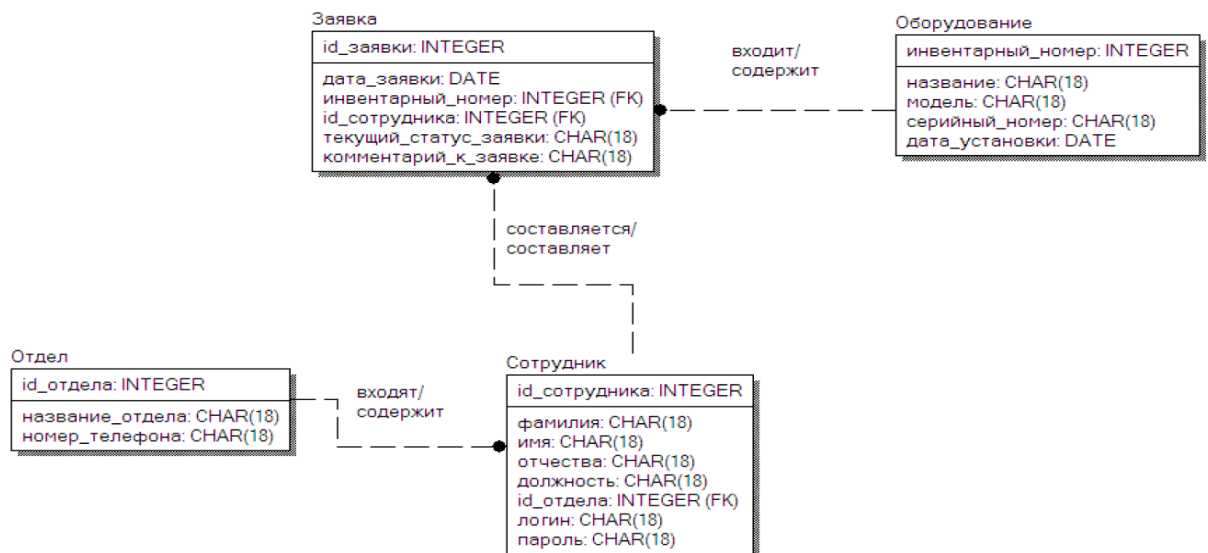


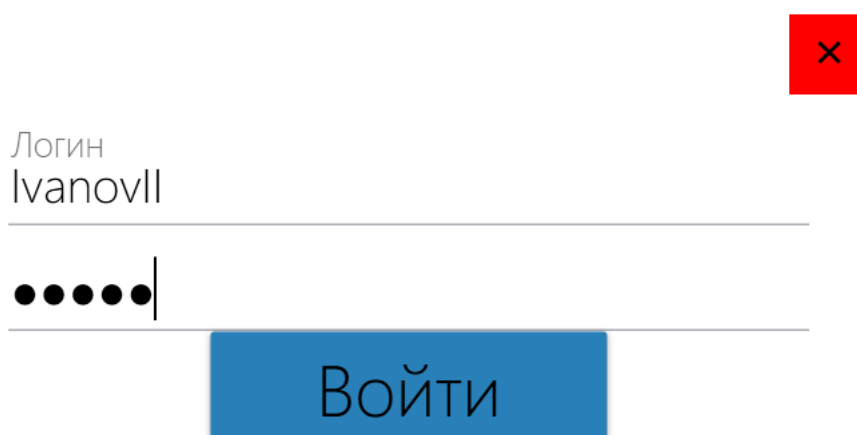
Рисунок 17 – Схема базы данных

Для дальнейшей работы с базой данных, дополнительных настроек не требуется, необходимо только подключить ее к информационной системе и вводить запросы, по средствам которых получать данные из базы данных.

3.5 Разработка автоматизированной информационной системы

Запуск приложения требует авторизации, чтобы пройти авторизация каждый пользователь будет иметь свой логин и пароль, которые им будет присваивать администратор системы при трудоустройте в компанию. На рисунке 18 представлена форма для ввода данных прохождения авторизации.

Программный код методов системы приведен в Приложении А.



The image shows a login form interface. At the top right, there is a red square button with a white 'X' icon. Below it, the text 'Логин' (Login) is displayed above the input field containing 'IvanovI'. Below the login field is a horizontal line. Underneath this line is a password field represented by five black dots and a vertical cursor line. Below the password field is another horizontal line. At the bottom center, there is a blue rectangular button with the white text 'Войти' (Login).

Рисунок 18 – Окно авторизации в системе

После прохождения авторизации сотрудник сервисного отдела попадает в окно со списком оборудования, рисунке 19.

ID заявки	Название	Фамилия	Имя	Отдел	Оборудование	Модель	S/N	Дата поступления	Текущий статус	Новый статус
1	Не работает монитор	Lukina	Veronika	Human resources	Keyboard	G505	12554211c	25.03.2024 00:00:00	В работе	В работе
2	Не включается принтер	Pronskiy	Sila	Sales	Printer	Hp1010	an1097	01.04.2024 00:00:00	Новая заявка	Новая заявка
3	Заменить картридж	Filatova	Olga	Human resources	Printer	Hp1010	an1097	01.04.2024 01:42:55	Новая заявка	Новая заявка
4	Заменить мышь	Filatova	Olga	Human resources	Mouse	G503	12536631c	01.04.2024 01:47:10	Новая заявка	Новая заявка
5	не работает клавиатура	Filatova	Olga	Human resources	Keyboard	G505	12555631c	01.04.2024 01:47:58	Новая заявка	Новая заявка
6	Почистить монитор	Filatova	Olga	Human resources	Monitor	UE51	2565411	02.04.2024 01:49:06	Новая заявка	Новая заявка
7	Не работает мышь	Lukina	Veronika	Human resources	Mouse	G503	12536631c	05.04.2024 01:51:35	Новая заявка	Новая заявка
8	Не работает	Filatova	Olga	Human resources	Monitor	UE51	2565411	07.04.2024 11:50:48	Новая заявка	Новая заявка
9	Не работает	Filatova	Olga	Human resources	Mouse	G503	12586611d	08.04.2024 13:48:59	Новая заявка	Новая заявка
10	Не работает	Filatova	Olga	Human resources	Monitor	UE51	2565412	05.04.2024 22:08:34	Новая заявка	Новая заявка
11	Не работает	Filatova	Olga	Human resources	Mouse	G503	12586611f	08.04.2024 22:12:31	Новая заявка	Новая заявка
12	Не работает	Filatova	Olga	Human resources	Mouse	G503	12586611c	10.04.2024 22:14:48	Новая заявка	Новая заявка
13	Не работает	Filatova	Olga	Human resources	Printer	Hp1010	an1097	12.04.2024 22:15:00	Новая заявка	Новая заявка
14	Не работает	Filatova	Olga	Human resources	Мышь	G503	12586611c	15.04.2024 22:17:16	Новая заявка	Новая заявка

Рисунок 19 – Список заявок

Перед тем как оформить заявку на обслуживания оборудования сотрудник сможет выбрать из списка необходимое устройство просто нажав на него два раза. Для этого в системе предусмотрен фильтр, который облегчит процесс создания заявки (рисунок 20), фильтрацию можно провести разными способами (рисунок 21).

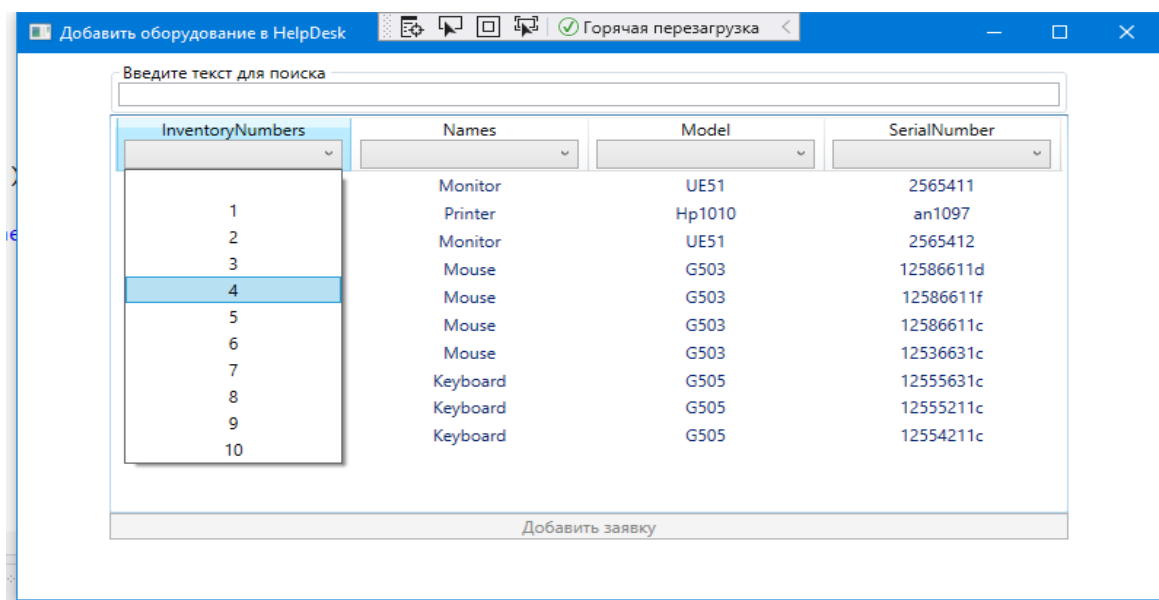


Рисунок 20 – Фильтр по инвентарному номеру оборудования

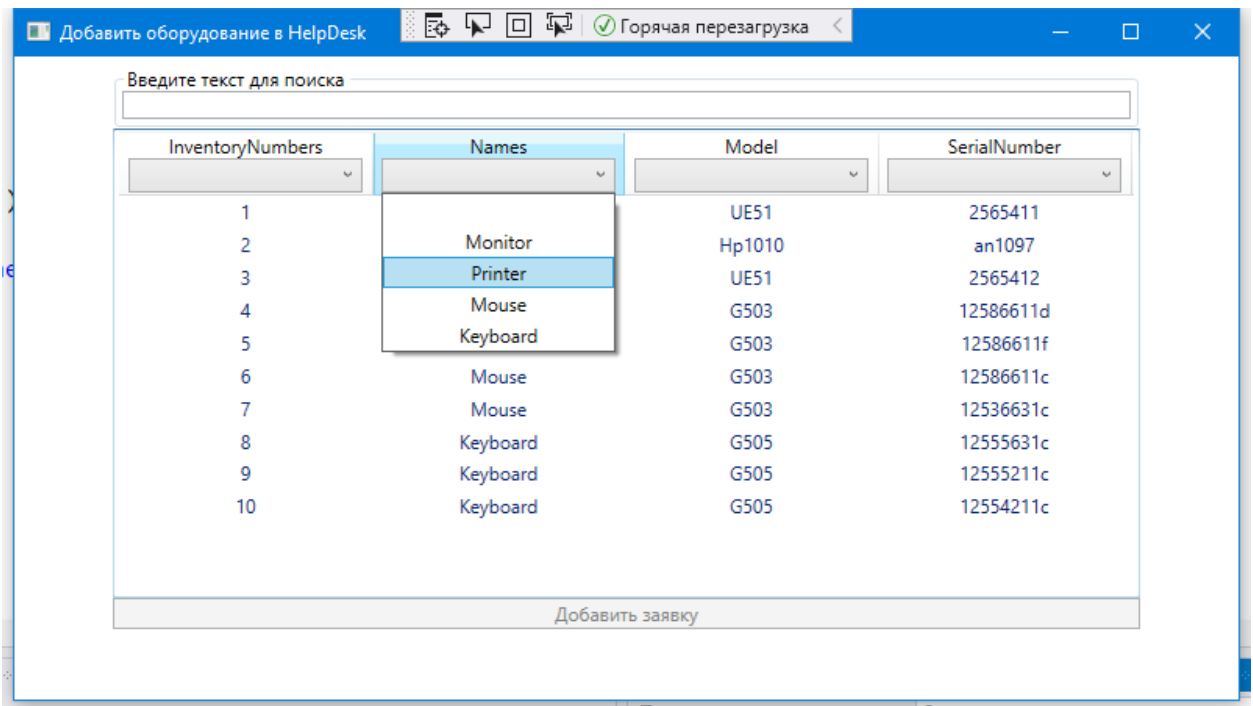


Рисунок 21 – Фильтр по названию оборудования

После выбора нужного оборудования необходимо нажать кнопку «Добавить заявку», оборудование добавится в заявку (рисунок 22).

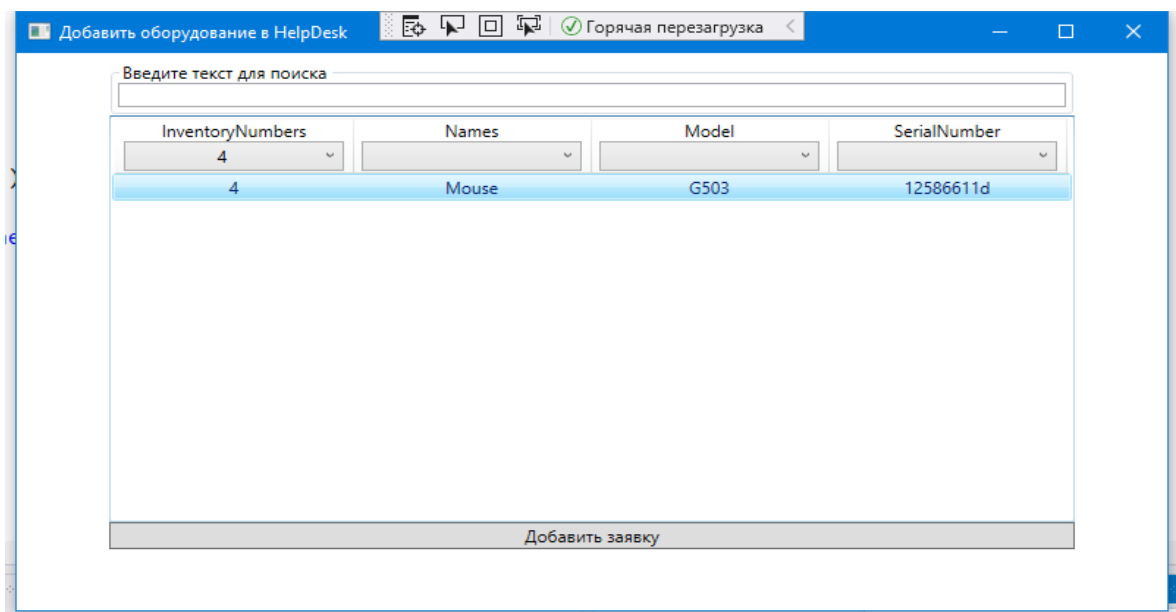


Рисунок 22 – Добавление данных в заявку

После создания заявки пользователь может оставить комментарий к заявке, если считает необходимым (рисунок 23).

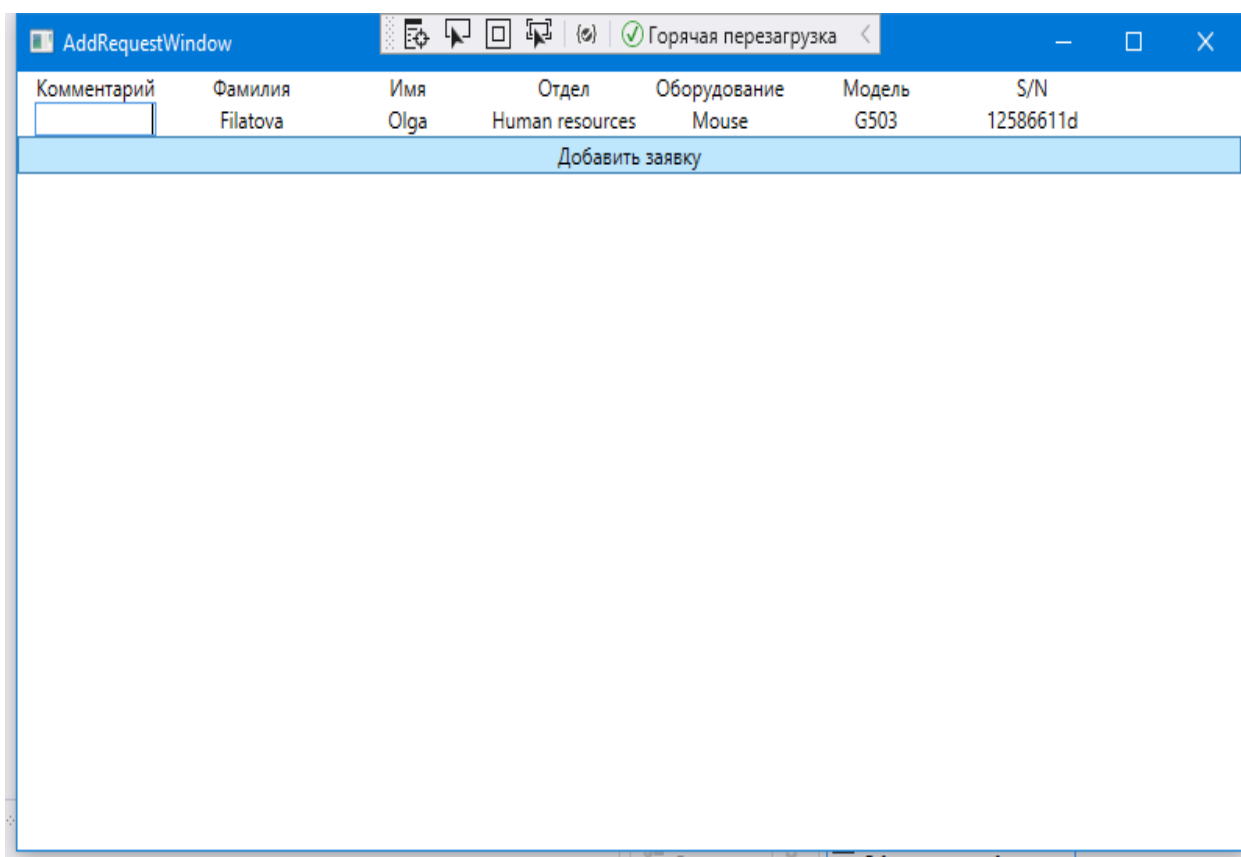


Рисунок 23 – Добавление комментария к заявке

Сотрудник сервисного отдела будет получать новые заявки после того, как обновит список заявок.

3.6 Руководство пользователя

Работа приложения начинается после того, как пользователь установит у себя на ПК систему и войдет в нее под своими данными, которые также присваивает администратор системы.

Система имеет два пользователя: сотрудник сервисного отдела и сотрудник, который будет отправлять заявки.

Сотрудник, создающий заявки на обслуживание оборудования увидит форму для заполнения сразу после того, как войдет в систему. После заполнения заявки при нажатии кнопки «Сохранить» пользователь подтверждает, что хочет добавить заявку на обслуживание.

После того как пользователь подтвердил добавление заявки она сохраняется в общий список заявок, который периодически обновляется, и сотрудники сервисного отдела могут видеть вновь поступившие заявки на обслуживание.

Завершение работы в системе подтверждается кнопкой «Выйти».

Выводы по третьей главе

Третья глава включает выполнения проектирования информационной системы. Проектирование подразумевает выбор необходимых инструментов для реализации информационной системы, которая будет помогать вести учет технического обслуживания компьютерной техники. Также были представлены скриншоты пользовательского интерфейса и показаны методы, осуществляемые в системе.

Заключение

В результате выполнения выпускной квалификационной работы были выполнены задачи, которые помогли достичь поставленной цели.

В работе был проведен глубокий обзор предметной области и деятельность компании ООО «Пласт». На основе обзора были выделены слабые места рассматриваемого процесса учета технического обслуживания компьютерной техники и сформированы функциональные требования к системе для дальнейшей разработки.

Также был проведен обзор аналогичных систем, которые представлены на российском рынке в готовом виде и выделены плюсы и минусы при внедрении таких систем, что помогло обосновать собственную разработку данной системы.

С помощью сравнительного анализа были выбраны ЯП C#, СУБД MSSQL Server и IDE Visual Studio 2022, выбранные инструменты будут реализованы при разработке информационной системы, также разработаны все необходимые модели, которые показывают архитектуру информационной систему и взаимосвязь пользователей с системой, построены таблицы для правильной реализации системы для сервисного отдела компании.

Разработанная автоматизированная информационная система успешно протестирована, в ходе которого показала корректную работу всех функций, которые были реализованы в ней. При вводе данных при ожидании получения определенного результата информационная система показывает ожидаемый результат при выводе информации на монитор пользователя.

Результатом работы является разработанная информационная система, автоматизирующая процесс учета технического обслуживания компьютерной техники в компании ООО «Пласт».

Список используемой литературы и используемых источников

1. Бистерфельд О.А. Методология функционального моделирования IDEF0: учебно-методическое пособие / О.А. Бистерфельд ; Ряз. гос. ун-т им. С.А. Есенина. — Рязань, 2008. — 48 с.
2. Горбаченко В. И. Проектирование информационных систем с СА ERwin Modeling Suite 7.3: учебное пособие / В. И. Горбаченко, Г. Ф. Убиенных, Г. В. Бобрышева – Пенза: Изд-во ПГУ, 2012. – 154 с
3. Кларис-Service Desk: Web система для работы с заявками онлайн [Электронный ресурс]. – URL: <http://lpsd.claris.su/?yclid> (дата обращения: 08.04.2024)
4. Онлайн-курс обучения программированию: методологии разработки. [Электронный ресурс]. – URL: <https://javarush.com/groups/posts/647-metodologii-razrabotki-po> (дата обращения: 08.04.2024)
5. ООО «Пласт». [Электронный ресурс]. – URL: <https://companium.ru/id/1136312001154-plast> (дата обращения: 08.04.2024)
6. Основы системного анализа и моделирование технологических процессов. [Электронный ресурс]. – URL: https://www.rsatu.ru/upload/medialibrary/e40/Konspekt-lektsiy_Osnovy-sistemnogo-analiza-i-modelirovanie-tekhnologicheskikh-protssesov.pdf (дата обращения: 08.04.2024)
7. Отличия, достоинства и недостатки инкрементной модели процесса. [Электронный ресурс]. – URL: <https://www.geeksforgeeks.org/advanategs-and-disadvanategs-of-incremental-process-model/> (дата обращения: 08.04.2024)
8. Ракитов А.И., Бондяев Д.А., Романов И.Б., Егерев С.В., Щербаков А.Ю. Системный анализ и аналитические исследования: руководство для профессиональных аналитиков / А.И. Ракитов, Д.А. Бондяев, И.Б. Романов, С.В. Егерев, А.Ю. Щербаков; [отв. ред. А.И. Ракитов]. - М.,2019. - 448 с

9. Системный анализ и моделирование информационных процессов и систем. [Электронный ресурс]. – URL: <https://books.ifmo.ru/file/pdf/2140.pdf> (дата обращения: 08.04.2024)
10. Черушева Т. В. Проектирование программного обеспечения: учеб. пособие / Т. В. Черушева. – Пенза: Изд-во ПГУ, 2014. – 172 с.
11. Access: База данных. [Электронный ресурс]. – URL: <https://www.microsoft.com/ru-ru/microsoft-365/access>(дата обращения: 08.04.2024)
12. GeekBrains – Язык программирования C#. [Электронный ресурс]. - URL: <https://geekbrains.ru/posts/yazyk-programmirovaniya-c-sharp-istoriya-spezifika-mesto-na-rynke> (дата обращения: 08.04.2024)
13. Helpiks.org: Достоинства и недостатки языка [Электронный ресурс]. - URL: <https://helpiks.org/6-21879.html> (дата обращения: 08.04.2024)
14. HubEx: облачная платформа управления сервисным обслуживанием. [Электронный ресурс]. – URL: <https://hubex.ru/features/inside/?yclid> (дата обращения: 08.04.2024)
15. KazEdu: принципы эффективности АИС. [Электронный ресурс]. – URL: <https://www.kazedu.kz/referat/98670/1> (дата обращения: 08.04.2024)
16. Naumen: программные продукты и решения Naumen. [Электронный ресурс]. – URL: <https://www.naumen.ru/products/> (дата обращения: 08.04.2024)
17. SoftClipper: что такое SQL Server. [Электронный ресурс]. – URL: <https://softclipper.net/foxpro-i-sql/sravnenie-baz-dannykh-microsoft-sql-server-i-microsoft-visual-foxpro.html> (дата обращения: 08.04.2024)
18. Download Visual Studio 2022 Web Installer / ISO (Community / Professional / Enterprise). [Электронный ресурс]. – URL: <https://developerinsider.co/download-visual-studio-2022-web-installer-iso-community-professional-enterprise/> (дата обращения: 08.04.2024)

19. Download and Install Visual Studio 2022. [Электронный ресурс]. – URL: <https://www.itechtics.com/visual-studio-2022/> (дата обращения: 08.04.2024)

20. SQL Server 2022. [Электронный ресурс]. – URL: <https://download-original-windows.ru/msdn-sql-server/sql-server-2022/> (дата обращения: 08.04.2024)

Приложение А

Программный код

```
/// <summary>
/// Логика взаимодействия для MainWindow.xaml
/// </summary>
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
    }
}
class EmployeeService
{
    static string connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;
    public static List<Employee> GetAllEmployees()
    {
        List<Employee> employees = new List<Employee>();
        //string sql = "SELECT * FROM Employee";

        string sql = "SELECT Employee.employeeId," +
            "Employee.employeeId," +
            "Employee.lastName," +
            "Employee.firstName," +
            "Employee.middleName," +
            "Employee.position," +
```

Продолжение приложения А

```
"Employee.login," +
    "Employee.password," +
    "Department.departmentId," +
    "Department.departmentName," +
    "Department.phoneNumber " +
    "FROM Employee " +
    "JOIN Department ON Department.departmentId =
Employee.departmentId";
    {
        DepartmentId = (int)row["departmentId"],
        DepartmentName = (string)row["departmentName"],
        PhoneNumber = (string)row["phoneNumber"]
    },
    Login = (string)row["login"],
    PasswordMd5 = (string)row["password"]
};
employees.Add(employee);
    }
}
return employees;
}
}
}
class EquipmentService
{
```

Продолжение приложения А

```
static string connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString
;

public static List<Equipment> GetAllEquipments()
{
    List<Equipment> equipments = new List<Equipment>();
    string sql = "SELECT * FROM Equipment";
    string tableName = "Equipment";
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        connection.Open();
        Equipment equipment;
        SqlDataAdapter adapter = new SqlDataAdapter(sql, connection);
        DataSet ds = new DataSet();
        adapter.Fill(ds, tableName);
        foreach (DataTable dt in ds.Tables)
        {
            foreach (DataRow row in dt.Rows)
            {
                equipment = new Equipment()
                {
                    InventoryNumber = (int)row["inventoryNumber"],
                    Name = (string)row["name"],
                    Model = (string)row["model"],
                    SerialNumber = (string)row["serialNumber"],
                    InstallationDate = (DateTime)row["installationDate"],
                };
                equipments.Add(equipment);
            }
        }
    }
}
```

Продолжение приложения А

```
}  
    }  
    }  
    return equipments;  
    }  
    }  
}  
internal class Request  
{  
    private int requestId;  
  
    public int RequestId  
    {  
        get { return requestId; }  
        set { requestId = value; }  
    }  
    private string requestName;  
    public string RequestName  
    {  
        get { return requestName; }  
        set { requestName = value; }  
    }  
    private Employee currentEmployee;  
    public Employee CurrentEmployee  
    {  
        get { return currentEmployee; }  
        set { currentEmployee = value; }  
    }  
}
```

Продолжение приложения А

```
private Equipment currentEquipment;
    public Equipment CurrentEquipment
    {
        get { return currentEquipment; }
        set { currentEquipment = value; }
    }
private DateTime requestDate;
public DateTime RequestDate
{
    get { return requestDate; }
    set { requestDate = value; }
}
private string currentRequestStatus;
public string CurrentRequestStatus
{
    get { return currentRequestStatus; }
    set { currentRequestStatus = value; }
}
private string newRequestStatus;
public string NewRequestStatus
{
    get { return newRequestStatus; }
    set { newRequestStatus = value; }
}
public List<string> Statuses { get; set; }
}
class RequestService
```

Продолжение приложения А

```
static string connectionString =
ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString
;

public static ObservableCollection<Request> GetAllRequests()
{
    ObservableCollection<Request> requests = new
ObservableCollection<Request>();
    string sql = "SELECT Request.requestId," +
        "Request.requestName," +
        "Request.requestDate," +
        "Request.currentRequestStatus," +
        "Employee.employeeId," +
        "Employee.lastName," +
        "Employee.firstName," +
        "Employee.middleName," +
        "Employee.position," +
        "Employee.login," +
        "Employee.password," +
        "Department.departmentId," +
        "Department.departmentName," +
        "Department.phoneNumber," +
        "Equipment.inventoryNumber," +
        "Equipment.name," +
        "Equipment.model," +
        "Equipment.serialNumber," +
        "Equipment.installationDate " +
```

Продолжение приложения А

```
"FROM Request " +  
    "JOIN Employee ON Employee.employeeId = Request.employeeId  
" +  
    "JOIN Department ON Department.departmentId =  
Employee.departmentId " +  
    "JOIN Equipment On Equipment.inventoryNumber =  
Request.equipmentId ";
```

```
using (SqlConnection connection = new SqlConnection(connectionString))  
{  
    connection.Open();  
  
    Request request;  
    SqlDataAdapter adapter = new SqlDataAdapter(sql, connection);  
  
    DataSet ds = new DataSet();  
  
    adapter.Fill(ds);  
  
    foreach (DataTable dt in ds.Tables)  
    {  
        foreach (DataRow row in dt.Rows)  
        {  
            request = new Request()  
            {  
                RequestId = (int)row["requestId"],  
                RequestName = (string)row["requestName"],
```

Продолжение приложения А

```
RequestDate = (DateTime)row["requestDate"],
    CurrentRequestStatus = (string)row["currentRequestStatus"],
    NewRequestStatus = (string)row["currentRequestStatus"],
    CurrentEmployee = new Employee()
    {
        EmployeeId = (int)row["employeeId"],
        LastName = (string)row["lastName"],
        FirstName = (string)row["firstName"],
        MiddleName = (string)row["middleName"],
        Position = (string)row["position"],
        CurrentDepartment = new Department()
        {
            DepartmentId = (int)row["departmentId"],
            DepartmentName = (string)row["departmentName"],
            PhoneNumber = (string)row["phoneNumber"]
        },
        Login = (string)row["login"],
        PasswordMd5 = (string)row["password"]
    },
    CurrentEquipment = new Equipment()
    {
        InventoryNumber = (int)row["inventoryNumber"],
        Name = (string)row["name"],
        Model = (string)row["model"],
        SerialNumber = (string)row["serialNumber"],
        InstallationDate = (DateTime)row["installationDate"],
    };
```


Продолжение приложения А

```
requests.Add(request);
    }
}
}
List<string> statuses = GetAllStatuses();
foreach (var oneRequest in requests)
{
    oneRequest.Statuses = statuses;
}
return requests;
}
public static void AddRequest(Request request)
{
}
}
public static void UpdateRequest(Request request)
{
}
private static List<string> GetAllStatuses()
{
    List<string> statuses = new List<string>();
    statuses.Add("Новая заявка");
    statuses.Add("В работе");
    statuses.Add("Закрыта");
    return statuses;
}
```