



МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Российский технологический университет»
МИРЭА

Физико-технологический институт
Кафедра оптических и биотехнических систем и технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА НА
ТЕМУ:

**«ПРИМЕНЕНИЕ ВОДОПАДНОЙ МОДЕЛИ ВНЕДРЕНИЯ
ИНФОРМАЦИОННЫХ СИСТЕМ ДЛЯ РЕАЛИЗАЦИИ
ЭЛЕКТРОННОЙ РЕГИСТРАТУРЫ БОЛЬНИЦЫ»**

Студент:
Воронова О. А.

Научный руководитель:
к.т.н., доц. МИРЭА Степанов Д.Ю.

Москва – 2020

АННОТАЦИЯ

Данная выпускная квалификационная работа посвящена автоматизации ключевых бизнес-процессов регистратуры больницы на основе каскадного метода внедрения в программной среде Microsoft Access.

В отчете представлены результаты детального анализа методологии, разработаны интерфейсы для работы с приложением, спроектировано само приложение по пользовательским и функциональным требованиям, детализированы процессы в нотациях IDEF0 и IDEF3 и проведено нагрузочное тестирование.

Объем выпускной квалификационной работы составляет 56 страниц пояснительной записки. Отчет содержит 41 рисунок, 7 таблиц, 21 формулу, в том числе для экономического расчета, а также приведены 20 литературных источников, включен обзор основных из них.

ОГЛАВЛЕНИЕ

| | |
|--|-----------|
| Введение | 4 |
| Раздел 1. Обзор литературных источников | 6 |
| Раздел 2. Теоретическая часть | 9 |
| 2.1. Устройство каскадной модели внедрения систем | 9 |
| 2.1.1. Этапы водопадной модели внедрения | 9 |
| 2.1.2. Преимущества и недостатки каскадной модели внедрения... | 11 |
| 2.2. Выработка требований к ПО | 14 |
| 2.2.1. Способы определения требований | 14 |
| 2.2.2. Пользовательские и функциональные требования | 15 |
| 2.3. Проектирование бизнес-процессов в AS-IS и TO-BE | 18 |
| 2.3.1. Концепция проектирования | 18 |
| 2.3.2. Описание ключевых бизнес-процессов в модели As-Is | 21 |
| 2.3.3. Описание ключевых бизнес-процессов в модели To-Be | 25 |
| 2.3.4. Архитектура данных | 31 |
| 2.3.5. Проектирование пользовательского интерфейса | 34 |
| Раздел 3. Программно-алгоритмическая часть | 38 |
| 3.1. Реализация основных функций системы | 38 |
| 3.2. Функциональное тестирование | 44 |
| 3.3. Нагрузочное тестирование | 46 |
| Раздел 4. Экономическая часть | 48 |
| 4.1. Затраты на оплату труда | 48 |
| 4.2. Амортизация | 50 |
| 4.3. Материальные затраты | 51 |
| 4.4. Страховые затраты | 52 |
| 4.5. Смета затрат | 52 |
| Заключение | 54 |
| Список использованных литературных источников | 55 |

Введение

С каждым днем все острее становится вопрос структуризации и доступности информации медицинских данных пациентов. Количество людей, обращающихся в медицинские учреждения не должно снижать качество оказания помощи. Однако если мы не имеем единой системы учета данных, эта задача становится затруднительной.

Для поддержания темпа стабильной работы необходимо иметь точную информацию о каждом обратившемся человеке, содержащую необходимые для специалиста параметры, а доступ к этой информации не должен быть ограничен для врача и не должен занимать много времени, а также должна быть исключена недостоверность показаний и присутствовать возможность моментального обновления актуальной информации.

Ряд факторов в данный момент может препятствовать желаемому уровню получения/предоставления данного функционала. Внедрение общей системы улучшит ситуацию, как для медицинского персонала, так и для пациентов. Базы данных оптимизируют работу больниц и других медицинских учреждений: время обработки каждого запроса предположительно сократится за счет упрощенной системы хранения информации, то есть замены бумажных носителей на электронные, появится возможность обратиться к БД из любой точки мира, имея устройство с выходом в интернет, исключается утеря, порча и подмена, появятся удобные инструменты в личном деле каждого пациента [3].

Цель данной работы – демонстрация применения водопадной модели внедрения медицинских информационных систем на примере реализации ключевых бизнес-процессов электронной регистратуры больницы в среде MS Access. Каскадная модель оптимальна в данном случае, так как определены

требования, для которых не предусматриваются изменения в процессе разработки [4]. Полученное в результате приложение поможет значительно оптимизировать работу больницы, обеспечив повышение качества и скорости оказываемых населению медицинских услуг. Для достижения поставленной цели будет использоваться среда MS Access. А также будут выполнены следующие задачи:

- детальный анализ водопадной методологии внедрения систем;
- идентификация требований и формирование матрицы требований;
- проектирование процессов и оргструктуры в моделях AS-IS и TO-BE нотации IDEF0 и IDEF3 до 3-4 уровней детализации;
- моделирование разрабатываемых пользовательских интерфейсов;
- проектирование структуры данных и нормализация таблиц данных;
- подготовка блок-схемы алгоритмов работы программы;
- реализация ключевых процессов в среде MS Access;
- тестирование и количественная оценка результатов тестирования.

Объектом исследования данной работы является недостаточная автоматизация работы регистратуры, в следствие не задействования информационных систем, или их неэффективного использования. Объектами исследования выступают три бизнес-процесса для отладки быстродействия, отвечающей современным стандартам работоспособности, регистратуры больницы. Таким образом, решение поставленных задач поспособствует выявлению необходимого программного обеспечения и его прикладное воплощение, что несет практическую ценность данных исследований.

Раздел 1. Обзор литературных источников

В самоучителе «Microsoft Access 2016» Бекаревича Ю. Б. рассматриваются следующие пункты:

- введение в MS Access;
- проектирование реляционной базы данных;
- создание базы данных;
- создание и работа с запросами;
- создание и работа с формами;
- разработка интерфейса для ввода, просмотра и корректировки документов;
- создание и использование отчётов;
- оперативный анализ данных реляционной базы access;
- разработка приложений с использованием макросов;
- разработка интерфейса для управления приложением;
- создание веб-приложений.

В учебном пособии «Модели жизненного цикла» авторы описывают:

- общее представление о моделях жизненного цикла;
- жизненный цикл экономических систем;
- жизненный цикл информационных систем, а именно: жизненный цикл продукта, управление жизненным циклом изделия, автоматизированные системы управления ЖЦИ, решения SAP по управлению ЖЦ продукта;
- актуальные ЖЦ в сфере информации и информационных систем;
- модели жизненного цикла в социально-экономических системах.

В книге «Бизнес-процессы: основные понятия, теории, методы» Шеера А.В. представлены:

- систематизированные знания по теоретическим основам бизнес-процессов и бизнес-информатике;
- различные подходы и методы описания и анализа бизнес-процессов;
- концепция архитектуры бизнес-инжиниринга.

В ГОСТ Р ИСО/МЭК ТО 15271-2002 приводятся:

- графическая система обозначений, описывающая процессы и этапы работы;
- описание основных и побочных процессов;
- детализация процессов и присвоение им приоритетности;
- связь между программным средством и запросом функциональных требований.

В учебном пособии «Методы и средства инженерии программного обеспечения» представлены:

- методы прикладного и теоретического проектирования;
- методы интеграции и преобразования программ и данных;
- подходы и методы, их аннотации для накопления, выбора и оценки применимости в новых программных проектах.

В статье Салихова Э.Ш. приводится разбор типовых медицинских информационных систем, их возможности и особенности построения.

Рассматриваются такие ключевые моменты как:

- автоматизированный сбор первичных данных;
- обеспечение защищенной передачи данных;
- эффективность концепций ПО на региональных уровнях.

В книге «Объектно-ориентированное моделирование и разработка» необходимая информация содержится в разделах:

- объектно-ориентированная методология и ее история;

- выделение сущности объектов;
- моделирование классов;
- обзор процесса разработки;
- жизненный цикл водопадной модели внедрения;
- анализ разрабатываемого приложения;
- разбиение системы на подсистемы;
- выбор стратегии управления программным обеспечением.

Раздел 2. Теоретическая часть

2.1. Устройство каскадной модели внедрения систем

Водопадная или каскадная модель – традиционная методология внедрения, появившаяся в 1970 году вследствие доработки Винстоном Ройсом “неэффективного способа обработки информации”. Каскадная методика управления проектами подразумевает построение многоуровневого процесса поэтапно, осуществляя переход со стадии на стадию без пропусков и возвратов на предыдущие этапы. Данная модель считается практичной и эффективной для недолговременных и достаточно сложных проектов. На начальных этапах требуется максимально детализированное понимание поставленных требований и точное формулирование задач после взаимодействия между заказчиком и исполнителем, поскольку внесение корректировок негативно скажется на сроках исполнения, и как возможное следствие, бюджете продукта. Однако после отладки данная модель способна существовать независимо и не нуждается во вмешательствах извне [7].

2.1.1. Этапы водопадной модели внедрения

Подробное описание каждого этапа водопадной модели внедрения [5]:

- Выработка системных требований. На первом этапе происходит сбор информации и исследование задачи, которая должна быть решена и резюмирование всех требований заказчика. По итогу первичного взаимодействия заказчика и исполнителя формируется техническое задание, согласованное с каждой участвующей в проекте стороной.
- Выработка требований к ПО. Далее осуществляется нахождение проектных решений, удовлетворяющих всем поставленным

условиям в техническом задании. Цель данного этапа – оформление проектной документации со всеми входными данными для реализации проекта.

- Анализ. Следующий шаг необходим для предотвращения возможных ошибок возникших при коммуникации сторон, неверно составленной документации или неточно распределенного ресурса. Проверяется соответствие исходным задачам, целостность и непротиворечивость запросов. Выстраивается логическая схема, доступная для понимания в равной степени для заказчиков, исполнителей и пользователей ведущейся разработки.
- Проектирование. Учитывая все полученные запросы, оформленные и согласованные документы, представленные ранее можно приступать непосредственно к проектированию. Решения, принятые на данном этапе показывают, как именно должна выглядеть система в разложении ее на простые составляющие - очевидно реализуемые процедуры.
- Кодирование. Ранее определенные процедуры реализуются с помощью наиболее подходящего языка программирования. Данным этапом занимаются разработчики, согласовывая свои действия с ранее заявленной документацией.
- Тестирование. Здесь, после проведенных мероприятий по выявлению соответствия ТЗ созданному ПО (программному обеспечению), происходит передача продукта в эксплуатацию.
- Эксплуатация. Фаза эксплуатации представляет работы по внедрению конечного продукта и оснащению рабочих мест пользователей, обеспечение эксплуатационной документацией, проведение обучения персонала и непосредственно эксплуатацию, в

том числе локализацию проблем и устранение их. Особую роль при этом играет стадия сопровождения (обслуживания). Она выполняется параллельно стадии эксплуатации.

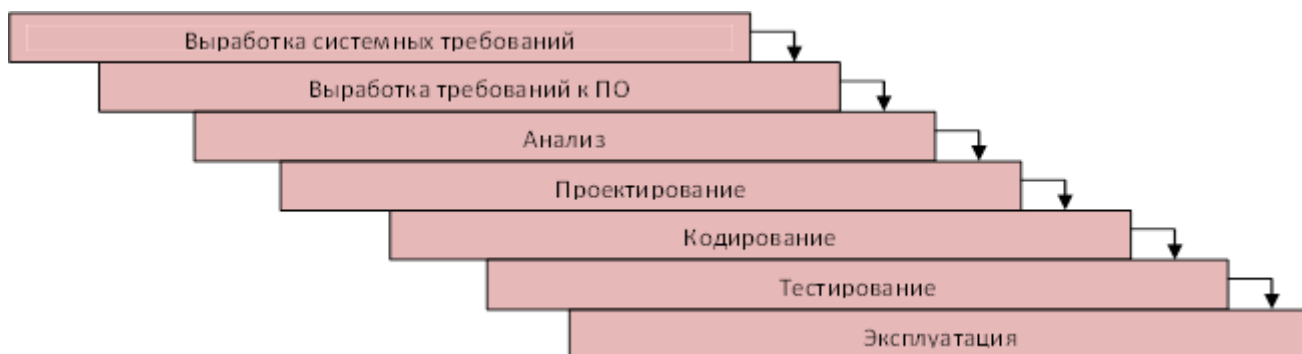


Рисунок 2.1 – *Этапы водопадной модели внедрения*

2.1.2. Преимущества и недостатки каскадной модели внедрения

Преимущества каскадной модели:

- поэтапно справляется с объемными задачами и подходит для тех проектов, которые аналитически ясны, но все же трудно разрешимы;
- она лаконична и интуитивно понятна в применении, так как процесс разработки выполняется поэтапно;
- ее структурой может руководствоваться даже слабо подготовленный в техническом плане или неопытный персонал;
- она отличается стабильностью требований;
- она представляет собой шаблон, в который можно поместить методы для выполнения анализа, проектирования, кодирования, тестирования и обеспечения;
- она хорошо срабатывает тогда, когда требования к качеству доминируют над требованиями к затратам и графику выполнения проекта;

- она способствует осуществлению строгого контроля менеджмента проекта;
- при правильном использовании модели дефекты можно обнаружить на более ранних этапах, когда их устранение еще не требует относительно больших затрат;
- стадии модели довольно хорошо определены и понятны;
- ход выполнения проекта легко проследить с помощью использования временной шкалы (или диаграммы Ганта), поскольку момент завершения каждой фазы используется в качестве стадии.

Недостатки каскадной модели:

- в основе модели лежит последовательная линейная структура, в результате чего каждая попытка вернуться на одну или две фазы назад, чтобы исправить какую-либо проблему или недостаток, приведет к значительному увеличению затрат и сбою в графике;
- она не может предотвратить возникновение итераций между фазами, которые так часто встречаются при разработке ПО;
- клиент не имеет возможности воспользоваться доступными промежуточными результатами, и отзывы пользователей нельзя передать обратно разработчикам;
- не поддается гибкому моделированию;
- все требования должны быть известны в начале жизненного цикла;
- модель не рассчитана на динамические изменения в требованиях на протяжении всего жизненного цикла;
- возникает необходимость в жестком управлении и контроле, поскольку в модели не предусмотрена возможность модификации требований;

- отсутствует возможность учесть переделку и итерации за рамками проекта.

Для реализации программной разработки был составлен план, соответствующий каскадной модели внедрения:

- Создание списка требований:
 - выделение пользовательских требований;
 - формирование функциональных требований;
 - приоритизация функциональных требований;
- Проектирование бизнес-процессов:
 - моделирование процессов по нотации IDEF0 в модели As-Is;
 - моделирование процессов по нотации IDEF3 в модели As-Is;
 - моделирование процессов по нотации IDEF0 в модели To-Be;
 - моделирование процессов по нотации IDEF3 в модели To-Be;
 - составление карты процессов;
 - проектирование архитектуры данных;
 - проектирование интерфейса;
- Реализация работающего приложения:
 - реализация основных функций системы;
 - функциональное тестирование;
 - нагрузочное тестирование.

2.2. *Выработка требований к ПО*

2.2.1. *Способы определения требований*

Важнейшим моментом в проектировании программного обеспечения является определение требований заказчика. Выработку требований можно разделить на четыре этапа, а именно:

- определение концепции продукта;
- сбор требований;
- анализ требований;
- формулировка конечных требований.

Таким образом, для выполнения поставленных задач необходимо в первую очередь встретиться с заказчиком для формулирования концепции конечного продукта. Затем вырабатываются конкретные требования заказчика к функциональности программного обеспечения, и обсуждается возможность реального его использования. Это является важнейшим этапом в разработке, в частности при использовании каскадной модели внедрения, где при неполной проработке требований значительно увеличивается риск повторного выполнения работ “с нуля”. После определения требований необходимо проанализировать полученные данные и сформулировать не избыточный список требований к продукту, что существенно повлияет на оптимизацию рабочего процесса.

В заключение прорабатывания происходит документирование требований, выстраивается система приоритетов для исключения лишних и невостребованных процессов, которые могли бы иметь негативное влияние на бюджет или сроки исполнения и, как результат, группа разработчиков получает законченное техническое задание и начинает его тестирование.

Первым делом необходимо определить требования заказчика. Регистратура является структурным подразделением поликлиники и врачебной амбулатории, обеспечивающим формирование и распределение потоков пациентов, своевременную запись и регистрацию больных на прием к врачу, в том числе с применением информационных технологий.

2.2.2. Пользовательские и функциональные требования

Пользовательские требования описывают цели и задачи, которые пользователям позволит решить система. После сбора и анализа требований мы получаем раздробленный список по данному профилю с конкретным отражением необходимых к предоставлению возможностей. В конкретном случае рассмотрения регистратуры больницы как заказчика и пользователя, разрабатываемого ПО, запрос пользовательских требований будет выглядеть следующим образом:

- возможность создания личного дела пациента;
- возможность добавления, редактирования, удаления данных о пациенте;
- возможность поиска пациента;
- возможность обновления статуса пациента;
- обеспечение доступа к анамнезу;
- возможность поиска направлений к специалисту;
- возможность просмотра списка специалистов;
- возможность записывать к специалисту пациента;
- возможность просматривать расписание специалистов;
- возможность поиска свободного специалиста;
- возможность формирования талона записи;
- возможность учета пациентов на стационарном режиме;
- возможность вести учет и классификацию палат;

- возможность учета пациентов на амбулаторном режиме.

Функциональные требования – требования, показывающие, какие элементы должны присутствовать в ПО, а также какие действия должны выполняться системой. Значит, функциональные требования будут иметь следующий вид:

- функция открытия личного дела пациента;
- функция редактирования личных и иных данных пациента;
- функция поиска по базе данных;
- функция учета пациентов на амбулаторном режиме;
- функция открытия анамнеза;
- функция поиска направлений к специалистам;
- функция формирования и просмотра списка специалистов;
- функция заполнения графика специалистов;
- функция доступа к расписанию специалистов;
- функция отбора специалиста по заданным параметрам;
- функция сбора нескольких (определенных) данных для распечатки пациенту;
- функция выбора амбулаторного или стационарного наблюдения;
- функция учета и классификации палат;
- функция учета пациентов на стационарном режиме.

После формирования и анализа списка требований была произведена приоритизация, результат которой представлен в таблице 2.1. Таким образом, можно выделить три ключевых бизнес-процесса, включающих в себя приоритетные требования, а именно: «Принять пациента», «Распределить пациента», «Оформить лечение пациента», эти бизнес-процессы впоследствии должны быть подвергнуты декомпозиции.

Таблица 2.1 – Приоритизация требований

| № | Приоритет требований | Пользовательские требования | Функциональное требование |
|----|----------------------|--|--|
| 1 | Высокий | Возможность создания личного дела пациента | Функция открытия личного дела пациента |
| 2 | Высокий | Возможность добавления, редактирования, удаления данных о пациенте | Функция редактирования личных и иных данных пациента |
| 3 | Высокий | Возможность поиска пациента | Функция поиска по базе данных |
| 4 | Средний | Возможность вести учет и классификацию палат | Функция учета и классификации палат |
| 5 | Средний | Обеспечение доступа к анамнезу | Функция открытия анамнеза |
| 6 | Средний | Возможность поиска направлений к специалисту | Функция поиска направлений к специалистам |
| 7 | Средний | Возможность просмотра списка специалистов | Функция формирования и просмотра списка специалистов |
| 8 | Высокий | Возможность записывать к специалисту пациента | Функция заполнения графика специалистов |
| 9 | Средний | Возможность просматривать расписание специалистов | Функция доступа к расписанию специалистов |
| 10 | Средний | Возможность поиска свободного специалиста | Функция отбора специалиста по заданным параметрам |
| 11 | Средний | Возможность формирования талона записи | Функция сбора нескольких (определенных) данных для распечатки пациенту |
| 12 | Высокий | Возможность учета пациентов на стационарном режиме | Функция учета пациентов на стационарном режиме |
| 13 | Средний | Возможность выбора режима наблюдения | Функция выбора амбулаторного или стационарного наблюдения |
| 14 | Средний | Возможность учета пациентов на амбулаторном режиме | Функция учета пациентов на амбулаторном режиме |

2.3. *Проектирование бизнес-процессов в AS-IS и TO-BE*

2.3.1. *Концепция проектирования*

Сформулировав требования, можно приступить к моделированию бизнес-процессов компании. Бизнес-процесс представляет из себя логически завершенную цепочку взаимосвязанных действий, целью которой является достижения измеримых результатов или создания продукта для удовлетворения требований внутренних или внешних потребителей. Для удобства разработки используется графическое описание деятельности, в качестве которого применяются блок-схемы бизнес-процессов.

Для четкого определения и систематизации текущих процессов компании, а также используемых информационных объектов применяется модель AS-IS (как есть). Такой подход помогает проанализировать слабые места в работе организации с целью оптимизации путем внедрения новых процессов [6].

Для устранения выявленных с помощью модели AS-IS недостатков создается модель TO-BE (как будет/как должно быть). Эта модель позволяет не только существенно сократить время внедрения новых процессов, но и значительно снижает риски, и в целом дает представления о последствиях оптимизации [8].

Ввиду необходимости наглядного описания работы компании используются различные нотации моделирования [9]. Нотацией называется набор графических объектов, используемых для моделирования.

Наиболее широко распространёнными являются нотации IDEF – методологии семейства ICAM (Integrated Computer-Aided Manufacturing). Каждая нотация определяется порядковым номером и используется для разных элементов. В работе будут применены нотации IDEF0 и IDEF3.

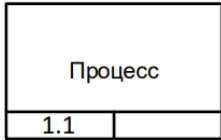
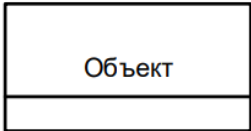
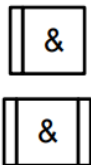
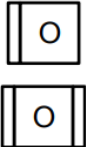
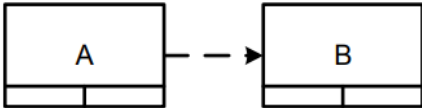
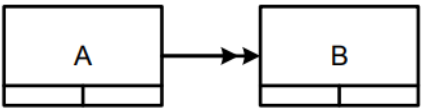

Нотация IDEF0 используется для отображения структуры и функций системы, а также связывающих эти функции потоков информации и материальных объектов. Графические символы, используемые в нотации, представлены в таблице 2.2 [1].

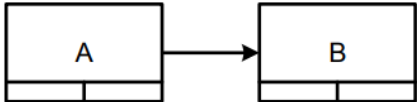
Таблица 2.2 – Графические элементы IDEF0

| Графический элемент | Описание |
|---------------------|---------------------------|
| | Процесс |
| | Входящие данные процесса |
| | Исходящие данные процесса |
| | Ограничение процесса |
| | Ресурс процесса |

В отличие от IDFE0, которая, несмотря на возможность в полной мере описать всю деятельность организации, обычно используется для описания верхних уровней, нотация IDFE3 чаще применяется для процессов нижнего уровня [14]. Графические символы, используемые в нотации, представлены в таблице 2.3.

Таблица 2.3 – Графические элементы IDEF3

| Графический элемент | Описание |
|---|---|
|  | Процесс |
|  | Ссылочный объект |
|  J1 | Асинхронный/синхронный разветвитель/соединитель «И» (все последующие/предшествующие работы должны быть запущены/завершены) |
|  J2 | Асинхронный/синхронный разветвитель/соединитель «ИЛИ» (несколько последующих/предшествующих работ должны быть запущены/завершены) |
|  | Связь отношения (процесс В может начаться и закончиться до завершения А) |
| <p>Документ</p>  | Связь потоков объектов (процесс В начинается после завершения А и использует Документ, полученный в А) |
|  J3 | Разветвитель/соединитель исключаяющий «ИЛИ» (только одна последующая/предшествующая работа должна быть запущена/завершена) |

| Графический элемент | Описание |
|---|--|
|  | <p>Связь предшествования (процесс В начинает выполняться после завершения А)</p> |

2.3.2. Описание ключевых бизнес-процессов в модели As-Is

Рассмотрим первый уровень описания процессов, применив нотацию IDEF0, он представлен на рисунке 2.2.

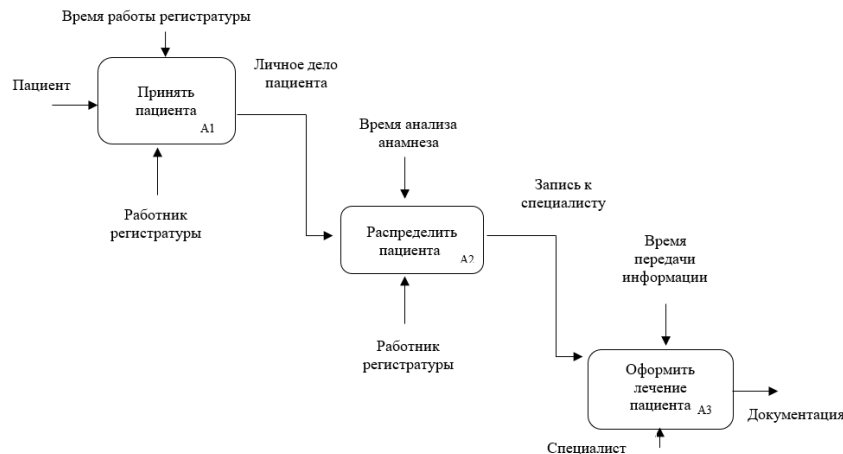


Рисунок 2.2 – Описание процесса «Обслужить пациента» 1 уровня по нотации IDEF0 в модели As-Is

Представленная диаграмма дает общее понимание о работе системы, но для полноценного анализа необходима декомпозиция. Для этого применима нотация IDEF3 [4]. Диаграмма представлена на рисунке 2.3.

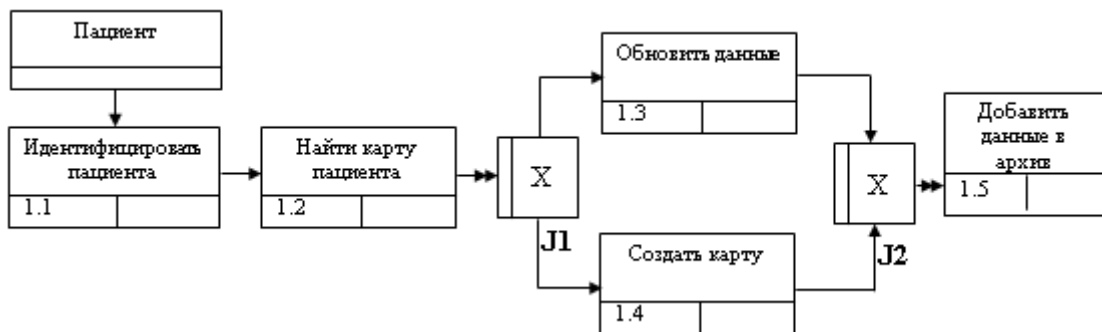


Рисунок 2.3 – Описание процесса «Принять пациента» 2 уровня по нотации IDEF3 в модели As-Is

Данная диаграмма значительно дополняет представление о происходящих процессах, однако для учета действий, необходимых для подробного описания ключевого бизнес-процесса необходима дополнительная декомпозиция [10]. Схема третьего уровня описания процессов представлена на рисунке 2.4.

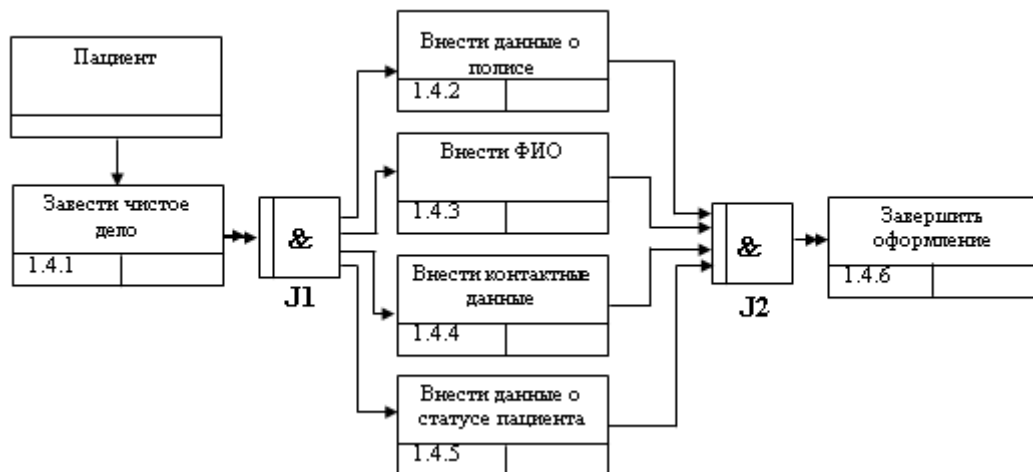


Рисунок 2.4 – Описание процесса «Создать карту» 3 уровня по нотации IDEF3 в модели As-Is

Детализация второго бизнес-процесса рисунок 2.5. Был выбран элемент, исключаящий выполнение двух последующих процессов одновременно, это позволяет ориентироваться по задачам, которые

необходимо реализовать на данном этапе, в зависимости от исходных данных [20].

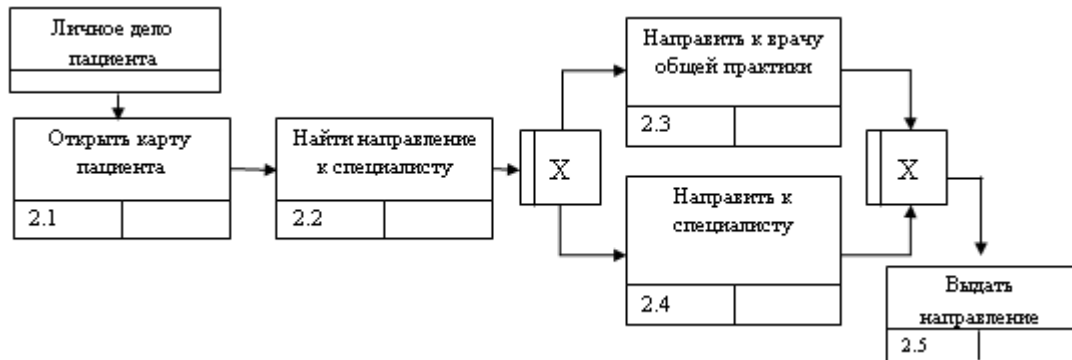


Рисунок 2.5 – Второй уровень описания процесса «Распределить пациента» по нотации IDEF3 в модели As-Is

Переходим к третьему уровню, в данном случае используем асинхронный разветвитель для параллельного выполнения нескольких процессов рисунок 2.6.

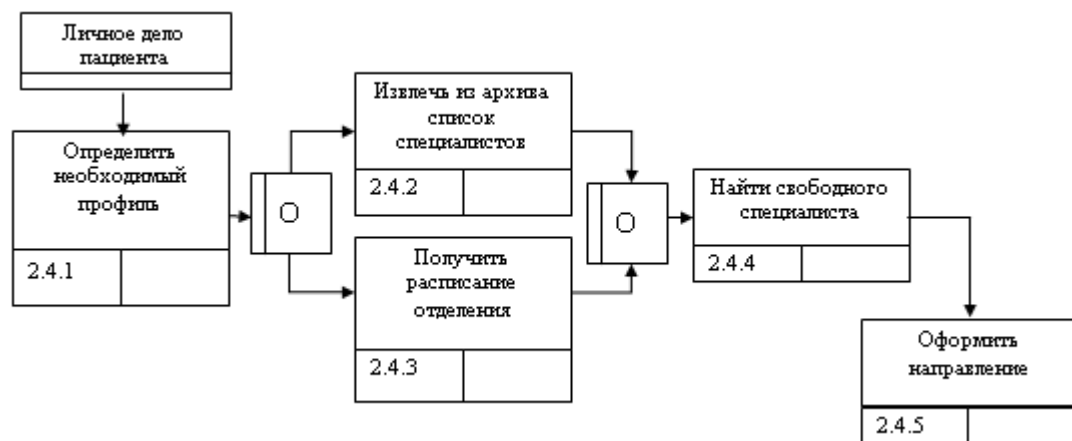


Рисунок 2.6 – Третий уровень описания процесса «Получить информацию о необходимом специалисте» по нотации IDEF3 в модели As-Is

Детализация третьего бизнес-процесса представлено на рисунке 2.7.

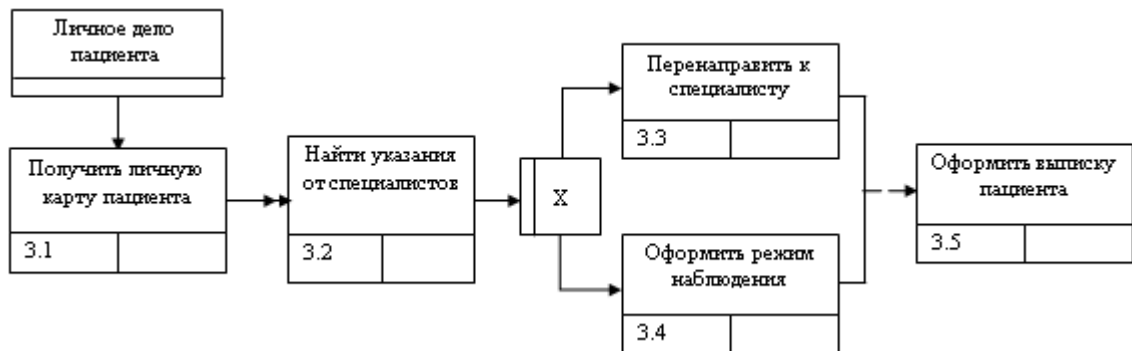


Рисунок 2.7 – Второй уровень описания процесса «Оформить лечение пациента» по нотации IDEF3 в модели As-Is

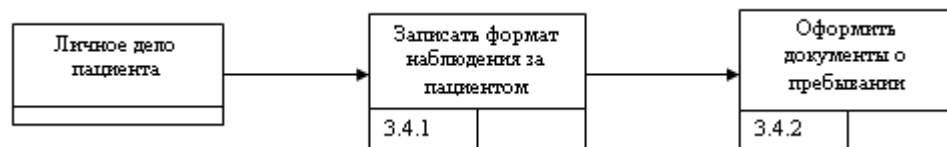


Рисунок 2.8 – Третий уровень описания процесса «Выписать» по нотации IDEF3 в модели As-Is

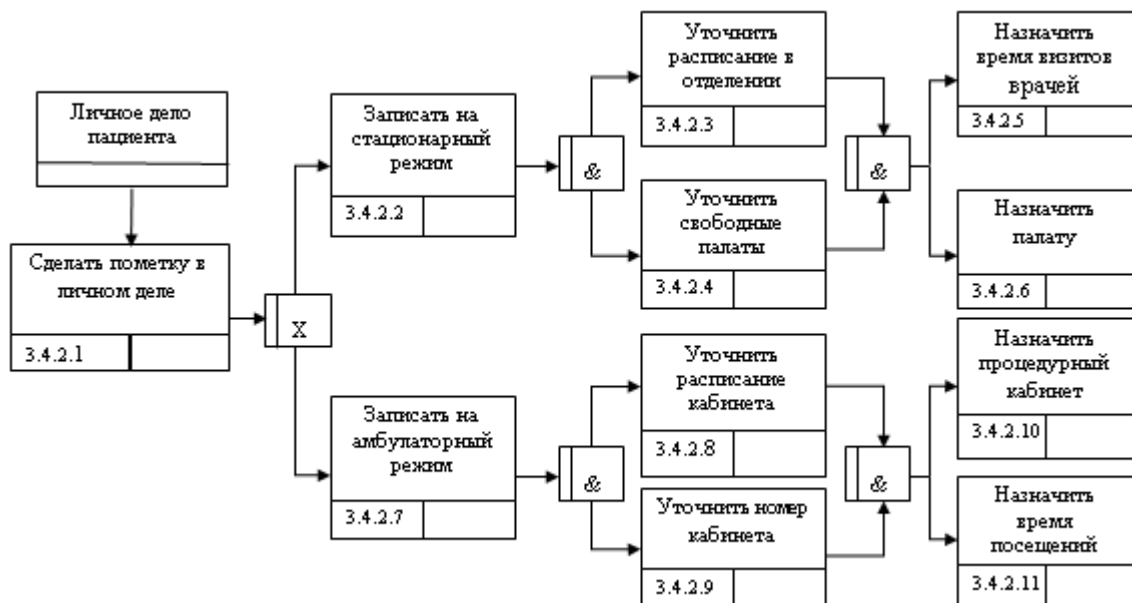


Рисунок 2.9 – Четвертый уровень описания процесса «Выбрать формат наблюдения» по нотации IDEF3 в модели As-Is

2.3.3. Описание ключевых бизнес-процессов в модели To-Be

Для наглядного отображения внедрения электронной базы данных используется модель To-Be. Схема первого уровня описания процессов после внедрения представлена на рисунке 2.10.

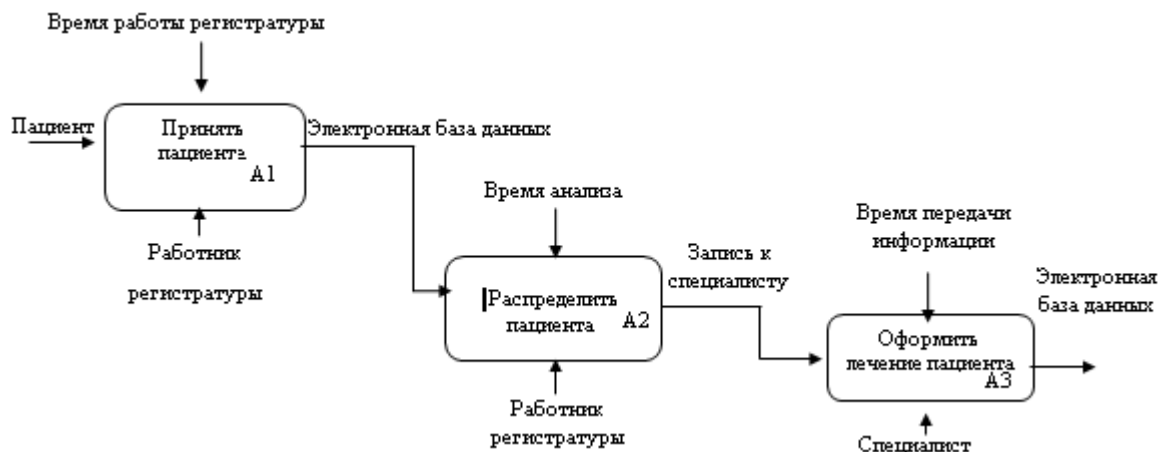


Рисунок 2.10 – Описание процесса «Обслужить пациента» 1 уровня по нотации IDEF0 в модели To-Be

Уже на первом уровне видны изменения и уход от бумажной документации. Второй уровень описания представлен на рисунке 2.11.

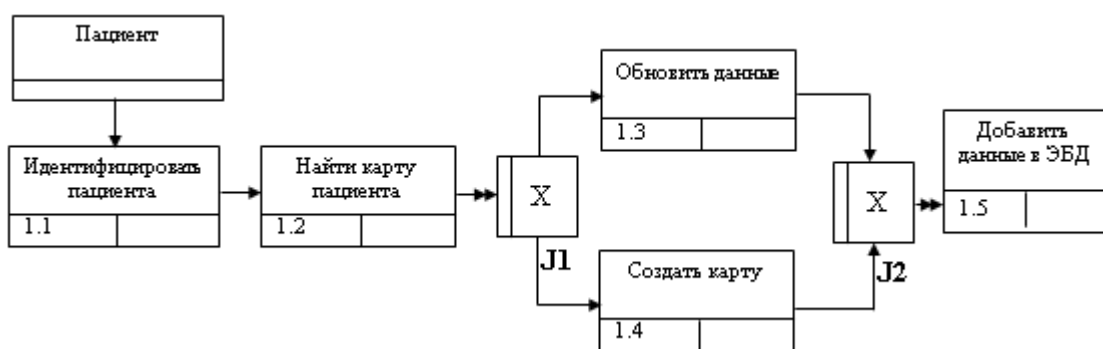


Рисунок 2.11 – Описание процесса «Принять пациента» 2 уровня по нотации IDEF3 в модели To-Be

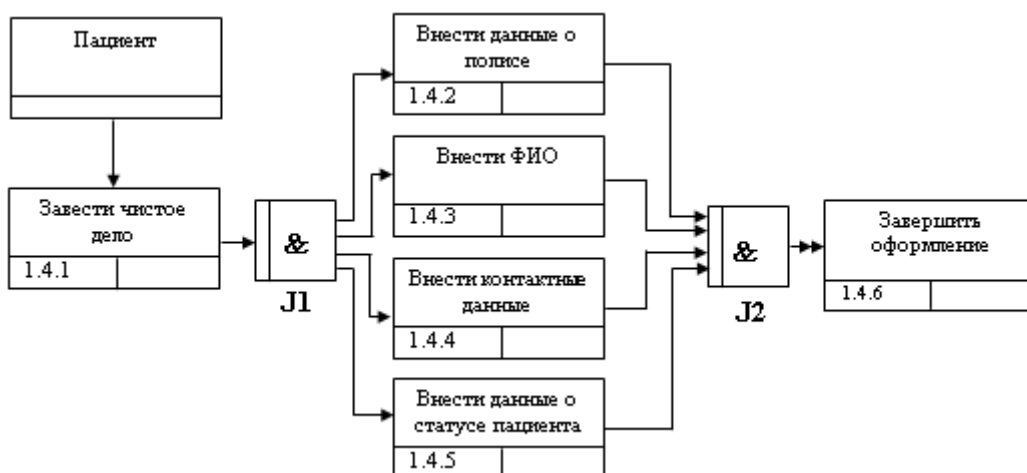


Рисунок 2.12 – Описание процесса «Создать карту» 3 уровня по нотации IDEF3 в модели To-Be

Переходим ко второму бизнес-процессу, на котором так же можно заметить модификации, теперь не требуется дополнительного времени для поиска вручную личных карт пациентов, доступ к анамнезу можно осуществить прямо с рабочего места регистратора [13]. Детализация будет представлена в четырех уровнях, так как этого достаточно для наглядной демонстрации. Второй уровень представлен на рисунке 2.13.

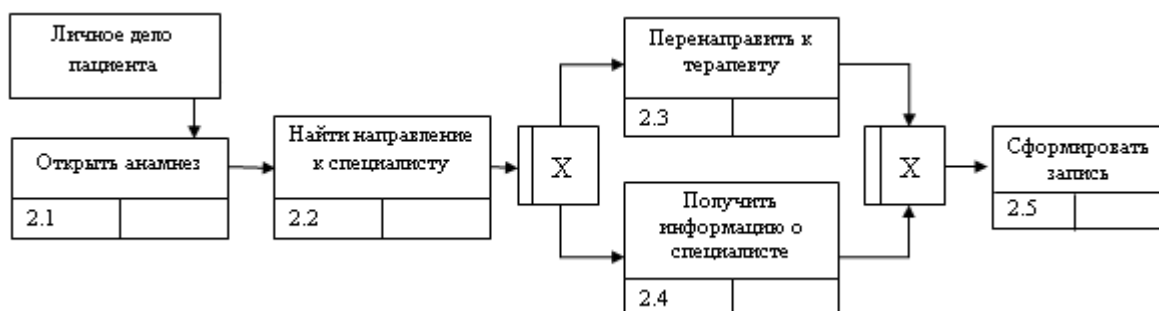


Рисунок 2.13 – Второй уровень описания «Распределить пациента» процесса по нотации IDEF3 в To-Be

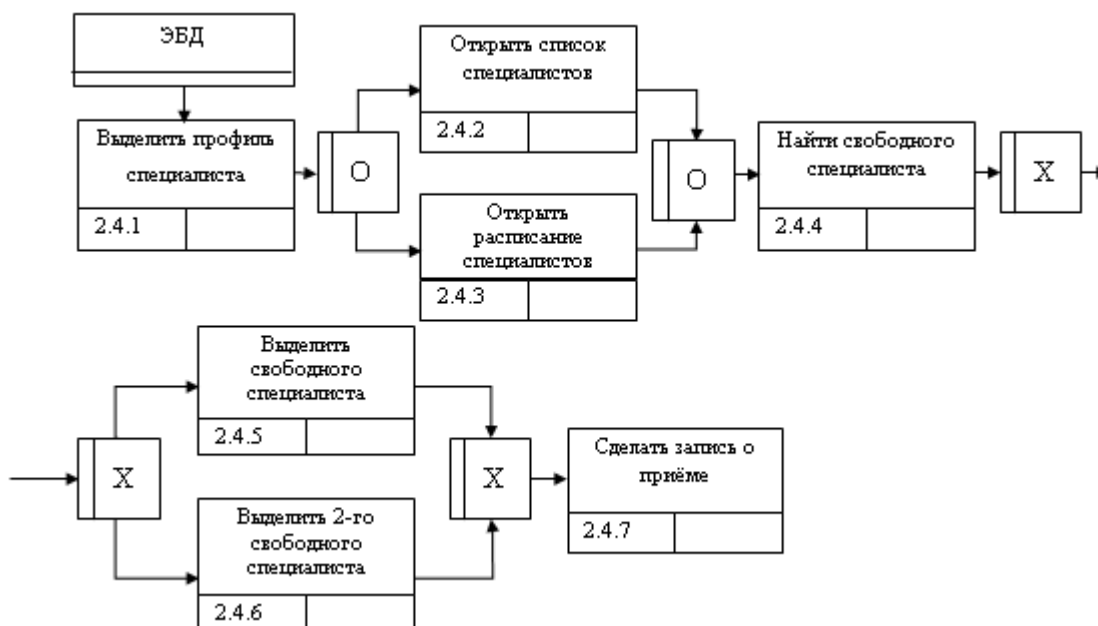


Рисунок 2.14 – Третий уровень описания процесса «Получить информацию о необходимом специалисте» по нотации IDEF3 в модели To-Be

Завершаем детализацию второго бизнес-процесса. Возможность распечатать уже сформулированный талон, хранение его в личном деле. Упрощает поиск, исключает ошибки. Заполнение графика специалиста предотвращает перегруженность. Переходим к третьему бизнес-процессу рисунок 2.15. Упрощение логистики сопутствующих документов. Надежное их хранение. Быстрый доступ.

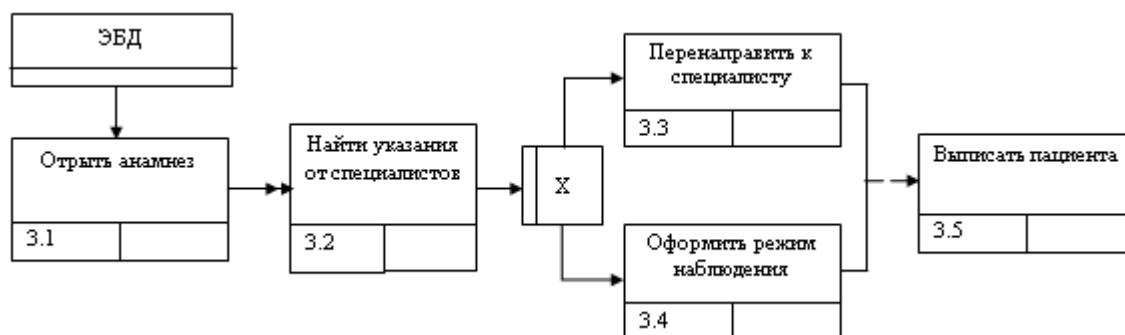


Рисунок 2.15 – Второй уровень описания процесса «Оформить лечение пациента» по нотации IDEF3 в To-Be

Процесс обновления информации в личном деле пациента намного проще, чем это было представлено на схемах As-Is. Возможность выполнения операций, не отходя от рабочего места, без необходимого ожидания документации рисунок 2.16.

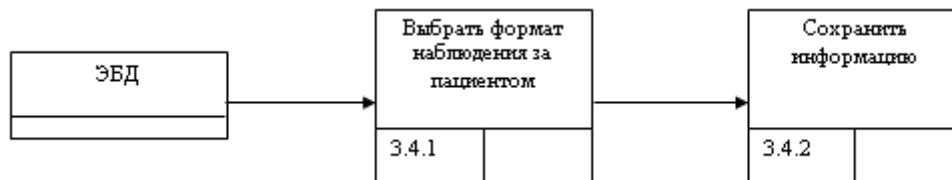


Рисунок 2.16 – Третий уровень описания процесса «Оформить режим наблюдения» по нотации IDEF3 в To-Be

На рисунке 2.17 прослеживается упрощение действий. Если сравнить с тем же уровнем детализации, но модели As-Is, разница весьма заметна и выражается в сокращении действий регистратора для выполнения той же функции. Из чего следует, что скорость обслуживания увеличится.

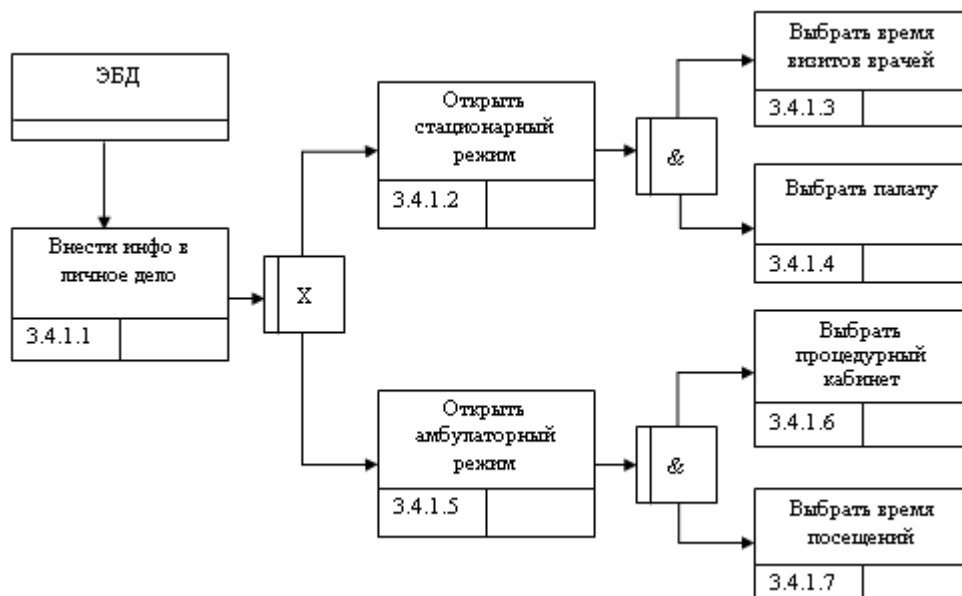


Рисунок 2.17 – Четвертый уровень описания процесса «Выбрать формат наблюдения» по нотации IDEF3 в To-Be

Для получения актуального, полного, детального вида бизнес-процессов используется карта процессов в модели То-Ве, представленная на рисунке 2.18 [15].

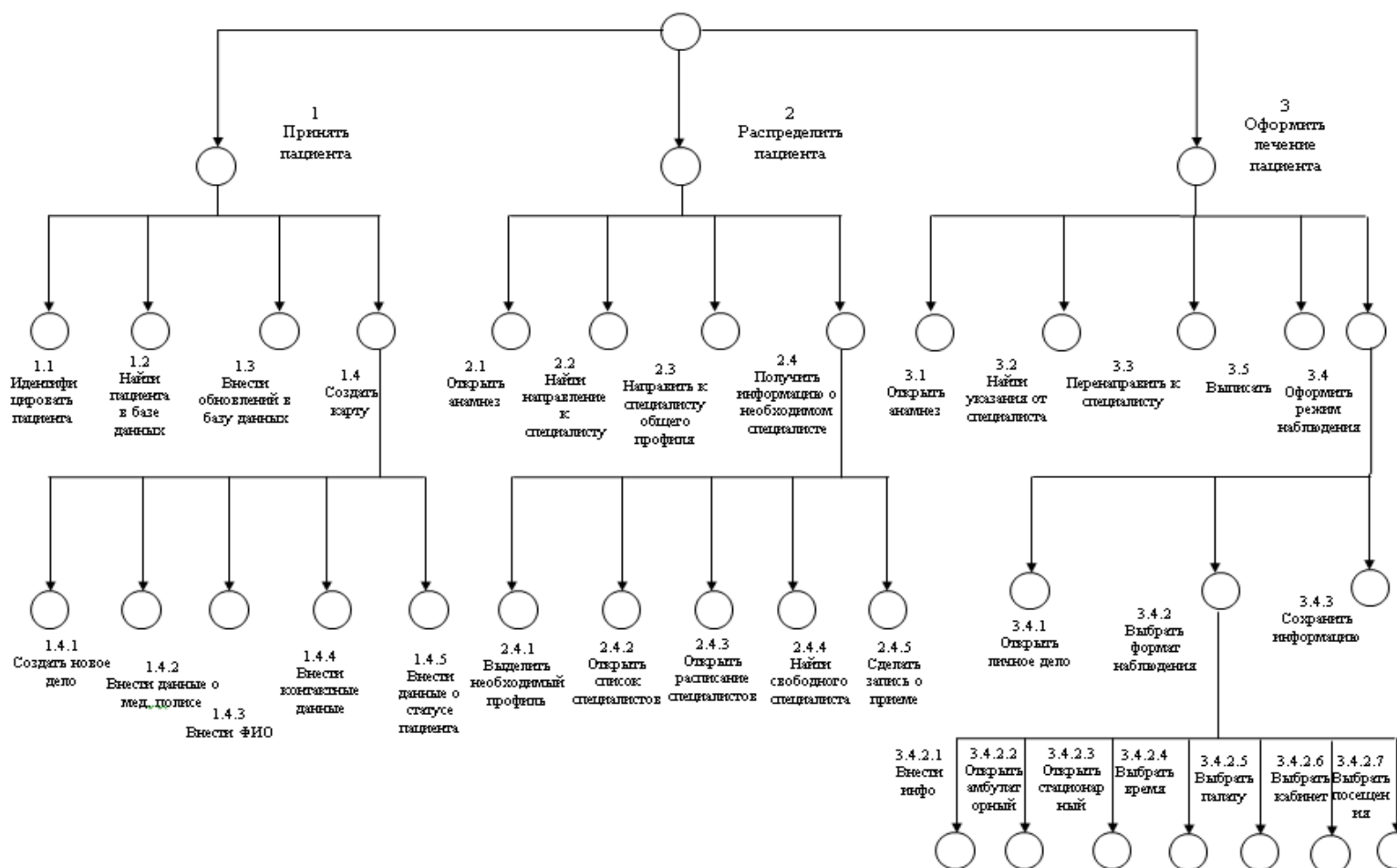


Рисунок 2.18 – Карта процессов в форме To-Be

2.3.4. Архитектура данных

Для реализации процессов будет создана база данных из трех таблиц: личные данные пациента, специалисты, анамнез [5]. Для удобства работы с данными, информация о классах и типах представлена в таблице 2.4.

Таблица 2.4 – Классы и типы данных

| Название класса | Поле | Тип | Количество символов |
|--------------------------|----------------------|-----------|---------------------|
| Личные данные сотрудника | Id ↑ | Числовой | 10 |
| | Фамилия | Текст | 25 |
| | Имя | Текст | 25 |
| | Отчество | Текст | 25 |
| Личные данные пациента | Id ↑ | Числовой | 4 |
| | Фамилия | Текст | 25 |
| | Имя | Текст | 25 |
| | Отчество | Текст | 25 |
| | Пол | Текст | 1 |
| | Телефон | Текст | 30 |
| | Мед. полис | Числовой | 36 |
| | Номер палаты | Числовой | 4 |
| | Статус | Текст | 10 |
| Анамнез | Id ↑ | Числовой | 10 |
| | Дата обращения | Дата | 8 |
| | Дата выписки | Дата | 8 |
| | Id_врача | Character | 30 |
| | Id_пациента | Числовой | 30 |
| | Диагноз | Текст | 30 |
| | Id_назначения | Текст | 30 |
| Рабочие данные | Id ↑ | Числовой | 10 |
| | Id_данные сотрудника | Числовой | 10 |
| | Id_профиль | Числовой | 10 |
| Палаты | Id ↑ | Числовой | 10 |
| | Id_Отделение | Числовой | 10 |
| | Кол-во коек | Числовой | 10 |
| Отделение | Id ↑ | Числовой | 10 |
| | Название | Текст | 30 |
| Прием | Id ↑ | Числовой | 10 |
| | Дата | Числовой | 10 |
| | Id_пациента | Числовой | 10 |
| | Время | Текст | 25 |
| | Id_пациента | Числовой | 10 |

| Название класса | Поле | Тип | Количество символов |
|-----------------|------------------|----------|---------------------|
| | Id_Специалиста | Числовой | 10 |
| Тип лечения | Id 🔑 | Числовой | 10 |
| | Название | Текст | 25 |
| Профиль | Id 🔑 | Числовой | 10 |
| | Название профиля | Текст | 30 |

На рисунке 2.19 приведена схема архитектуры данных и связей между ними. Из схемы видно, что таблица «Личные данные пациента» связаны с таблицей «Специалисты» связью «один ко многим» и с таблицей «Анамнез» связью «один-к-одному».

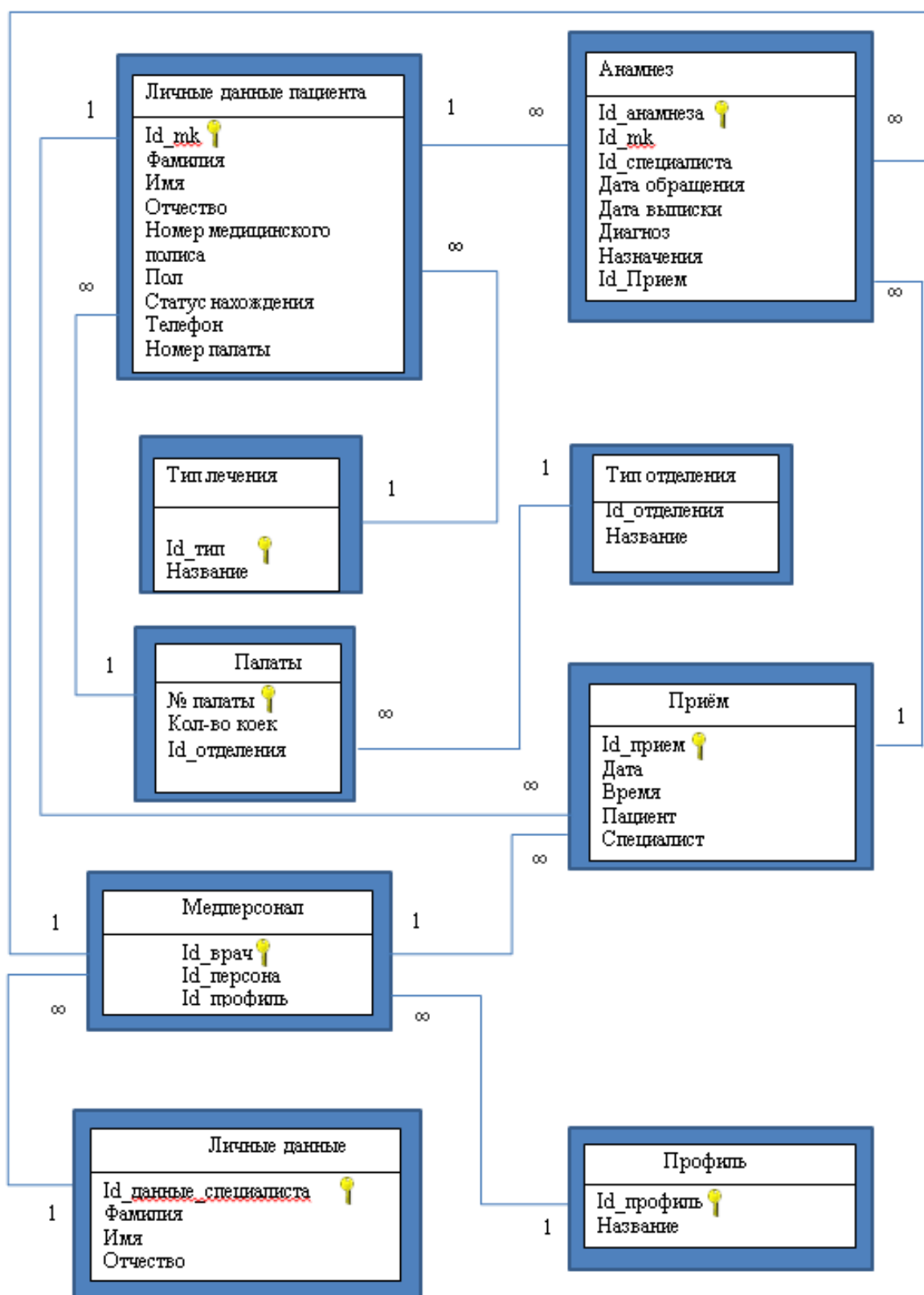


Рисунок 2.19 – Архитектура данных

2.3.5. Проектирование пользовательского интерфейса

Для удобства пользователей необходимо разработать интуитивно понятный интерфейс, отвечающий всем требованиям заказчика [12]. На рисунке 2.20 представлена схема работы приложения.

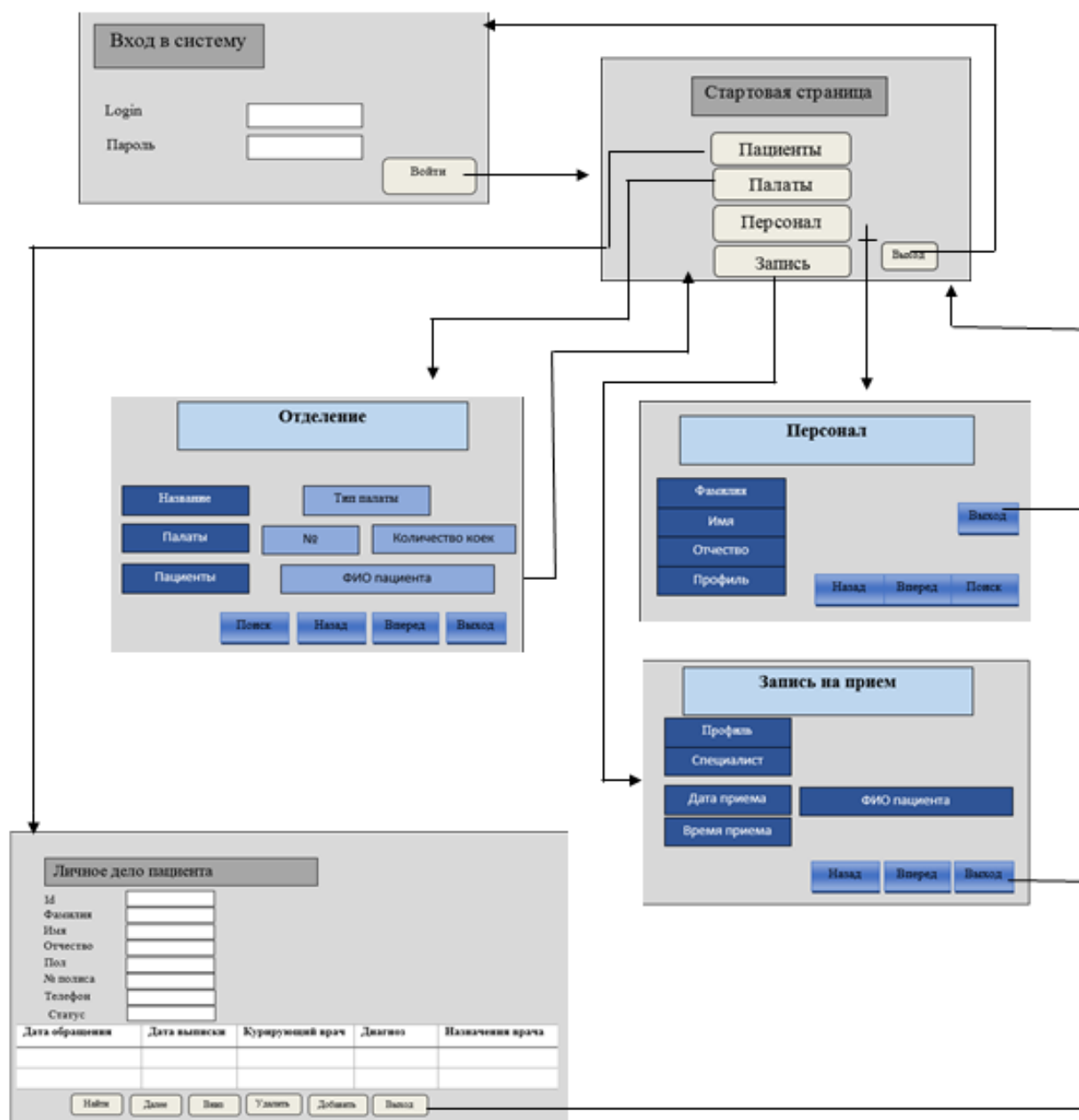


Рисунок 2.20 – Схема работы приложения

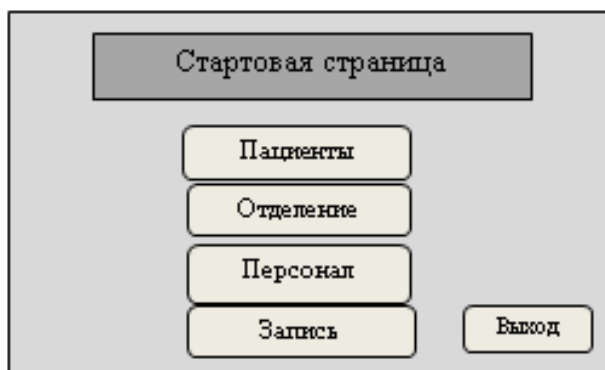


Рисунок 2.21 – *Главное меню*

Главное меню дает доступ к трем основным рабочим формам программы. Форма «Пациенты», позволяющая работать с данными посетителей больницы, представлена на рисунке 2.22. За просмотр и редактирование данных о персонале, а также за работу с расписанием отвечает форма «Персонал», представленная на рисунке 2.23. Форма «Отделение» позволяет обрабатывать данные о нахождении пациентов в стационаре. Форма схематично изображена на рисунке 2.24. Для записи на прием используется форма, представленная на рисунке 2.25. Для лучшего понимания взаимодействия с приложением, была создана блок-схема, подробно описывающая каждый шаг пользователя в процессе работы. Она представлена на рисунке 2.26 [18].

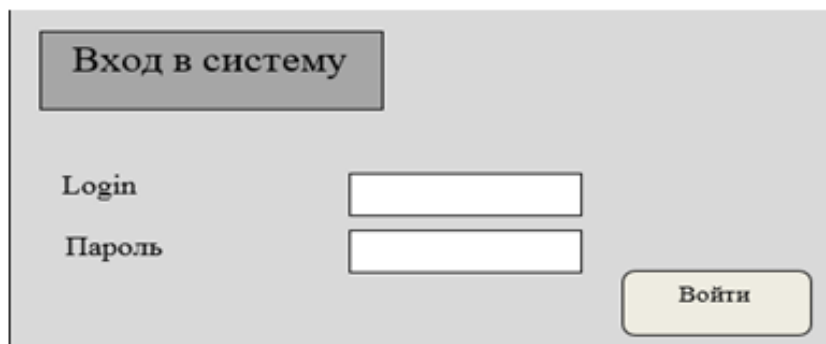


Рисунок 2.22 – *Пациенты*

Персонал

Фамилия

Имя

Отчество

Профиль

Выход

Назад Вперед Поиск

Рисунок 2.23 – Персонал

Отделение

Название

Тип палаты

Палаты

№

Количество коек

Пациенты

ФИО пациента

Поиск Назад Вперед Выход

Рисунок 2.24 – Палаты

Запись на прием

Профиль

Специалист

Дата приема

Время приема

ФИО пациента

Назад Вперед Выход

Рисунок 2.25 – Запись на приём

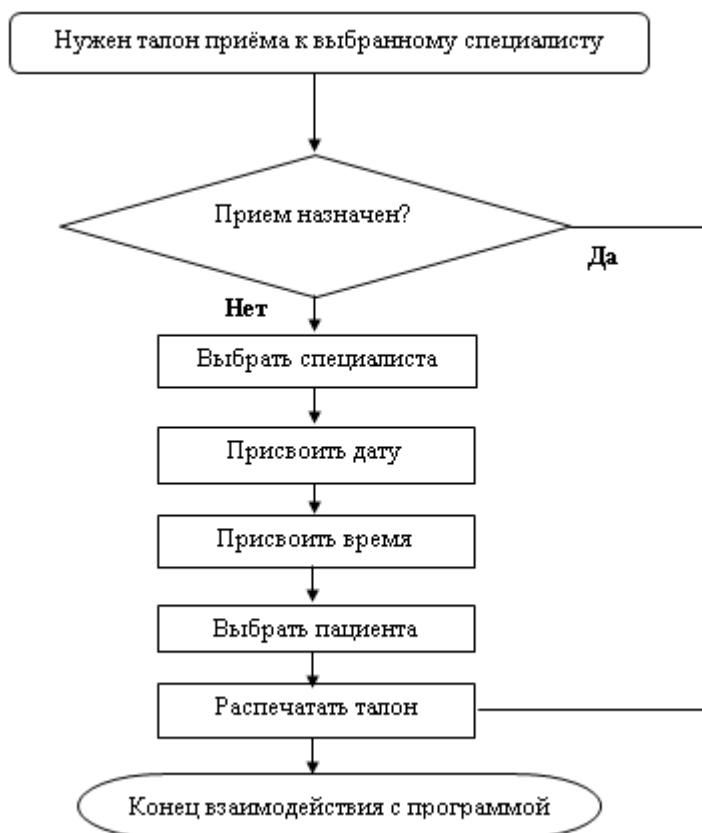


Рисунок 2.26 – Алгоритм работы с формой приёма

Далее на рисунке 2.27 приведен пример использования языка Visual basic for application для внесения данных о приеме в соответствующую таблицу, создания фильтра по дате и времени с последующим выводом информации о приеме на экран, а также выхода в главное меню [11], [17].

```

Option Compare Database
Private Sub Добавить_Click()
    Dim Rst As Recordset
    Set Rst = CurrentDb.OpenRecordset("Приём")
    Rst.AddNew
    Rst.Fields("ID_прием") = IdField
    Rst.Fields("Дата") = Date
    Rst.Fields("Время") = Time
    Rst.Fields("Пациент") = PaciField
    Rst.Fields("Специалист") = SpecField
End Sub
Private Sub Печатьталона_Click()
    DoCmd.OpenReport "талон", acViewNormal, , [Date] = Me.Date And [Time] = Me.Time
End Sub
Private Sub Выход_Click()
    DoCmd.OpenForm "Меню"
    DoCmd.Close
End Sub
  
```

Рисунок 2.27 – Пример использование языка Visual basic for applications

Раздел 3. Программно-алгоритмическая часть

3.1. Реализация основных функций системы

После составления схемы приложения можно приступить к этапу разработки. Для выполнения поставленных задач используется среда MS Access [16]. Согласно требованиям, в первую очередь необходимо создать таблицу для хранения данных о пациенте. Фрагмент таблицы представлен на рисунке 3.1.

| id_mk | Фамилия | Имя | Отчество | Номер мед | Пол | Статус нахождения | Телефон | Номер палаты |
|-------|------------|-----------|----------------|-----------|-----|-------------------|------------------|--------------|
| 1 | Калинин | Игорь | Степанович | 1 | М | Стационарный | 8(909)-965-13-12 | 101 |
| 2 | Егорова | Ирина | Владимировна | 2 | Ж | Стационарный | 8(925)-100-58-36 | 101 |
| 3 | Бермудский | Степан | Борисович | 3 | М | Стационарный | 8(910)-581-64-56 | 101 |
| 4 | Филькина | Любовь | Ивановна | 4 | Ж | Амбулаторный | 8(999)-475-81-03 | 0 |
| 5 | Громов | Василий | Константинович | 5 | М | Амбулаторный | 8(910)-601-47-58 | 0 |
| 6 | Фомин | Лаврентий | Артемович | 6 | М | Амбулаторный | 8(986)-761-35-49 | 0 |
| 7 | Неуклюжева | Валентина | Андреевна | 7 | Ж | Стационарный | 8(909)-232-78-65 | 102 |
| 8 | Протасов | Степан | Викторович | 8 | М | Стационарный | 8(919)-293-48-10 | 103 |
| 9 | Григорьев | Николай | Александрович | 9 | М | Стационарный | 8(989)-387-71-07 | 103 |
| 10 | Астафьева | Мария | Андреевна | 10 | Ж | Стационарный | 8(979)-989-86-65 | 102 |

Рисунок 3.1 – Таблица «Личные данные пациента»

Далее создаются таблицы для дополнения данных о специалистах. Фрагменты таблицы представлены на рисунке 3.2.

| Id_личные | Фамилия | Имя | Отчество |
|-----------|-----------|----------|-------------|
| 1 | Кузовкина | Вероника | Викторовна |
| 2 | Чичваркин | Андрей | Евгеньевич |
| 3 | Волков | Федор | Прокопьевич |
| 4 | Любимова | Светлана | Игоревна |
| 5 | Клинцова | Дарья | Олеговна |
| 6 | Алкиньев | Олег | Петрович |
| 7 | Савилов | Илья | Павлович |

Рисунок 3.2 – Таблица «Данные сотрудников»

Для оптимизации расписания, была создана таблица «Дата приема». Ее фрагмент изображен на рисунке 3.3.

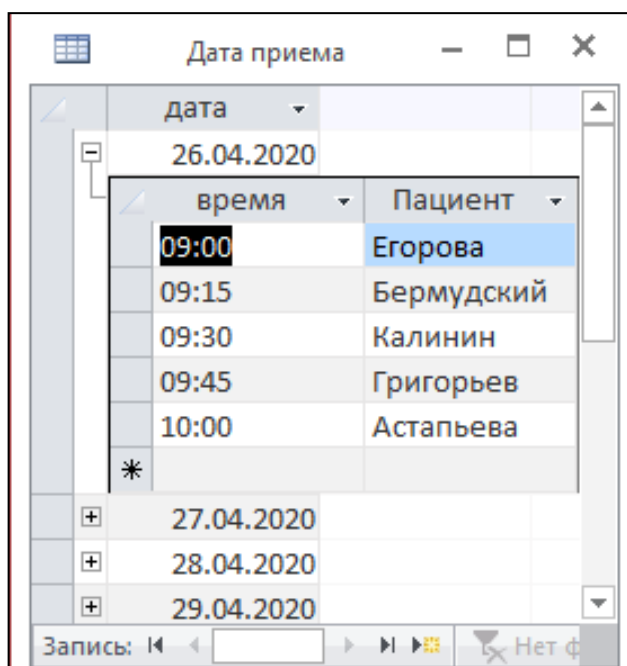


Рисунок 3.3 – Таблица «Дата приема»

Для распределения стационарных больных, была создана таблица «Палаты», фрагмент которой представлен на рисунке 3.4.

| 101 Инфекционная | | 3 | | | | | |
|-----------------------|------------|-----------|-------------|-----------|-----|-------------------|------------------|
| id_mk | Фамилия | Имя | Отчество | Номер мед | Пол | Статус нахождения | Телефон |
| 1 | Калинин | Игорь | Степанович | 1 | М | Стационарный | 8(909)-965-13-12 |
| 2 | Егорова | Ирина | Владимировн | 2 | Ж | Стационарный | 8(925)-100-58-36 |
| 3 | Бермудский | Степан | Борисович | 3 | М | Стационарный | 8(910)-581-64-56 |
| * | 0 | | | | 0 | | |
| 102 Послеоперационная | | 5 | | | | | |
| id_mk | Фамилия | Имя | Отчество | Номер мед | Пол | Статус нахождения | Телефон |
| 7 | Неуклюжева | Валентина | Андреевна | 232312121 | Ж | Стационарный | 8(909)-232-78-65 |
| 10 | Астапова | Мария | Андреевна | | 10 | Ж | 8(979)-989-86-65 |
| * | 0 | | | | 0 | | |

Рисунок 3.4 – Таблица «Палаты»

Для визуализации связей в базе данных на рисунке 3.5 представлена архитектура данных.

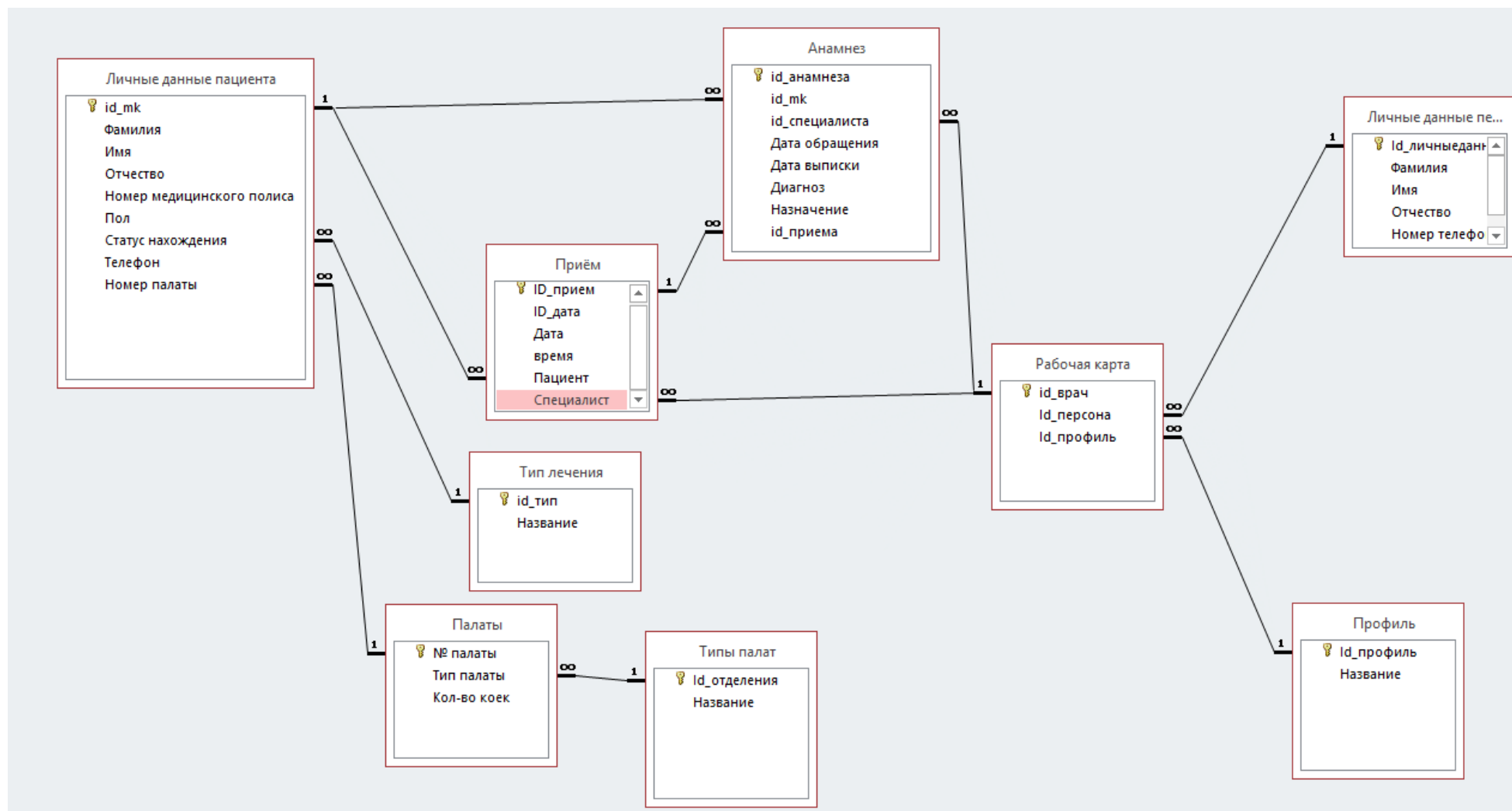


Рисунок 3.5 – Архитектура данных

Для успешной работы с созданными таблицами необходимо создать интерфейс, обеспечивающий, согласно требованиям, создание, редактирование и просмотр данных о пациенте. Для отображения личного дела пациента, а также детализированной истории болезни была создана форма, представленная на рисунке 3.6.

The screenshot shows a web application window titled 'Пациент'. It contains a form with the following fields and values:

| Field | Value |
|---------------------------|------------------|
| id_mk | 7 |
| Фамилия | Неуклюжева |
| Имя | Валентина |
| Отчество | Андреевна |
| Номер медицинского полиса | 232312121 |
| Пол | Ж |
| Статус нахождения | 2 |
| Телефон | 8(909)-232-78-65 |

Below the form is a table with the following data:

| Дата обращения | Дата выписки | Диагноз | Назначенный режим |
|----------------|--------------|---------|-------------------|
| 18.04.2020 | 19.04.2020 | Диарея | Амбулаторно |
| 20.04.2020 | 21.04.2020 | Рвота | Амбулаторно |

At the bottom of the window, there are five icons (back, forward, print, delete, and check) and a red button labeled 'Выход'.

Рисунок 3.6 – Личное дело пациента

Затем была создана форма «Персонал», для реализации таких требований, как: «функция поиска направлений к специалистам», «функция формирования и просмотра списка специалистов», «функция заполнения графика специалистов», «функция доступа к расписанию специалистов» и «функция отбора специалиста по заданным параметрам». Форма представлена на рисунке 3.7.

Для удобной работы с данными пациентов, находящихся на стационарном лечении, была создана форма «Палаты», позволяющая

систематизировать информацию о нахождении больных. Форма представлена на рисунке 3.8.

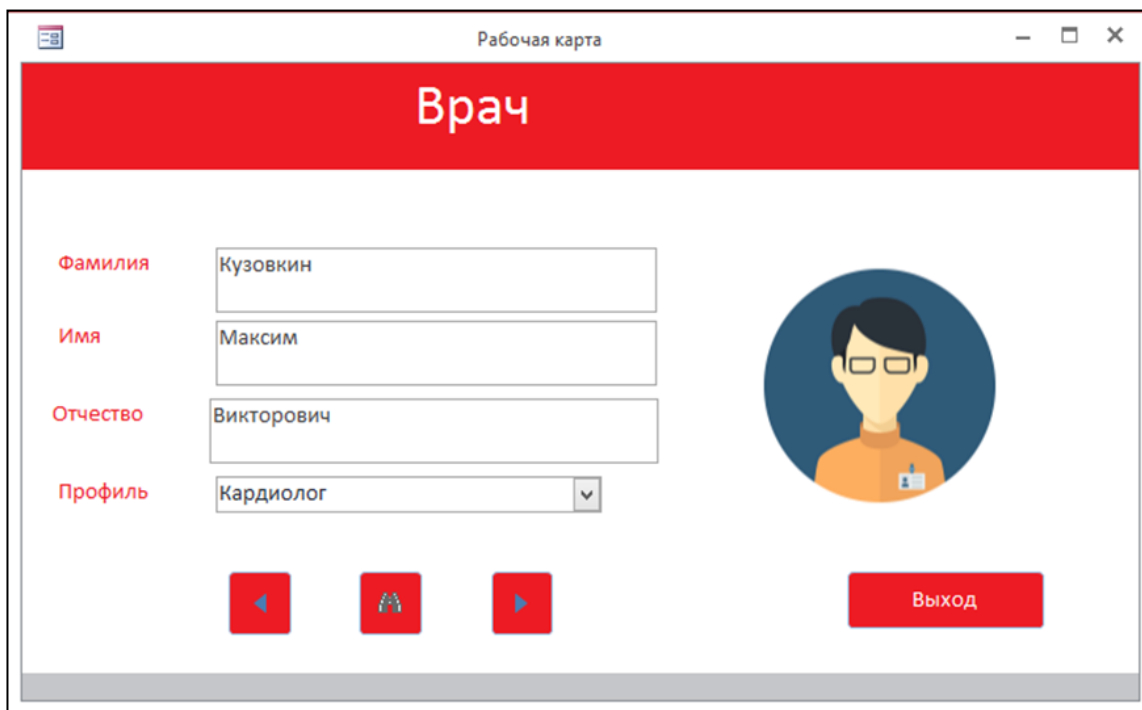


Рисунок 3.7 – Персонал

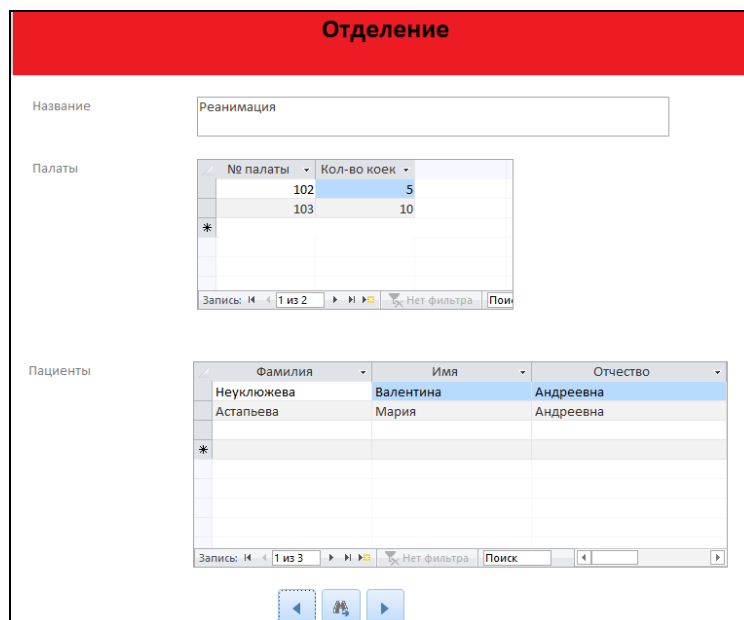


Рисунок 3.8 – Палаты

Запись на прием представлена на рисунке 3.9. Для обеспечения безопасного доступа к данным создается форма входа в систему с паролем. Меню входа представлено на рисунке 3.10, а для удобной навигации, было создано главное меню, представленное на рисунке 3.11.

Профиль

Кардиолог

Специалист

| id_врач | Фамилия | Имя | Отчество |
|---------|-----------|--------|------------|
| 1 | Кузовкин | Максим | Викторович |
| 5 | Чичваркин | Андрей | Евгеньевич |
| * | (№) | | |

Записи: 14 1 из 2

Дата и время приема

| дата | время | Пациент |
|------------|-------|------------|
| 26.04.2020 | 09:00 | Егорова |
| 26.04.2020 | 09:15 | Бермудский |
| 26.04.2020 | 09:30 | Калинин |
| 26.04.2020 | 09:45 | Григорьев |
| 27.04.2020 | 14:00 | Филькина |

Записи: 14 1 из 11

Рисунок 3.9 – Запись на прием

Вход

Логин:

Пароль:

Вход

Рисунок 3.10 – Меню входа

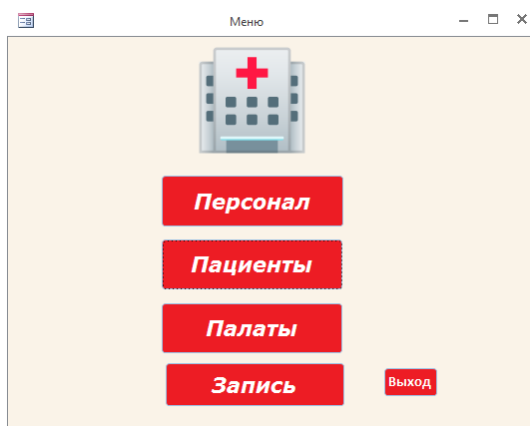


Рисунок 3.11 – *Главное меню*

3.2. Функциональное тестирование

Для анализа адекватности работы программы, а также соответствия финальной версии программы требованиям заказчика, согласно выбранной методологии проводится функциональное тестирование. Рассмотрим реализацию требований:

- Функция открытия личного дела пациента:
 - таблица, содержащая информацию о персональных данных пациента, форма личного дела с информацией из таблицы.
- Функция редактирования личных и иных данных:
 - таблица, содержащая информацию анамнеза пациента, форма личного дела с возможностью редактирования.
- Функция поиска по базе данных:
 - форма с личными делами пациентов с возможностью поиска.
- Возможность вести учет и классификацию палат:
 - форма учета и классификации палат.
- Обеспечение доступа к анамнезу:
 - форма личного дела, содержащая анамнез.
- Возможность поиска направлений к специалисту:

- форма персонала, где ведется расписание врачей с заполненными посещениями.
- Функция формирования и просмотра списка специалистов:
 - форма с информацией о всех специалистах в больнице.
- Функция заполнения графика специалистов:
 - форма персонала с возможность дополнять расписание специалиста.
- Функция доступа к расписанию специалистов:
 - форма персонала с возможностью просмотра расписания специалистов.
- Функция отбора специалиста по заданным параметрам:
 - форма персонала с возможностью поиска по параметрам.
- Функция сбора нескольких (определенных) данных для распечатки пациенту:
 - различные формы для отображения и печати.
- Функция учета пациентов на стационарном режиме:
 - форма поиска пациентов, способная посчитать количество пациентов на стационарном режиме.
- Функция выбора амбулаторного или стационарного наблюдения:
 - форма личного дела пациента с возможностью выбора режима наблюдения.
- Функция учета пациентов на амбулаторном режиме:
 - форма поиска пациентов, способная посчитать количество пациентов на амбулаторном режиме.

3.3. Нагрузочное тестирование

Нагрузочное тестирование помогает оценить быстродействие системы, что, несомненно, является ключевым фактором в работе регистратуры [19].

Тестирование проводилось путем подсчёта времени, необходимого для выполнения операций [2]. Было произведено по 5 измерений с помощью секундомера для разного количества записей. Определив среднее время и отклонения времени для различного количества пациентов определим время отклика.

Среднее арифметическое для всех измерений рассчитывается:

$$t_{\text{ср.арифм.}} = \frac{\sum_{i=1}^n t_i}{n}, \quad (3.1)$$

где t_i – время отклика, n – количество измерений при тестировании.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (t_i - t_{\text{ср.арифм.}})^2}{n}}. \quad (3.2)$$

Среднеквадратическое отклонение имеет вид:

Погрешность измерения выражается:

$$\Delta t = \sqrt{\left(\frac{\sigma}{n} \cdot t_{\alpha(N-1)}\right)^2 + \Delta t_p^2}, \quad (3.3)$$

где $t_{\alpha(N-1)} = 2,571$ – доверительный коэффициент Стьюдента, $\Delta t_p = 0,005$ – абсолютная погрешность измерительного прибора (секундомера).

Итоговое время отклика равно:

$$t_{\text{отк}} = t_{\text{ср.ар.}} \pm \Delta t. \quad (3.4)$$

Таблица 3.1 – Нагрузочное тестирование

| Количество личных дел | Время отклика | |
|-----------------------------|----------------|---------------------------|
| | Функция поиска | Вывод информации на экран |
| 1 | 0,12±0,02с | 0,12±0,02с |
| 25 | 0,2±0,03с | 0,2±0,03с |
| 50 | 0,25±0,04с | 0,25±0,03с |
| 100 | 0,3±0,07с | 0,35±0,06с |

Таким образом, в результате проведения измерений с помощью секундомера для разного количества записей, а именно: 1, 25, 50, 100, можно утверждать, что приложение успешно прошло нагрузочное тестирование.

Раздел 4. Экономическая часть

В данной ВКР производилась разработка электронной регистратуры больницы с применением водопадной модели внедрения информационных систем. Для обоснования экономической целесообразности проведения исследования произведем затрат по статьям:

- материальные затраты;
- заработная плата исполнителей работы с учетом страховых взносов;
- амортизация оборудования;
- прочие затраты.

Исходя из наличия необходимого технического оборудования и отсутствия программного обеспечения, следует необходимость совершить покупку лицензии на MS Access. Так же будет произведен учет затраченных средств на электроэнергию.

4.1. Затраты на оплату труда

Для выполнения поставленной задачи необходим штат из двух сотрудников: разработчик-программист и аналитик. Основная заработная плата начисляется исходя из ставки нанятого сотрудника и времени затрачиваемого на выполнение работы. Для данного расчета примем, аналитик имеет ставку 40000 руб., и разработчик-программист имеет ставку 45000 руб. В таблице 3.1 представлены этапы проведения работ с распределением сроков.

Таблица 4.1 – План выполнения

| Этап | Срок | Исполнитель |
|--|--------|-------------|
| Сбор информации, взаимодействие с заказчиком проекта | 7 дней | Аналитик |

| Этап | Срок | Исполнитель |
|---|--------|-----------------------------------|
| Выработка системных требований | 2 дня | Аналитик |
| Формирование технического задания | 4 дня | Аналитик |
| Оформление проектной документации со всеми входными данными для реализации проекта | 2 дня | Аналитик |
| Проверяется соответствие исходным задачам, целостность и непротиворечивость запросов. Взаимодействие сторон заказчика и исполнителя | 1 день | Аналитик, разработчик-программист |
| Проектирование. Разложение задачи на реализуемые процессы | 7 дней | Разработчик-программист |
| Реализация программы в соответствии с составленной документацией | 7 дней | Разработчик программист |
| Тестирование получившегося продукта и передача в эксплуатацию | 2 дня | Разработчик-программист |

План выполнения работ составлен исходя из поочередного задействования сотрудников в проекте, т.е. первую половину работ совершает аналитик, а вторую разработчик-программист. Существует один рабочий день на этапе проверки соответствия исходным задачам и целостности запросов, когда требуется задействование всех имеющихся исполнителей.

Таким образом, исходя из затрат времени на разработку (разработчик-программист – 17 дней, аналитик – 16 дней, количество рабочих дней в одном месяце составляет в среднем 22), заработная плата будет рассчитана по формуле 4.1.

$$ЗП = (\text{Ставка} / \text{Рабочие дни в месяце}) * \text{Рабочие дни фактические} \quad (4.1)$$

Теперь рассчитаем заработную плату для каждого специалиста и общую сумму затрат по данному пункту.

$$ЗП_{\text{аналитик}} = (40000 / 22) * 16 = 29\,090,9 \text{ руб.}, \quad (4.2)$$

$$ЗП_{\text{разработчик}} = (45000 / 22) * 17 = 34\,772,7 \text{ руб.}, \quad (4.3)$$

$$З_{\text{осн}} = ЗП_{\text{разработчик}} + ЗП_{\text{аналитик}} = 34\,772,7 + 27\,272,7 = 63\,863,6 \text{ руб.} \quad (4.4)$$

4.2. Амортизация

Амортизационные отчисления рассчитывается по формуле 4.5.

$$A_{\text{нир}} = \frac{\Phi_{\text{перв}} \times T_{\text{и}} \times H_{\text{а}}}{\Phi_{\text{эф}}}, \quad (4.5)$$

$\Phi_{\text{перв}}$ – первоначальная стоимость оборудования, руб., $T_{\text{и}}$ – время использования оборудования при проведении работ, дн., $H_{\text{а}}$ – норма амортизации, $\Phi_{\text{эф}}$ – годовой эффективный фонд времени работы оборудования, для односменной работы он составляет 256 дня.

$$H_{\text{а}} = \frac{1}{T_{\text{ни}}}, \quad (4.6)$$

$T_{\text{ни}}$ – срок службы оборудования, лет.

Средний срок службы компьютера составляет 5 лет, тогда норма амортизации рассчитывается по формуле 4.7.

$$H_{\text{а}}(\text{комп.}) = 1 / 5 = 0,2. \quad (4.7)$$

Средний срок службы принтера составляет 5 лет, тогда норма амортизации рассчитывается по формуле 4.8.

$$H_{\text{а}}(\text{принт.}) = 1 / 5 = 0,2. \quad (4.8)$$

Амортизационные отчисления для компьютера стоимостью в 43 530 руб. (моноблок LENOVO V530-24ICB, интернет магазин www.citilink.ru) за 32 дня его использования составят:

$$A_{\text{нир}}(\text{комп.}) = (43\,530 * 32 * 0,2) / 256 = 1088,3 \text{ руб.} \quad (4.9)$$

Амортизационные отчисления для принтера стоимостью в 12 490 руб. (принтер струйный EPSON L132, интернет магазин – www.citilink.ru) за 32 дня его использования рассчитываются по формуле 4.10:

$$A_{\text{нпр}}(\text{принт}) = (12\,490 * 32 * 0,2) / 256 = 312,2 \text{ руб.} \quad (4.10)$$

$$A_{\text{нпр}} = 1088,3 + 312,2 = 1400,5 \text{ руб.} \quad (4.11)$$

4.3. Материальные затраты

Необходимая компьютерная техника как инструмент для работы имеется в наличии, а соответственно не приобреталась дополнительно. Однако, программное обеспечение отсутствует. Поэтому к материальным затратам необходимо добавить затраты на приобретение лицензии на MS Access. Стоимость Microsoft Office 365 Бизнес Премиум составляет 10 990 руб. В процессе работы будет задействован принтер, как средство для печати информации и документов. Сам принтер, как и компьютерная техника, присутствует, однако в дополнительных затратах необходимо указать расходные материалы, а именно сменный картридж (EPSON T6641) для печати и три упаковки офисной бумаги (бумага Xerox Performer A4): $360 + 235 * 3 = 1\,065$ руб. Кроме этого, к материальным затратам отнесем сумму на оплату электроэнергии.

Затраты на электроэнергию зависят от стоимости машинного часа и времени эксплуатации оборудования и определяются по формуле

$$Z_{\text{эл}} = P \times C_{\text{эл}} \times T_{\text{и}}, \quad (4.12)$$

P – потребляемая мощность оборудования, кВт/ч, $C_{\text{эл}}$ – стоимость 1 кВт/ч, руб., $T_{\text{и}}$ – время использования оборудования при проведении работ, ч.

Для выполнения работы использовался персональный компьютер моноблок LENOVO V530-24ICB с потребляемой мощностью 90 Вт/ч (информация представлена на официальном сайте торговой марки –

www.lenovo.com) и принтер струйный EPSON L132 потребляемой мощностью 10 Вт (данные взяты с официального сайта производителя – epson.ru). Время работы ПК составило 32 дня по 8 часов в день, а принтера – 32 дня по 1 часу в день.

Стоимость 1 кВт в г. Москве в здании без газового оснащения на 01.05.2020 г., составляет 4,65 руб./кВт (по данным с официального сайта правительства города Москвы – mosenergosbyt.ru).

$$З_{эл} = 0,09 * 4,65 * 32 * 8 + 0,01 * 4,65 * 32 * 1 = 107,1 + 1,5 = 108,6 \text{ руб.} \quad (4.13)$$

Следовательно, материальные затраты составили:

$$10\,990 + 1\,065 + 108,6 = 12\,163,6 \text{ руб.} \quad (4.14)$$

4.4. Страховые затраты

Рассчитаем прочие затраты на разработку АРМ ($З_{пр}$) как сумму затрат на страховые взносы ($З_{страх}$). Ставка страховых взносов в 2020 г. составляет 30,2% от величины фонда оплаты труда. Таким образом:

$$З_{страх} = 63\,863,6 * 0,302 = 19\,286,8 \text{ руб.} \quad (4.15)$$

4.5. Смета затрат

Общие затраты на разработку АРМ составят:

$$З = З_{прям} + З_{страх} = 77\,427,7 + 19\,286,8 = 96\,714,5 \text{ руб.} \quad (4.16)$$

В таблице 4.2 представлена смета затрат на разработку электронной регистратуры в городской больнице на основе водопадной методологии внедрения.

Таблица 4.2 – Смета затрат

| Наименование статьи затрат | Сумма, руб. | Удельный вес, % |
|----------------------------|-------------|-----------------|
| Материальные затраты | 12 163,6 | 12.58 |
| Затраты на оплату труда | 63 863,6 | 66.03 |

| Наименование статьи затрат | Сумма, руб. | Удельный вес, % |
|----------------------------|-------------|-----------------|
| Амортизация оборудования | 1400,5 | 1.45 |
| Страховые затраты | 19 286,8 | 19.94 |
| Общие затраты | 96 714,5 | 100 |

На рисунке 4.1 представлена диаграмма затрат на разработку автоматизированного рабочего места.



Рисунок 4.1 – *Распределение затрат на реализацию проекта по статьям*

Таким образом, 66,03% затрат, необходимых на разработку АРМ, составляют затраты на заработную плату исполнителей. Страховые затраты составляют 19,94%, материальные затраты – 12,58%. Наименьшая сумма затрат приходится на амортизацию оборудования – 1,45%.

Заключение

В данной работе был изучен и проработан каскадный метод внедрения информационной системы. Была дана оценка целесообразности применения данного метода для разработки приложения электронной регистратуры больницы. Ключевые бизнес-процессы были спроектированы и детализированы в моделях AS-IS и TO-BE с помощью нотаций IDEF0 и IDEF3 для достаточной наглядности изменений, которые принесет разрабатываемая система. Убедившись в корректном исполнении требуемых улучшений, можно сделать вывод о возможности дальнейшей разработки данного проекта. На следующем этапе был произведён сбор пользовательских требований и выявление функциональных. Чтобы структурировать предстоящую к выполнению задачу были выявлены этапы разработки, спроектированы все сопутствующие таблицы.

Далее было создано непосредственно программное обеспечение. Был разработан интерфейс и проработаны взаимосвязи компонентов. Стабильность работы приложения была проверена, благодаря функциональному и нагрузочному тестированию.

Проект можно считать успешным, так как все поставленные задачи выполнены в полном объеме, система не дала сбоев при тестировании, а значит, может быть интегрирована в область применения. Также можно утверждать о несомненной актуальности продукта и, как следствие, возможности дальнейшего его развития.

Список использованных литературных источников

1. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: основные термины и определения. [Электронный ресурс] // https://stepanovd.com/documents/training/1_erp/7-processlevel/processlevel.pdf (Дата обращения 20.12.2018).
2. Куликов С.С. Тестирование программного обеспечения. Базовый курс. – Минск: Четыре четверти – 2017 – 312 с.
3. Берг Д.Б., Ульянова Е.А., Добряк П.В. Модели жизненного цикла. Учебное пособие – Екб.: Издательство Уральского университета, 2014. – 74 с.
4. Шеер А.В. «Бизнес-процессы: основные понятия, теории, методы» пер. с англ. и ред. Рыбьянец А.А. – 3-е изд. – М.: Вильямс, 2009.
5. Кондратьев В.В., Кузнецов М.Н. Показываем бизнес-процессы: от модели процессов компании до регламентов процедур – М.: Эксмо, 2008.-256 с.
6. Чеботарев В.Г., Громов А.И. Роль субъектности в бизнес-процессах. Бизнес-информатика. – 2013. – № 1 (23). – С. 3-9.
7. Катасонова Н.С. Автоматизация ключевых бизнес-процессов городской больницы на основе каскадной модели внедрения: диплом бакалавра. [Электронный ресурс] // https://stepanovd.com/documents/training/3_vkr/2018/2_katasonova/katasonova_diploma.pdf (Дата обращения 20.12.2018).
8. Harmon P. The scope and evolution of business process management. Handbook on Business Process Management 1. – Berlin: Springer Heidelberg, 2010.-P.37-81.
9. ГОСТ Р ИСО/МЭК ТО 15271-2002.

10. Боронина Л.Н., Сенук З.В. Основы управления проектами. Учебное пособие. – Екатеринбург: Изд-во Урал. ун-та. 2015. – 112 с.
11. Лаврищева Е.М., Петрухин В.А. Методы и средства инженерии программного обеспечения. Учебное пособие. – М.: МФТИ, 2007. – 415 с.
12. Дейт К.Дж. Введение в системы баз данных / пер. с англ. и ред. Птицына К.А. – 8-е изд. – М.: Вильямс, 2016.
13. Ермакова С.Э. Управление бизнес-процессами в медицинской организации. – Самара, 2010 – 142 с.
14. Рамбо Дж., Блаха М. UML 2.0. Объектно-ориентированное моделирование и разработка. 3-е изд. – Спб.: Питер, 2011. – 544 с.
15. Шеер А.В. Моделирование бизнес-процессов – М.: Серебряные нити, 2014. – 219 с.
16. Бекаревич, Ю. Б. Самоучитель Access 2016 – СПб.: БХВ-Петербург, 2017. – 480 с.
17. Кент Б. Экстремальное программирование: разработка через тестирование – СПб.: Питер – 2003.
18. Салихова Э.Ш. Информатизация здравоохранения регионального уровня на основе типовых медицинских информационных систем// Врач и информационные технологии. – 2009. – №1. – С. 36-41.
19. Романов С.В., Дзюбак С.А., Абаева О.П. Пути повышения удовлетворенности пациентов при обслуживании в регистратуре поликлиники медицинской организации системы ФМБА России – ФБУЗ «Приволжский окружной медицинский центр» ФМБА России, – г.Нижний Новгород.
20. ГОСТ 19.701–90 (ИСО 5807–85) «Единая система программной документации».