



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

БАКАЛАВРСКАЯ РАБОТА

На тему

Разработка автоматизированного рабочего места (на примере
Муниципального казенного учреждения «Многофункциональный
центр предоставления государственных услуг Туапсинского района»)

Исполнитель

Середа Дарья Владимировна

(фамилия, имя, отчество)

Руководитель

кандидат технических наук доцент

(ученая степень, ученое звание)

Котиков П. Е.

(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой

(подпись)

кандидат технических наук

Слесарева Людмила Сергеевна

«22» 06 2016г.

Санкт-Петербург
2016



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ
ГИДРОМЕТЕОРОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Кафедра Прикладной информатики

БАКАЛАВРСКАЯ РАБОТА

На тему

Разработка автоматизированного рабочего места (на примере
Муниципального казенного учреждения «Многофункциональный
центр предоставления государственных услуг Туапсинского района»)

Исполнитель

Середа Дарья Владимировна

(фамилия, имя, отчество)

Руководитель

кандидат технических наук доцент

(ученая степень, ученое звание)

Котиков П. Е.

(фамилия, имя, отчество)

«К защите допускаю»

Заведующий кафедрой

(подпись)

кандидат технических наук

Слесарева Людмила Сергеевна

«__» _____ 20__ г.

Санкт-Петербург
2016

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
1. Исследование предметной области.....	6
1.1 Общая характеристика объекта	6
1.2 Характеристика основных служб	11
2. Методический раздел	13
2.1 Анализ существующих разработок по автоматизации выбранной задачи.....	13
2.2 Выбор и обоснование стратегии автоматизации	14
Глава 3. Проектирование и реализация информационной системы	27
3.1 Построение модели «Как должно быть»	27
3.2 Построение блок-схем алгоритмов	31
3.3 Программная реализация	35
3.4 Экономическая эффективность	48
ЗАКЛЮЧЕНИЕ	55
СПИСОК ЛИТЕРАТУРЫ	58
ПРИЛОЖЕНИЕ 1.....	61
Программный код	61

ВВЕДЕНИЕ

С появлением и широким распространением ЭВМ у человечества появилась возможность автоматизировать процессы сбора, обработки и модификации информации, а также различные информационно-справочные и расчётные задачи и бизнес-процессы предприятия.

Во всех сферах человеческой деятельности предполагается широкое применение принципов обработки и управления информацией в процессе развития общества. Значительную роль в данном случае играет процесс внедрения в сферу массового обслуживания и различные области производства автоматизированных рабочих мест по управлению и обработке информации. Данные системы позволяют снизить затраты времени и трудовых ресурсов, повысить качество результатов труда, уменьшить вероятность ошибок, просчетов и аварийных ситуаций на производстве.

Одна из важнейших проблем экономических преобразований в Российской Федерации в настоящий момент – это внедрение и рациональное использование углублённых возможностей ЭВМ. Широкое использование на практике методов алгоритмического описания задач и их программирования высокого уровня является наиболее эффективным путём решения указанной проблемы.

Деятельность предприятия представляет собой общее число отношений. Это отношения, которые возникают в процессе производства и реализации продукции, и включают формирование и использование денежных доходов, организацию взаимоотношений с другими предприятиями, бюджетом, банками, страховыми организациями и др., а также обеспечение кругооборота средств в воспроизводственном процессе. Деятельность многофункционального центра заключается в предоставлении государственных и муниципальных услуг населению.

Объектом исследования в данной выпускной квалификационной работе (ВКР) является Муниципальное казенное учреждение «Многофункциональный центр предоставления государственных услуг Туапсинского района».

Предметом исследования в настоящей ВКР выступает процесс подготовки документов для органов ФМС (Федеральной Миграционной Службы).

Объектом деятельности многофункционального центра является предоставление информационных услуг, поэтому автоматизация бизнес-процессов является основополагающим фактором функционирования предприятия.

Целью ВКР является разработка автоматизированного рабочего места оператора предоставляющего услуги населению по подготовке документации для получения/замены паспорта Российской Федерации.

Для реализации поставленной цели необходимо решить задачи:

1. Провести анализ действующих систем автоматизации документооборота на предприятии;
2. На основе проведенного анализа выявить слабые звенья в общей цепи документооборота;
3. Углубить и закрепить теоретические знания, полученные в ходе профессиональной и специальной подготовки;
4. Продемонстрировать практические навыки и компетенции в сфере профессиональной деятельности путем решения практических задач, а также опыта в реальных условиях деятельности предприятия.
5. Разработать автоматизированное рабочее место оператора по подготовке документации для органов Федеральной Миграционной службы;
6. Выявить существующие недостатки и разработать рекомендации по их исправлению.

Основная часть выпускной квалификационной работы состоит из трех Глав:

Глава 1. Исследование предметной области – содержит общую характеристику объекта и характеристику основных служб. В данной главе представлена деятельность предприятия, организационная структура и используемые в процессе деятельности вспомогательные информационные системы и инструменты.

Глава 2. Методологический раздел – содержит анализ существующих разработок по автоматизации выбранной задачи. Описан функционал разработок, которые применяются на аналогичных предприятиях для автоматизации процесса подготовки документов для ФМС. Так же данная глава содержит выбор и обоснование стратегии автоматизации.

Глава 3. Проектирование и реализация информационной системы – содержит построение модели «Как должно быть», которая наглядно описывает все процессы, выполняемые АРМ. Так же в этой главе представлены блок-схемы алгоритмов, программная реализация и экономическая эффективность разработанного АРМ.

1. Исследование предметной области

1.1 Общая характеристика объекта

Целью усовершенствования систем государственного управления одной из главных задач, стоящих перед органами государственной власти, является создание и развитие сети многофункциональных центров, которые призваны предоставлять государственные и муниципальные услуги населению, в том числе по принципу «одного окна». Такая практика широко применяется во многих странах и способствует сокращению сроков предоставления услуг, снижению очередей и повышению уровня удовлетворенности граждан работой органов государственной власти.

В городе Туапсе с 11.03.2011г. действует казенное муниципальное учреждение «Многофункциональный центр предоставления государственных и муниципальных услуг Туапсинского района», в соответствии с Федеральным законом от 27.07.2010 N 210-ФЗ «Об организации предоставления государственных и муниципальных услуг». [17]

Объектом исследования ВКР является государственное учреждение: Муниципальное казенное учреждение «Многофункциональный центр предоставления государственных услуг Туапсинского района» (МФЦ). Предприятие находится в ведомственной подчиненности у администрации муниципального образования Туапсинский район. Основной вид деятельности предприятия заключается в предоставлении государственных и муниципальных услуг населению.

С целью максимально высокой производительности многофункционального центра, используется комплекс систем автоматизации бизнес-процессов с наименованием АИС «МФЦ».

Структура АИС «МФЦ». Система АИС «МФЦ» состоит из следующих компонентов:

- Подсистема автоматизации процессов оказания услуг

Данная подсистема позволяет централизованно управлять административными регламентами и контролировать процесс оказания услуг. В основе подсистемы лежит база системы управления рабочими процессами (workflow).

- Подсистема информационно-справочного обеспечения.

Данная подсистема служит для информирования населения о деятельности многофункционального центра, участвующих в процессе оказания услуги организациях, оказываемых ими услугах, ходе предоставления услуг и другой информации, связанной с предоставлением муниципальных и государственных услуг в МФЦ. Также подсистема осуществляет централизованное хранение справочно-нормативной информации и предоставляет доступ к ней из других подсистем.

- Подсистема электронной очереди

Данная подсистема призвана служить для автоматизации процесса управления потоком посетителей в МФЦ, обеспечения комфортных условий ожидания и оптимизации загрузки специалистов многофункционального центра.

- Подсистема СМЭВ (Система межведомственного электронного взаимодействия).

Данная подсистема позволяет обеспечивать возможности обмена данными между АИС «МФЦ» и автоматизированными системами заинтересованных ведомств. Так же СМЭВ служит для обмена данными между подсистемами АИС «МФЦ» на одном уровне процессов автоматизации; синхронизации баз данных разных уровней процессов автоматизации; импорта и экспорта данных; а также обработки пересылаемой информации и доставки системных сообщений.

- Подсистема обработки и хранения документов.

Данная подсистема является ядром архитектуры данных АИС «МФЦ». Она выполняет следующие функции:

- Хранение и обработка информации, поступающей из внешних источников, в соответствии с заключенными соглашениями о межведомственном информационном обмене;

- Форматно-логический и семантический контроль, очистка и преобразования исходной информации с целью повышения качества предоставляемых данных;

- Управление метаданными;

- Оптимальная организация данных с точки зрения скорости выполнения операций загрузки информации и запросов пользователей;

- Разграничение доступа пользователей к данным;

- Обеспечение возможности доступа пользователей к данным ЕХД (Единое хранилище данных) через слой аналитических приложений.

- Подсистема администрирования

Данная подсистема администрирования осуществляет общее конфигурирование системы, формирование информационных массивов, настройку параметров системы и ее технических характеристик.

- Подсистема обеспечения внутренней работы МФЦ

Данная подсистема призвана автоматизировать внутреннюю работу многофункционального центра, хранение нормативных и иных документов, а также для организации совместной работы над документами в МФЦ.

- Аналитическая подсистема

Данная подсистема служит для формирования статистических и аналитических отчетов по работе многофункционального центра. Она представляет информацию в различных срезах с использованием технологий OLAP.

- Подсистема безопасности

Подсистема безопасности предназначена для обеспечения информационной безопасности АИС «МФЦ». [5]

Для автоматизации документооборота на предприятии используется система «АИС МФЦ СКАТ». Она представляет собой программно-технический комплекс, предназначенный для автоматизации основных административно-управленческих и технологических процессов, протекающих в рамках МФЦ, и обеспечения эффективной информационной поддержки деятельности организатора и участников МФЦ. Так же «АИС МФЦ СКАТ» осуществляет функции автоматизации обращения и своевременного получения заявителем полной и актуальной информации о порядке предоставления государственных и муниципальных услуг в соответствии со стандартом, в том числе в электронной форме, повышение качества и снижение административных барьеров при оказании государственных и муниципальных услуг, обеспечение открытой деятельности органов, предоставляющих государственные и муниципальные услуги, повышение эффективности межведомственного взаимодействия и обмена информацией. В состав «АИС МФЦ СКАТ» входит программное обеспечение руководителя, операциониста, канцелярии, информационного киоска. [16]

Для автоматизации документооборота на предприятии используется система «Синкопа-Документ». Данная система служит для того, чтобы предоставлять руководителю подразделения возможность контролировать прохождение всей документации, мгновенно получать отчеты по количеству выполненных (в том числе с нарушением сроков исполнения), находящихся в работе и просроченных документов.

Система электронного документооборота «СИНКОПА – ДОКУМЕНТ» разработана департаментом по транспорту и связи. Она является платформой для решения различных прикладных задач, а также управления документооборотом.

Благодаря этой системе появляется возможность формализовать различные процессы, такие как:

- обработка документов различными подразделениями и исполнителями;

- задание маршрутов движения документов;
- отслеживание фактического перемещения документов.

«СИНКОПА – ДОКУМЕНТ» обеспечивает решение следующих задач:

- Автоматизация процессов делопроизводства – возможность перевода документооборота администрации в электронный вид;

- Организация электронного документооборота;

- Формирование банков данных входящих, исходящих и внутренних документов;

- Разграничение прав к предоставлению информации оперативного доступа;

- Фильтрация документов по необходимым атрибутам;

- Система отслеживания движения документа со своевременным оповещением сотрудников и ответственных лиц;

- Формирование индивидуального задания сотруднику на работу с документами и осуществление контроля за его выполнением;

- Составление плана работы над документами и формирование отчета по их фактическому выполнению;

- Формирование плановых и оперативных отчетов, связанных с делопроизводством и документооборотом. Данная система позволяет обеспечить полный охват сотрудников подразделений, что приведет к полному и замкнутому циклу автоматизации документооборота. А также сэкономит рабочее время персонала за счет использования современной техники, такой как поточные сканеры, принтеры и считыватели штрих-кодов. Каждое входящее письмо при помощи поточного сканера будет сканироваться и регистрироваться в системе;

- Система позволяет получить доступ для удаленной работы через сеть Интернет, в том числе с использованием GPRS-подключений.;

- Применение Интернет-технологий обеспечивает одновременную работу любого количества сотрудников на любом расстоянии, находящихся в системе. [9]

1.2 Характеристика основных служб

Организационная структура сильно влияет на способность предприятия адаптироваться к изменениям внешней среды. Она представляет собой совокупность структурных подразделений и связей между ними. Важную роль в данном случае играет структура организации, посредством которой и через которую это взаимодействие осуществляется.

Организационно – управленческую структуру предприятия можно представить в виде схемы (рисунок 1):

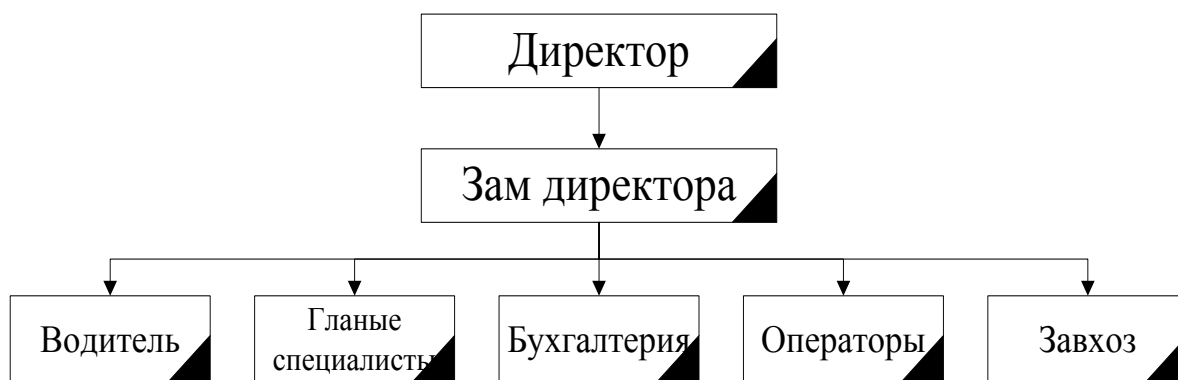


Рисунок 1 Организационно-управленческая структура предприятия¹

Предприятие состоит из:

- Директора – руководителя предприятия, который руководствуется уставом предприятия;
- Заместителя директора – Заместитель директора относится к категории руководителей, подчиняется, непосредственно, директору;

¹ Рисунок составлен в ходе исследования

- Главных специалистов – Эксперты филиала МФЦ относятся к категории специалистов и непосредственно подчиняются директору муниципального казенного учреждения Туапсинского района «Многофункциональный центр предоставления государственных и муниципальных услуг в Туапсинском районе.

- Бухгалтерии – Должность главного бухгалтера относится к категории специалистов, в своей деятельности руководствуется Конституцией РФ, федеральными законами, постановлениями Правительства РФ, инструктивными материалами Министерства финансов РФ и Управления финансов Краснодарского края, законодательством Краснодарского края, нормативными правовыми актами Собрания представителей Туапсинского района и приказами директора МФЦ, настоящей инструкцией.

- Операторов – Оператор филиала МФЦ относится к категории специалистов и непосредственно подчиняется директору муниципального казенного учреждения «Многофункциональный центр предоставления государственных и муниципальных услуг в Туапсинском районе»

- Водителя – водитель подчиняется заместителю директора и непосредственно директору.

- Технического персонала (завхоз) – технический персонал состоит из уборщиц. Подчиняются заместителю директора.[17]

2. Методический раздел

2.1 Анализ существующих разработок по автоматизации выбранной задачи

Автоматизированное рабочее место (АРМ) — представляет собой программно-технический комплекс, который позволяет автоматизировать деятельность определенного вида автоматизированной системы. Как правило, АРМ является частью автоматизированной системы управления.

АРМ объединяет программно-аппаратные средства, обеспечивающие взаимодействие человека с компьютером, предоставляет возможность ввода информации и её вывод на экран монитора, принтер, графопостроитель, звуковую карту — динамики или иные устройства вывода.

На выбранном предприятии функция подготовки документов для замены/получения паспорта не осуществляется. Но в других МФЦ для автоматизации данного процесса используется, описанная ранее, информационная система «АИС МФЦ»

АИС «МФЦ» - это программный комплекс, который направлен на автоматизацию основных административно-управленческих процессов, происходящих в рамках МФЦ, и обеспечивающий эффективную информационную поддержку его деятельности.

Применение АИС «МФЦ» в качестве средства информатизации деятельности МФЦ:

- повышает информационную прозрачность и открытость органов государственной власти и органов местного самоуправления;
- повышает осведомленность граждан и юридических лиц об услугах, оказываемых органами власти,
- повышает удобство и комфорт, а также снизить финансовые и временные затраты физических/юридических лиц при получении государственных и муниципальных услуг в соответствии с требованиями, закрепленными в административных регламентах соответствующих услуг,

- повышает эффективность межведомственного информационного обмена и взаимодействия в части обеспечения доступа к информации о деятельности органов исполнительной власти и предоставления государственных и муниципальных услуг несколькими органами власти,
- повышает производительность труда специалистов МФЦ при оказании государственных и муниципальных услуг,
- повышает эффективность контроля над ходом оказания государственных и муниципальных услуг. [5]

2.2 Выбор и обоснование стратегии автоматизации

Разрабатываемая информационная система для автоматизации процесса подготовки документации для замены/получения паспорта Российской Федерации будет представлять собой базу данных (БД), состоящую из нескольких таблиц. Для реализации данной работы, а также для подключения к ней пользовательского интерфейса, будет использоваться язык Java.

Java — объектно-ориентированный язык программирования, разработанный компанией Sun Microsystems. Не зависимо от компьютерной архитектуры приложения Java могут работать на абсолютно любой виртуальной Java-машине (Java Virtual Machine – JVM). Это программа, которая обрабатывает байтовый код и передаёт инструкции оборудованию как интерпретатор. Это связано с тем, что они транслируются в специальный байт-код.

Байт-код полностью независим от оборудования и операционной системы. Java-приложение может быть выполнено на любом устройстве, для которого существует соответствующая виртуальная машина. Эта же машина полностью контролирует исполнение программы, чем обеспечивает гибкую систему безопасности. Это является одним из важнейших достоинств подобного способа выполнения программ и главной отличительной особенностью технологии.

JVM вызывает немедленное прерывание работы программы в случае выполнения операций, которые превышают установленные полномочия.

К недостаткам концепции виртуальной машины часто относят снижение производительности. С этой проблемой призван бороться ряд усовершенствований, который несколько увеличил скорость выполнения программ на Java. К этим усовершенствованиям относятся:

- применение ЛТ-технологии. Это технология трансляции байт-кода в машинный код непосредственно во время работы программы с возможностью сохранения версий класса в машинном коде,
- широкое использование платформенно-ориентированного кода (native-код) в стандартных библиотеках,
- аппаратные средства, обеспечивающие ускоренную обработку байт-кода.

Множество людей, вдохновившихся виртуальной машиной Java, стали расширять перечень языков, которые могли бы быть использованы для создания программ, исполняемых на виртуальной машине. [12, 208]

Для отображения процесса автоматизации были построены несколько диаграмм с использованием CASE-средств. Для наглядного представления был выбран Rational Rose т.к. данное CASE-средство предназначено для отображения процесса автоматизации, используя язык UML.

UML — это графический язык, который позволяет описывать параметры документирования и конструирования при помощи диаграмм. Диаграммы можно создавать при помощи специальных CASE средств, например, Rational Rose и Enterprise Architect. Единая информационная модель строится на основе технологии UML. Некоторые CASE средства, такие как, например, приведенные выше, способны генерировать код на различных объектно-ориентированных языках, а также обладают очень полезной функцией реверсивного инжиниринга. [14, 736]

Rational Rose - CASE-средство позволяющее автоматизировать этапы анализа и проектирования ПО, а также способно генерировать код на различных языках и выпуск проектной документации.

Принципиальное отличие Rational Rose от других средств заключается в объектно-ориентированном подходе. Графические модели, которые создаются в этой среде, основаны на объектно-ориентированных принципах и языке UML (Unified Modeling Language). Создание целостной архитектуры процессов предприятия, сохраняя все взаимосвязи и управляющие воздействия между различными уровнями иерархии, становится возможным благодаря инструментам моделирования Rational Rose. [2]

Благодаря использованию Rational Rose появляется возможность максимально приблизить представления различных специалистов, которые осуществляют моделирование бизнес процессов и работают с моделями к одному общему стандартизированному виду. Так происходит благодаря объектному принципу моделирования. Это является основным преимуществом данного CASE средства. Помимо этого, есть и преимущества, обусловленные удобством работы с программным пакетом Rational Rose.

Преимуществами Rational Rose являются:

- Поддержка командной работы.
- Управление моделями.
- Контроль ошибок.
- Документирование моделей.
- Управление конфигурацией.

Для отображения процесса автоматизации было разработано несколько показательных диаграмм, а именно: диаграмму прецедентов, диаграмму последовательности действий, диаграмма вариантов использования разрабатываемой системы и диаграмму взаимодействия.

Процесс проектирования БД может быть представлен в виде модели бизнес-процессов. Бизнес-модель процесса проектирования позволяет:

- отобразить субъективное мнение проектировщика баз данных на процесс проектирования конкретной БД;
- учесть особенности ИТ-проекта, в рамках которого проектируется база данных;
- достаточно быстро составить план проектирования конкретной БД;
- просчитать длительность проектных работ (создать временную модель проектирования).

Для этого используются нотации IDEF0 и IDEF3.

Описание процессов моделирования верхнего уровня обычно принято выполнять в нотации IDEF0. При помощи данной нотации появляется возможность описать всю деятельность компании. Отличительная особенность нотации – это возможность отображения «управления» и «механизмов» каждого блока. Основана данная нотация на методологии SADT.

SADT (Structured Analysis and Design Technique, технология структурного анализа и проектирования) – это графические обозначения и подход к описанию систем. Самыми популярными разновидностями семейства IDEF, из тех, что применяются в бизнесе, являются нотации IDEF0 и IDEF3. Главной особенностью нотации является возможность декомпозиции, т.е. возможность разбивать каждый отдельный блок на отдельные процессы. [1]

Несмотря на наличие дополнительных свойств, в виде «управления» процессом, нотация IDEF0 остается статичной и не способна отразить, как именно меняется ход выполнения процесса под воздействием этого самого «управления».

Одним из главных недостатков нотации IDEF0 является ограниченное количество блоков на схеме, как правило, не превышает 9. Из-за чего особо крупные процессы приходится дробить на несколько диаграмм из-за того, что такого количества описываемых процессов превышает допустимые 9.

Во время построения процессов в нотации IDEF0 рекомендуется рисовать блоки в порядке доминирования: от самых важных до второстепенных, а не в очередности их выполнения. Но это правило не является обязательным, и, зачастую, многие специалисты пренебрегают им.

Нотация IDEF0 является одной из самых популярных среди консультантов и специалистов в сфере управленческого консалтинга, несмотря на рассмотренные выше недостатки.

Самым известным российским продуктом, поддерживающим построение процессов в нотации IDEF0, является Business Studio, поддержку данной нотации имеет также Microsoft Visio. [11]

Нотация IDEF3 чаще применяется для построения процессов нижнего уровня, могут также использовать при декомпозиции блоков процесса IDEF0. Нотация IDEF3 не поддерживает отображение «механизмов» и «управления», но при этом отображает очередность выполнения работ персоналом. Это является одним из главных отличий от IDEF0. IDEF3 очень схожа с нотацией FlowChart, но при этом имеет ряд существенных отличий.

- Весь процесс строится слева направо и при этом, как правило, ограничен количеством используемых блоков на одну диаграмму.
- Нотация изначально предназначалась для технических специалистов, поэтому содержит специальные перекрёстки, такие как, «XOR», «Synchronous OR», «Asynchronous OR», «Synchronous AND» и «Asynchronous AND», знакомые программистам, но требующие дополнительное пояснения менеджерам предприятия.

Нотация IDEF3, в отличие от IDEF0, не получила такого широкого распространения, большинство самых популярных программных продуктов

предпочли использовать альтернативные нотации для отображения процессов нижнего уровня. [13]

Для описания процесса проектирования баз данных используется нотация IDF0, так как она чаще всего, используется для описания процессов верхнего уровня. Так же позволяет описать всю деятельность компании.

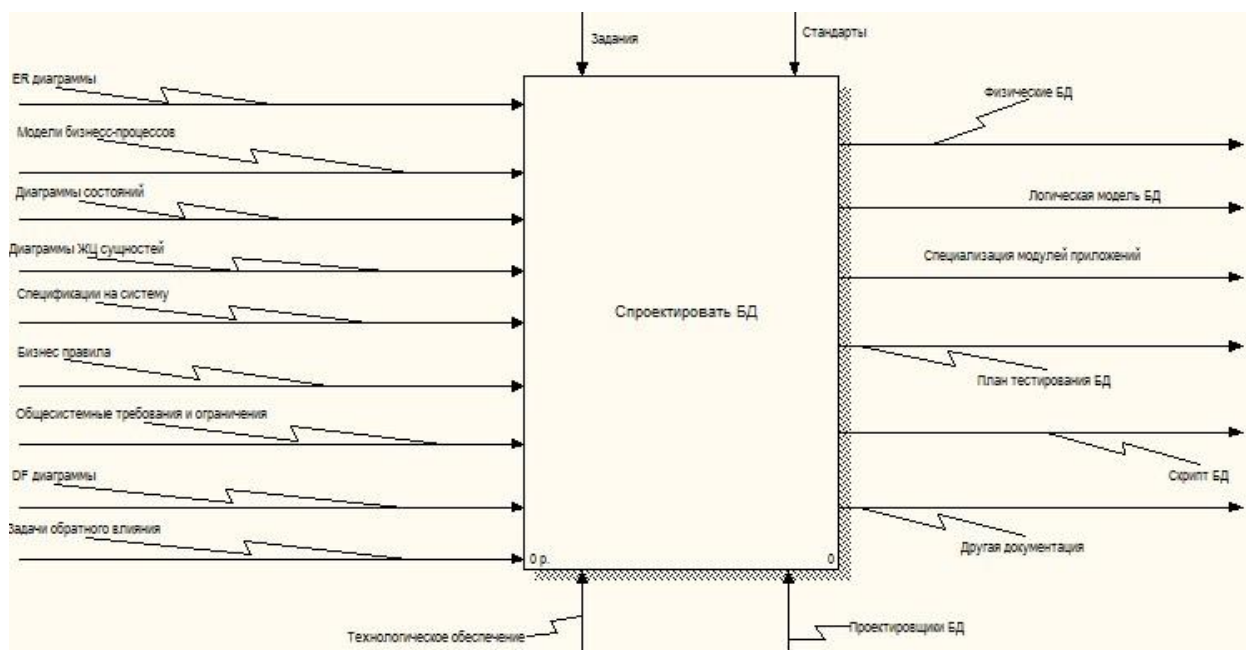


Рисунок 2 Контекстная диаграмма процесса проектирования базы данных²

Процесс проектирования баз данных наглядно представлен на рисунке 2. На вход подаются следующие данные:

- ER –диаграммы. Данные диаграммы иллюстрируют информационную модель предметной области базы данных;
- Функциональная модель предметной области базы данных. В её состав входят DF-диаграммы (диаграммы потока данных), бизнес-модель процессов, диаграммы жизненных циклов сущностей, требования и бизнес-правила;

² Рисунок составлен в ходе исследования

- Общесистемные требования и ограничения;
- Задачи обратного влияния.

Так же по необходимости можно представить и другие необходимые документы.

В результате на выходе процесса проектирования базы данных можно получить:

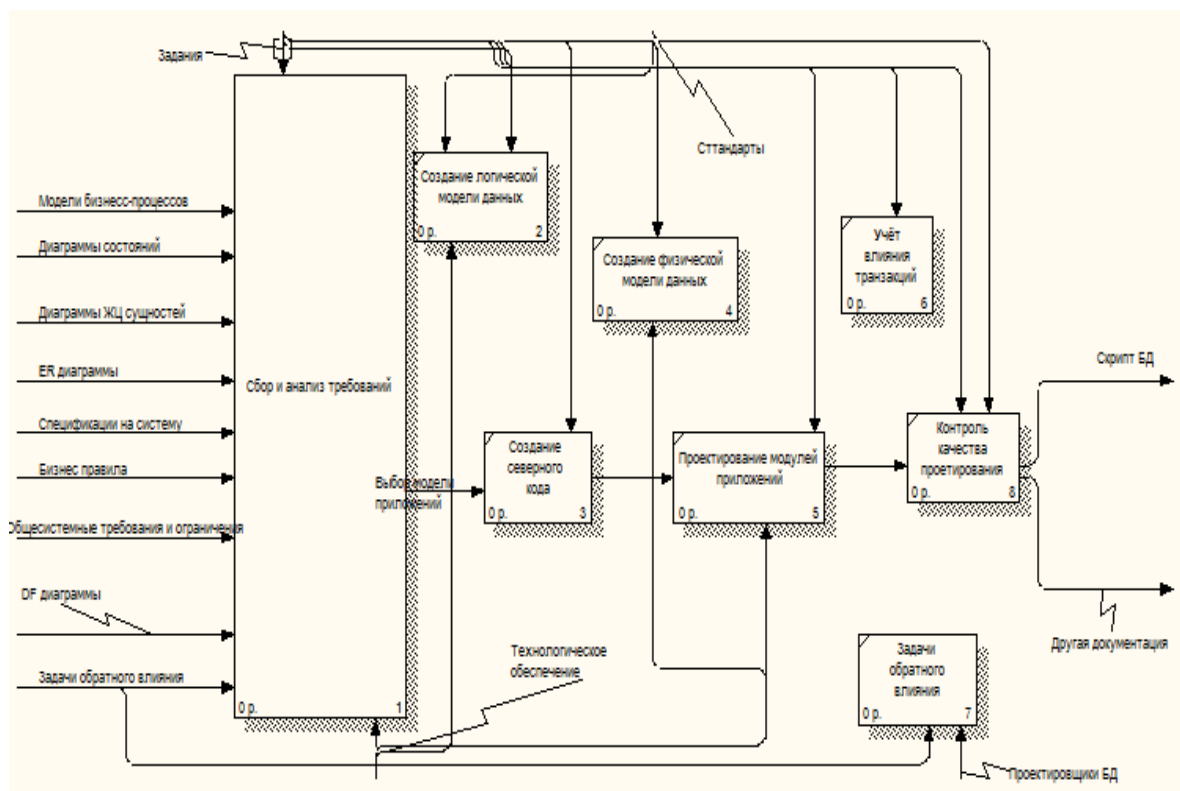
- Физическую модель БД;
- Физическую БД;
- Спецификацию модулей приложений БД;
- План тестирования БД.

Как и на входе, может быть разработана и другая необходимая документация.

Процесс проектирования БД можно декомпозировать на несколько процессов разного уровня. На рисунке 3 изображена диаграмма декомпозиции процесса проектирования базы данных первого уровня, которая отражает основные этапы проектирования, которые являются наиболее крупными и профессиональными.

Таковыми задачами являются:

- Сбор и анализ входных данных;
- Создание логической модели БД;
- Создание физической модели БД: внутренняя схема;
- Проектирование модулей приложений БД;
- Контроль качества проектирования БД;
- Задачи обратного влияния.



**Рисунок 3 Диаграмма декомпозиция процесса проектирования
базы данных: первый уровень³**

Диаграммой декомпозиции процесса проектирования БД второго уровня является декомпозиция этапа сбора и анализа входных данных (рисунок 4). Она изображает основные задачи, которые необходимо выполнить для реализации данного этапа.

³ Рисунок составлен в ходе исследования

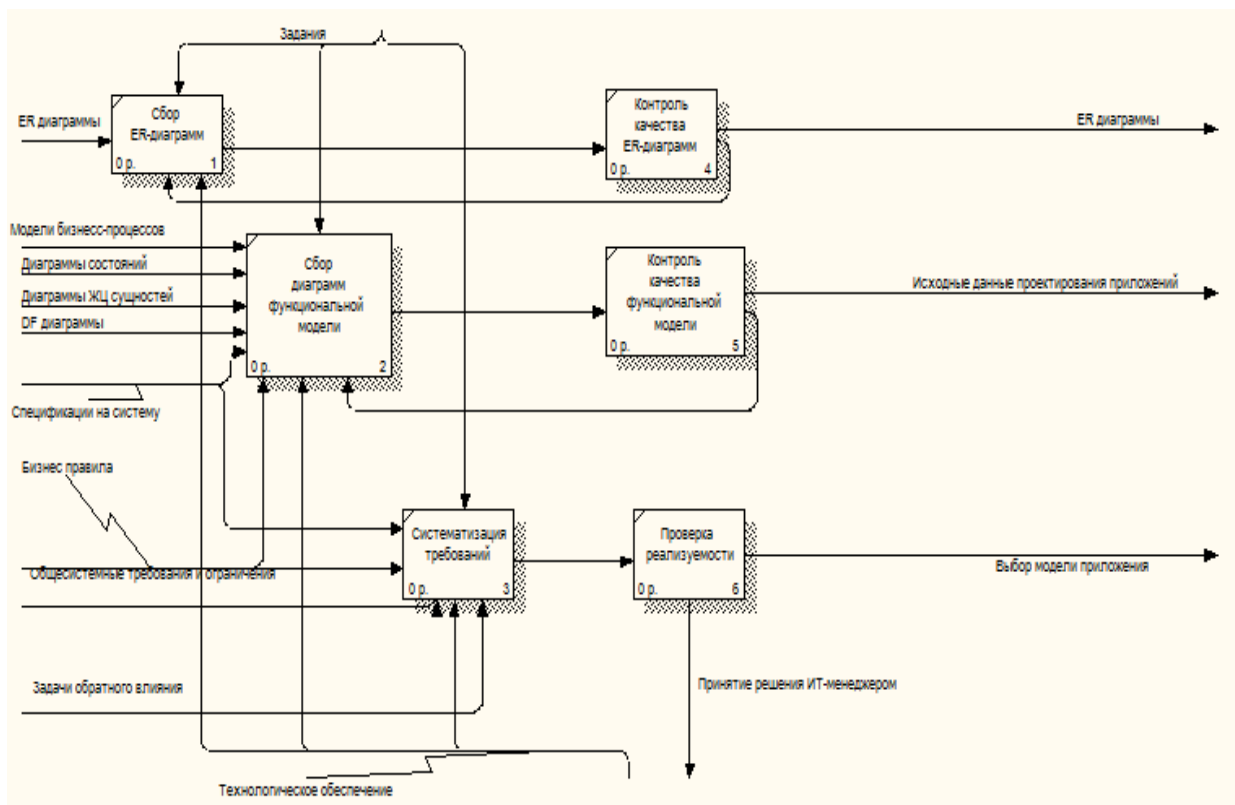


Рисунок 4 Диаграмма декомпозиции процесса проектирования базы данных: второй уровень. Сбор и анализ входных данных⁴

Таковыми задачами являются:

- Сбор документации с результатами анализа предметной области БД в виде диаграмм, спецификаций и требований;
- Контроль качества результатов анализа предметной области БД;
- Систематизация требований и спецификаций заказчика к базе данных;
- Подготовка плана проектирования БД.

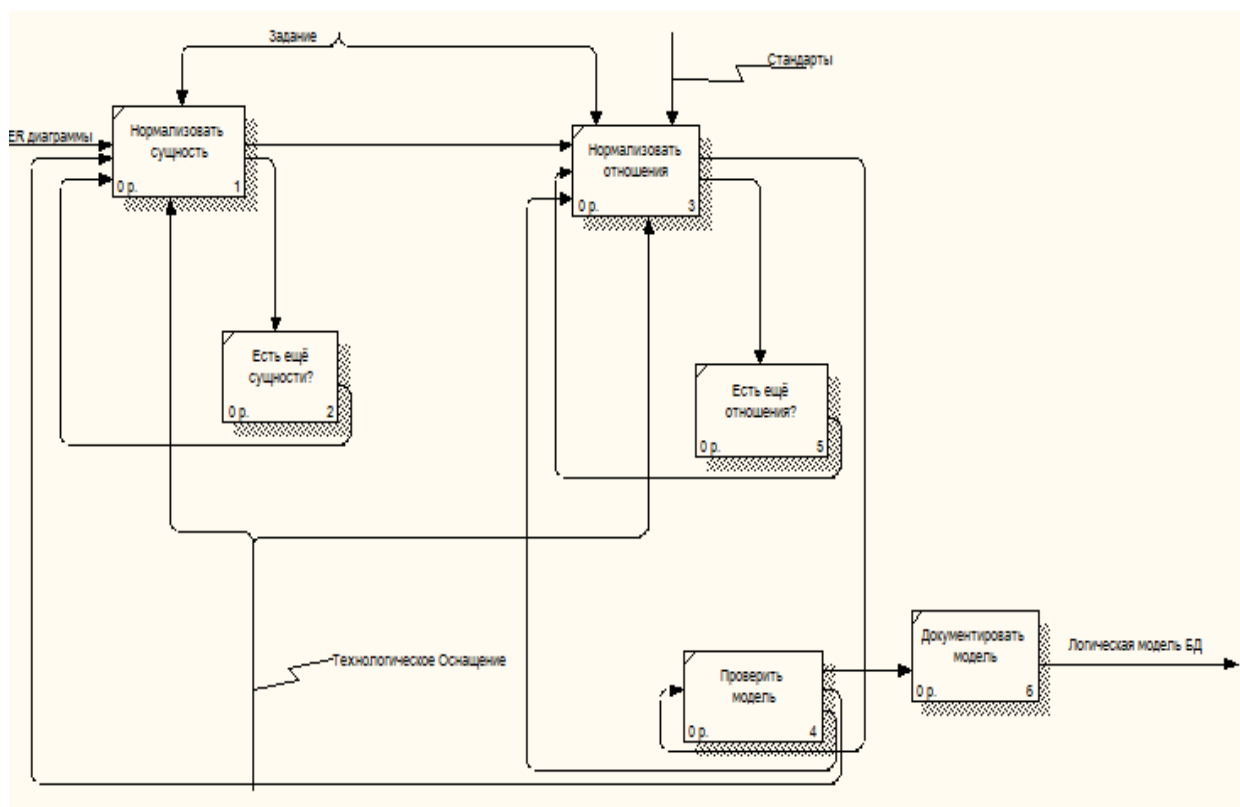
Логическая модель БД создаётся в рамках второго этапа функциональной декомпозиции процесса проектирования реляционной БД.

Основная цель этапа создания логической модели БД заключается в преобразовании информационной модели предметной области БД в логическую модель реляционной БД.

⁴ Рисунок составлен в ходе исследования

В результате создания логической модели БД можно получить нормализованную схему отношений БД.

Следующий процесс проектирования БД заключается в разработке бизнес-модели процессов создания логической модели БД, нормализации сущности предметной области и нормализации отношений логической модели БД соответственно. Данные действия представлены на следующих рисунках (рисунок 5-7).



**Рисунок 5 Бизнес-модель процесса создания логической модели
базы данных⁵**

⁵ Рисунок составлен в ходе исследования

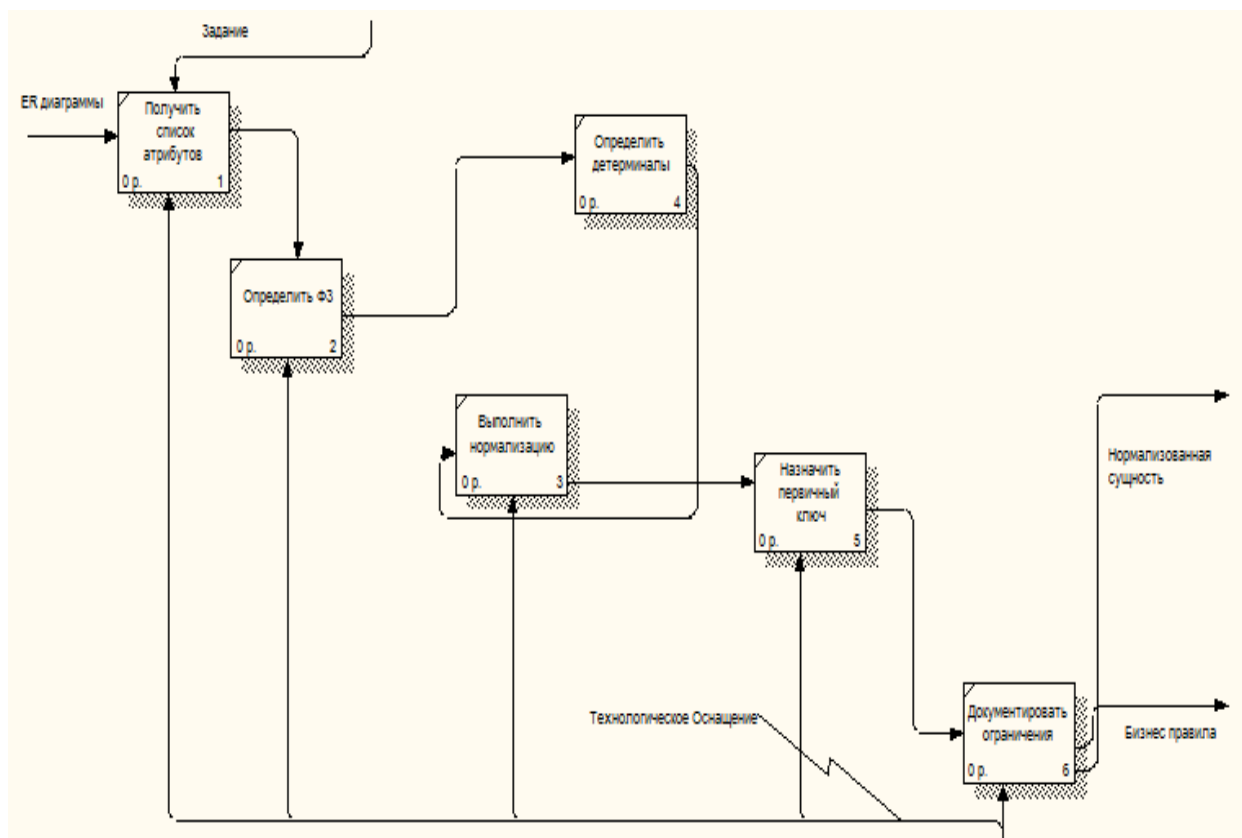


Рисунок 6 Бизнес-модель процесса нормализации сущности⁶

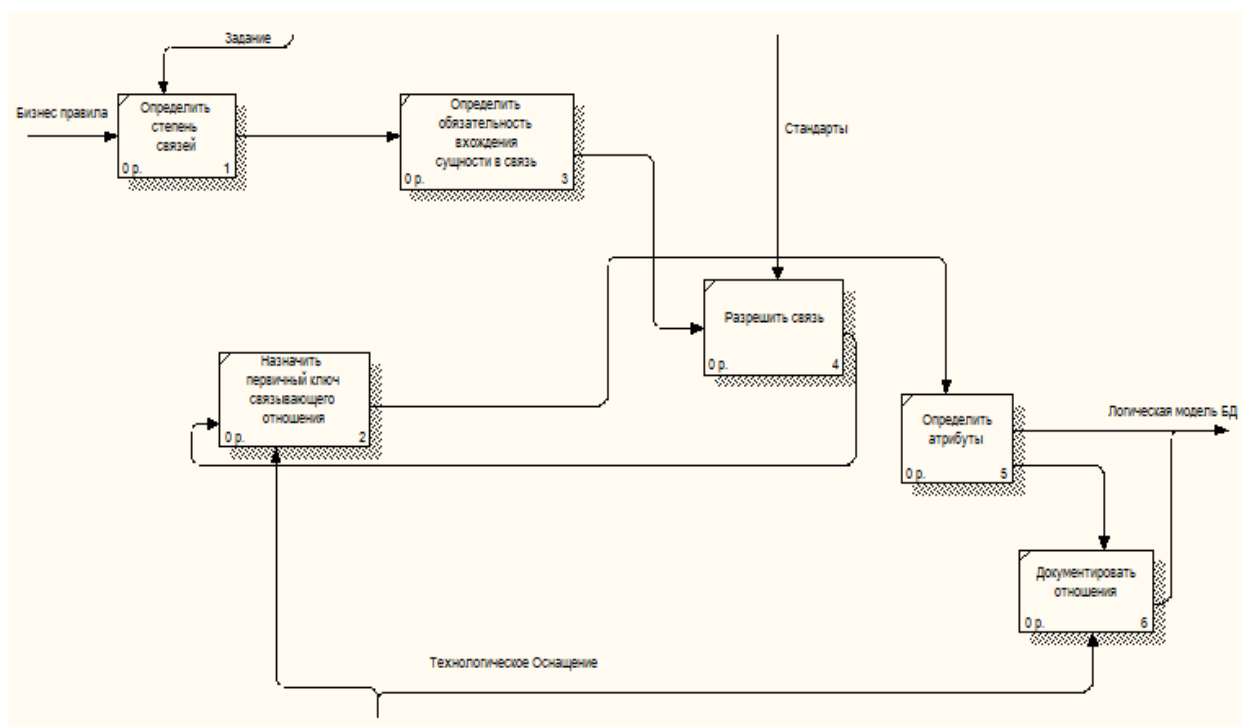


Рисунок 7 Бизнес-модель процесса нормализации отношения⁷

⁶ Рисунок составлен в ходе исследования

Создание первой итерации внутренней схемы БД является начальным этапом построения бизнес-модели этапа проектирования физической модели реляционной БД.

Модель бизнес-процессов первой итерации физической модели БД представлена на рисунке 8.

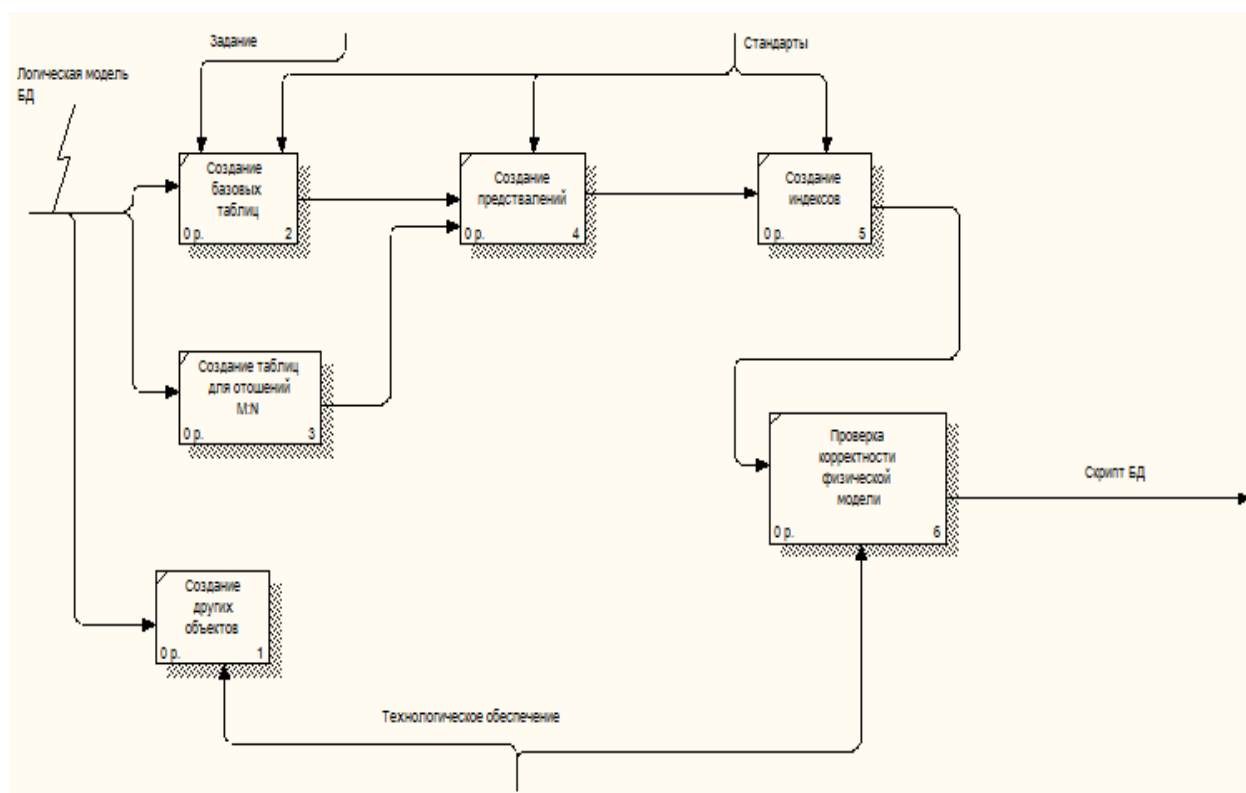


Рисунок 8 Декомпозиция этапа проектирования - создание первой итерации физической модели базы данных: внутренняя схема⁸

⁷ Рисунок составлен в ходе исследования

⁸ Рисунок составлен в ходе исследования

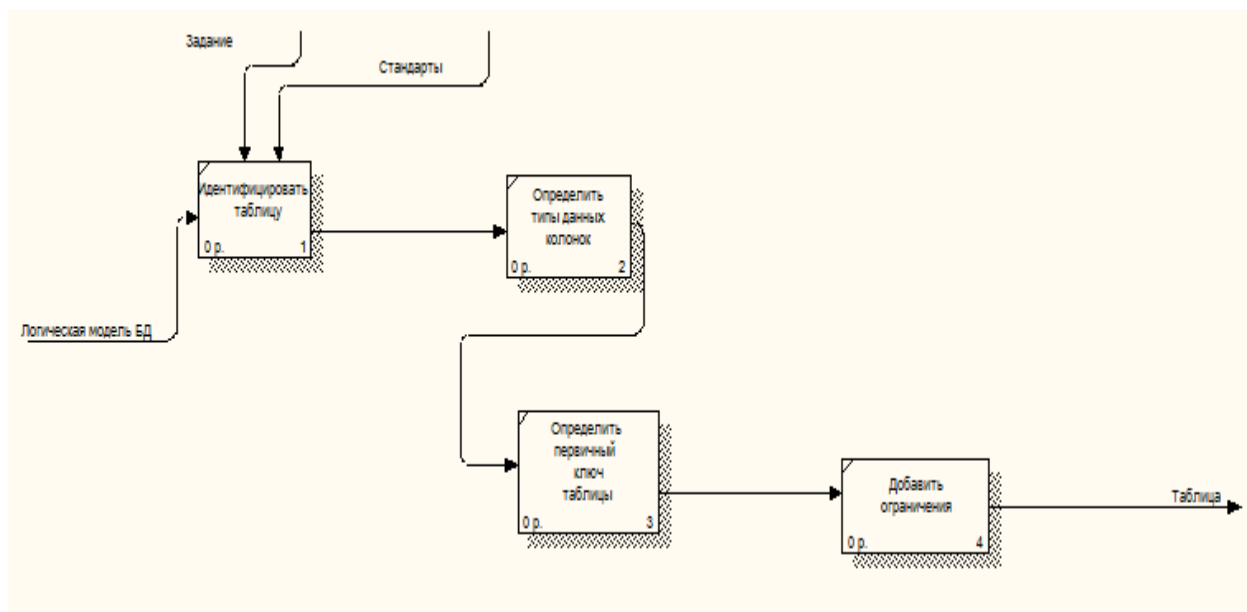


Рисунок 9 Декомпозиция работы по созданию базовой таблицы⁹

На рисунке 9 изображена декомпозиция работы по созданию базовой таблицы.

⁹ Рисунок составлен в ходе исследования

Глава 3. Проектирование и реализация информационной системы

3.1 Построение модели «Как должно быть»

Для демонстрации работы разрабатываемого автоматизированного рабочего места (далее АРМ) представлена диаграмма прецедентов.

Диаграмма прецедентов (англ. use case diagram, диаграмма вариантов использования) в UML — это диаграмма, которая призвана отражать отношения между актёрами и прецедентами. Диаграмма прецедентов является одной из частей модели прецедентов, которая позволяет описать систему на концептуальном уровне.

Прецедент — часть функциональности моделируемой системы, которая позволяет пользователю получить конкретный, измеримый и нужный ему результат. Прецедент позволяет определить один из вариантов использования проектируемой системы и описывает типичный способ взаимодействия пользователя с системой. Варианты использования обычно применяются для спецификации внешних требований к системе. [8]

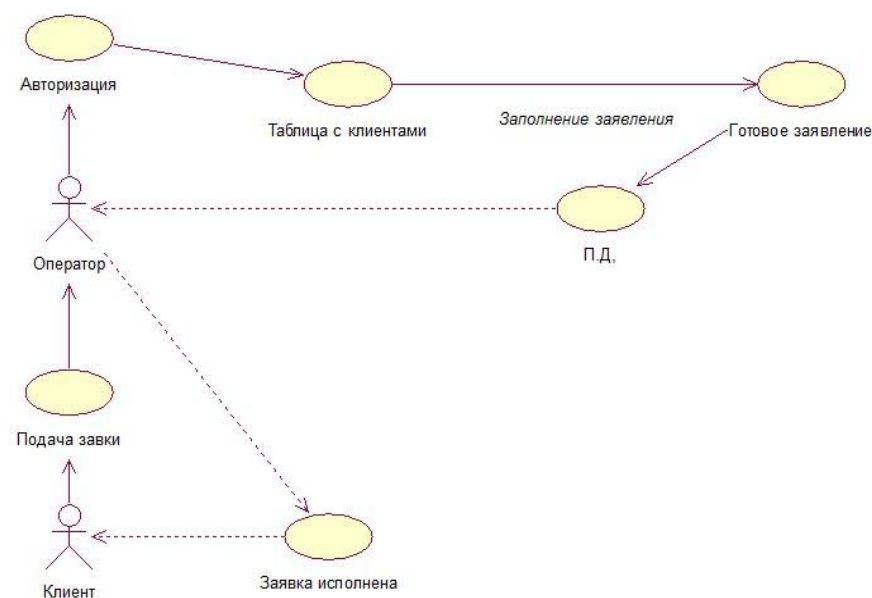


Рисунок 10 Диаграмма прецедентов «Общая схема»¹⁰

¹⁰ Рисунок составлен в ходе исследования

На рисунке 10 изображен следующий процесс:

- Клиент подаёт заявку о замене/получении паспорта оператору;
- Оператор выполняет авторизованный вход в систему;
- Оператор обращается к таблице с клиентами. Данная таблица хранит данные о посетителях МФЦ. В неё вносятся данные о посетителе.
- Оператором заполняется заявление;
- Оператор собирает готовый комплект документов, исходя из причины замены/получения паспорта РФ:

1. Утрата/хищение паспорта;
2. Достижение 14-летнего возраста;
3. Приобретения гражданства Российской Федерации;
4. Изменение имени, фамилии, отчества, сведений о дате и месте рождения (Смена персональных данных);
5. Достижение 20/45-летнего возраста;
6. Изменение пола;
7. Изменение внешности;
8. Обнаружение ошибочности или неточности произведенных записей;
9. Непригодность для дальнейшего использования.

Каждой причине соответствует набор необходимых документов для получения паспорта.

- Оператор распечатывает необходимые бланки и передаёт их клиенту.

Теперь, когда видна общая схема работы АРМ, можно более детально рассмотреть её структуру. Для этого используется диаграмма вариантов использования разрабатываемой информационной системы (Рисунок11).

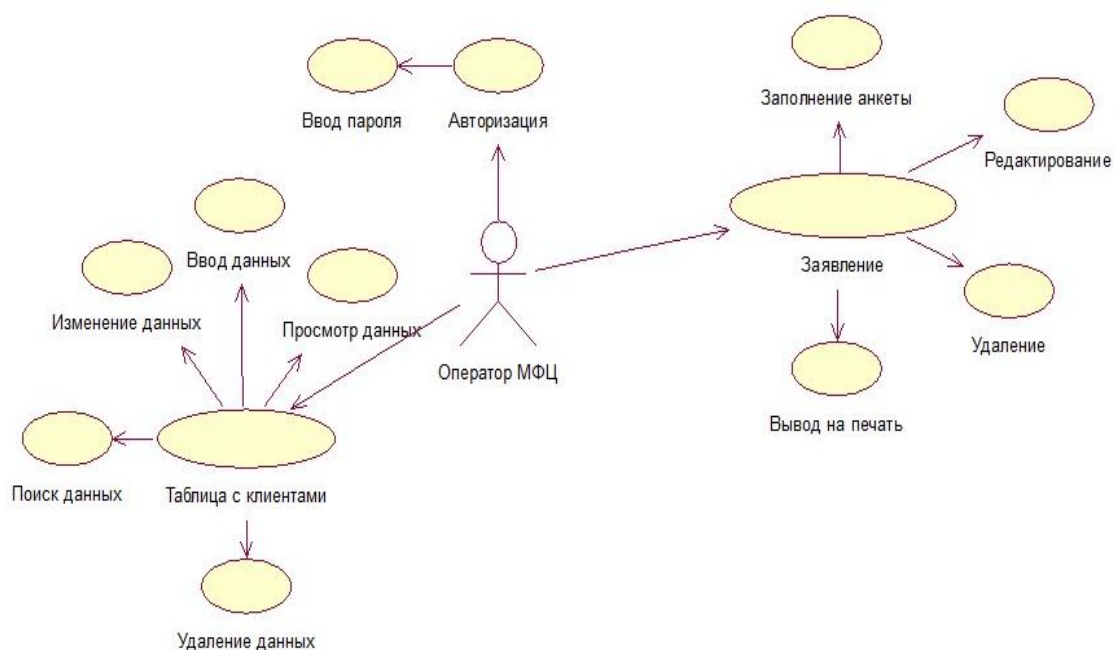


Рисунок 11 Диаграмма вариантов использования разрабатываемой системы¹¹

На диаграмме вариантов использования можно рассмотреть, какие процессы могут выполняться внутри системы.

Оператор начинает работу с процесса авторизации. В данном случае выбран сложный пароль, изменять который может только администратор системы.

Далее оператор переходит к работе с таблицей клиентов. Данная таблица является обязательным элементом системы, так как хранит в себе данные обо всех посетителях многофункционального центра. У данной таблицы есть поисковая система, которая позволяет облегчить поиск клиента в случае, если он уже посещал МФЦ раньше. Так же в эту таблицу можно вносить новые данные, подвергать изменениям уже существующие и удалять ненужные. Каждому посетителю в этой таблице присваивается индивидуальный номер, который связывает его с остальными элементами системы.

¹¹ Рисунок составлен в ходе исследования

После внесения изменений в таблицу клиентов, оператор приступает к заполнению заявления. Заявление заполняется в виде анкеты. В процессе так же можно редактировать и удалять данные. После заполнения заявления, оно выводится на печать. На этом обращение к системе можно считать законченным.

Для наглядной демонстрации того, какие действия выполняет тот или иной участник АРМ, используется диаграмма последовательности действий.

Диаграмма последовательности действий — диаграмма, отражающая упорядоченное по времени взаимодействие объектов, отражая продолжительность обработки и последовательности их проявления.

Основными элементами диаграммы последовательности являются обозначения объектов (прямоугольники с названиями объектов), вертикальные «линии жизни» (англ. *lifeline*), отображающие течение времени, прямоугольники, отражающие деятельность объекта или исполнение им определенной функции (прямоугольники на пунктирной «линии жизни»), и стрелки, показывающие обмен сигналами или сообщениями между объектами. [8]

На данной диаграмме объекты располагаются слева направо (рисунок 12).



Рисунок 12 Диаграмма последовательности действий¹²

Процесс, изображенный на рисунке 12 можно описать следующим образом:

- Посетитель подаёт заявку о замене/получении паспорта оператору;
- Оператор вводит данные посетителя в таблицу клиентов;
- После заполнения таблицы оператор запускает программу формирования шаблона, которая заполнит существующий шаблон данными из таблицы;
- Оператор получает заполненный шаблон, распечатывает его вместе с необходимыми недостающими документами (например, квитанцией об оплате);
- Оператор передаёт готовый пакет документов посетителю.

3.2 Построение блок-схем алгоритмов.

Для наиболее полного описания функционирования разрабатываемой системы используются блок-схемы алгоритмов (Рисунок 13, 14).

¹² Рисунок составлен в ходе исследования

Блок-схема — графический способ описания алгоритмов или процессов, удобный для восприятия человеком.

Блок-схемы строго структурированы, каждый отдельный процесс должен быть представлен в виде блоков различной формы, которые должны быть соединены между собой линиями, указывающими направление последовательности.

Для пояснения некоторых сокращений вводится таблица 1, которая относится ко всем блок-схемам.

Таблица 1 - Таблица сокращений, применяемых в блок-схемах.¹³

Сокращение	Пояснение
МФЦ	Многофункциональный центр
БД	База данных клиентов, применяемая в МФЦ для учёта посетителей
АРМП	Кодовое название разрабатываемой системы (Автоматизированное Рабочее Место Паспортного контроля)

¹³ Таблица составлена в ходе исследования

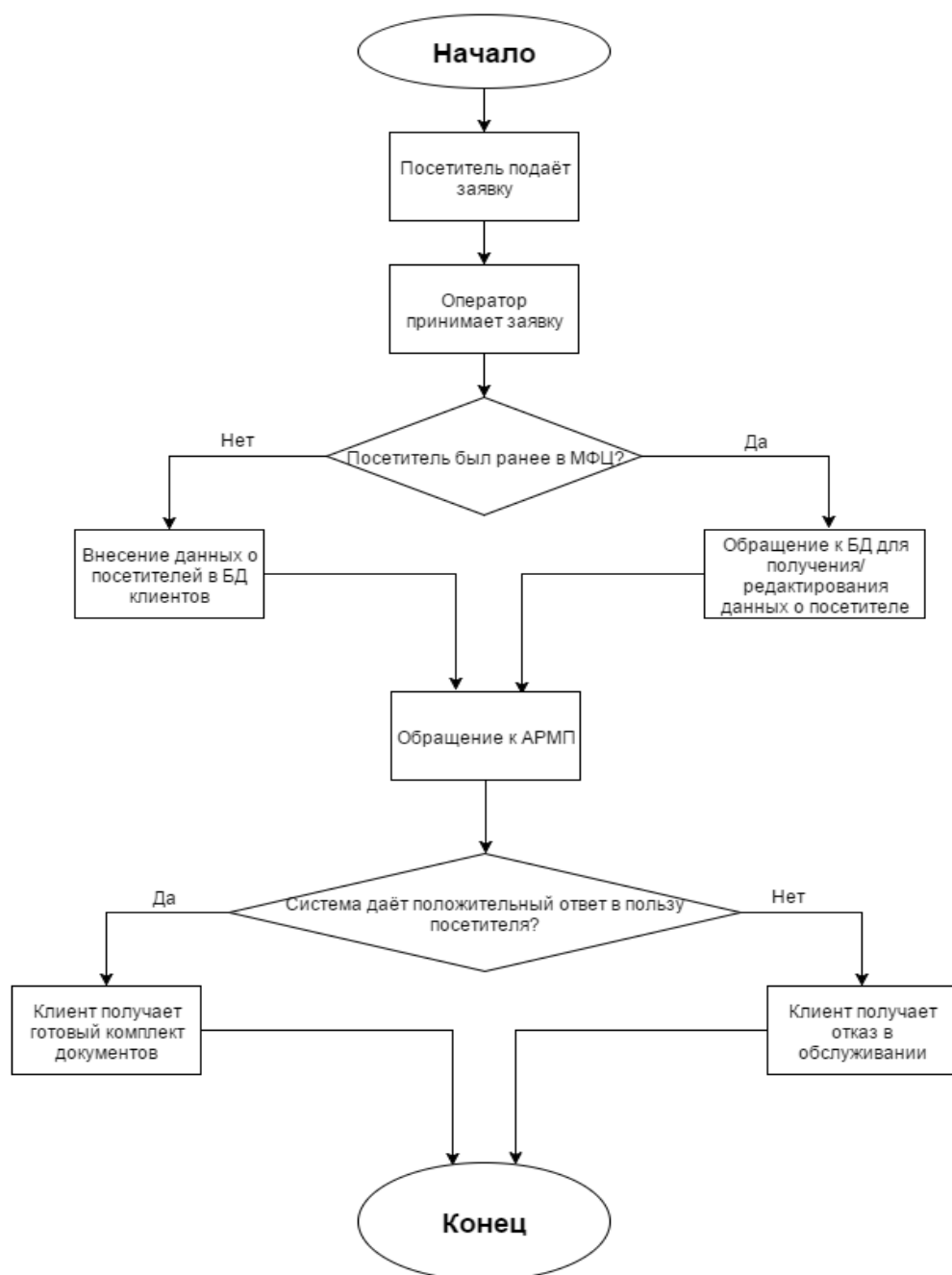


Рисунок 13 Блок-схема алгоритма взаимодействия посетителя МФЦ, оператора и системы¹⁴

¹⁴ Рисунок составлен в ходе исследования

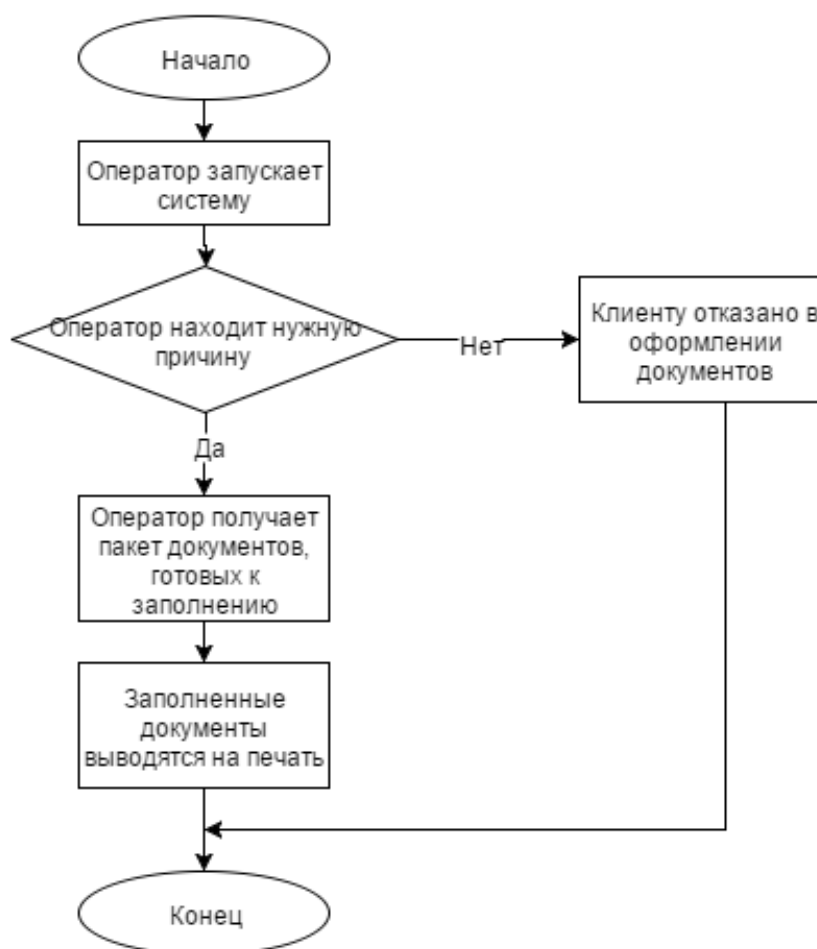


Рисунок 14 Блок-схема алгоритма стандартной работы разрабатываемой системы¹⁵

3.3 Программная реализация

Программная реализация АРМ была выполнена на языке Java. Для разработки была использована среда NetBeans IDE 8.1. При помощи IDE NetBeans у разработчика появляется возможность быстро и легко разрабатывать настольные, мобильные и веб-приложения Java, а также приложения HTML5 с использованием технологий HTML, JavaScript и CSS. IDE также предоставляет многофункциональные наборы средств для разработчиков PHP и C/C++. Это бесплатное ПО с открытым исходным

¹⁵ Рисунок составлен в ходе исследования

кодом, которое имеет большое сообщество пользователей и разработчиков по всему миру. [4]

Данная система состоит из двух частей: клиентской и серверной. Рассмотрим клиентскую.

Система представляет собой три таблицы: персоны, документы и обращения. В каждой из таблиц есть свой первичный ключ: ID_USER, ID_DOCUMENT, ID_ORDER. Все таблицы привязаны к таблице пользователей, то есть все ID_USER привязаны к записям в таблице «Документы» связью многие к одному. Данная связь представлена на рисунке 15.

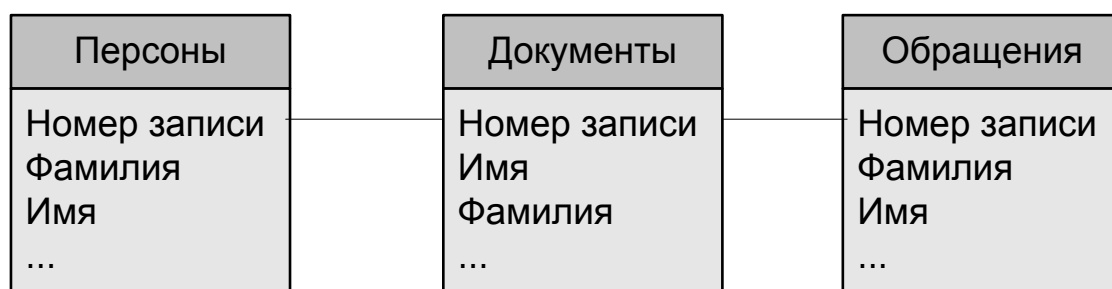


Рисунок 15 Связи таблиц информационной системы¹⁶

Суть данной связи заключается в том, что невозможно добавить документ или заявку пользователя, которого нет, и, если удалить пользователя, то удалятся все его заявки и обращения. Это обеспечивает достоверность и полноту информации в реляционном представлении.

¹⁶ Рисунок составлен в ходе исследования

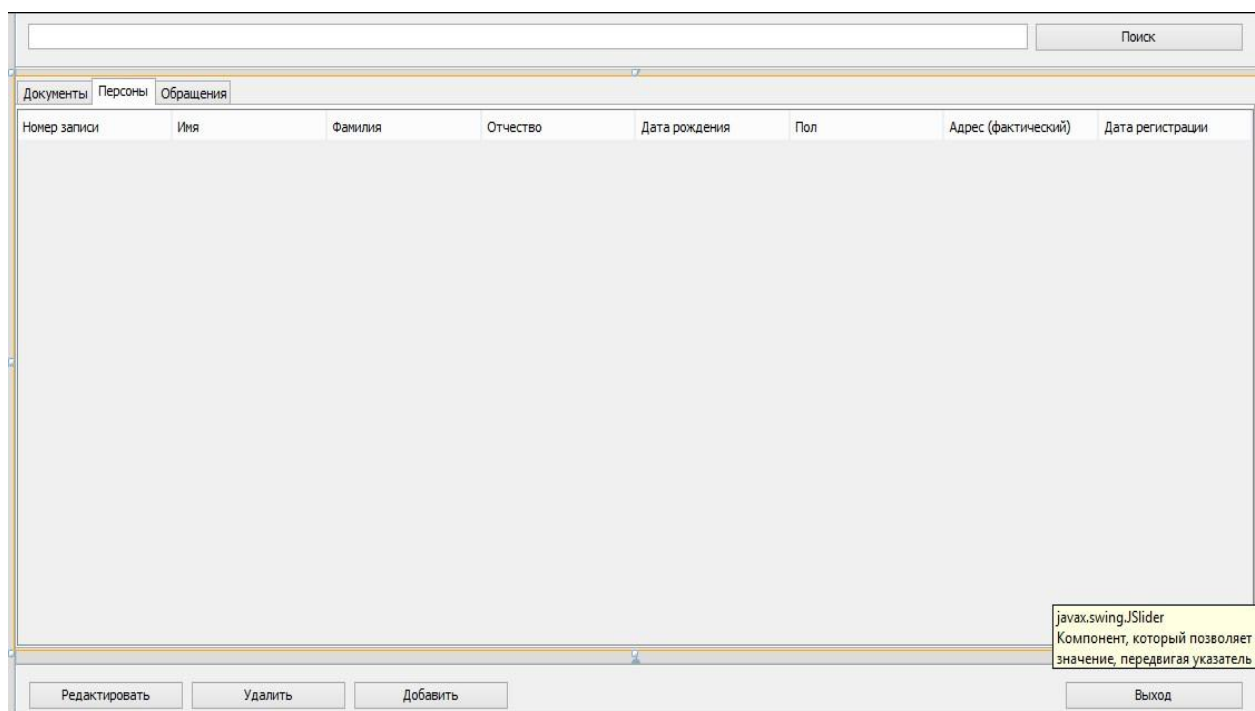


Рисунок 16 Таблица: Персоны¹⁷

На рисунке 16 представлен интерфейс таблицы «Персоны».

Таблица 2 - Типы данных таблицы «Персоны»¹⁸

Имя поля	Тип данных
Номер записи	Счётчик
Имя	Короткий текст
Фамилия	Короткий текст
Отчество	Короткий текст
Дата рождения	Дата и время
Пол	Короткий текст
Адрес (фактический)	Короткий текст
Дата регистрации	Дата и время

¹⁷ Рисунок составлен в ходе исследования

¹⁸ Таблица составлена в ходе исследования

Данная таблица является аналогом информационного киоска. При внедрении АРМ в предприятие, она станет частью общей системы, в которой реализуется данная функция. В данном случае, эта таблица нужна для того, чтобы отметить человека, посещающего предприятие. Данные вводятся со слов посетителей без подтверждения документами, удостоверяющими личность. Состоит из следующих полей:

- Номер записи;
- Имя;
- Фамилия;
- Отчество;
- Дата рождения;
- Пол;
- Адрес (фактический);
- Дата регистрации – дата посещения МФЦ.

Для того, чтобы внести данные из документов, удостоверяющих личность посетителя, нужно обратиться к таблице «Документы» (рисунок 17).

Рисунок 17 Таблица: Документы¹⁹

Таблица 3 - Типы данных таблицы «Документы»²⁰

Имя поля	Тип данных
Номер записи	Числовой
Имя	Короткий текст
Фамилия	Короткий текст
Отчество	Короткий текст
Тип документа	Короткий текст
Серия и номер	Числовой
Дата рождения	Дата и время
Место рождения	Короткий текст
Пол	Короткий текст
Дата сохранения	Дата и время

¹⁹ Рисунок составлен в ходе исследования

²⁰ Таблица составлена в ходе исследования

Таблица «Документы» хранит в себе данные о посетителе. В отличие от таблицы «Персоны», здесь данные вносятся, основываясь на документах, удостоверяющих личность клиента. В данной таблице клиенту присваивается индивидуальный код, который связывает данные клиента в разных таблицах.

В данном случае она состоит из следующих полей:

- Номер записи;
- Фамилия;
- Имя;
- Отчество;
- Тип документа – тип документа, удостоверяющего личность.

Например, паспорт;

- Серия, номер – серия и номер документа, удостоверяющего личность;
- Дата рождения;
- Место регистрации;
- Пол;
- Дата сохранения – дата обновления данных о посетителе.

Именно эта таблица, в дальнейшем, будет основой для заполнения существующего шаблона заявления.

Таблица 3 - Типы данных таблицы «Документы»²¹

Имя поля	Тип данных
Номер записи	числовой
Причина обращения	Короткий текст
Имя	Короткий текст
Фамилия	Короткий текст
Отчество	Короткий текст
Дата обращения	Дата и время

²¹ Таблица составлена в ходе исследования

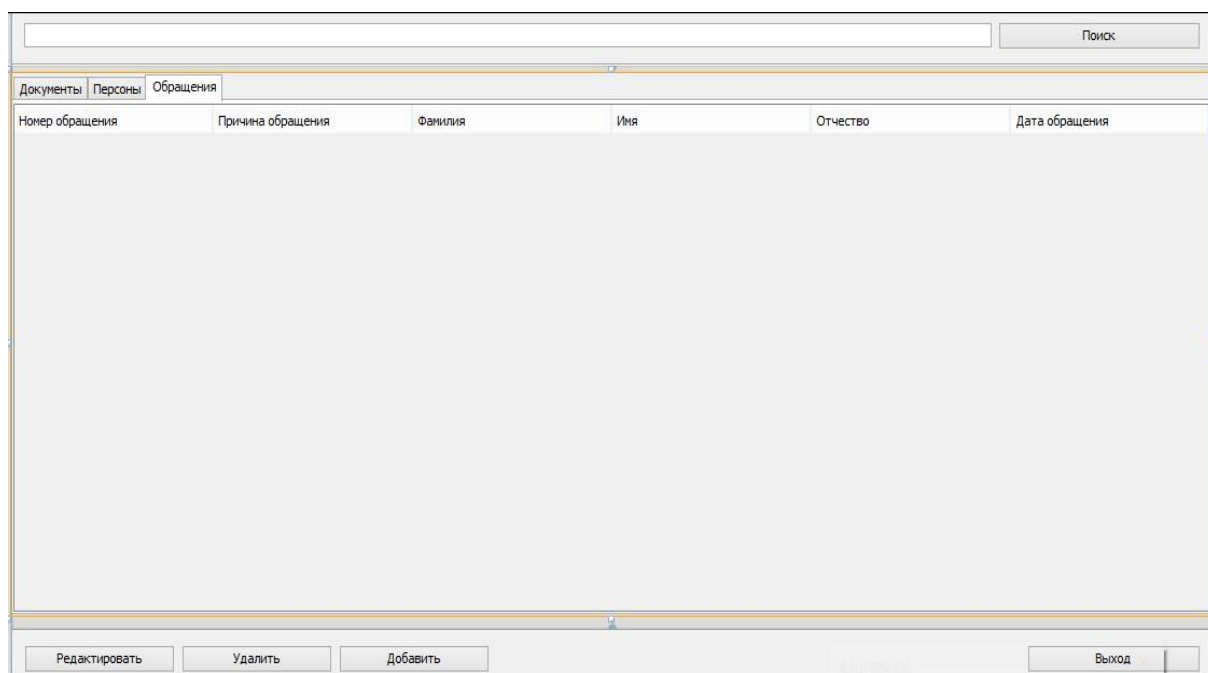


Рисунок 18 Таблица: Обращения²²

Таблица «Обращения» (рисунок 18) хранит в себе данные о всех обращениях и посетителях МФЦ. Она состоит из:

- Номера обращения;
- Причины обращения;
- Фамилии;
- Имени;
- Отчества;
- Даты обращения.

Каждую из таблиц можно редактировать, добавлять новые данные и удалять ненужные. Так же таблицы оснащены поиском, который значительно упрощает работу, в случае, когда посетитель уже обслуживался в МФЦ.

Перед началом работы с системой, оператор должен подключиться к серверу и авторизоваться (рисунок 19).

²² Рисунок составлен в ходе исследования

Подключение к базе данных

Адрес: 127.0.0.1 Название БД: DWP

Пароль: Имя пользователя: root

Подключиться

Рисунок 19 Авторизация в системе²³

Рассмотрим код разрабатываемой системы. После инициализации переменных необходимо выбрать метод подключения к базе данных (БД) и создать объект для работы ней (рисунок 20).

```
private void connectDatabase() {
    //Метод подключения к базе
    this.sql = new MySQL(); //Содаем объект для работы с БД
```

Рисунок 20 Метод подключения к базе данных²⁴

Далее необходимо задать формам параметры подключения (рисунок 21)

```
//Устанавливаем параметры подключения
//Берем их с формы
String ip = SettingsForm.AddressField.getText();
String username = SettingsForm.UsernameField.getText();
String password = String.valueOf(SettingsForm.PasswordField.getPassword());
String database = SettingsForm.DatabaseField.getText();

sql.SetConnectionSettings(ip, username, password, database);
```

Рисунок 21 Установка параметром подключения²⁵

²³ Рисунок составлен в ходе исследования

²⁴ Рисунок составлен в ходе исследования

²⁵ Рисунок составлен в ходе исследования

Следующим шагом делаем тестовый запрос к базе. Используя набор данных результата (result set) выполняем запрос. Так же прописываем условие неудачного подключения (рисунок 22).

```
private boolean testConnection(){
    //Тестовый запрос к базе
    //Получаем текст тестового запроса
    String currentQuery = this.query.GetTestQuery();
    //Выполняем запрос
    ResultSet rs = sql.GetResultSet(currentQuery);

    try {
        while (rs.next()){
            Integer count = rs.getInt(1);
            System.out.println("Test: " + count);
        }
        //Закрываем резалтсет
        rs.close();
    } catch (SQLException ex) {
        //Если запрос упадет или нет подключения, упадем сюда
        Logger.getLogger(Std_Antonenkova.class.getName()).log(Level.SEVERE, null, ex);
        return false;
    }
    return true;
}
```

Рисунок 22 Параметры тестового запроса²⁶

Теперь рассмотрим класс для работы с базой MySQL.Java. Он и подключается к БД, и выполняет запросы. В нём находятся вспомогательные методы, такие, например, как, рассмотренные ранее, преобразования result set в массив. Данный класс можно считать некой «прослойкой» между логикой и источником данных. Для начала следует задать параметры подключения (рисунок 23).

²⁶ Рисунок составлен в ходе исследования

```

public class MySQL {
    //Параметры подключения
    private String url;
    private String username;
    private String password;
    private String ip;
    private String database;

    //Соединение и стейтмент
    private Connection con;
    private Statement stmt;

    MySQL() {
        System.out.println("MySQL class initialize...");
    }
}

```

Рисунок 23 Параметры подключения²⁷

Далее необходимо описать отключение от СУБД (Системы управления баз данных) (Рисунок 24).

```

private boolean connect(){
    //Подключение к СУБД
    boolean connectResult = true;
    System.out.println("MySQL: connecting...");
    try {
        //Формируем строку подключения
        this.url = "jdbc:mysql://" + this.ip + "/" + this.database;

        //Подключаемся только если подключение не активно или ещё не подключались
        if (this.stmt != null) {
            if (this.stmt.isClosed()){
                this.con = (Connection) DriverManager.getConnection(this.url, this.username, this.password);
                this.stmt = (Statement) con.createStatement();
                System.out.println("MySQL: Connected!");
            }
        } else {
            this.con = (Connection) DriverManager.getConnection(this.url, this.username, this.password);
            this.stmt = (Statement) con.createStatement();
            System.out.println("MySQL: Connected!");
        }
    } catch (SQLException sqlEx) {
        System.out.println("MySQL connection error!" + sqlEx.getMessage());
        this.Disconnect();
        connectResult = false;
    }

    return connectResult;
}

```

Рисунок 24 Подключение к СУБД²⁸

²⁷ Рисунок составлен в ходе исследования

²⁸ Рисунок составлен в ходе исследования

```

public String[][] FetchStringArray(ResultSet rs){
    //Преобразует резалтлист в массив
    int columnCount = 0;
    int rowsCount = 0;

    ArrayList<ArrayList<String>> tableList = new ArrayList();

    try {
        //Получаем кол-во полей
        ResultSetMetaData metaData = rs.getMetaData();
        columnCount = metaData.getColumnCount();

        //Перебираем все строки резалтлиста
        while (rs.next()){
            //Итерируем кол-во строк
            rowsCount++;
            //Арэйлис со строкой
            ArrayList <String> row = new ArrayList();
            for (Integer i = 1; i <= columnCount; i++){
                row.add(rs.getString(i));
            }
            //Добавляем строку в арэйлист
            tableList.add(row);
        }
    } catch (SQLException ex) {
        Logger.getLogger(Controller.class.getName()).log(Level.SEVERE, null, ex);
        return null;
    }
}

```

Рисунок 25 Путь получения итогового массива²⁹

После всех вышеперечисленных преобразований можно получить итоговый массив (рисунок 25, 26).

```

//Итоговый массив
String[][] tableArray = new String[rowsCount][columnCount];

//Переливаем данные с арэйлиста в массив
for (int i = 0; i < rowsCount; i++){
    for (int j = 0; j < columnCount;j++){
        tableArray[i][j] = tableList.get(i).get(j);
    }
}

//Возвращаем то, что получилось
return tableArray;

```

Рисунок 26 Итоговый массив³⁰

²⁹ Рисунок составлен в ходе исследования

³⁰ Рисунок составлен в ходе исследования

В результате разработки мы получаем функционирующее АРМ. Рабочий интерфейс представлен на рисунке 27.

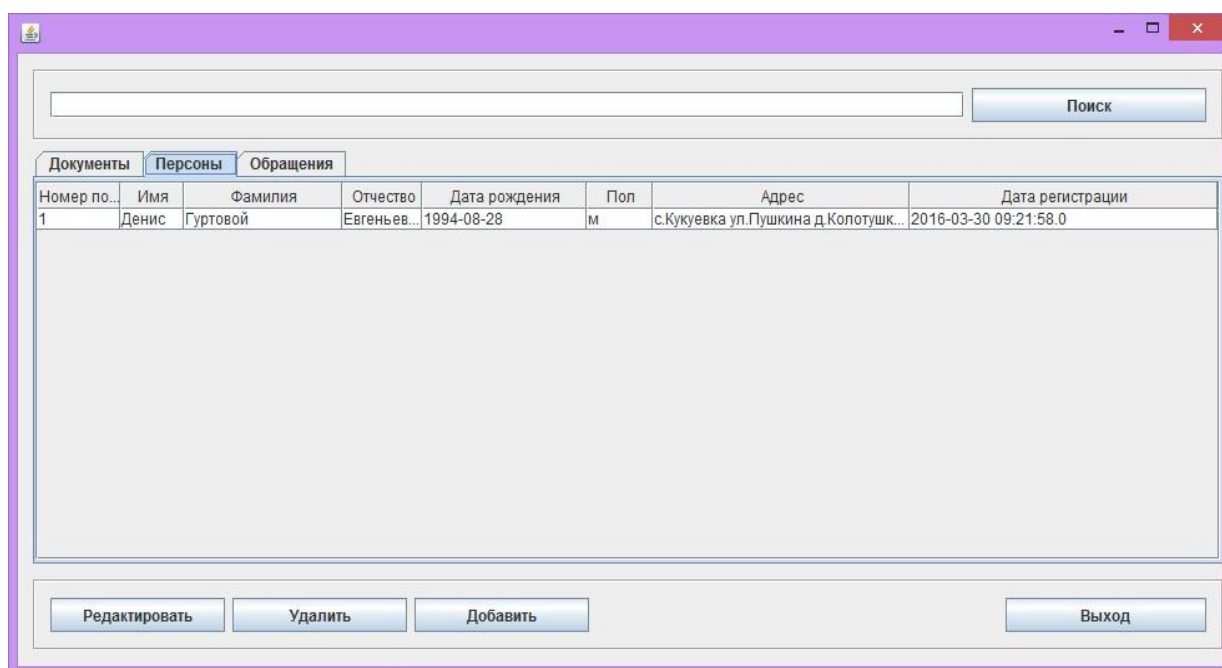


Рисунок 27 Рабочий интерфейс АРМ³¹

Интерфейс достаточно простой, позволяющий интуитивно разобраться в работе с АРМ. Таблицы оснащены поиском по ключевым словам, что облегчит работу оператора. Так же есть возможность добавлять данные, нажав на кнопку «Добавить».

³¹ Рисунок составлен в ходе исследования

Параметр	Значение
Имя	
Фамилия	
Отчество	
Дата рождения	
Тип документа	
Серия, номер	
Адрес регистрации	
Пол	

Сохранить Отмена

Рисунок 28 Форма ввода данных³²

Нажав на кнопку «Редактировать», оператор получает возможность исправить любую из граф. При этом ему не придётся заново заполнять все поля, так как вызванная форма открывается уже заполненной (рисунок 28, 29).

³² Рисунок составлен в ходе исследования

Параметр	Значение
Имя	Денис
Фамилия	Евгеньевич
Отчество	Гуртовой
Дата рождения	1994-08-28
Тип документа	Паспорт РФ
Серия, номер	1234 567890
Адрес регистрации	Россия, Краснодарский край, г.Туапсе, ул.Ереванская 2
Пол	М

Сохранить

Рисунок 29 Форма редактирования данных³³

3.4 Экономическая эффективность

Экономическая эффективность (эффективность производства) — это отношение между затраченными ресурсами (трудовыми, денежными, человеческими и т.д.) и результатами производства (продукцией и услугами). Этот показатель служит одним из основополагающих факторов при запуске нового товара или услуги на рынок. [15, 392]

Если фирма в процессе производства не может увеличить выпуск продукции не увеличивая расходы на ресурсы и при этом обеспечивая тот же объём выпуска, используя меньшее количество ресурсов одного типа и не увеличив при этом затраты на другие ресурсы, то предприятие можно считать экономически эффективным.

Экономическая эффективность показывает воздействие процесса реализации инвестиционного проекта на внешнюю для проекта среду и

³³ Рисунок составлен в ходе исследования

учитывает соотношение результатов и затрат по инвестиционному проекту, которые прямо не связаны с финансовыми интересами участников проекта и могут быть количественно оценены. [24]

Для расчёта экономической эффективности разрабатываемого АРМ необходимо создать проект, который позволит выделить все задачи и этапы разработки, внедрения и обслуживания. Удобнее всего это сделать с использованием программного обеспечения Microsoft Project. Это ПО предназначено для управления проектами.

Microsoft Project призван помогать менеджеру проекта при разработке планов, распределении ресурсов по задачам, отслеживании прогресса и анализе объёмов работ. Microsoft Project создаёт расписания критического пути. Расписания могут быть составлены с учётом используемых ресурсов. Цепочка визуализируется в диаграмме Ганта.

Возможность адаптировать продукт под свою специфику путём программирования или покупки готовых решений, созданных на базе Visual Basic или Microsoft .Net. Также данная программа направлена на уменьшение проблем продукта с технической поддержкой. [20, 800]

Разработку информационной системы можно разделить на несколько суммарных задач. В данном случае можно выделить 3 основные суммарные задачи:

1. Разработка системы.

Включает в себя различные этапы разработки АРМ. Состоит из:

- Анализа ТЗ и составления плана-графика разработки;
- Разработки приложения;
- Отладки приложения;
- Формирования руководства пользователя;
- Формирования руководства разработчика;
- Тестирования системы.

2. Внедрение системы на предприятие;

Включает в себя ряд мероприятий, проводимых на предприятии, для внедрения и успешного функционирования системы на предприятии.

Состоит из:

- Установки системы на ПК сотрудников;
 - Обучения персонала.
3. Обслуживание системы.

Включает в себя ряд мероприятий, проводимых на предприятии регулярно, с промежутком определённого периода времени. Состоит из обновления системы на всех ПК сотрудников (рисунок 30).

ДИАГРАММА ГАНТА С ОТСЛЕЖИВАНИЕМ

	и	за	Название задачи	Длительность	Начало	Окончание	Г	Использование ресурсов	а
1			Разработка системы	43 дней	Пн 18.01.16	Ср 16.03.16			
2			Анализ ТЗ и составление плана-графика разработки	3 дней?	Пн 18.01.16	Ср 20.01.16		Разработчик-аналитик	
3			Разработка приложения	30 дней?	Чт 21.01.16	Ср 02.03.16	2	Разработчик	
4			Отладка приложения	5 дней?	Чт 03.03.16	Ср 09.03.16	3	Разработчик	
5			Формирование руководства пользователя	3 дней?	Чт 10.03.16	Пн 14.03.16	4	Технический писатель	
6			Формирование руководства разработчика	2 дней?	Чт 10.03.16	Пт 11.03.16	4	Разработчик	
7			Тестирование системы	3 дней?	Пн 14.03.16	Ср 16.03.16	6	Разработчик	
8			Внедрение системы на предприятие	26 дней?	Чт 17.03.16	Чт 21.04.16	1		
9			Установка системы на ПК сотрудников	5 дней?	Чт 17.03.16	Ср 23.03.16	7	Специалист IT-отдела	
10			Обучение персонала	21 дней	Чт 24.03.16	Чт 21.04.16	9	Специалист IT-отдела	
11			Обслуживание системы	14 дней	Пт 22.04.16	Ср 11.05.16	8		
12			Обновление системы	14 дней	Пт 22.04.16	Ср 11.05.16	10	Специалист	

Рисунок 30 Суммарные задачи этапов разработки АРМ³⁴

³⁴ Рисунок составлен в ходе исследования

После заполнения всех суммарных задач, заполнения всех условий и распределения ресурсов, можно просмотреть диаграмму Ганта (рисунок 30, 31):

Диаграмма Ганта— это разновидность столбчатых диаграмм, которые используются для наглядного представления плана, графика работ по какому-либо проекту. Представляет собой один из методов планирования проектов. Диаграммы Ганта приятно использовать в приложениях по управлению проектами.

Первый формат диаграммы был разработан Генри Л. Гантом в 1910 году. [19]

Представлена диаграмма Ганта в виде полос, ориентированных вдоль оси времени. Каждая отдельная задача в проекте представляется в виде отдельной полосы на диаграмме. Начало и конец полосы означают, соответственно, начало и конец выполнения работы, а длина полосы отражает протяженность работы по времени. Вертикальной осью координат на диаграмме является список всех задач. Также на диаграмме могут отмечаться совокупные (суммарные) задачи, проценты их завершения, указатели последовательности и зависимости работ, метки ключевых моментов (вехи), метка текущего момента времени «Сегодня» и др.

Веха является ключевым понятием диаграммы Ганта. Она представляет собой метку значимого момента в ходе выполнения работы, другими словами, это общая граница двух или более задач. Благодаря вехам появляется возможность наглядно отобразить необходимость синхронизации, последовательности в выполнении различных работ. Вехи, как и другие границы на диаграмме, не являются календарными датами. Если сдвинуть веху, то сдвинется и весь проект. [18, 10]

Диаграмма Ганта с отслеживанием применяется для того, чтобы сравнить запланированные сроки выполнения проекта и реальные в процессе

выполнения проектных работ. Для каждой задачи и фазы на диаграмме отображается запланированный срок выполнения и реальный. [19]

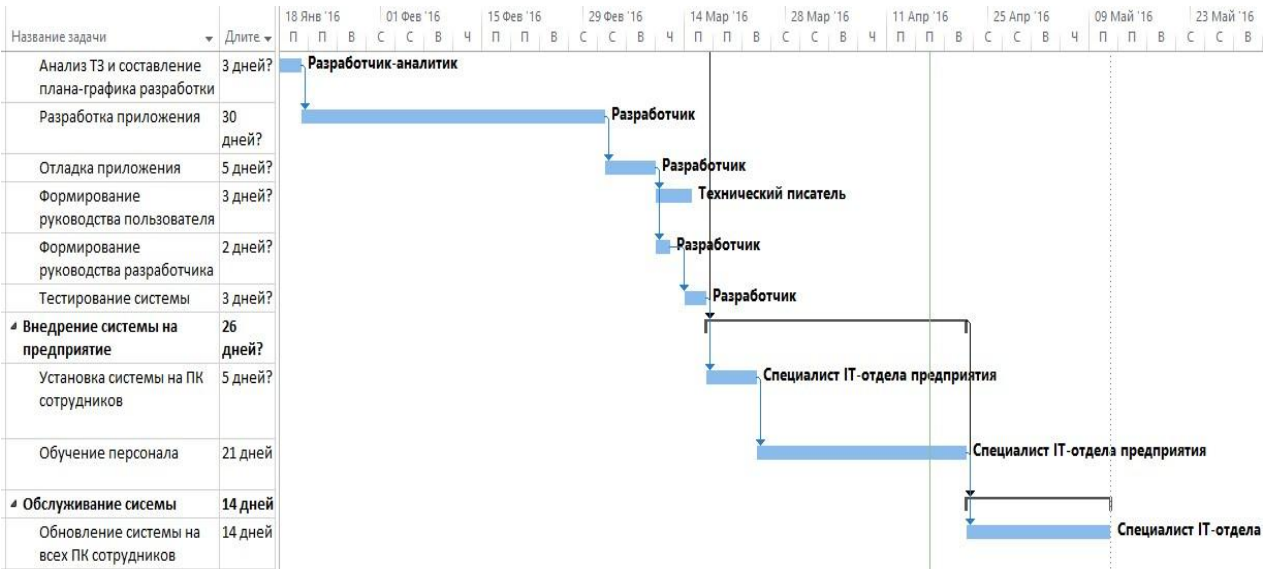


Рисунок 31 Диаграмма Ганта³⁵

Для подсчёта стоимости разработки, внедрения и обслуживания АРМ ресурсов (в данном случае работникам) была присвоена их часовая ставка. Расчёт часовой ставки взят из средней заработной платы в Туапсинском районе. Последующие расчёты так же базируются на этих показателях, при расчёте следует учитывать погрешность. Запросив отчёт по затратам, можно увидеть стоимость разработки, внедрения и обслуживания АРМ (рисунок 32).

Из отчёта по затратам видно, что стоимость разработки данной АРМ составит 113924,08 рублей. Эта сумма является примерной, так как основана на часовой ставке сотрудников, которая, в свою очередь, была взята по среднему уровню заработной платы в данном регионе.

На внедрение системы на предприятие будет затрачено 28912 рублей. Данная статья расходов плавающая, так как может быть уменьшена. Это может произойти в случае высокой квалификации работников, которые

³⁵ Рисунок составлен в ходе исследования

смогут ознакомиться с системой быстрее. Следовательно, меньше рабочих часов это займет у специалиста, занимающегося обучением персонала.

Последняя статья расходов – «обслуживание системы» является ежегодной. То есть, один раз в год предприятие выплачивает 15568 рублей за обновление, исправление неполадок и полное обслуживание.

Таким образом, разработка и внедрение системы обойдутся предприятию в 142836,08 рублей.

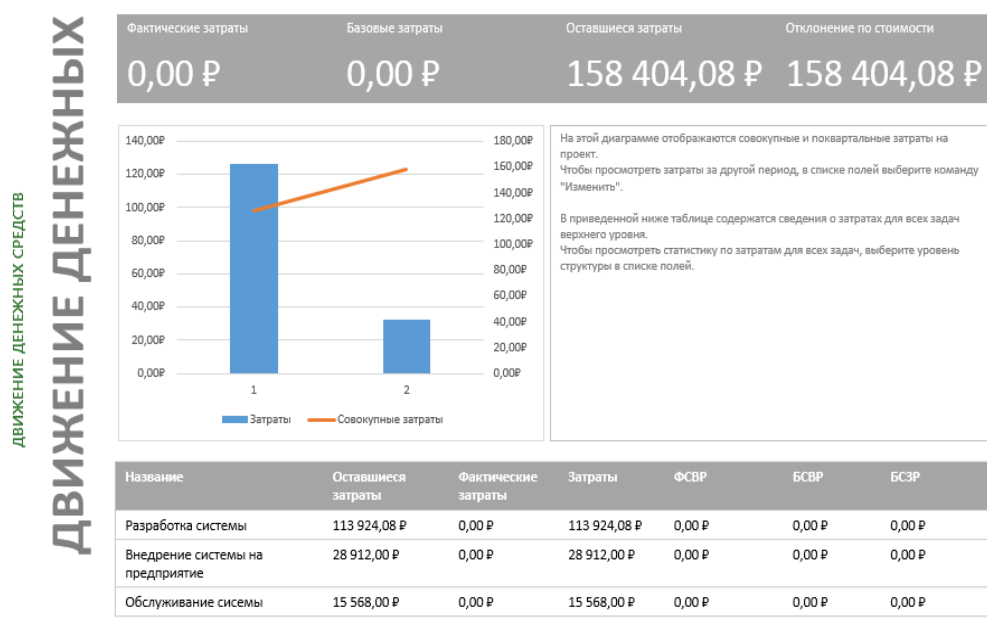


Рисунок 32 Отчёт по затратам «Движение денежных средств»³⁶

На исследуемом предприятии не реализуется функция по предоставлению услуги по подготовке документов для замены/получения паспорта Российской Федерации. Средняя стоимость данной услуги в других МФЦ города Туапсе и Туапсинского района составляет 230 рублей. За один рабочий день МФЦ посещают около 280 человек, из них примерно 15% обращаются с просьбой о подготовке документов для ОУФМС. Получается, в среднем, 5 человек в день. Отсюда можно посчитать, что внедрение данной услуги будет приносить предприятию прибыль в замеры 1150 рублей в день. С учетом графика работы предприятия 5/2, получаем 23 рабочих дня в месяц,

³⁶ Рисунок составлен в ходе исследования

которые принесут предприятию 26450 рублей. При подобной прибыли предприятие уже через 5,4 месяцев сможет полностью перекрыть затраты на разработку и внедрение системы, и сможет получать чистую прибыль.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы были решены задачи:

1. Проведен анализ действующих систем автоматизации документооборота на предприятии;
2. На основе проведенного анализа выявлены слабые звенья в общей цепи документооборота;
3. Углублены и закреплены теоретические знания, полученные в ходе профессиональной и специальной подготовки;
4. Продемонстрированы практические навыки и компетенции в сфере профессиональной деятельности путем решения практических задач, а также опыта в реальных условиях деятельности предприятия.
5. Разработано автоматизированное рабочее место оператора по подготовке документации для органов Федеральной Миграционной службы;
6. Выявлены существующие недостатки и разработаны рекомендации по их исправлению.

В ходе выполнения ВКР были сделаны следующие выводы:

1. Было проведено исследование действующих систем автоматизации документооборота и бизнес-процессов предприятия, в ходе которого был выявлен процесс, который требовал автоматизации. Таким процессом оказался процесс подготовки документов для получения/замены паспорта Российской Федерации.
2. В ходе разработки АРМ были продемонстрированы навыки работы с языком Java, умения в области разработки баз данных, использование различных приложений, созданных, для помощи в работе с программным кодом.
3. Были углублены и закреплены теоретические знания, полученные в ходе профессиональной и специальной подготовки, продемонстрированы

практические навыки и компетенции в сфере профессиональной деятельности путем решения практических задач, обусловленных индивидуальным планом, а также получен опыт в реальных условиях деятельности предприятия.

4. Были получены знания и навыки о методиках проектирования, внедрения и эксплуатации отдельных задач и подсистем информационной системы, изучение автоматизированных средств и систем, реализующих информационные системы, приобретение навыков исследования и проектирования подсистем информационных систем.

5. Были освоены на практике методы предпроектного обследования объекта – исследование системы информационного обеспечения организации, ее организационной, методической, алгоритмической, лингвистической, программной и технической реализацией, исследование системной архитектуры информационной системы (данные, приложения и оборудование), оценка возможностей повышения эффективности функционирования информационной системы; проведён системный анализ результатов обследования при определении путей эффективного использования информационных ресурсов; изучена техническая и экономическая документация предприятия, получены знания по оформлению технических и рабочих проектов построения информационных систем; приобретен практический опыт разработки систем информационного обеспечения, навыки работы с локальными и глобальными вычислительными сетями, обслуживания информационных систем.

6. Так же был проведен анализ экономической эффективности, разработанного АРМ, результаты которого показали, что при внедрении АРМ предприятие будет получать среднюю прибыль в размере 26450 рублей в месяц.

После проведения всех исследований были разработаны следующие **рекомендации:**

1. Внедрение разработанного АРМ на предприятие привлечет новых посетителей, а, следовательно, принесет предприятию прибыль.
2. Для корректной работы АРМ на протяжении всего срока службы предприятия необходимо регулярно проводить обслуживание и обновление.

СПИСОК ЛИТЕРАТУРЫ

1. BPM Soft. Нотации серии IDEF. [Электронный ресурс]. – URL: <http://bpmsoft.org/idef0-and-idef3/> - (дата обращения: - 24.03.2016)
2. Cit Forum. Объектно-ориентированные CASE-средства (Rational Rose). [Электронный ресурс]. – URL: http://citforum.ru/database/case/glava5_5.shtml - (дата обращения: - 18.03.2016)
3. Java. Пособие для проведения практических занятий. [Электронный ресурс]. – URL: <http://math.sgu.ru/sites/chairs/prinf/materials/java/lesson10.htm> - (дата обращения: - 20.03.2016)
4. NetBeans IDE. Среда разработки на языках Java, JavaScript, HTML5, PHP, C/C++. [Электронный ресурс]. – URL: https://netbeans.org/features/index_ru.html - (дата обращения: - 19.03.2016)
5. АИС «МФЦ» Информационная система автоматизации деятельности многофункциональных центров. – [Электронный ресурс]. – URL: http://www.evolenta.ru/ais_mfc.html - (дата обращения: - 29.03.2016)
6. Википедия. Свободная энциклопедия. Блок – схемы алгоритмов. [Электронный ресурс]. – URL: <https://ru.wikipedia.org> - (дата обращения: - 24.03.2016)
7. Википедия. Свободная энциклопедия. Диаграмма последовательности. [Электронный ресурс]. – URL: <https://ru.wikipedia.org> - (дата обращения: - 24.03.2016)
8. Википедия. Свободная энциклопедия. Диаграмма прецедентов. [Электронный ресурс]. – URL: <https://ru.wikipedia.org/wiki/> - (дата обращения: - 18.03.2016)
9. Государственный реестр информационных систем Краснодарского края. ИСАЭД «СИНКОПА-ДОКУМЕНТ» [Электронный

ресурс]. – URL: http://ris.krasnodar.ru/?ELEMENT_ID=742 - (дата обращения: - 29.03.2016)

10. Джеймс Гослинг, Билл Джой, Гай Стил, Гилад Брача, Алекс Бакли. Язык программирования Java SE 8. Подробное описание, 5-е издание = The Java Language Specification, Java SE 8 Edition (5th Edition) (Java Series). — М.: «Вильямс», 2015. — 672 с.

11. Заметки управленца: бизнес-моделирование, Business Studio, СМК. Нотация IDEF0. [Электронный ресурс]. – URL: http://mymanager.com.ua/bp/bs/overview/notation_idef0.php - (дата обращения: - 24.03.2016)

12. Кей С. Хорстманн. Java SE 8. Вводный курс = Java SE 8 for the Really Impatient. — М.: «Вильямс», 2014. — 208 с.

13. Корпоративный менеджмент. Библиотека управления. Геннадий Верников «Основы IDEF3». [Электронный ресурс]. – URL: <http://www.cfin.ru/vernikov/idef/idef3.shtml> - (дата обращения: - 24.03.2016)

14. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с.

15. Орехов А.М. Методы экономических исследований. М.: ИНФРА-М, 2009 – 392 с.

16. Официальный портал АИС МФЦ – С.К.А.Т. [Электронный ресурс]. – URL: http://www.skate-vending.com/aic_mfc_skate.php# - (дата обращения: - 29.03.2016)

17. Официальный сайт предприятия МКУ «МФЦ ТУАПСИНСКОГО РАЙОНА» МНОГОФУНКЦИОНАЛЬНЫЙ ЦЕНТР ПРЕДОСТАВЛЕНИЯ ГОСУДАРСТВЕННЫХ И МУНИЦИПАЛЬНЫХ УСЛУГ ТУАПСИНСКОГО РАЙОНА [Электронный ресурс]. URL: <http://e-mfc.ru/mfc/77/> (дата обращения: 02.03.2016).

18. Рассел Д., Кохн Р.: Диаграмма Ганта – Книга по Требованию, 2012 – 10с.
19. Самоучитель MS Project «Таурион». Диаграмма Ганта. [Электронный ресурс]. – URL: <http://www.taurion.ru/project/4/2> - (дата обращения: - 24.03.2016)
20. Сингаевская Галина Ивановна. Управление проектами в Microsoft Project 2007. — М.: «Диалектика», 2008. — С. 800.
21. Управление ФМС России по г. Москве. Официальный сайт. Перечень документов, необходимых для получения паспорта гражданина РФ. [Электронный ресурс]. – URL: https://77.fms.gov.ru/Perechen_dokumentov - (дата обращения: - 21.03.2016)
22. Федеральный закон от 27.07.2010 N 210-ФЗ (ред. от 31.12.2014) "Об организации предоставления государственных и муниципальных услуг" (с изм. и доп., вступ. в силу с 31.03.2015).
23. Хассан Гома: UML-проектирование систем реального времени параллельных и распределенных приложений. – ДМК Пресс, 2011. – 31с.
24. Экономика БГЭУ — Блог. Оценка экономической эффективности. [Электронный ресурс]. – URL: <http://www.economy-web.org/?p=430> - (дата обращения: - 29.03.2016)
25. Ясенов В.Н: Информационные системы и технологии в экономике: учебное пособие. – Юнити-Дана, 2012. – 560 с.

ПРИЛОЖЕНИЕ 1

Программный код

```
/*
 * To change this license header, choose License Headers in Project
Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package std_antonenkova;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Controller {
    private MySQL sql;
    private final SettingsForm setingsForm;
    private final MainGUI mainGui;
    private final Queries query;
    private final EditForm editForm;

    Controller(){
        this.query = new Queries();
        this.setingsForm = new SettingsForm();
        this.mainGui = new MainGUI();
        this.editForm = new EditForm();
    }
}
```

```

    }

    private void connectDatabase() {
        //Метод подключения к базе
        this.sql = new MySQL();

        String ip = SettingsForm.AddressField.getText();
        String username = SettingsForm.UsernameField.getText();
        String password =
String.copyOfValueOf(SettingsForm.PasswordField.getPassword());
        String database = SettingsForm.DatabaseField.getText();

        sql.SetConnectionSettings(ip, username, password, database);
    }

    private boolean testConnection(){
        String currentQuery = this.query.GetTestQuery();
        ResultSet rs = sql.GetResultSet(currentQuery);

        try {
            while (rs.next()){
                Integer count = rs.getInt(1);
                System.out.println("Test: " + count);
            }
            rs.close();
        } catch (SQLException ex) {
            Logger.getLogger(Std_Antonenkova.class.getName()).log(Level.SEVERE,
null, ex);
            return false;
        }
    }

```

```

        return true;
    }

    public void RunApp(){
        this.runSettingsInterface();
        this.sql.Disconnect();
    }

    private void runSettingsInterface(){
        ActionListener actionListener = new ConnectButtonListener();
        SettingsForm.ConnectButton.addActionListener(actionListener);

        ActionListener editButtonListener = new EditButtonListener();
        MainGUI.EditButton.addActionListener(editButtonListener);

        this.setingsForm.setVisible(true);
    }

    private void runMainInterface(){
        this.setingsForm.setVisible(false);
        this.mainGui.setVisible(true);
        this.loadOrders();
        this.loadDocuments();
        this.loadPersons();
    }

    //=====Данные=====
    private String[][] getArrayForEditForm(int tabIndex){
        switch(tabIndex){
            case 0:

```



```

return new String [][] {
    {"Имя", ""},
    {"Фамилия", ""},
    {"Отчество", ""},
    {"Дата рождения", ""},
    {"Тип документа", ""},
    {"Серия, номер", ""},
    {"Адрес регистрации", ""},
    {"Пол", ""}
};

```

case 1:

```

return new String [][] {
    {"Имя", ""},
    {"Фамилия", ""},
    {"Отчество", ""},
    {"Дата рождения", ""},
    {"Пол", ""},
    {"Адрес(фактический)", ""}
};

```

case 2:

```

return new String [][] {
    {"Причина обращения", ""},
    {"Фамилия", ""},
    {"Имя", ""}
};

```

default: return null;

```

}

```

```

}

```

//=====Формы=====

```
private void loadOrders(){
    String currentQuery = this.query.GetOrders();
    ResultSet rs = sql.GetResultSet(currentQuery);

    String[][] tableArray = sql.FetchStringArray(rs);

    MainGUI.OrdersTable.setModel(new
javax.swing.table.DefaultTableModel(
    tableArray,
    new String [] {
        "Номер заявки"
        ,"Причина обращения"
        ,"Фамилия"
        ,"Имя"
        ,"Отчество"
        ,"Дата"
    }
));
    MainGUI.OrdersTable.setAutoResizeMode(5);
}
```

```
private void loadDocuments(){
    String currentQuery = this.query.GetDocuments();
    ResultSet rs = sql.GetResultSet(currentQuery);

    String[][] tableArray = sql.FetchStringArray(rs);
```

```

MainGUI.DocumentsTable.setModel(new
javax.swing.table.DefaultTableModel(
    tableArray,
    new String [] {
        "Номер документа"
        , "Имя"
        , "Фамилия"
        , "Отчество"
        , "Дата рождения"
        , "Тип документа"
        , "Серия, номер"
        , "Адрес регистрации"
        , "Пол"
        , "Дата добавления"
    }
));
MainGUI.DocumentsTable.setAutoResizeMode(5);
}

```

```

private void loadPersons(){
    String currentQuery = this.query.GetPersons();
    ResultSet rs = sql.GetResultSet(currentQuery);

    String[][] tableArray = sql.FetchStringArray(rs);

```

```

MainGUI.PersonsTable.setModel(new
javax.swing.table.DefaultTableModel(
    tableArray,
    new String [] {

```

```

        "Номер пользователя"
        , "Имя"
        , "Фамилия"
        , "Отчество"
        , "Дата рождения"
        , "Пол"
        , "Адрес"
        , "Дата регистрации"
    }
});
MainGUI.PersonsTable.setAutoResizeMode(5);
MainGUI.PersonsTable.doLayout();
}

```

```

private void showEditForm(){
    int index = MainGUI.MainTabGroup.getSelectedIndex();
    System.out.println("Index: " + index);
    this.editForm.setVisible(true);

    EditForm.EditTable.setModel(new javax.swing.table.DefaultTableModel(
        this.getArrayForEditForm(index),
        new String [] { "Параметр", "Значение" }
    ));
}

```

```

//=====Обработчики=====

public class ConnectButtonListener implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        Controller.this.connectDatabase();
    }
}

```

```
boolean connectionResult = Controller.this.testConnection();
```

```
if (connectionResult) {
```

```
    Controller.this.runMainInterface();
```

```
}
```

```
}
```

```
}
```

```
public class EditButtonListener implements ActionListener {
```

```
    public void actionPerformed(ActionEvent e) {
```

```
        Thread t = new Thread(new Runnable() {
```

```
            @Override
```

```
            public void run() {
```

```
                Controller.this.showEditForm();
```

```
            }
```

```
        });
```

```
        t.start();
```

```
    }
```

```
}
```

```
}
```