

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 Автоматизация управления заявками на обслуживание оргтехники банка

Обучающийся

Ю. В. Леонов

(И.О. Фамилия)

(личная подпись)

Руководитель

канд. пед. наук, доцент, О.В. Оськина

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

Тольятти 2024

Аннотация

Выпускная квалификационная работа посвящена автоматизации обработки запросов на техническое обслуживание офисного оборудования, эксплуатируемого в банке.

Целью данной работы является изучение бизнес-процессов компании и разработка системы для автоматизации управления запросами на техническое обслуживание офисного оборудования в ПАО "Росбанк".

Актуальность выбранной темы заключается в целесообразности сокращения времени в получении информации о необходимости проведения обслуживания оргтехники, также есть необходимость управлять заявками распределяя их по исполнителям и следить за их выполнением.

Задачи работы:

- изучить деятельность банка ПАО «Росбанк»;
- изучить процесс приема заявок на обслуживание оргтехники банка;
- провести проектирование информационной системы, которая позволит автоматизировать процесс управления заявками на обслуживание оргтехники банка;
- разработать информационную систему для ПАО «Росбанк».

Результатом работы будет информационная система автоматизации процесса управления заявками на обслуживание оргтехники, в рамках банковского бизнес-процесса. Система будет позволять сотрудникам банка разных отделов создавать самостоятельно заявку и отправлять ее с помощью системы в отдел обслуживания оргтехники. Инженеры отдела будут получать заявки также в системе и распределять их между сотрудниками согласно нагрузке.

Первая глава работы содержит в себе изученную предметную область деятельности банка ПАО «Росбанк», на базе отдела обслуживания

технических средств банка. Так же в организации среди существующих информационных систем был проведён анализ для последующего сравнения их между собой. В процессе исследования подверглись изучению бизнес-процессы банка, а также изучение нормативной базы, с которой работают инженеры в процессе обслуживания оргтехники банка. Результатом анализа предметной области, является формализация протекающего в компании бизнес-процесса в виде модели процесса с использованием нотации IDEF0.

Вторая глава посвящена проектированию разрабатываемой информационной системы, с учётом проведенного анализа предметной области. В ней описывается структура информационной системы, для автоматизации процесса управления заявками на обслуживание оргтехники. С использованием нотации UML, для всех основных участников бизнес-процесса была создана модель взаимодействия, а также выделены основные функции, каждого из участников, которые будут использоваться в процессе разработки и последующего использования разрабатываемой системы. Далее, на основании полученных данных, была спроектирована модель в виде модели сущность-связь, на основании которой была разработана логическая модель данных создаваемой информационной системы. Безусловно были учтены и системные требования к создаваемой информационной системе.

Третья глава, посвящается информационной системе, ее разработке, с обоснованием выбора и применения технологических решений, моделированию интерфейса для пользователей, экранных форм, меню, навигации. Для успешной интеграции данной системы в работу была посчитана и обоснована экономическая эффективность проекта.

Данная выпускная квалификационная работа состоит из 76 страниц текста, 27 рисунков, 3 таблиц и 23 источника.

Оглавление

Введение	6
Глава 1 Исследование деятельности компании ПАО «Росбанк»	8
1.1 Изучение и анализ структуры компании ПАО «Росбанк»	8
1.2 Описание и моделирование исследуемого процесса.....	11
1.2.1 Обзор и выбор существующих инструментов для моделирования бизнес-процессов.....	13
1.2.2 Разработка модели бизнес-процесса «как есть»	15
1.2.3 Обоснование необходимости автоматизированного варианта решения.....	19
1.3 Сравнительный анализ существующих аналогов информационных систем и их соответствие требованиям.....	20
1.4 Реинжиниринг модели процесса «как есть»	25
1.5 Постановка задачи на разработку автоматизированной системы управления заявками.....	28
Глава 2 Логическое проектирование системы управления заявками.....	31
2.1 Выбор технологий проектирования	31
2.2 Логическая модель приложения	32
2.3 Информационное обеспечение системы	39
2.3.1 Характеристика нормативно-справочной, входной и оперативной информации	39
2.3.2 Характеристика выходной информации	39
2.4 Проектирование базы данных.....	40
2.4.1 Выбор технологии проектирования базы данных	40
2.4.2 Разработка концептуальной модели данных.....	41
2.4.3 Обоснование вида логической модели	43
2.4.4 Разработка логической модели данных.....	44

2.5 Требования к аппаратно-программному обеспечению	47
Глава 3 Физическое проектирование системы управления заявками на обслуживание оргтехники	49
3.1 Выбор архитектуры информационной системы	49
3.2 Выбор технологии разработки программного обеспечения	51
3.3 Выбор системы управления базой данных информационной системы	52
3.4 Разработка физической модели данных	54
3.5 Разработка информационной системы	57
3.6 Описание функциональности информационной системы	62
3.7 Обоснование экономической эффективности разработки приложения	66
Заключение	72
Список используемой литературы и используемых источников	73

Введение

Основной целью данной выпускной квалификационной работы является создание информационной системы, предназначенной для автоматизации управления заявками на техническое обслуживание офисного оборудования банка.

Банк осуществляет надзор как за наличными, так и за безналичными платежными операциями, а также мобилизует средства за счет депозитов вкладчиков и выдает кредиты для содействия росту бизнеса или личных потребностей. В процессе обслуживания клиентов сотрудники банка часто используют различное офисное оборудование для предоставления клиентам дополнительной информации о вкладах, кредитных операциях, а также об открытии или закрытии депозитных счетов. Такое интенсивное использование неизбежно приводит к износу офисного оборудования, что в дальнейшем неизбежно потребует его технического обслуживания или ремонт.

Внедрение автоматизированной системы значительно сократит время для получения и обработки информации, позволит оптимально использовать ресурсы, упростить работу отдела технического обслуживания [1].

Цель работы – разработать систему управления заявками, поступающими на обслуживание оргтехники в банке ПАО «Росбанк», которая должна функционировать в автоматическом режиме.

Задачи работы:

- изучить деятельность банка ПАО «Росбанк»;
- изучить процесс приема заявок на обслуживание оргтехники;
- провести проектирование информационной системы, которая позволит автоматизировать процесс управления заявками на обслуживание оргтехники;
- разработать информационную систему для ПАО «Росбанк».

Объектом исследования является процесс управления заявками на обслуживание оргтехники ПАО «Росбанк».

Исследование посвящено автоматизации управления запросами на техническое обслуживание офисного оборудования банка.

Выпускная квалификационная работа включает введение, три раздела, заключение и список использованной литературы и источников.

В первой главе рассматривается деятельность ПАО "Росбанк". Данное исследование включает обзор бизнес-процессов банка и внутренних административных документов, которыми руководствуются сервисные инженеры при обслуживании офисного оборудования банка. На основе собранных данных была разработана модель бизнес-процессов в нотации IDEF0.

Во второй главе описывается проектная составляющая работы. В ней подробно описывается структура информационной системы, ее функции и свойства, а также описывается автоматизация управления запросами на техническое обслуживание офисного оборудования. Для обеспечения надлежащего внедрения новой системы были созданы схемы. По результатам исследований была разработана структура базы данных и определены основные ресурсы, необходимые для реализации проекта.

В третьей главе подробно описывается проектирование информационной системы, включая выбор архитектуры приложения, предварительные требования к системе и распределение ресурсов. Кроме того, в нем рассматриваются вопросы создания физической модели данных и функционирования системы, разработанной для контроля запросов на техническое обслуживание офисного оборудования банка.

Глава 1 Исследование деятельности банка ПАО «Росбанк»

1.1 Изучение и анализ структуры банка ПАО «Росбанк»

В 1993 году банк был основан как КН «Независимость», а в 1998 году был переименован как в АКБ «Росбанк».

В 2000 году стратегическим направлением развития банка стала работа с корпоративным сегментом рынка.

В 2005 году была успешно проведена консолидация бизнеса Росбанка и шести банков ОВК [2].

Банк осуществляет контроль как за наличными, так и за безналичными платежными операциями, привлекая средства вкладчиков через депозиты и выдавая кредиты для содействия росту бизнеса или удовлетворения личных потребностей. Во время взаимодействия с клиентами сотрудники банка часто используют различное офисное оборудование, чтобы предоставить клиентам дополнительную информацию о депозитных и кредитных операциях, часто в виде печатных материалов.

На данный момент в банке не автоматизирован процесс подачи заявки на обслуживание тех или иных технических устройств, заявка принимается сервисным отделом по телефону, сотрудники, подающие заявку, как правило не могут четко и понятно сформулировать проблему и суть поломки устройства. Таким образом специалисту сервисного отдела становится сложнее принять быстрые меры по устранению неполадки, и выезжать в отдел для исследования и устранения поломки, на что тратится очень много времени [3].

Для кросс-функционального взаимодействия между сотрудниками сервисной службы, а также пользователями оргтехники банка нужно

разработать наиболее удобную и эффективную систему для взаимодействия с технической поддержкой.

Система для взаимодействия с отделом технической поддержки должна отвечать следующим требованиям:

- предоставлять интуитивно понятные пользовательские интерфейсы для прохождения регламента сопровождения заявки;
- персистентное хранение всех данных, связанных с регламентом прохождения заявки на всех этапах выполнения;
- стабильный доступ к системе, разграниченный между отдельными участниками процесса;
- обеспечивать возможность взаимодействия между всеми участниками процесса;
- функциональная доступность для любых пользователей данной системы.

Пользователи для взаимодействия с технической поддержкой должны быть уверены, что оформленная ими заявка в полной мере содержит необходимую информацию и в неизменном виде она дойдет до персонала технической поддержки, после чего будет принято решение по устранению инцидента с последующим решением или ответом, либо возврат заявки на доработку.

Один из главных элементов системы для взаимодействия с технической поддержкой это – база данных. В базе хранятся все типы объектов, для которых в последующем будут применены действия, на основании которых служба технической поддержки будет выполнять поставленные задачи.

Рассмотрим более подробно работу банка изнутри. Структура банка изображена на рисунке 1.

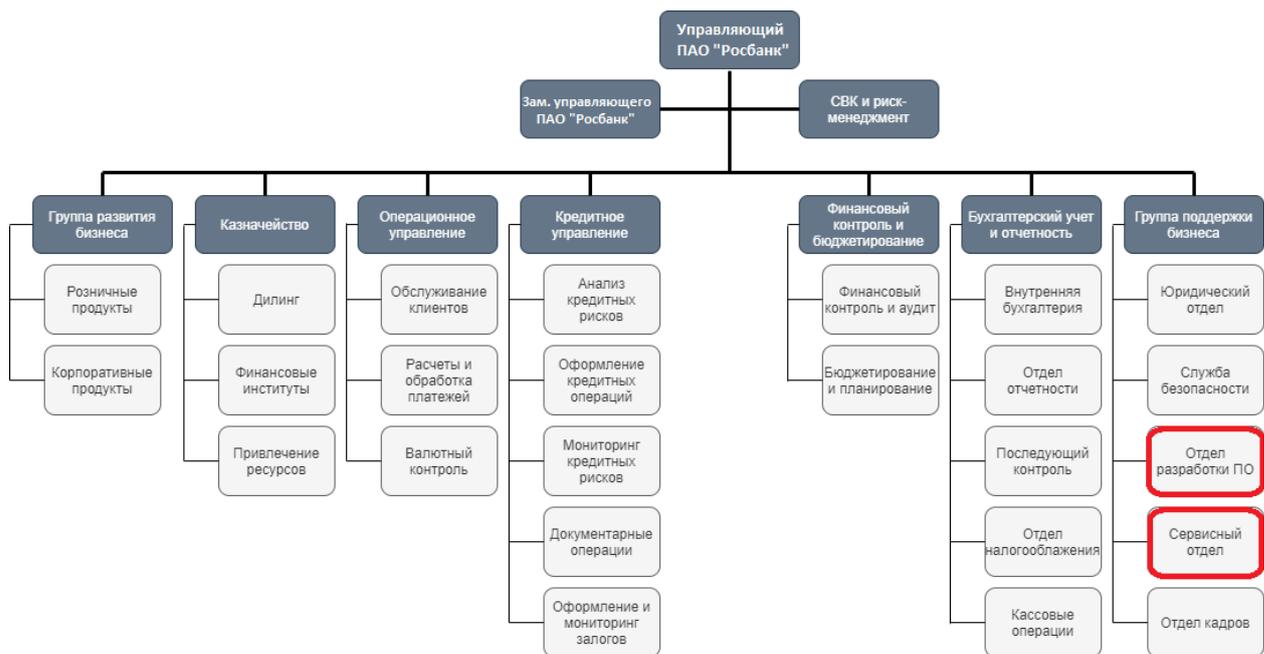


Рисунок 1 – Организационная структура ПАО «Росбанк»

Банк занимается регулированием платежного оборота в наличной и безналичной форме, привлечением денежных средств на депозиты и выдает деньги в виде кредитов.

Разработкой и внедрением программного обеспечения занимается отдел информационных технологий.

Пользователями системы будут все сотрудники банка кроме сотрудников внешних компаний, которые оказывают услуги клининга и охраны.

Основной целью практически любой банковской организации является привлечение средств на депозиты, обеспечивая тем самым достаточный уровень капитализации для поддержки своей предпринимательской деятельности. Такая ресурсная база позволяет банку предоставлять кредиты различным категориям клиентов и осуществлять инвестиционную деятельность с конечной целью максимизации прибыли для самой коммерческой организации [5].

Банк разрабатывает свою депозитную политику, ориентируясь на свои собственные потребности и цели, а также с учетом стратегического плана, который был предварительно продуман и просчитан. Этот документ определяет методы и стратегии, которые банк использует для привлечения вкладов и управления депозитами клиентов.

Право на привлечение денежных средств во вклады принадлежит как центральному офису банка, так и его филиалам. Однако для этого в каждом офисе должны быть обеспечены необходимые условия, включая соответствующий комплект технических средств и наличие подготовленной правовой базы. Важно также учитывать, что для клиентов, желающих обратиться в финансовую организацию, должны быть созданы все необходимые условия для удобства их обслуживания.

Еще одной важной задачей банка, занимающегося выдачей кредитов, является максимизация количества выданных кредитов и обеспечение их возврата. Для того чтобы заёмщик мог доказать свою надежность, ему необходимо предоставить определенный набор документов. Этот пакет документов служит для подтверждения наличия стабильного дохода, наличия имущества, которое может быть использовано в качестве залога, а также отсутствия задолженностей.

На основании вышеизложенного, чтобы все эти операции выполнялись быстро и успешно необходимо вовремя и эффективно обслуживать офисную технику, которая облегчает и ускоряет делопроизводство.

1.2 Описание и моделирование исследуемого процесса

Рассмотрим бизнес-процессы, которые происходят в банке.

Клиент подает заявку на кредит в онлайн-банке, указывая свои данные и прикрепляя необходимые документы. Кредитный специалист проверяет

данные; при несоответствии или наличии плохой кредитной истории следует отказ. При успешной проверке данные передаются кредитным инспекторам для проверки заемщика на способность выплатить кредит. При выполнении всех условий банк одобряет заявку и заключает с клиентом кредитный договор. Если данные прошли проверку, то они передаются на проверку кредитным инспекторам для того, чтобы удостовериться, что заемщик способен выплатить кредит.

Открытие электронного депозитного счета. Клиент выбирает на сайте или в своем личном кабинете какой именно вклад планирует открыть, затем знакомится с действующими условиями. Если все устраивает, он подает заявку на открытие счета, затем клиенту предлагают выполнить ряд операций:

- предлагают заполнить анкету;
- внести деньги на лицевой счет;
- подписать договор, предварительно внимательно изучив его условия.

Прием заявки на обслуживание оргтехники. Поступление заявок происходит чаще всего по телефону или в письменном виде в сервисный отдел. Для того, чтобы заявку приняли, сотруднику отдела, где была обнаружена неисправность того или иного технического устройства, необходимо составить заявку, где четко и внятно расписать все признаки неисправности, указать модель и инвентарный номер устройства.

После этого заявка передается в сервисный отдел, где ее фиксируют путем регистрации в журнале заявок.

Из всего вышеперечисленного можно с точностью сказать, что без своевременного обслуживания оргтехники банка, бизнес-процессы будут выполняться значительно медленнее.

Так как этот процесс является очень важным рассмотрим его более подробно и выявим все слабые места.

1.2.1 Обзор и выбор существующих инструментов для моделирования бизнес-процессов

В распоряжении пользователя имеется множество методов и инструментов для моделирования банковских процессов, включая и бизнес-процессы. В данной работе будут рассмотрены несколько часто используемых инструментов для моделирования бизнес-процессов в банковском секторе.

Предприятия, занимающиеся моделированием бизнес-процессов, используют множество популярных инструментов таких как:

- IDEF (Integration Definition for Function Modeling): Совокупность методов и языков, используемых для моделирования бизнес-процессов и систем, широко используемых в системном анализе и проектировании;
- UML (Unified Modeling Language): Хотя UML в первую очередь известен своими программными средствами моделирования, его применимость распространяется и на моделирование бизнес-процессов, особенно при описании взаимодействий между различными системами;
- диаграммы потоков данных (DFD): Этот инструмент предназначен для отображения потока данных в рамках процесса. DFD помогают понять движение информации по всей системе или организации;
- EPC (Event-driven Process Chain): Эти диаграммы отображают процессы, выделяя события, которые запускают различные действия и этапы в рамках процесса;
- BPMN (Business Process Model and Notation) - это еще один широко используемый язык для моделирования различных бизнес-процессов. Он позволяет создавать графические схемы любой сложности, упрощая моделирование этапов процесса, взаимосвязей потоков, данных и ролей;

- ARIS (Architecture of Integrated Information Systems) отличается от своих предшественников не только контролем бизнес-процессов, но и моделированием организационных структур. Она может похвастаться надежными инструментами для тщательного анализа и совершенствования процессов;
- Microsoft Visio предоставляет полный набор инструментов, предназначенных для создания множества диаграмм, в том числе для бизнес-процессов. Благодаря предварительно разработанным диаграммам Visio предлагает гармоничную основу для иллюстрации структурированных потоков данных, что делает его предпочтительным вариантом для моделирования процессов;
- IBM Blueworks Live - это интегрированная платформа, созданная для моделирования и автоматизации бизнес-процессов. Оснащенная набором инструментов, она облегчает создание, анализ и оптимизацию процессов, обеспечивая всестороннюю поддержку на протяжении всего жизненного цикла процесса.

В сценариях, связанных со сложными процессами, такими как производство или логистика, часто незаменимо специализированное программное обеспечение для моделирования, такое как AnyLogic, Simul8 или Arena. Эти инструменты позволяют пользователям создавать сложные модели и проводить численные эксперименты для уточнения процессов, оценки их эффективности и прогнозирования результатов.

Однако для небольших компаний или на начальном этапе моделирования процессов достаточно более доступных инструментов, таких как электронные таблицы Excel или Google Sheets. Они позволяют создавать простые диаграммы рабочего процесса или таблицы процессов, что позволяет визуализировать процессы и проводить анализ эффективности. Это особенно

полезно для компаний с ограниченными ресурсами или для тех, кто только начинает знакомиться с моделированием процессов.

1.2.2 Разработка модели бизнес-процесса «как есть»

Чтобы проанализировать слабые места в организационной структуре банка, давайте построим схему процессов "КАК ЕСТЬ", чтобы понять текущее состояние дел. Эта схема проиллюстрирует рабочий процесс, определит действия, которые могут замедлять процесс, и выделит области для улучшения за счет автоматизации в отделе обслуживания.

Для целей моделирования мы выбрали BPwin в качестве основного инструмента. Он облегчает реализацию моделей бизнес-процессов в соответствии с выбранными обозначениями, а также позволяет проводить глубокую декомпозицию различных бизнес-процессов.

На рисунке 2 представлена IDEF0 диаграмма процесса управления заявками на обслуживание оргтехники.

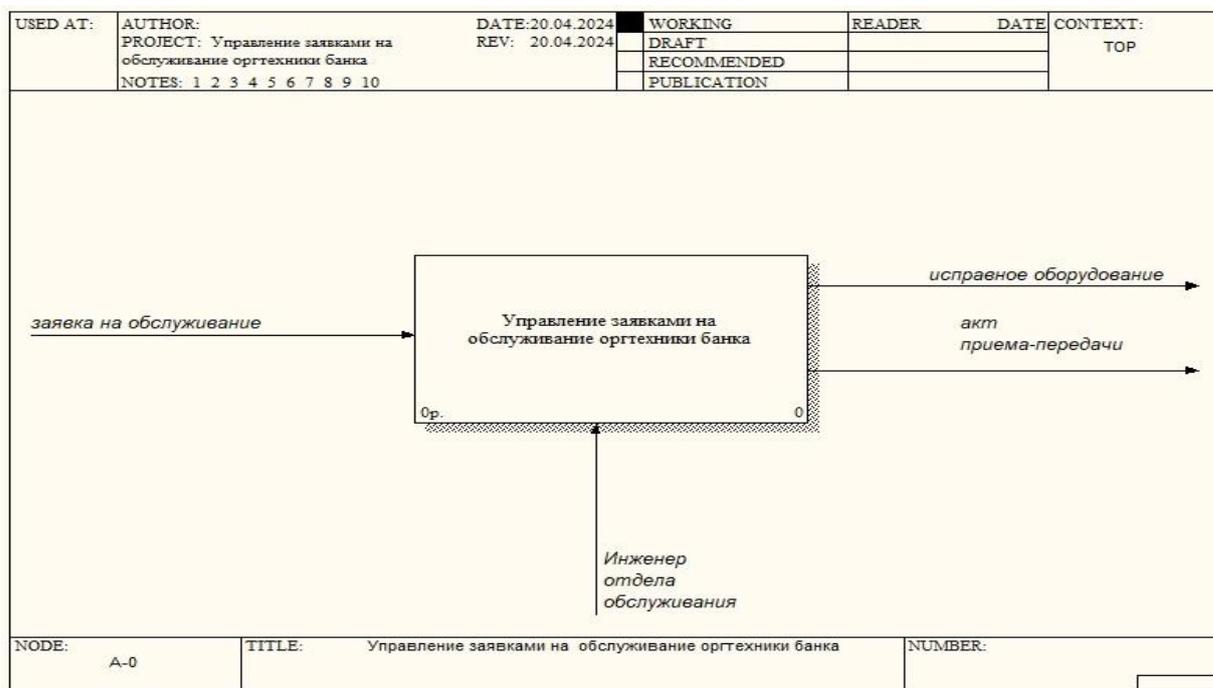


Рисунок 2 – IDEF0 диаграмма процесса управления заявками на обслуживание оргтехники

Для более подробного анализа представим процесс в декомпозиции для этого разобьём его на четыре блока:

- приемка заявки,
- обработка заявки,
- поиск причины неисправности,
- устранение неисправности,

Диаграмма декомпозиции представлена на рисунке 3.

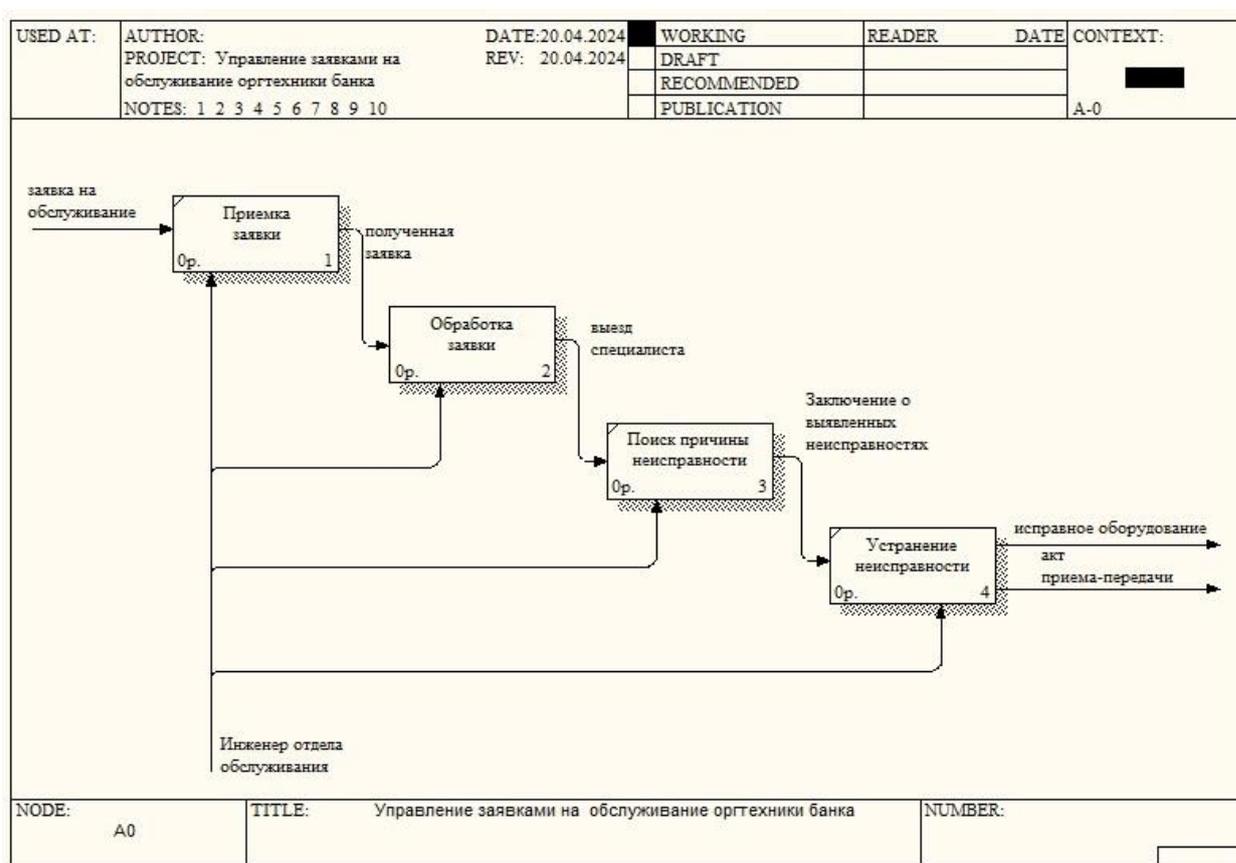


Рисунок 3 – Дальнейшая декомпозиция бизнес-процесса

Блок «Приемка заявки» имеет входную функцию в виде заявки от сотрудника какого-то отдела или подразделения, так же есть управляющие функции такие как корпоративные инструкции, пользователями процесса

будут инженеры отдела, осуществляющие ремонт оборудования. Выходной функцией из этого блока будет полученная заявка, которая послужит входной функцией во второй блок.

Блок «Обработка заявки» имеет входную функцию в виде полученной заявки, так же есть управляющие функции такие как корпоративные стандарты, пользователем процесса будет инженер отдела, непосредственно обрабатывающий этот процесс. Выходными функциями из этого блока будет выезд специалиста, который послужит входной функцией в третий блок.

Блок «Поиск причины неисправности» имеет входную функцию в виде выезда специалиста, так же есть управляющие функции такие как корпоративные инструкции, пользователем процесса будет инженер отдела. Выходной функцией из этого блока будет заключение выявленных неисправностей.

Блок «Устранение неисправности» имеет входную функцию в виде заключения выявленных неисправностей, так же есть управляющие функции такие как корпоративные инструкции и наличие расходных материалов, пользователем процесса будет инженер отдела. Выходной функцией из этого блока будут исправное оборудование и акт приема-передачи.

Такие процессы как прием заявок, выяснение причин неисправности и устранение неисправности не требует пристального внимания, рассмотрим более подробно блок «Обработка заявки». Опустимся на уровень ниже и представим декомпозицию блока, разбив его на процессы такие как:

- ввод данных в журнал,
- ожидание обработки,
- работа с заявкой,

На диаграмме рассмотрим процесс углубив внимание на то, как происходит регистрация заявок при поступлении, кто этим занимается, как в

дальнейшем меняется статус заявки и что влияет на то, чтобы заявка была обработана, все это показано на рисунке 4.

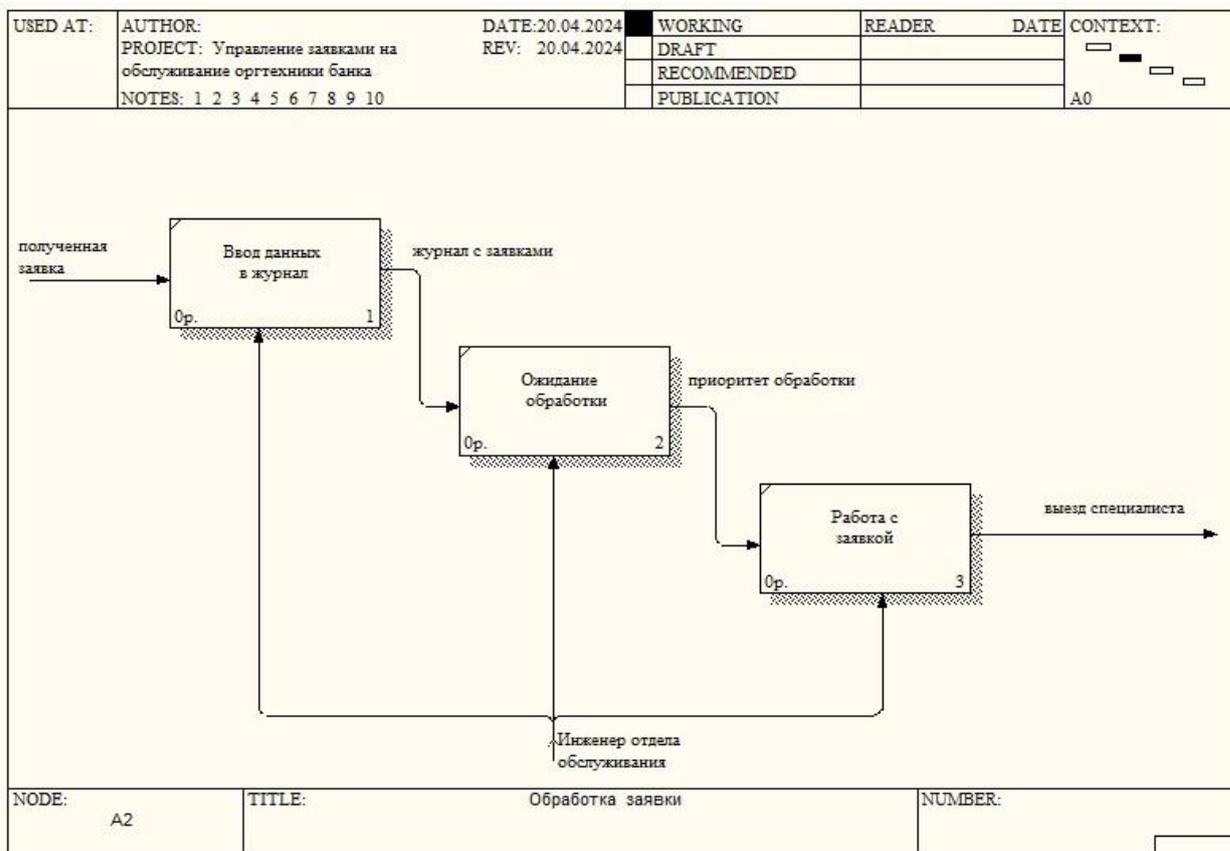


Рисунок 4 – Декомпозиция процесса «Обработка заявки»

Исходя из данных диаграммы, видно, что заявки поступают на бумажных носителях и фиксируются только в журнале, так же заявка может очень долго обрабатываться по причине не корректно составленных пунктов заявки и малыми сведениями о неисправном устройстве. Каждая заявка обрабатывается только после того, как специалист отдела обслуживания осуществит выезд на место и выяснит причины неисправности. Таким образом было принято решение автоматизировать процесс.

В связи с тем, что исправное техническое оборудование помогает быстро и эффективно осуществлять все бизнес-процессы банка, необходимо в

кратчайшие сроки обрабатывать заявки по устранению неисправностей оргтехники на рабочем месте сотрудника банка.

Таким образом необходимо автоматизировать процесс управления заявками на обслуживание оргтехники и хранить все данные в одном месте.

1.2.3 Обоснование необходимости автоматизированного варианта решения

Чтобы упростить процесс передачи оборудования на обслуживание в сервисный отдел, необходимо упростить выдачу и согласование заявок, поскольку в настоящее время это отнимает значительное рабочее время. При создании заявки сотрудник банка должен эффективно сообщить специалистам сервисного отдела о проблеме, связанной с неисправностью офисного оборудования.

Процесс обработки заявок на обслуживание включает несколько этапов.

Сотрудник банка создает заявку, где указывает все данные об оборудовании и причину отправки оборудования на обслуживание.

После получения заявки сотрудником отдела обслуживания, инженер уточняет информацию у сотрудника подавшего заявку все подробности, после чего регистрирует ее в журнале регистрации.

В зависимости от важности заявки инженер должен определить приоритетность заявки и сроки ее выполнения, после чего назначается исполнитель на заявку, который будет ее исполнять. Если для выяснения неисправности необходимо выехать специалисту на место установки оборудования то он выезжает и производит осмотр оргтехники на месте, на основе чего составляется заключение о выявленных неисправностях.

Заключение о выявленных неисправностях является основанием для ремонта оргтехники.

Документы, такие как акт приема-передачи оргтехники является подтверждением того, что оно было отремонтировано и проверена его работа, после чего ставятся подписи специалиста и сотрудника банка.

Все данные о поступившей оргтехнике на ремонт и выполнение обслуживания или ремонта должны сохраняться для дальнейшего анализа и заказа комплектующих.

1.3 Сравнительный анализ существующих аналогов информационных систем и их соответствие требованиям

Мы можем изучить несколько доступных на рынке готовых программных решений для автоматизации управления запросами на техническое обслуживание офисного оборудования, учитывая следующие критерии:

Программное решение должно быть стабильным и надежным, чтобы обеспечить бесперебойное функционирование системы управления заявками.

Механизмы защиты данных, такими как шифрование и управление доступом, чтобы предотвратить утечку конфиденциальной информации.

Интуитивно понятный и удобный пользовательский интерфейс облегчит работу с системой и повысит эффективность использования.

Решение должно предоставлять возможности по управлению заявками, включая создание, назначение исполнителей, отслеживание статусов и формирование отчетности.

Сравним три схожие системы: «Итилиум», «ИнфраМенеджер», «AstroSoft».

Итилиум - платформа на базе 1С с Web-интерфейсом. Автоматическая система обращений как по средствам электронной почты, так и с помощью WEB сайта. Обращения могут создаваться не только физическими лицами, но

и самим оборудованием. Поддерживается управление на уровне сервиса, контроль, учет затрат, управление настройками и изменениями. Есть возможность аренды виртуального размещения ресурсов.

ИнфраМенеджер – инженерно-техническая серверная платформа с возможностью автоматической обработки заявок, учета инженерно-технических ресурсов, управления настройками, мониторинга и отслеживания оборудования. Используется в варианте с веб-интерфейсом как для пользователей, так и для инженеров службы технической поддержки.

AstroSoft - AstroSoft HelpDesk классифицируется как система из разряда HelpDesk, также включает в себя быструю и эффективную техническую поддержку пользователей, имеет возможность сохранять историю обращений пользователей, а также проводить анализ входящих заявок и создавать прогнозы возникших нештатных ситуаций.

«Итилиум». Управление пользовательскими заявками:

- регистрация заявок, мониторинг сроков решения выполненных заявок;
- поддержка схем делегаций с правом переназначения (перераспределение ответственности, уведомления о распределении);
- управление созданными или имеющимися задачами;
- поддержка базы часто задаваемых вопросов по решению инцидентов.

В системе Итилиум создан удобный процесс для обращений пользователей услуг — интеграция с мессенджерами Telegram и Viber. Есть возможность зарегистрировать обращение, ввести в него корректировку, посмотреть историю сообщений в ленте, запросить статус обращений, оценить исполнителя за выполненное обращение и иные функции.

Обработка потенциальных нештатных ситуаций:

- регистрация и идентификация аварийных ситуаций;
- документирование списка "выявленных ошибок".

Управление конфигурациями и изменениями:

- ведение системного журнала изменений и конфигураций;
- содействие обмену информацией о проблемах, развертываниях и инцидентах;
- хранение документации, относящейся к инженерным и техническим активам (сконфигурированным элементам) и текущим изменениям.

Управление уровнем обслуживания:

- поддержка различных каталогов услуг;
- разработка Соглашения об уровне обслуживания (SLA) и управление им;
- создание отчетов об уровнях обслуживания (SLA reporting).

«Итилиум» значительно упрощает работу инженерно-технических специалистов, делает работу технического отдела прозрачной для бизнес-клиентов, позволяет эффективно оценивать качество инженерно-технических услуг, делает «прозрачной» работу службы технической поддержки для пользователей.

«ИнфраМенеджер».

Заявки регистрируются автоматически с любого канала, при этом сразу же рассчитывается регламентное время решения.

Возможности системы позволяют проводить такие действия как:

- создание, техническая поддержка, вывод из эксплуатации сервисов;
- разделение по зонам ответственности;
- разделение по правам доступа;
- автоматические процессы обработки задач;
- взаимные связи сервисов друг с другом и от элементов инфраструктуры;
- взаимосвязи с ресурсами;
- финансовая стоимость сервисов;

- настраиваемые поля для заявок в структуре сервисов.

Автоматизация процессов предоставляет возможность:

- осуществлять автоматическое предоставление заданий на работу;
- конфигурировать действия по истечению расчетного времени;
- конфигурировать систему оповещений по этапам, времени контроля, событиям;
- мониторинг и запуск процессов по сообщениям мониторинга;
- производить фильтрацию входящих сообщений электронной почты.

Система содержит такие новшества как:

- база ресурсов информации;
- метки у контента;
- поиск;
- соответствие с задачей;
- ссылки на группу документов в базах информации.

«AstroSoft».

Данная система позволяет выполнять сложные задачи путем обработки больших массивов данных поступающих как от пользователей, так и от администраторов. Все настройки системы сводятся к конфигурации параметров путем административных распределений между ними. Настройка системы довольно гибкая, что само по себе дает ряд преимуществ перед конкурентами. Вся информация распределяется по группам для удобной обработки, а также перераспределения между категориями.

AstroSoft обладает удивительно гибкой системной архитектурой, обеспечивающей плавную интеграцию с внешними приложениями. Используя алгоритмы, разработанные AstroSoft, она легко интегрировалась с Microsoft MOM 2005 и Microsoft SMS 2003. Кроме того, открытый интерфейс веб-сервисов и поддержка сторонних программных продуктов на уровне архитектуры упрощают установление пользовательских подключений.

Создание и выполнение запроса определяются различными атрибутами, связанными с самим запросом, такими как ключевые слова, тип услуги или категория запроса. Кроме того, атрибуты, относящиеся к пользователю, который инициировал и зарегистрировал запрос, влияют на его перемещение. Операторы и исполнители продвигают запрос путем внесения изменений, причем действия, предпринимаемые на каждом этапе, влияют на выполнение запроса по всему его маршруту.

Довольно мощный встроенный движок подсистемы формирования отчетной информации с легкостью производит анализ сообщений и возможных инцидентов, находит и распознает проблемные места корпоративной информационной системы. Так же возможна обратная связь путем оценки выполненных работ по обслуживанию техническим персоналом, с последующим ее доведением до инициатора запроса.

База в которой содержится вся информация по движению, обработке, внесенным изменениям, завершению и архивированию запросов и сообщений как пользователей, так и исполнителей, является неотъемлемой частью информационной системы, которая может послужить для повторного применения полезной информации. Для более удобного использования данной информации протоколы и журналы хранятся в формате HTML, что дает возможность использовать ее в различных вариантах и комбинациях с последующим редактированием.

Результаты сравнения функциональных возможностей систем приведены в таблице 1.

Таблица 1 – Результаты сравнения программ

Функции	Итилиум	ИнфраМенеджер	AstroSoft
Понятный интерфейс пользователя	+	-	-
Мобильная версия системы	-	-	+
СМС оповещения	+	+	+
Интеграция с другими сервисами	+	+	+
Понятный портал самообслуживание	-	+	+

Исходя из всего вышесказанного, можно сделать выбор, что существует множество продуктов на рынке, способных решить данную задачу, но все они, являясь коммерческими, требуют дополнительных ежемесячных трат от компании. При том, что функционал аналогов, значительно выходит за рамки задачи. Целесообразно будет провести разработку информационной системы для управления заявками на обслуживание оргтехники в отделе IT-разработок.

1.4 Реинжиниринг модели процесса «как есть»

В рамках анализа существующего бизнес-процесса, описанного в главе 1.2.2, можно провести оптимизацию, чтобы определить границы, в пределах которых должна функционировать разработанная информационная система. Эта оптимизация направлена на устранение ненужных этапов обработки заявок и оптимизацию информационных потоков, необходимых для эффективного завершения процесса [6].

Определим место разрабатываемой системы в рамках нашего бизнес-процесса. На рисунке 6 представлена система, как механизм уже существующего бизнес-процесса.

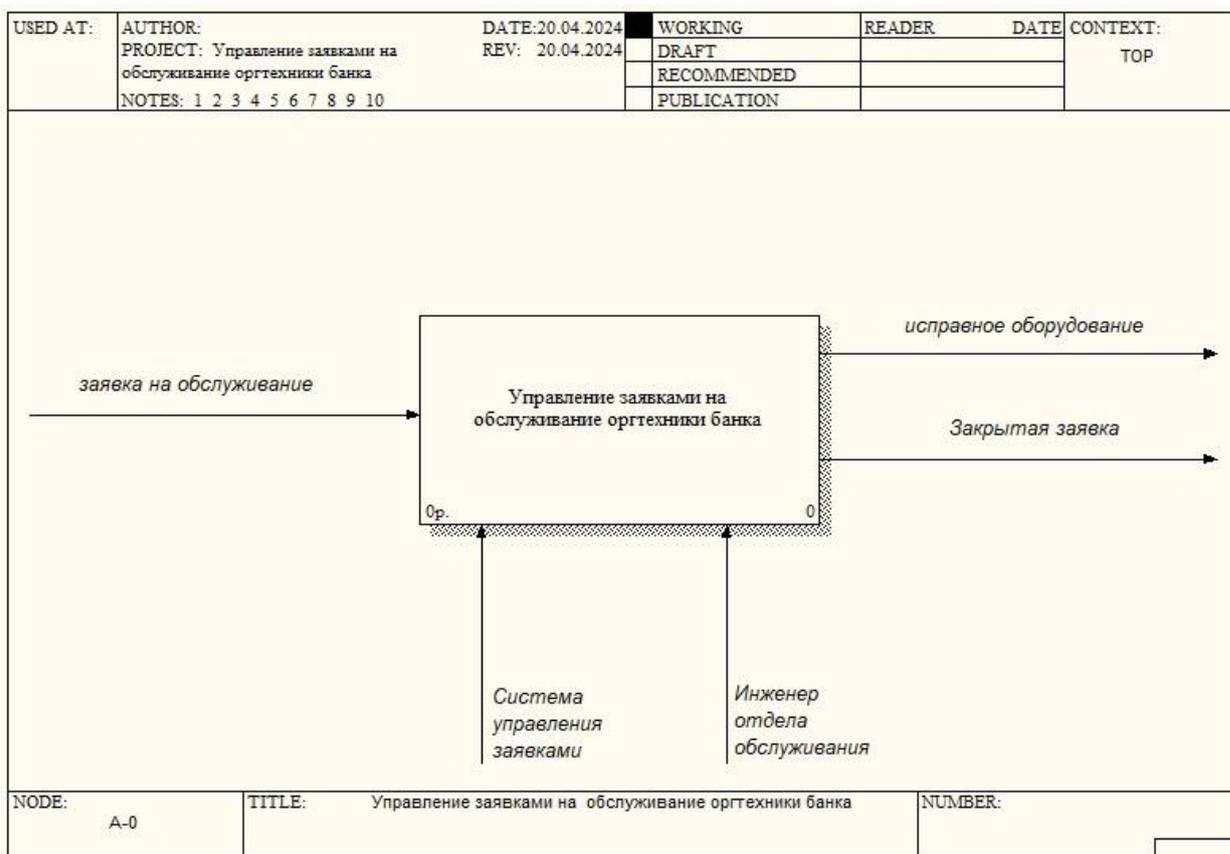


Рисунок 6 – IDEF0 диаграмма процесса управления заявками на обслуживание оргтехники

Входные и выходные данные, в свою очередь, представляют из себя саму заявку, в произвольной форме, которая, при дальнейшей декомпозиции, будет внесена в систему ответственным за это лицом. В качестве управляющих элементов выступают всё те же, корпоративные инструкции и информация о наличии расходных материалов, которая могла бы быть рассмотрена, как входные данные (вместе с заявкой), но ввиду того, что разрабатываемая система, должна быть в рамках отдела обслуживания, эта информация в бизнес-процесс поступает опосредовано со склада.

На рисунке 7 дальнейшая декомпозиция иллюстрирует размещение разработанной информационной системы непосредственно в отделе обслуживания. Процесс управления заявками включает в себя несколько ключевых элементов:

- принятие заявки включает в себя ввод исходных данных о заявке в информационную систему;
- обработка заявки, это когда технический специалист оценивает сложность выполнения, формализует предполагаемые технические неисправности оборудования и направляет ее конкретному специалисту по оборудованию.
- расследование причин неисправности, это когда специалист по данному типу оборудования проводит расследование причины неисправности в рамках ремонтных работ. Это включает в себя определение конкретной причины неисправности и оценку стоимости ремонта с возможностью отмены заявки, если ремонт будет сочтен ненужным.
- процесс ремонта необходим, если неисправность может быть устранена и ремонт является целесообразным, специалист приступает к ремонту. При необходимости он принимает решение о приобретении дополнительных компонентов, необходимых для ремонта, и соответствующим образом обновляет статус заявки.

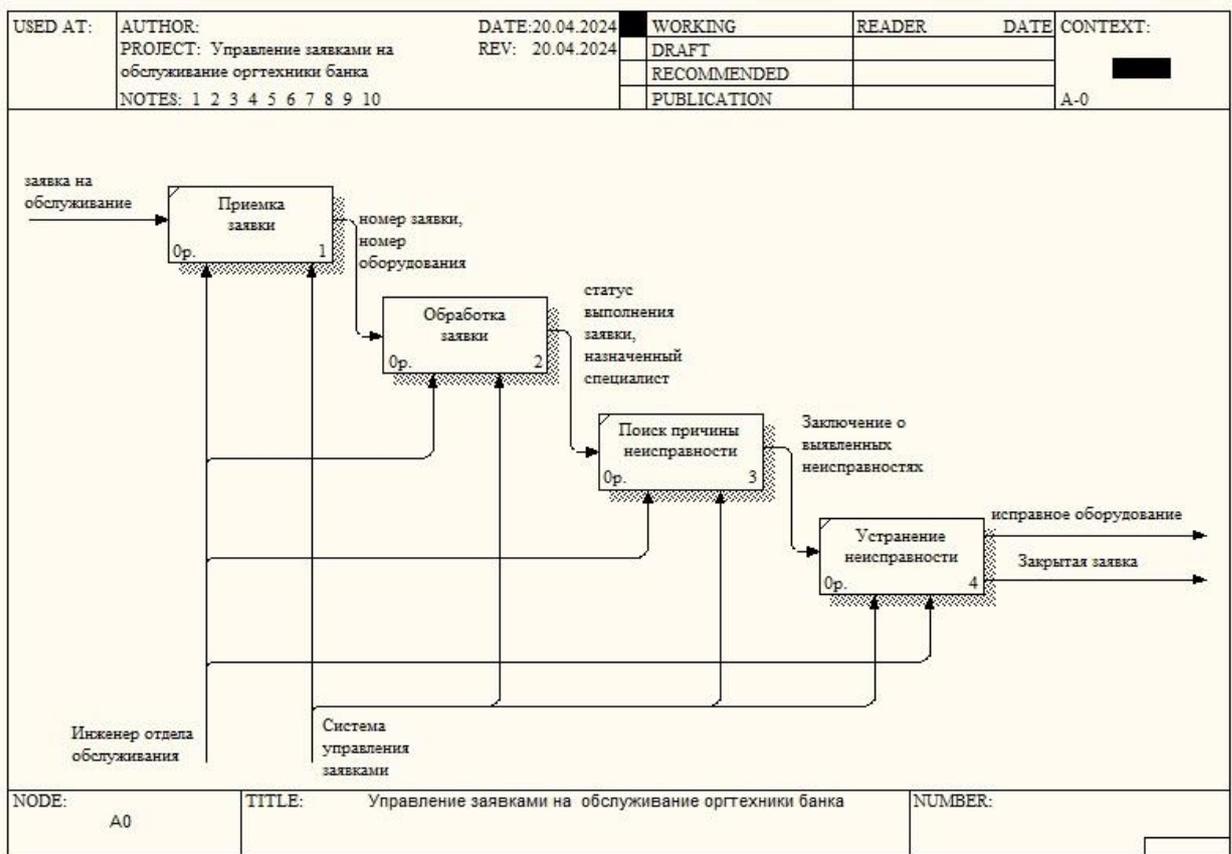


Рисунок 7 – Декомпозиция процесса, модель «как должно быть»

Все сотрудники сервисной службы могут работать с системой под своими учетными записями, в зависимости от своих функциональных обязанностей работают с заявкой согласно внутренних распорядительных документов банка.

1.5 Постановка задачи на разработку автоматизированной системы управления заявками

Проектирование автоматизированной информационной системы важно для эффективной работы в любой сфере деятельности предприятия. Ошибки на стадии проектирования, являются наиболее критичными в рамках дальнейшего её развития и требуют наибольшего количества ресурсов и затрат

на их устранение, потому, для реального использования системы в рамках предприятия, может потребовать дополнительных усилий, но в рамках данной работы, мы рассматриваем прототип системы [6].

Основные требования для разрабатываемой системы:

- предоставление простых и понятных пользовательских интерфейсов для отображения, внесения и изменения данных о заявке в систему;
- механизм управления ключевыми статусами заявки и навигации между ними;
- доступ к журналу событий и истории движения заявки;
- база данных оборудования для обслуживания и сотрудников сервисного отдела.

Функциональные требования к системе:

- ввод и хранение информации о заявках;
- ввод и хранение информации о сотрудниках;
- ввод и хранение информации об оргтехнике;
- обновление статуса заявки;
- поиск необходимой информации.

Нефункциональные требования к системе:

- дружелюбный и понятный интерфейс;
- отклик системы в течении заданного короткого промежутка времени, согласно техническим требованиям и в пределах допустимого по UX (User experience) соображениям требованиям задержки;
- интерфейс системы должен быть на целевом по техническим требованиям языке, а именно - русском языке;
- дизайн пользовательского интерфейса должен быть выполнен согласно техническим требованиям и лучшим практикам UI (User interface) и в спокойных тонах;

– обновление системы должно осуществляться не реже одного раза в год.

В процессе проведённого нами исследования и освоения теоретического материала, нами был проведен сравнительный обзор аспектов и основных принципов, относящихся к теме исследования. Вся проделанная нами работа, помогла нам в принятии решения для конкретных областей, которые еще не были затронуты или некоторые из них требуют дополнительного анализа и изучения.

Исследование модели существующего в рамках ПАО «Росбанк» бизнес-процесса, связанного с учетом заявок на обслуживание, позволило обнаружить недостатки, включая использование бумажных носителей для подачи заявок, что создавало риск потери данных и требовало много времени. В связи с тем, что данная технология работы с бумажными носителями теряет свою актуальность по мере появления цифровых технологий, автоматизация процесса путем разработки и интеграции, разрабатываемой нами информационной системы, в существующей предметной области является необходимостью для оптимизации и экономии времени, как рабочего персонала, так и пользователей, являющихся сотрудниками всевозможных отделов банка, от которых поступают заявки на обслуживание всевозможной техники банка, что является крайне необходимым для осуществления банком всех своих основных функций.

Глава 2 Логическое проектирование системы для управления заявками

2.1 Выбор технологий проектирования

Разработка логического системного проекта предполагает разработку понятных моделей бизнес-процессов с различных точек зрения, уделяя особое внимание таким важным аспектам, как потоки данных, взаимосвязанная структура систем и подсистем, а также их взаимоотношения с участниками.

Логическое моделирование описывает основные компоненты приложения, их взаимосвязи, операционную логику, механизмы обработки данных, бизнес-правила и взаимодействие с пользователями. Цель состоит в создании абстрактного представления, понятного всем заинтересованным сторонам проекта, включая разработчиков, дизайнеров, менеджеров и заказчиков. Существуют различные инструменты для формализации каждого аспекта информационной системы на различных этапах ее развития.

К числу наиболее известных инструментов проектирования относятся:

Унифицированный язык моделирования (UML): «Широко используемый программистами при разработке программного обеспечения, UML позволяет создавать разнообразные диаграммы, включая диаграммы классов, варианты использования и другие, разъясняющие логику программного обеспечения».

Диаграммы взаимосвязей сущностей: «Используемые для отображения уровня данных приложения, диаграммы взаимосвязей сущностей помогают определить взаимосвязи между ключевыми сущностями, тем самым помогая в проектировании структуры базы данных».

Для логического моделирования системы, направленного на автоматизацию процесса управления запросами на техническое обслуживание

офисного оборудования в банке, подходит язык моделирования UML. Он обеспечивает четкое представление логики приложения с помощью различных диаграмм в рамках этой нотации, включая:

- диаграмма вариантов использования, иллюстрирует функциональные требования системы, отображая взаимодействие между пользователями (участниками) и самой системой.
- диаграмма потоков данных, показывает поток данных в системе, показывая, как информация перемещается между различными компонентами и процессами.
- диаграмма классов, представляет структуру системы путем описания классов, их атрибутов, методов и взаимосвязей, обеспечивая схему реализации на языках программирования высокого уровня.

Для визуализации уровня данных мы будем использовать диаграмму "сущность-взаимосвязь", которая помогает изобразить взаимосвязи между сущностями в структуре базы данных системы [4].

2.2 Логическая модель приложения

Моделировать и рассматривать процесс будем с точки зрения сотрудника отдела, так как процесс принятия и обработки заявок происходит именно в отделе, где занимаются обслуживанием оргтехники.

Разработаем диаграммы процесса в программе BPWin 7.0.

Составим блок-схему процесса управления заявками на обслуживание оргтехники представленную на рисунке 8.

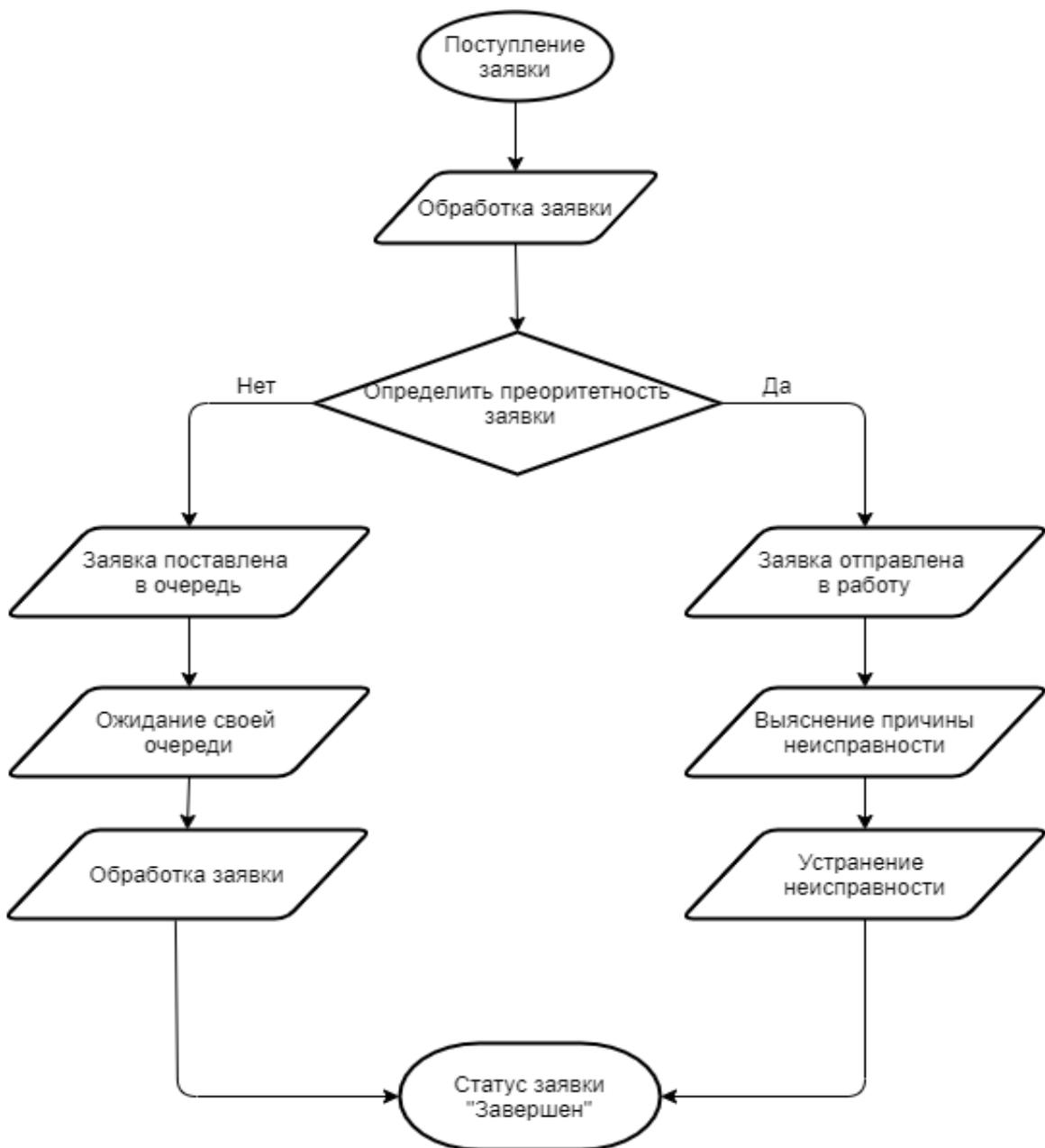


Рисунок 8 – Блок-схема процесса «Управление заявками»

Рассмотрим еще один процесс «Устранить неисправности», этот процесс представим в нотации DFD, выделим процессы, хранилища данных и взаимосвязи между процессами, рисунок 9.

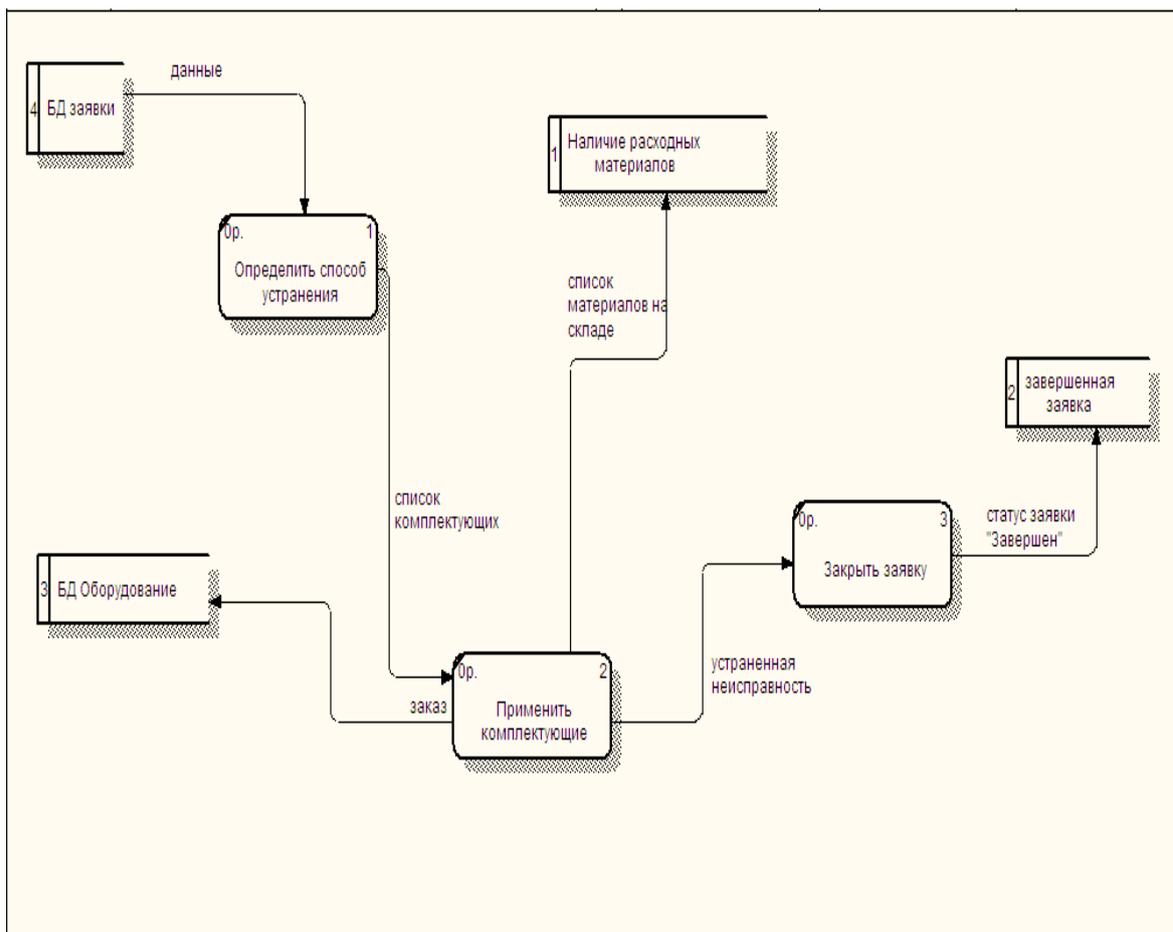


Рисунок 9 – Диаграмма DFD процесса «Устранить неисправности»

На диаграмме показано взаимодействие процессов «Определить способ устранения», «Применить комплектующие» и «Закрыть заявку». Взаимодействие происходит с хранилищами данных БД оборудование, БД заявки, Наличие расходных материалов, Завершенные заявки.

После автоматизации процесса добавилась база данных, где будет храниться вся информация, так же можно расширить функционал и создать интерфейс для пользователя и сотрудника отдела, которые будут обращаться за данными к одной базе данных.

Для проведения логического моделирования системы, предназначенной для автоматизации управления запросами на обслуживание офисного оборудования в банке, в качестве приемлемого варианта предлагается язык

моделирования UML. С помощью ряда диаграмм UML может детально описать логику системы, обеспечивая всестороннее представление [15].

На рисунке 10 представлена диаграмма вариантов использования разрабатываемой системы для разрабатываемой информационной системы.

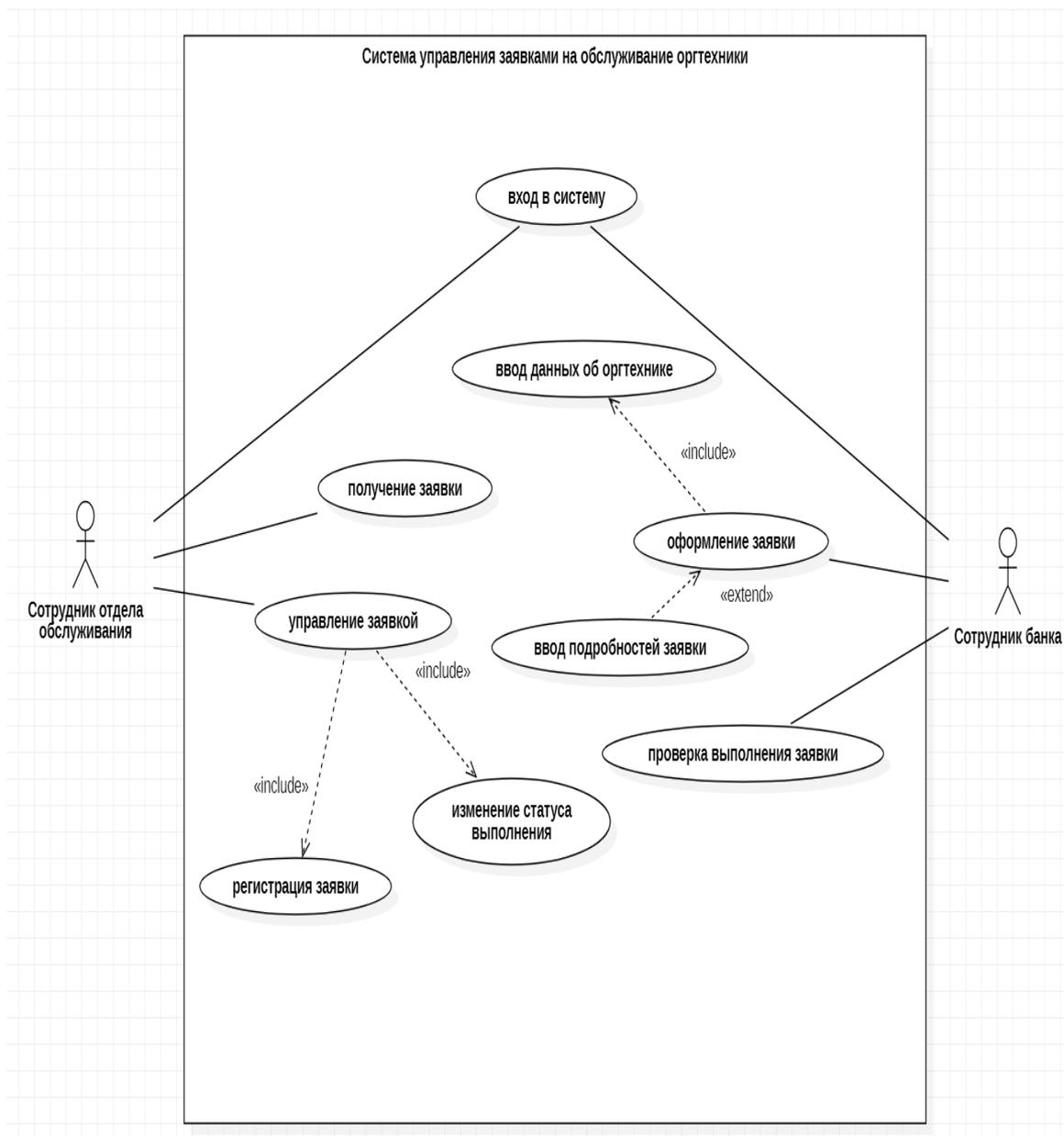


Рисунок 10 – Диаграмма вариантов использования

Построим диаграмму последовательности процесса получения и обработки заявки на обслуживание оргтехники. На рисунке 11 изображена диаграмма, на которой отражается весь процесс по этапам.

На диаграмме показано три объекта, это сотрудник банка, инженер отдела обслуживания и система.

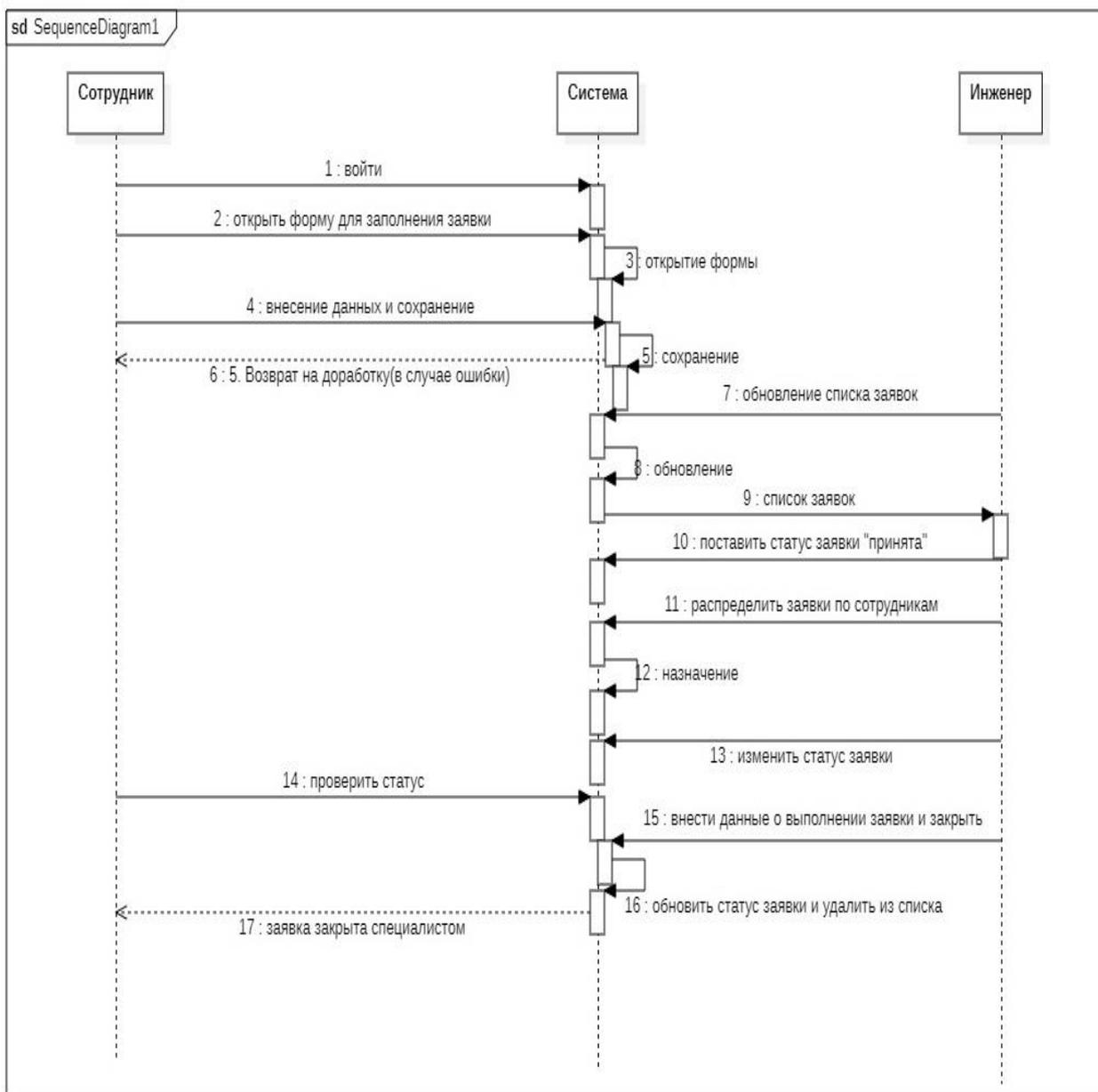


Рисунок 11 – Диаграмма последовательности процесса приема и обработки заявок в отделе обслуживания оргтехники

На схеме показан процесс, при котором сотрудник банка создает заявку на обслуживание офисного оборудования, вводит данные об офисном оборудовании в систему и записывает сведения о поломке или проблеме. Необходимые данные уже имеются в базе данных и доступны сотруднику при заполнении стандартной формы заявки.

Когда сотрудник банка вводит и выбирает данные из доступных списков, у него есть возможность сохранить заявку с помощью системы. Эта сохраненная заявка будет отображаться как "новая заявка" для обработки сотрудниками отдела обслуживания.

Для того чтобы осуществлять процесс обслуживания оргтехники не нужны специальные документ или соблюдение федеральных законов.

Обслуживание оргтехники в банке производится на основании должностных обязанностей инженеров отдела обслуживания и заявки, полученной от сотрудника банка.

Входящими документами в данном процессе нужно считать заявку на обслуживание, которая должна содержать информацию об оборудовании, которое нуждается в ремонте. Под информацией, подразумевается инвентарный номер оборудования, название, модель и причина, по которой данная заявка создается.

На основании полученной заявки сотрудники отдела обслуживания должны проверить оргтехнику и выяснить причину неисправности.

Процесс закрытия заявки будет считаться завершенным, когда все действия по ремонту или техническому обслуживанию оргтехники будут выполнены в полном объеме.

Процесс закрытия заявки включает в себя несколько этапов:

- сервисный отдел получает заявку на техническое обслуживание офисного оборудования;
- проводится предварительная проверка офисного оборудования

для оценки выявленных проблем;

- составление заключения о выявленных неисправностях: На основании предварительного осмотра составляется заключение о выявленных неисправностях или сбоях в работе;
- для устранения выявленных неисправностей выполняются необходимые ремонтные работы;
- после завершения ремонта и приведения оборудования в рабочее состояние заявка официально закрывается.

Весь процесс технического обслуживания офисной техники находится под пристальным контролем руководителя отдела, что гарантирует своевременный ремонт офисной техники и обеспечивает бесперебойную работу подразделений банка.

Рассмотрим методы моделирования данных, которые пользуются наибольшей популярностью.

"ER-моделирование": Этот тип моделирования эффективно иллюстрирует объекты в модели данных, их взаимосвязи и названия связей, которые помогают понять логику модели.

"Логическая модель данных": Эта модель фокусируется на визуальном представлении построения модели структуры данных. Это не зависит от выбранного языка программирования или среды разработки разрабатываемого продукта.

Давайте определим сущности, необходимые для формулирования концептуальной и логической модели, которая будет охватывать данные информационной системы.

Для разработки системы, нацеленной на автоматизацию управления приложениями в отделе технического обслуживания офисного оборудования, необходимы несколько объектов. В этих объектах будет храниться информация, относящаяся к сотрудникам, отделам и офисному оборудованию

[17]. Кроме того, крайне важно поддерживать данные о статусе приложений и самих приложениях.

2.3 Информационное обеспечение системы

2.3.1 Характеристика нормативно-справочной, входной и оперативной информации

Для того чтобы осуществлять процесс обслуживания оргтехники не нужны специальные документ или соблюдение федеральных законов [9].

Обслуживание оргтехники в банке производится на основании должностных обязанностей инженеров отдела обслуживания и заявки, полученной от сотрудника банка.

Входящими документами в данном процессе нужно считать заявку на обслуживание, которая должна содержать информацию об оборудовании, которое нуждается в ремонте. Под информацией подразумевается инвентарный номер оборудования, название, модель и причина, по которой данная заявка создается.

На основании полученной заявки сотрудники отдела обслуживания должны проверить оргтехнику и выяснить причину неисправности [10].

2.3.2 Характеристика выходной информации

Процесс закрытия заявки будет считаться завершенным, когда все действия по ремонту или техническому обслуживанию оргтехники будут выполнены в полном объеме.

Весь жизненный цикл заявки и ее закрытие разбивается на несколько этапов:

- получение новой заявки от пользователя системы;
- внешний осмотр поступившей техники на возможные неисправности;
- формирование решения о выявленных неисправностях;

- обслуживание или устранение неисправности;
- отметка о закрытии заявки.

Все работы по выполнению технического обслуживания или устранения неисправностей оргтехники полностью возлагается на руководителя сервисного отдела банка, чтобы обеспечить своевременный ремонт оргтехники и обеспечить бесперебойную работу отделов банка.

2.4 Проектирование базы данных

2.4.1 Выбор технологии проектирования базы данных

Существует множество решений для реализации и проектирования. Давайте воспользуемся простой диаграммой "Сущность-взаимосвязь", чтобы проиллюстрировать основные сущности и их взаимосвязи.

«ER-моделирование подходит для проектирования базы данных, так как с помощью моделирования можно показать сущности и описать их взаимосвязи в базе данных. ER-диаграммы помогают понять структуру данных и их взаимосвязи» [11].

«Логическая модель данных является визуальным графическим представлением структур данных, их атрибутов и связей. Логическая модель представляет данные таким образом, чтобы они легко воспринимались бизнес-пользователями. Проектирование логической модели должно быть свободно от требований платформы и языка реализации или способа дальнейшего использования данных» [23].

«При разработке используются требования к данным и результаты анализа для формирования логической модели данных. Логическую модель приводят к третьей нормальной форме, и проверяют ее на соответствие модели процессов» [12].

2.4.2 Разработка концептуальной модели данных

В процессе разработки автоматизированной системы управления приложениями в отделе обслуживания офисной техники банка потребуется некоторая информация. Эта информация должна содержать данные о следующих объектах: сотрудники, отделы, офисное оборудование, заявки [13].

Построим ER-модель данных, рисунок 12.

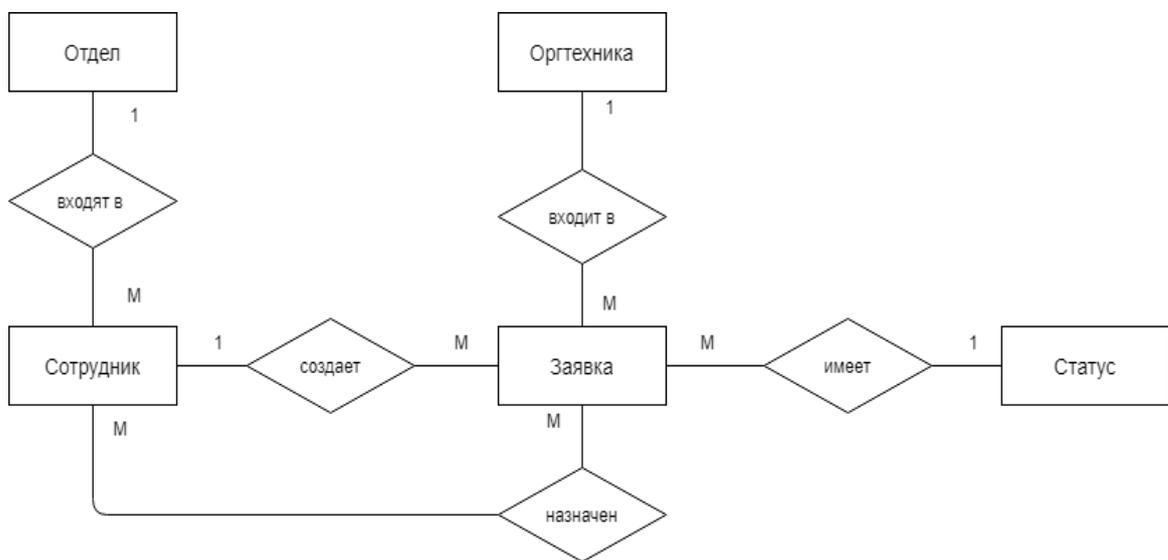


Рисунок 12 – модель сущность-связь

Давайте определим N соединения и представим их как слабый элемент на диаграмме ER.

Из слабых сущностей будет представлен сотрудник и заявка.

Объект заявка имеет бинарную связь со следующими объектами: сотрудник, офисное оборудование и сотрудник-заявка.

Используя специализированное программное обеспечение для моделирования моделей баз данных ER_WIN, можно сгенерировать следующую модель данных ER, показанную на рисунке 13.

Объект Статус использует суррогатный кодовый ключ статус, что позволяет использовать идентичные названия статусов, которые также служат ключевыми атрибутами. В объекте отсутствуют внешние ключи.

Сущность отдел использует суррогатный ключ, представленный кодом отдел, который служит атрибутом первичного ключа. В сущности, отдел отсутствуют внешние ключи.

Объект «Оргтехника» использует суррогатный ключ, такой как инвентарный номер, который служит атрибутом первичного ключа. Поскольку возможно повторение одних и тех же моделей офисного оборудования, этот ключ является ключевым. Внешние ключи не включены в объект «Оргтехника».

Объекту заявка присваивается суррогатный ключ, обозначаемый как номер заявки, который служит ключевым атрибутом из-за потенциального повторения заявки. Кроме того, в объект включены внешние ключи, включая код статуса, код сотрудника и номер оргтехника, что позволяет осуществлять связь со связанными объектами, такими как статус, сотрудник и офисное оборудование.

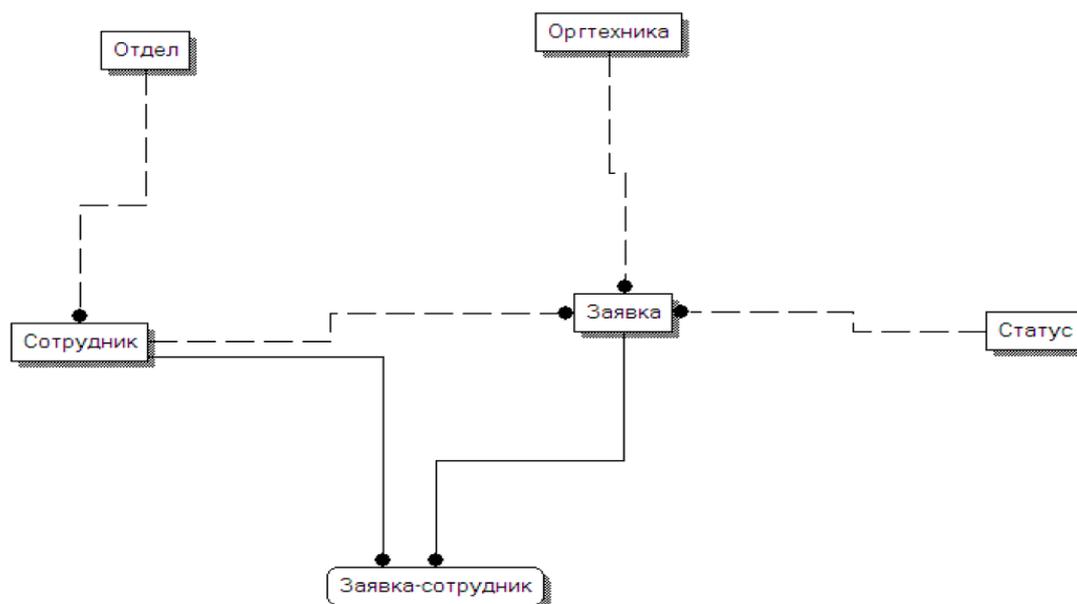


Рисунок 13 – ER-модель

ER-модель данных отображает исключительно только те сущности которые будут использоваться для хранения информации. Теперь необходимо для каждой из сущностей определить первичные ключи.

2.4.3 Обоснование вида логической модели

Рассмотрим спецификации соединений сущностей.

Сотрудник может не создавать заявку на обслуживание оргтехники.

Оргтехника может не входить в заявку на обслуживание, если не будет ломаться.

Сотрудник может и не назначаться на выполнение заявок по обслуживанию оргтехники, если его должность это не предусматривает.

На рисунке 14 показаны сущности и выбранные первичные ключи для них.

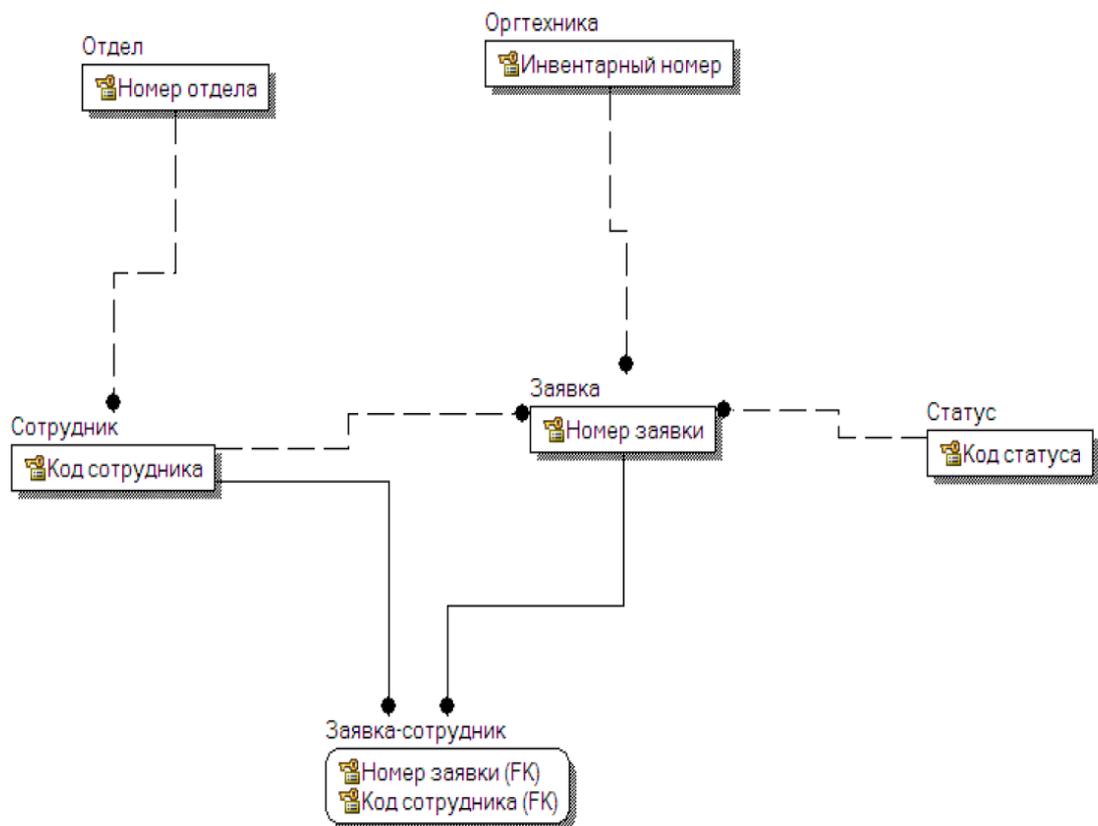


Рисунок 14 – Сущности и выбранные первичные ключи

В процессе исследования наличие новых атрибутов выявлено не было. Разработаем логическую модель данных, создадим поля для каждой сущности, а также определим их типы [13].

2.4.4 Разработка логической модели данных

В таблицах, которые должны содержать данные, должны быть поля для заполнения, тип данных и ограничения по количеству вводимых символов. Разработанная форма представлена в таблице 2, где видна вся необходимая структура.

Таблица 2 – Атрибуты и типы данных

Название	Описание	Тип данных	Название сущности	Ограничение
название	название отдела	текст	отдел	18 знаков
модель	модель	текст	оргтехника	18 знаков
серийный номер	серийный номер	текст	оргтехника	25 знаков
название	название	текст	оргтехника	18 знаков
фамилия	фамилия	текст	сотрудник	18 знаков
имя	имя	текст	сотрудник	18 знаков
отчество	отчество	текст	сотрудник	18 знаков
логин	логин	текст	сотрудник	18 знаков
пароль	пароль	текст	сотрудник	18 знаков
номер отдела	номер отдела	число	сотрудник	4 знака
название	название	текст	заявка	18 знаков
инвентарный номер	инвентарный номер	число	заявка	4 знака
сотрудник	сотрудник	число	заявка	4 знака
статус	статус	число	заявка	4 знака
дата создания	дата создания	дата	заявка	дата
название	название	текст	статус	18 знаков
дата начала	дата начала	дата	сотрудник-заявка	дата
дата завершения	дата завершения	дата	сотрудник-заявка	дата

После определения типов данных, а также установки ограничений, можно выстроить логическую модель представленных данных, изображенных на рисунке 15.

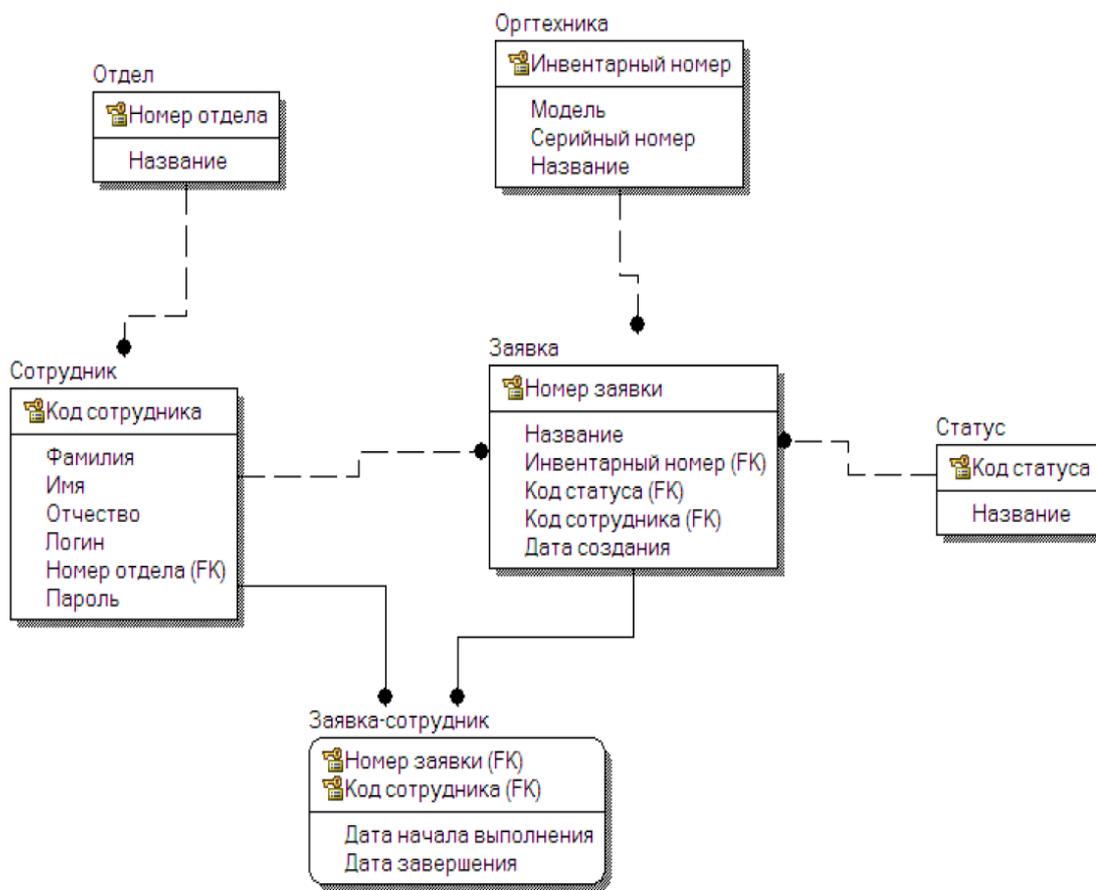


Рисунок 15 – Логическая модель данных

На основании логической модели данных теперь можно перейти к созданию физической модели. Требования к аппаратно-программному комплексу играют довольно значимую роль, так как производительность системы, скорость обработки данных, поиск нужной информации являются главными факторами для создания и реализации системы, а также последующей ее интеграции в рабочий процесс организации.

2.5 Требования к аппаратно-программному обеспечению

Чтобы новая система обеспечивала надежность и стабильность на протяжении всего срока службы информационного технического центра, оборудование должно соответствовать современным стандартам и техническим средствам. Серверный компьютер организации должен обладать высокой вычислительной производительностью, достаточным объемом оперативной памяти, надежным периферийным оборудованием с возможностью использования масштабируемой системой съемных носителей [7].

На серверной стороне должны размещаться, как приложение являющегося системой управления базами данных (СУБД), так и сама база данных в плане файлов [7]. Кроме того, в случае выхода из строя одного или нескольких жестких дисков на них должна быть предусмотрена возможность горячей замены без потери данных, а также системы резервного хранения данных, таких как RAID и соответствующих технологий резервного хранения и копирования физических данных устройства.

Система должна гарантировать быстрое и надежное взаимодействие между пользователями и между ее компонентами [14].

Операционная система выбирается с учетом ее доступности, поддержке, открытости, а также учитываются особенности архитектуры создаваемой системы. Наиболее популярными операционными системами в части физической серверной части, где располагается СУБД и база данных предпочтительна LINUX подобная платформа (Операционная система в широком смысле).

На основе разработанных моделей данных для эффективного хранения информации следует использовать системы управления реляционными базами

данных (Системами управления базами данных), такие как MySQL или PostgreSQL.

В процессе разработки программного обеспечения рекомендуется выбирать наиболее популярный язык программирования. Это обеспечивает гибкость при внесении возможных изменений в программный код в будущем и плавную интеграцию дополнительных функциональных возможностей в работу приложения. Популярным языкам программирования, таким как C#, Python, Java или JavaScript, часто отдают предпочтение из-за их широкого распространения и надежной поддержки сообществом разработчиков и программистов, что упрощает разработку, обслуживание и масштабируемость приложения. Наличие фреймворков и библиотек для различных областей разработки, также является огромным плюсом, для развития проекта и добавления нового функционала в будущем, для развития проекта и реализации новых требований [23].

Что касается интерфейса системы, то он должен быть в спокойных тонах, интуитивно понятен, логически построена система навигации по меню, а также своей информативностью и удобством в работе, с соблюдением всех наиболее общепринятых практик в рамках разработки пользовательских интерфейсов и опыта пользователей.

Таким образом, нами было проведено проектирование информационной системы на уровне данных. Были найдены в процессе исследования и описаны связи между основными сущностями, типами данных отдельных полей, сущностями и типами связей между ними.

Глава 3 Физическое проектирование и программная реализация системы автоматизации управления заявками

Выполнив логическое проектирование разрабатываемой информационной системы, мы можем приступить к её физической реализации, но для этого, необходимо выбрать наиболее подходящий, с нашей точки зрения, стек технологий.

В предыдущей главе, мы определились со структурой данных и сделали выбор в пользу реляционных баз данных. Но, помимо этого, необходимо выбрать набор технологий, который мог бы позволить нам, осуществить взаимодействие с реляционной БД, реализовать собственную бизнес логику и предоставить пользовательский интерфейс для взаимодействия с системой [8].

3.1 Выбор архитектуры информационной системы

Под архитектурой информационных систем подразумевают организацию элементов информационной системы и взаимосвязей между ними.

Архитектура приложения, состоит из нескольких логических слоёв:

- архитектура на уровне бизнес-логики;
- архитектура данных;
- программная архитектура;
- техническая архитектура.

Техническая архитектура – это архитектура аппаратных средств, т.е. непосредственно физические программных устройств.

Программная архитектура – это архитектура программных средств, которые могут быть распределены в рамках одного или нескольких физических устройств.

Архитектура данных – это архитектура средств управления данными и непосредственно хранилищ этих данных в рамках информационной системы.

Архитектура бизнес-логики – это архитектура, отражённая в рамках языка программирования, реализующего основную бизнес-логику информационной системы.

В зависимости от распределения инфраструктуры информационной системы между физическими устройствами, выделяют следующие типы архитектур информационных систем:

- централизованная;
- сервис-ориентированная;
- облачная.

Под централизованной архитектурой подразумевают такую архитектуру, в которой существует один сервер, в рамках которого существует, как хранилище данных, так и приложение с его зависимостями и программными компонентами. Централизованная архитектура, в свою очередь, бывает в нескольких вариантах:

- файл-серверная, которая подразумевает простое хранение данных на сервере;
- двухуровневая, которая подразумевает разделение на клиентский и серверный уровни. Клиентский уровень реализует интерфейсы взаимодействия с системой. Серверный уровень отвечает за хранение данных;
- трёхуровневая, которая подразумевает наличие клиента, сервера приложения и пользовательского интерфейса.

Сервис-ориентированная архитектура оперирует такой абстракцией, как сервис. Сервисом – может быть, как отдельный программный компонент, так и любая, из выделенных при проектировании, подсистем приложения, в виде независимого приложения и интерфейса взаимодействия с ним. По средствам

интерфейса (например, API) сервисы взаимодействуют между собой и формируют единую инфраструктуру, в рамках которой функционирует вся система.

Облачная-архитектура – подразумевает взаимодействие приложения с готовыми внешними системами, предоставляющих услуги своих собственных типовых сервисов, и способы конфигурирования их в рамках облачных технологий [21].

С учётом того, что разрабатываемая информационная система является достаточно тривиальной и одним из требований является простота развёртывания, нами была выбрана централизованная клиент-серверная архитектура с двумя уровнями (уровнем представления и уровнем данных).

3.2 Выбор платформы и инструментов разработки программного обеспечения

Под инструментами разработки, мы подразумеваем языки программирования, при помощи которых, реализуются бизнес-логика и пользовательские интерфейсы.

В прошлой главе, мы определились с архитектурой нашего приложения и выбрали двухуровневую клиент-серверную архитектуру. Это подразумевает наличие клиентского приложения и сервера некоторой СУБД, к которому осуществляет подключение каждый из клиентов, для доступа к хранимым данным.

Стоит отметить, что, основными требованиями к языку программирования, является его актуальность, на данный момент времени, развитость, с точки зрения наличия библиотек и фреймворков, и соответствие технической архитектуре, то есть возможность полноценного использования на целевой аппаратной платформе. В нашем случае, платформой будут выступать компьютеры, с семейством процессоров x86 / x64, ввиду их

распространённости [16]. Ещё одним, зачастую, решающим аспектом в выборе языка программирования, являются, как банальные предпочтения команды разработчиков, так и навыки владения этими языками.

Нами было принято решение, реализовать бизнес-логику и интерфейсы приложения с использованием объектно-ориентированного языка высокого уровня C#.

В свою очередь ввиду того, что за развитие и поддержку языка отвечает компания Microsoft (чьей разработкой он и является) и разрабатывался он под семейство ОС Microsoft Windows, то целевой операционной системой, было логично выбрать Windows Server.

Таким образом, для запуска клиентского приложения, необходимо иметь физический сервер с архитектурой x86/64, на котором необходимо развернуть ОС Windows Server. Также, необходимо установить пакет .NET под архитектуру x86/64 и произвести конфигурирование сетевых интерфейсов для сетевого доступа к серверу хранения данных (серверу СУБД).

3.3 Выбор целевой СУБД разрабатываемой информационной системы

В качестве персистентного хранилища данных, в главе 2, нами был сделан выбор в пользу реляционных баз данных. В настоящий момент, к наиболее популярным реляционным СУБД можно отнести:

- MySQL;
- PostgreSQL;
- MariaDB.

Стоит сразу отметить, что все из представленных СУБД являются бесплатными или условно-бесплатными решениями, которые используются во многих коммерческих и не коммерческих продуктах и являются высококачественными продуктами на рынке. Стоит отметить, что в Январе

2010 года, СУБД MySQL, была приобретена коммерческой компанией Oracle, которая в свою очередь, имеет свою собственную коммерческую СУБД, тем самым, устранила, на тот момент, одного из крупнейших некоммерческих конкурентов [16]. Со временем сильно сократила возможности бесплатной версии MySQL, особенно в задачах, связанных с хранением и операциям над большими объёмами данных [20]. Проведём же короткий сравнительный анализ в рамках предоставляемого ими функционала и отличиями в таблице 3.

Таблица 3 – Критерии сравнения СУБД MariaDB и PostgreSQL

Критерий	MariaDB	PostgreSQL
Наличие типов для неструктурированных данных (JSON и т.д)	-	+
Индексирование	+	+
Оптимизация запросов	+	+
Соблюдение строгой типизации на уровне СУБД	-	+
Резервное сохранение данных	+	+

Исходя из выбранных нами критериев сравнения систем управления базами данных, нами было принято решение отдать предпочтение PostgreSQL, ввиду того что нами уже было принято решение использовать язык программирования C#, для реализации приложения. Язык программирования C#, в свою очередь, является языком со строгой типизацией данных, что обеспечивает дополнительную безопасность на уровне данных, связанную с его строгой, статической типизацией [22]. Таким образом, разработчику ещё на этапе компиляции (или же в случае с СУБД, на этапе выполнения запроса), будет выдана ошибка типов. Отсюда, считаем уместным выбрать и СУБД со строгой типизацией данных PostgreSQL.

Для обеспечения взаимодействия между разрабатываемым приложением на языке C# и СУБД PostgreSQL необходимо убедиться, что существует, так называемый, драйвер. Под драйвером, подразумевается библиотека, реализующая интерфейс взаимодействия между системой управления базами данных на конкретном языке программирования. Для C# и PSQL, используется NPSQL драйвер, имеющий поддержку со стороны Microsoft и предоставляющий интерфейсы взаимодействия для спецификации их основных языков, таких как C#, F#, Visual Basic [22].

Исходя из вышесказанного, абстрактное представление слоя данных и технологий, входящих в него, будет выглядеть следующим образом: Приложение на языке C# - NPSQL Driver – PostgreSQL. На рисунке 16 представлена структурная схема слоя данных в рамках разрабатываемой нами системы.

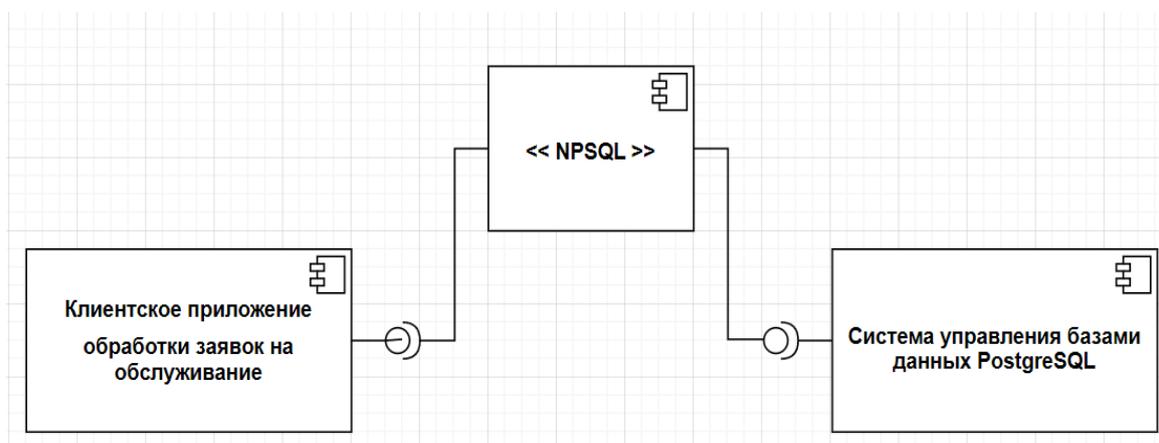


Рисунок 16 – Структурная схема слоя данных информационной системы

3.4 Проектирование и реализация физической модели данных

На основании, созданной ранее логической модели хранения данных информационной системы и выбранной системы управления базами данных, мы можем приступить к физическому проектированию и реализации. Исходя

из выделенных нами в рамках диаграммы сущность-связь сущностей и связей между ними, а также типов данных предоставляемых PostgreSQL создадим логическую модель данных, которая будет изображена на рисунке 17.

Центральной сущностью, нашей информационной системы является сущность «Заявка», которая имеет следующие поля: Поле «Название» типа массив символов для хранения вербального названия заявки; поле «Инвентарный номер», которое является внешним ключом и осуществляет связь с соответствующей моделью; поле «Код статуса», которое также является внешним ключом и обеспечивает связь с одноименной моделью; поле «Код сотрудника», связывающее заявку с моделью «Сотрудник»; поле «Дата создания», имеющее тип «DateTime» и хранящее дату и время создания заявки.

Сущность «Оргтехника», в свою очередь, описывает всю номенклатуру техники организации, которая внесена в систему и может быть передана в отдел обслуживания. Данная сущность включает в себя следующие поля: поле «Инвентарный номер», выступающее первичным ключом и уникальным идентификатором данной сущности; поле «Модель» типа char, которое необходимо для хранения вербального названия модели устройства; поле «серийный номер» типа char, которое хранит серийный номер оборудования для возможности дополнительной фильтрации и поиска; поле «Название» типа Char, которое хранит текстовое название устройства.

Сущность «Статус», которая отражает возможные статусы заявки и служащее для управления регламентом прохождения заявки. Имеет первичный ключ «Код-статуса» и поле «Название» типа char, необходимое для вербального отражения статуса заявки (например: «Заявки принята в отдел» или «Заявка отклонена»).

Сущность «Сотрудник», является второй по важности сущностью и отражает сотрудника, оставившего запрос на ремонт техники в наш отдел обслуживания. Данная сущность имеет следующие поля: поле «Фамилия»

типа char, которое служит для хранения фамилии сотрудника; поле «Имя», типа char, которое служит для хранения имени сотрудника; поле «Отчество» типа char, которое хранит в себе отчество сотрудника; поле «Номер отдела», которое является внешним ключом к модели «Отдел». Для обеспечения режима доступа к системе, модель «Сотрудник» предоставляет поля «Логин» и поле «Пароль» типа char. Поле «Логин» хранит текстовое значение уникального имени пользователя. Поле «Пароль» хранит хэшированный алгоритмом SHA-1 пароль пользователя системы в виде набора символов. Хэширование пароля необходимо для защиты пользовательских данных, даже в случае утечки данных из нашей системы в пользу злоумышленников.

Модель «Отдел» состоит из двух полей: «Номер отдела» - первичный ключ. «Название» - вербальное название отдела.

Для связи между заявкой и сотрудником, оставившим заявку, используется связь многие ко многим, реализованная при помощи создания промежуточной таблицы «Заявка-Сотрудник». Данная сущность имеет связи с сущностями «Заявка» и «Сотрудник» в виде внешних ключей к соответствующим таблицам. Также, сущность «Заявка-Сотрудник» имеет поля для хранения информации о времени начала выполнения заявки и времени её завершения, имеющие тип DateTime.

Также, все модели, имеют свой первичный ключ «Код» или «Номер» сущности, который выступает уникальным идентификатором экземпляра каждой из сущностей.

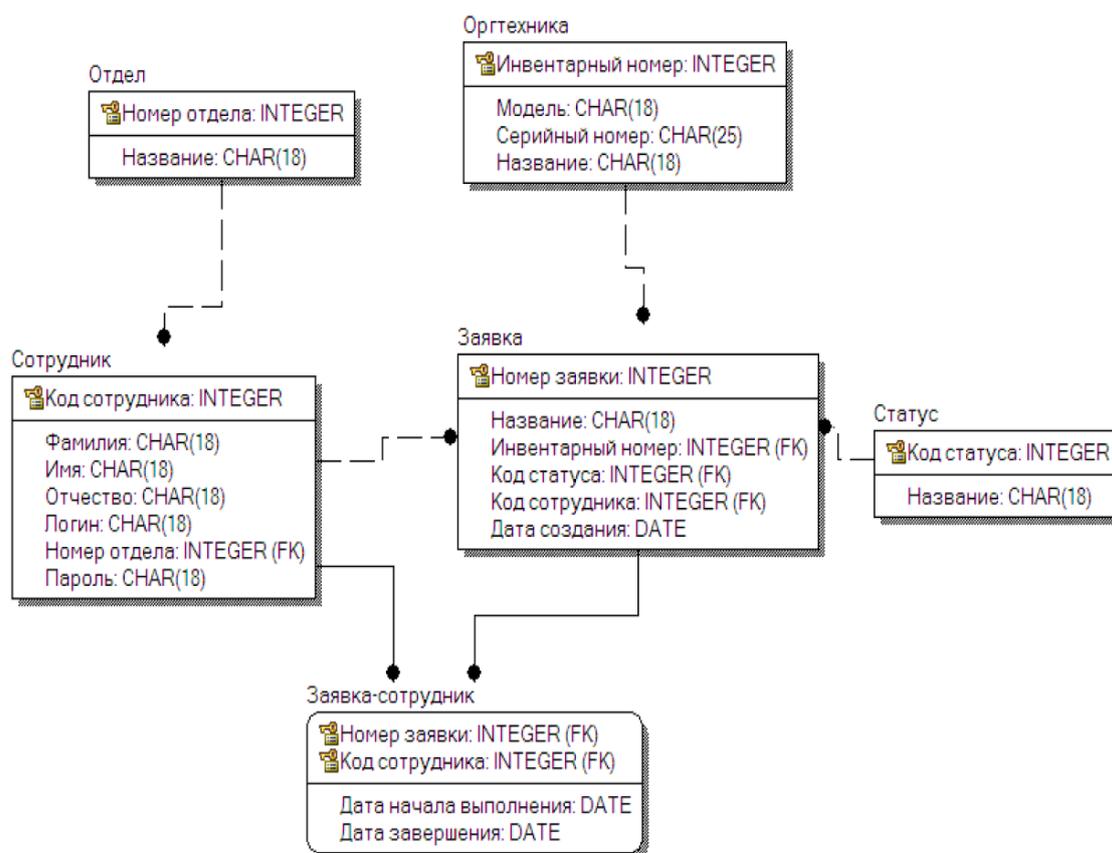


Рисунок 17 – Физическая модель данных

3.5 Реализация клиентского приложения информационной системы

Для удобства разработки на языке C# компания Microsoft предоставляет инструмент IDE Microsoft Visual Studio 2017 Community Edition, который мы и будем использовать для реализации нашего клиентского приложения и основной бизнес логики приложения [19]. В качестве основных зависимостей нашего приложения, будут выступать NPSQL драйвер и базовые средства взаимодействия с API Windows Forms предоставляемые в рамках пакетного решения Microsoft Visual Studio.

Npgsql — это ADO.NET Data Provider с открытым исходным кодом для PostgreSQL, который позволяет программам, написанным на C#, Visual Basic

и F#, получать доступ к серверу базы данных PostgreSQL. Он полностью написан на 100% C# и является бесплатным и открытым.

Также доступен провайдер Entity Framework Core, который предоставляет пользователям доступ к некоторым функциям, уникальным для базы данных PostgreSQL.

При внесении новых данных администратором приложения, они автоматически сохраняются в базу данных, откуда их можно получить в любое время с помощью запросов.

В рамках реализации информационной системы, выбрав инструменты и определившись с архитектурными особенностями информационной системы автоматизации проведения заявки на обслуживание, с учётом языка высокого уровня с объектно-ориентированным подходом, создадим диаграмму классов, на основании которой и будет разработана клиентская часть информационной системы и сопутствующая бизнес-логика [8]. На рисунке 18 представлена диаграмма классов.

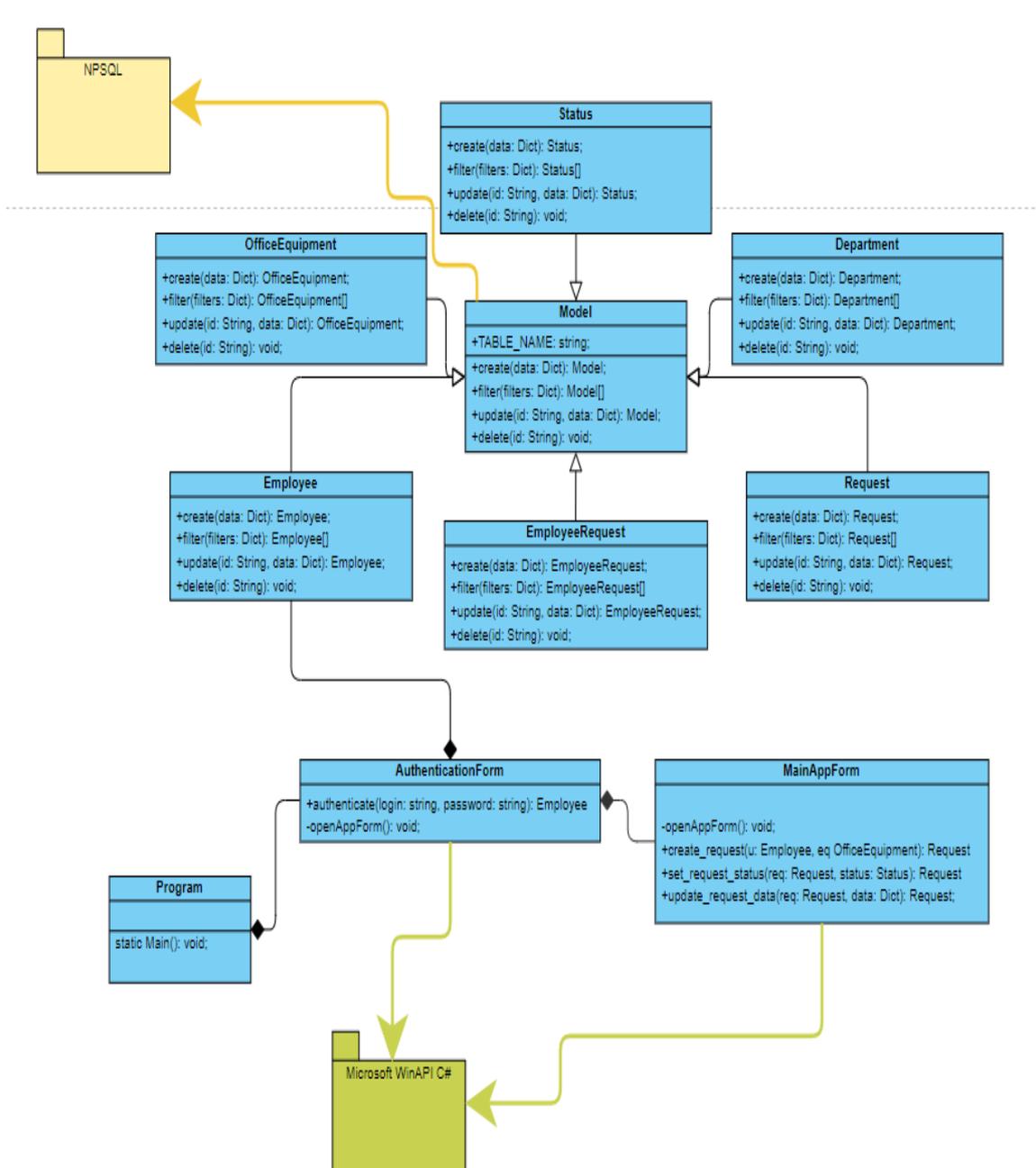


Рисунок 18 – Диаграмма классов приложения

Как видно из диаграммы классов, представленной выше, уровень данных приложения представлен в виде моделей, которые и реализуют внутри себя методы доступа к БД. Центральной сущностью на уровне данных является класс базовый Model, который реализует базовые методы доступа к

таблицам БД по средствам NPSQL драйвера. Все остальные абстракции, такие как: Request, Employee, EmployeeRequest, OfficeEquipment, Status, Department, которые соответствуют определённым нами сущностям на ER диаграмме; все эти сущности наследуются от базового класса Model.

Для предоставления графического интерфейса, мы используем базовые средства .NET платформы для взаимодействия с WinAPI и создания окон приложения с различными компонентами: такими как Таблицы, Компоненты ввода (input, buttons, checkboxes). Приложение состоит из двух основных экранов. Экрана предоставления учётных данных (логина и пароля) AuthenticationForm и экрана управления заявкой и представления заявки в виде таблицы MainAppForm. Доступ ко второму экрану предоставляется только после подтверждения соответствующих прав, которые закреплены за пользователем и записаны в таблицу БД.

Хорошим тоном, является выделение трёх слоёв абстракций приложения таких как: слой данных (Models), слой представления (Windows Forms .Net) и слоя бизнес-логики. На диаграмме выше, мы выделили только слой моделей и представлений, но основная бизнес-логика замешана в слой представления, что в случае отказа, например, от платформы Windows, будет требовать дополнительных издержек на выделения логики. Попробуем сразу учесть это и выделим дополнительную абстракцию Service, которая и будет хранить в себе нашу бизнес-логику. На рисунке 19 представлена диаграмма классов с учётом разделения на абстракции моделей, представления и бизнес-логики.

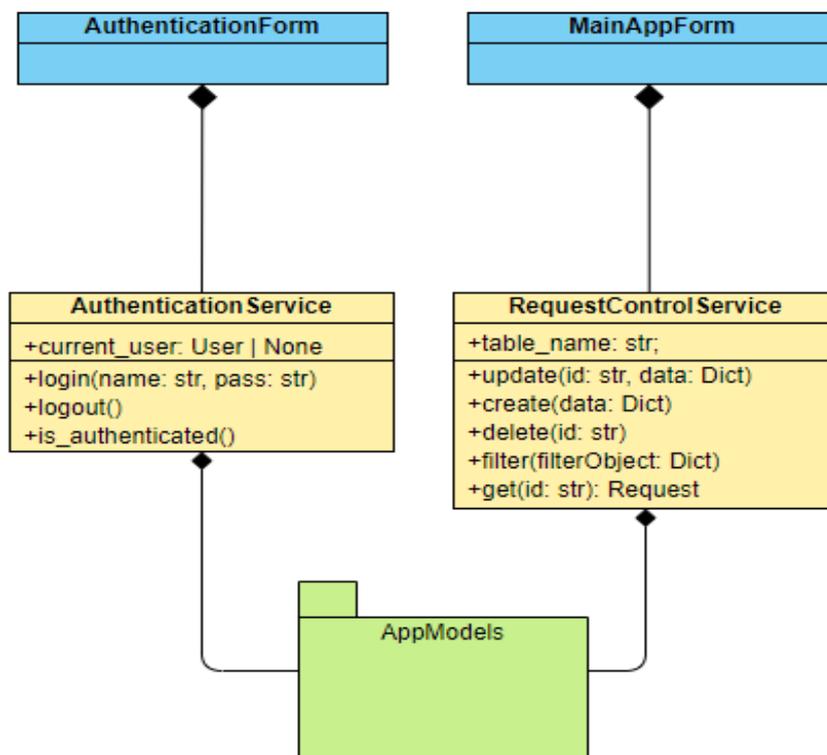


Рисунок 19 – Слой бизнес-логики приложения

В качестве основных классов, содержащих бизнес-логику приложения, нами выделены два сервиса Authentication Service и Request Control Service. Как следует из названий, первый хранит в себе метод аутентификации пользователя системы, завершения сессии и методы проверки текущего статуса пользователя. В свою очередь, сервис управления заявками, описывает основные методы взаимодействия с заявкой и связанными с ней сущностями.

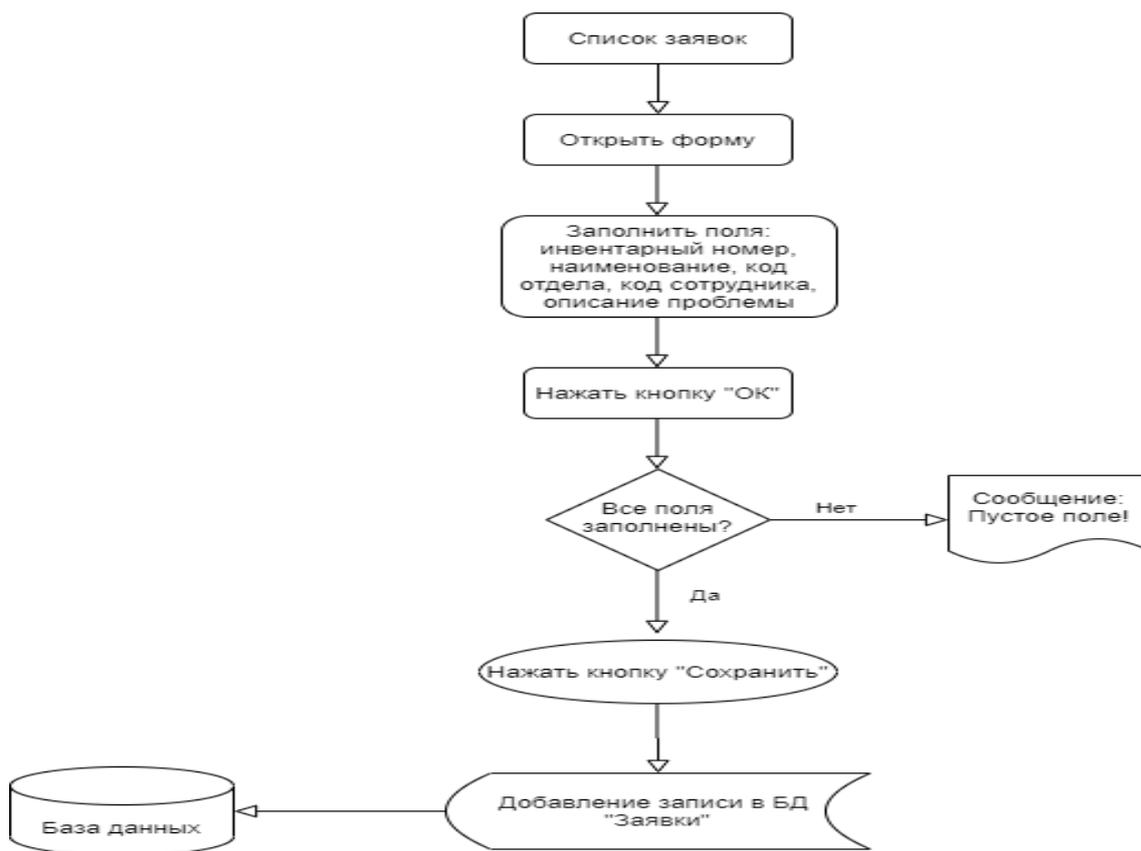


Рисунок 20 – Блок-схема алгоритма функции заполнения формы «Заявка на обслуживание оргтехники»

Алгоритм заполнения заявки начинается, когда пользователь открывает форму для создания заявки и выбирает данные оргтехники, которые предлагает система. Затем вводит данные о неисправности и отправляет заявку [18].

3.6 Описание функциональности информационной системы

Открытие пользователем информационной системы запросит авторизоваться. Пользователь не зависимо от того это сотрудник банка или сотрудник отдела обслуживания оргтехники должен ввести свои логин и пароль, чтобы авторизоваться в системе.

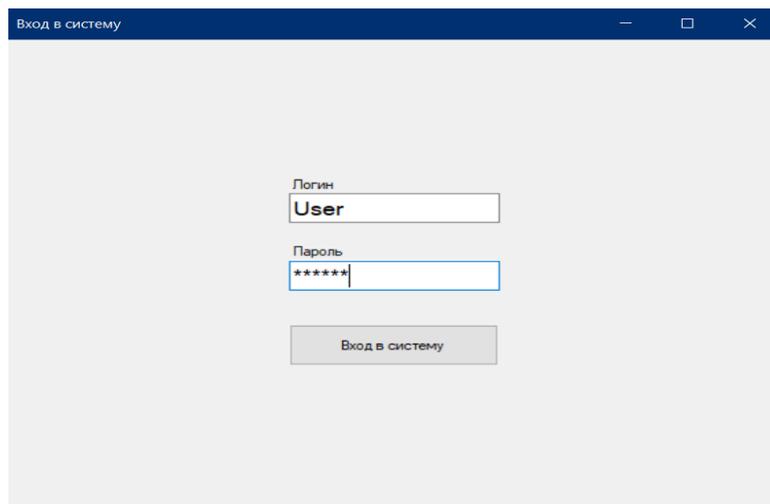


Рисунок 21 – Форма входа в систему

В случае, ввода некорректной пары логина и пароля, пользователь получит соответствующее сообщение о недостоверных учётных данных. Стоит отметить, что перед отправкой пароля, нам необходимо выполнить хэширование значения и осуществить сопоставление с паролем именно хэшированного значения.

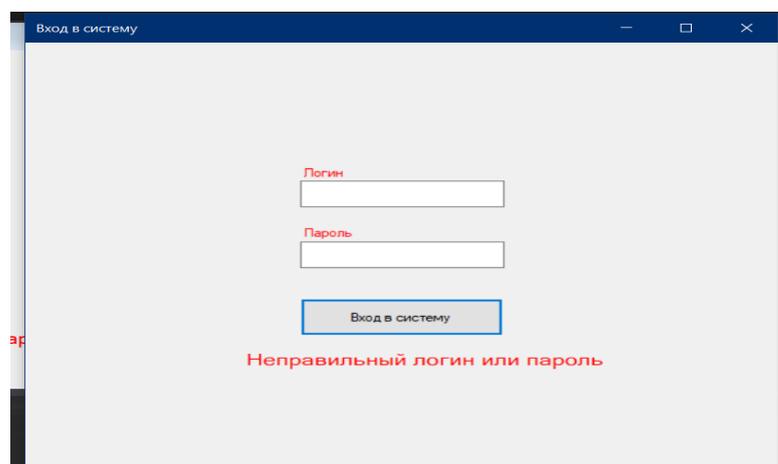


Рисунок 22 – Валидация учётных данных

После входа в систему сотрудник банка увидит форму для заполнения заявки. Форма содержит данные, сохраненные в базе данных той оргтехники,

которая стоит на учете в банке. Сотрудник выбирает данные из списков и добавляет заявку, на рисунке 23 представлена форма.

Инвентарный номер	Название	Модель	Серийный номер
1	Monitor	UE51	2565411
2	Printer	Hp1010	an1097
3	Monitor	UE51	2565412
4	Mouse	G503	12586611d
5	Mouse	G503	12586611f
6	Mouse	G503	12586611c

Добавить заявку

2565411
an1097
2565412
12586611d
12586611f
12586611c
12536631c
12555631c
12555211c
12554211c

Рисунок 23 – Форма создания заявки на обслуживание оргтехники

После выбора всех данных оргтехники сотрудник отправляет заявку на обслуживание, рисунок 24.

Инвентарный номер	Название	Модель	Серийный номер
6	Monitor	UE51	2565411

Добавить заявку

Рисунок 24 – Готовая заявка

Сотрудник отдела обслуживания оргтехники, войдя в систему видит список заявок, который может обновить и получить новые заявки, заявки располагаются в порядке поступления, рисунок 25.

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	25.03.2020 00:00:00	В работе	В работе
2	Не включается принтер	Pronskiy	Sila	Sales	Printer	Hp1010	an1097	23.04.2020 00:00:00	В работе	В работе
3	Заменить картридж	Filatova	Olga	Human resources	Printer	Hp1010	an1097	26.11.2020 01:42:55	Новая заявка	Новая заявка
4	Заменить мышь	Filatova	Olga	Human resources	Mouse	G503	12536631c	26.11.2020 01:47:10	Новая заявка	Новая заявка
5	не работает клавиатура	Filatova	Olga	Human resources	Keyboard	G505	12555631c	26.11.2020 01:47:58	В работе	В работе
6	Почистить монитор	Filatova	Olga	Human resources	Monitor	UE51	2565411	26.11.2020 01:49:06	Новая заявка	Новая заявка
7	Не работает мышь	Lukina	Veronika	Human resources	Mouse	G503	12536631c	26.11.2020 01:51:35	Новая заявка	Новая заявка
8	Не работает	Filatova	Olga	Human resources	Monitor	UE51	2565411	26.11.2020 11:50:48	В работе	В работе
9	Не работает	Filatova	Olga	Human resources	Mouse	G503	12586611d	27.11.2020 13:48:59	Новая заявка	Новая заявка
10	Не работает	Filatova	Olga	Human resources	Monitor	UE51	2565412	18.12.2020 22:08:34	Новая заявка	Новая заявка
11	Не работает	Filatova	Olga	Human resources	Mouse	G503	12586611f	18.12.2020 22:12:31	В работе	В работе
12	Не работает	Filatova	Olga	Human resources	Mouse	G503	12586611c	18.12.2020 22:14:48	Новая заявка	Новая заявка

Рисунок 25 – Список заявок на обслуживание оргтехники

Специалист отдела обслуживания может управлять заявка разными способами, может фильтровать заявки по поломке, к примеру мониторов, или процессоров, где будут отображаться только те заявки, где необходим ремонт мониторов, рисунок 26.

ID заявки	Название	Фамилия	Имя	Отдел	Оборудование	Модель	S/N	Дата поступления	Текущий статус	Новый статус
3	Заменить картридж	Filatova	Olga	Human resources	Printer	Hp1010	an1097	26.11.2020 01:42:55	Новая заявка	Новая заявка
13	Не работает	Filatova	Olga	Human resources	Printer	Hp1010	an1097	18.12.2020 22:15:00	Новая заявка	Новая заявка

Рисунок 26 – Фильтрация заявок

Также сотрудники могут присваивать заявке статус, в котором она находится на данный момент, после чего обновлять список с заявками. Новый статус будет виден также и сотруднику, который подал заявку на обслуживание оргтехники.

3.7 Обоснование экономической эффективности разработки приложения

Проведем расчет, согласно которому будет наглядно видно об эффективности разработки и внедрения информационной системы. Произведем расчеты по формуле (1).

$$U = U_{it} + U_{alg} + U_{conf} + U_{int} \quad (1)$$

где U_{it} – затраты на оборудование и материалы;

U_{alg} – затраты на написание алгоритмов;

U_{conf} – затраты на техническую документацию и отладку;

U_{int} – затраты на интеграцию системы.

Объем затрат необходимых на написание исходного кода программного продукта возможно посчитать по следующей формуле (2).

$$U_{alg} = \frac{Z_{razr}}{d} \cdot T + Ps \quad (2)$$

где Z_{razr} – зарплата программиста в месяц;

d – количество рабочих дней в месяце;

T – диапазон времени необходимый на разработку продукта;

Ps – прочие расходы и отчисления.

Произведем необходимый подсчет общих затрат на разработку программного продукта. Средняя зарплата по региону у программиста составляет 127 000

рублей, на разработку будет затрачено 17 дней, прочие расходы и отчисления будут равняться 16 000 рублей. Итого общие затраты на разработку ПО будут равняться: $127\,000 / 23 \times 17 + 16\,000 = 109\,869$ рублей.

Расходы на закупку оборудования и материалов можно рассчитать по формуле (3).

$$U_{it} = U_{pc} + U_{print} + U_{en} \quad (3)$$

где U_{pc} - стоимость АРМ (автоматизированное рабочее место) ПК;

U_{print} - стоимость принтера;

U_{en} – расходы на прочее оборудования.

Средняя цена ПК составляет 75 000 рублей, принтера 17 000 рублей, другого оборудования 16 000 рублей. В таком случае затраты составят: $75\,000 + 17\,000 + 16\,000 = 108\,000$ рублей.

Для интеграции системы в рабочий процесс так же необходимы определенные расходы. Рассчитаем их по формуле (4).

$$U_{int} = V_{int} + Z_{int} \quad (4)$$

где V_{int} – затраты на интеграцию системы;

Z_{int} – зарплата специалиста по интеграции системы в рабочий процесс.

Затраты времени на внедрение программного продукта были оценены как 294 400 рублей, зарплата разработчика 91 500 рублей, общие затраты на процесс внедрения составили 385 500 рублей.

Расходы на время для интеграции программного продукта рассчитываются по следующей формуле (5).

$$V_{int} = T \cdot X \cdot M \quad (5)$$

где T – время кратное одному часу работы;

X – всего необходимо дней;

M – стоимость одного часа работы.

Среднее время работы в день составляет 8 часов, число дней в месяце которое отработывает сотрудник - 23 дня, час работы стоит 1 600 рублей. Если произвести расчет по формуле затраты на внедрение, в итоге получим:
 $8 \times 23 \times 1\,600 = 294\,400$ рублей.

Заработная плату рассчитаем по формуле (6).

$$Z_{\text{int}} = O_{\text{rz}} \cdot \left(\frac{D}{T}\right) + Ws \quad (6)$$

где O_{rz} - оклад разработчика;

D – количество дней;

T – число рабочих дней в месяц;

Ws – отчисления на соц. страхование.

Оклад равен 75 000 рублей, число рабочих дней 23, отчисления по налоговой ставке составили 16 500 рублей. Итого заработная плата будет равна: $75\,000 \times (23/23) + 16\,500 = 91\,500$ рублей.

Затраты на техническую документацию, отладку и тестирование программного продукта рассчитаем по следующей формуле (7).

$$U_{\text{conf}} = L_{\text{otl}} + L_{\text{zp}} \quad (7)$$

где L_{otl} – стоимость одного часа, затраченного на техническую документацию, отладку и тестирование программного продукта;

L_{zp} – оплата работы программиста.

Затраты на техническую документацию, отладку и тестирование программного продукта равны: $385\,500 + 544\,730 = 930\,230$ рублей.

Размер оплаты услуг программиста, занятого отладкой и написанием программы, можно рассчитать по формуле (8).

$$L_{zp} = D_m \cdot (Z_{razr} + Oz) \quad (8)$$

где D_m – количество дней;

Z_{razr} – оклад;

Oz – отчисления.

Разработка программного продукта составляет 94 дня, ежедневная оплата работы программиста составляет 4 750 рублей, отчисления равны 1 045 рублей, зарплата рассчитанная на 94 дня равна: $94 \times (4\,750 + 1\,045) = 544\,730$ рублей.

Сумму расходов на эксплуатацию системы можно рассчитать по формуле (9).

$$W_{exp} = T_v \cdot N_{ch} \quad (9)$$

где T_v – время;

N_{ch} – стоимость часа.

Эксплуатационные расходы, связанные с использованием программного продукта, составляют: $1\,330 \times 180 = 239\,400$ рублей.

Выгода в год от автоматизации определяется нормами эксплуатационных расходов, связанных с использованием программного продукта, ежемесячным начислением на зарплату и основной зарплатой работников, решающих задачу вручную, коэффициентом, учитывающим дополнительную зарплату. Рассчитать можно по формуле (10).

$$dW_{exp} = (1 + Zr) \cdot (1 + Zg) \cdot Z_{osn} - W_{exp} \quad (10)$$

где W_{exp} – эксплуатационные расходы, связанные с использованием программы;

Zr – начисление на оклад работника;

Z_g – коэффициент;

$Z_{осн}$ – основной оклад.

Ежемесячные начисления на зарплату составили 0,35, коэффициент, учитывающий дополнительную зарплату равен 0,15, основная зарплата работников составляет: $4 \times 12 \times 35\,500 = 1\,704\,000$ рублей. Расходы на эксплуатацию составили 239 400 рублей. В итоге годовой прирост прибыли организации составил: $(1+0,35) \times (1+0,15) \times 1\,704\,000 - 239\,000 = 2\,406\,460$ рублей.

Стоимость программного продукта можно рассчитать по формуле (11).

$$H = \frac{M}{X} \quad (11)$$

где M – затраты на разработку программы;

X – число пользователей.

Затраты на разработку программного продукта составили 1 533 599 рублей для двух пользователей. Тогда стоимость программного продукта составит 766 799,5 рублей.

Срок окупаемости определяется отношением капитальных вложений к экономии годовых эксплуатационных расходов. Расчет можно произвести по следующей формуле (12).

$$J = \frac{M}{fY} \quad (12)$$

где M – расходы на разработку программного продукта;

fY - экономия годовых эксплуатационных расходов.

Затраты на разработку программного продукта составили 1 533 599 рублей, экономическая выгода составила 2 406 460 рублей, таким образом срок окупаемости равен: $1\,533\,599 / 2\,406\,460 = 0,637$ лет (~7 месяцев).

На этапе расчетов была представлена экономическая выгода от внедрения программного продукта.

На рисунке 27 представлен график окупаемости предлагаемой информационной системы.

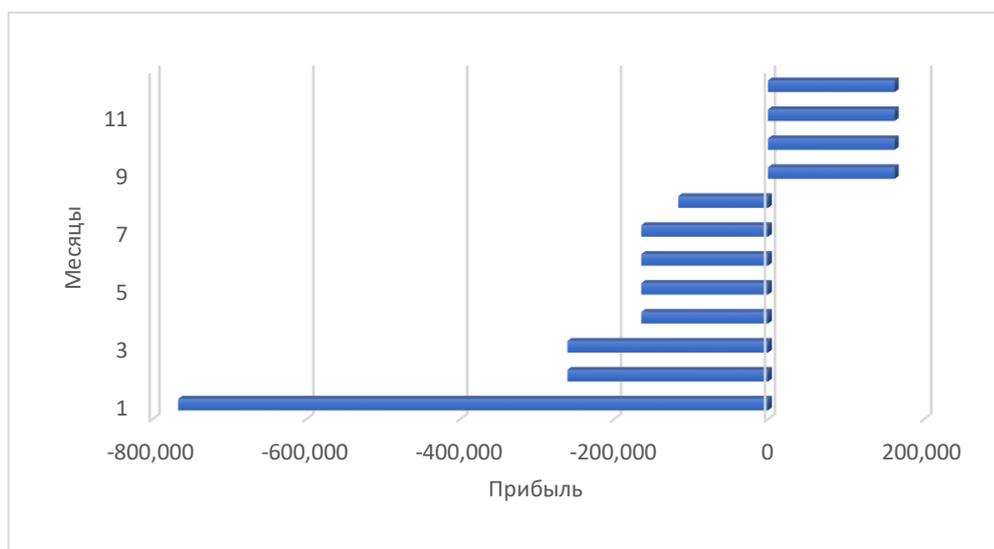


Рисунок 27 – График окупаемости предлагаемой информационной системы

С помощью расчетов наглядно видно, что период окупаемости интеграции системы составляет не более восьми месяцев.

Выводы по главе: в третьей главе была завершена физическая реализация информационной системы, включающая создание физической модели данных и разработку пользовательского интерфейса. Используя выбранную системную архитектуру и инструменты разработки, информационная система была успешно развернута. Это поможет сотрудникам сервисного отдела эффективно управлять запросами на техническое обслуживание офисного оборудования банка, а также вести их учет и хранение в системе для последующего доступа по мере необходимости. Экономическая эффективность разработки была обоснована расчетами.

Заключение

В ходе выполнения выпускной квалификационной работы был представлен всесторонний обзор процесса разработки информационной системы для управления запросами на техническое обслуживание офисного оборудования в ПАО «Росбанк».

На начальном этапе проекта основное внимание уделялось анализу организационной структуры и изучению операционных процессов внутри компании. Это включало в себя выявление неавтоматизированных бизнес-процессов, которые могли бы выиграть от автоматизации за счет внедрения информационных технологий.

На основании результатов анализа было определено, что сервисный отдел, отвечающий за ремонт и техническое обслуживание офисной техники банка, больше всего выиграет от автоматизации процессов.

Кроме того, был проведен сравнительный анализ существующих программных решений, разработанных сторонними компаниями, который выявил недостатки, присущие этим продуктам.

Для реализации приложения были выбраны средства разработки на C# и СУБД Microsoft MySQL как наиболее подходящие и эффективные варианты для данного проекта.

После проведения экономических расчетов было установлено, что проект является экономически выгодным для компании, с предполагаемым сроком окупаемости примерно в восемь месяцев.

Разработанная информационная система позволит повысить эффективность работы сотрудников отдела обслуживания, а также повысит контроль над процессом исполнения заявок и последующего их учета.

Список используемой литературы и используемых источников

1. Алексеев А.В. Современные технологии разработки программного обеспечения. – М.: Бином, 2020.
2. Арнольд К. «Python для сложных задач. Наука о данных и машинное обучение» // М.: ООО «Издательство «Эксмо-Пресс», 2019. - 304 с.
3. Богданов Д. «Проектирование баз данных» // М.: ООО «Издательство Астрель», 2019. - 272 с.
4. Фаулер, М. UML Основы [Текст] / М. Фаулер. - СПб: Символ-Плюс, 2005. - 184 с. - ISBN: 5-93286-060-X.
5. Галимова Р.Х. "Программирование на языке Python. Разработка приложений" // М.: Издательство "Бином. Лаборатория знаний", 2017. - 480 с.
6. Макконнелл С., Совершенный код, Мастер-класс / Пер. с англ. – М.: Издательство «Русская редакция», 2012. – 896 стр.: ил., ISBN 978-5-7502-0064-1
7. Вигерс, К. Разработка требований к программному обеспечению[Текст]/ К.Вигерс, Дж. Битти. - 3-е изд. - Издательство: русская редакция, 2014. - 737 с. - ISBN: 978-5-7502-0433-5.
8. Мандел, Т. Разработка пользовательского интерфейса: практическое пособие [Текст] / Т. Мандел. Издательство: ДМК Пресс, 2008. - 412 с. - ISBN: 5-94074-069-3.
9. Козлов В.А., Козлова Н.В. Программирование на Python 3: подробное руководство. – М.: ДМК Пресс, 2020.
10. Колесникова, А. А., & Колесников, А. С. (2019). Проектирование системы управления проектами в условиях цифровой экономики. Информационные технологии в науке, управлении, социальной сфере и медицине, (4), 128-131.

11. Кравченко О.В., Кравченко А.В. «Разработка программного обеспечения на платформе .NET Framework» // М.: БХВ-Петербург, 2019. - 352 с.
12. Кравченко О.В., Кравченко А.В. «С# 7.0 и платформа .NET Core 2.0. Программирование для профессионалов» // М.: БХВ-Петербург, 2018. - 848 с.
13. Крамер Р., Лемер М. «Современный JavaScript» // М.: ООО «Издательский дом «Питер», 2018. - 472 с.
14. Крылов А.В. Основы программирования на JavaScript. – М.: Издательство Юрайт, 2018.
15. Кузнецов, А. Ю., & Смирнова, И. В. (2016). Системы управления проектами в области IT-технологий. Информационные технологии в науке, управлении, социальной сфере и медицине, 151-155.
16. Куликов А.В. Методология разработки программного обеспечения. – М.: Издательство Юрайт, 2017.
17. Ларионова, И. А., & Трухин, А. В. (2017). Проектирование архитектуры и выбор технологий реализации системы управления проектами. Управление проектами и программами, (4), 62-74.
18. Лескова О. «Проектирование информационных систем» // М.: Издательство Юрайт, 2018. - 352 с.
19. Макконнелл С. Совершенный код. Мастер-класс. – М.: Бином, 2020.
20. SQL Server: руководство по SQL Server. [Электронный ресурс]. – URL: <https://docs.microsoft.com/ru-ru/sql/sql-server/tutorials-for-sql-server-2016?view=sql-server-ver15> (дата обращения: 15.04.2024)
21. SQL Server: программное обеспечение. [Электронный ресурс]. – URL: <https://www.microsoft.com/ru-ru/sql-server/sql-server-downloads> (дата обращения: 15.04.2024)

22. Visual Studio 2020: программное обеспечение. [Электронный ресурс]. – URL: <https://visualstudio.microsoft.com/ru/vs/> (дата обращения: 15.04.2024)

23. Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика [Текст] / Т. Коннолли, К. Бегг.- 3-е изд. - Издательство: Вильямс, 2008. - 1440 с. - ISBN: 5-8459-0527-3.