



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

---

Физико-технологический институт  
Кафедра оптических и биотехнических систем и технологий

---

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА НА ТЕМУ:

**«АНАЛИЗ, ПРОЕКТИРОВАНИЕ, РАЗРАБОТКА  
И ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ ДЛЯ АВТОМАТИЗАЦИИ  
ГОРОДСКОЙ БОЛЬНИЦЫ»**

Студент:  
Орешкина А.М.

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

Москва – 2017

## Аннотация

Выпускная квалификационная работа на тему «Анализ, проектирование, разработка и тестирование приложения для автоматизации городской больницы» содержит 49 страниц пояснительной записки, которая включает в себя четыре главы, введение, заключение и список литературы.

В «Введении» доказываемся актуальность выбранной темы, ставится цель и обозначаются задачи работы.

В главе «Анализ требований» описываются возможные способы сбора требований и выбираются подходящие для работы, анализируются пользовательские требования, на основе которых составляются функциональные требования и матрица отслеживания.

В главе «Проектирование приложения» приводится обоснование выбранных способов проектирования, описываются ключевые бизнес – процессы, строится архитектура данных разрабатываемой программы и пользовательские интерфейсы.

В главе «Разработка программы» описываются методы разработки, а также приводятся копии экранов разработанной программы с указанием порядка обработки данных для трёх бизнес-процессов.

В главе «Тестирование разработанного приложения» приводится описание трёх видов тестирования для проверки работоспособности программы, а также экономическая целесообразность работы.

В «Заключении» подводятся итоги работы, а в «Списке литературы» – список использованных печатных источников и интернет – ресурсов.

## Оглавление

<b>Введение .....</b>	<b>1</b>
<b>Глава 1. Анализ требований .....</b>	<b>3</b>
1.1. Способы анализа требований .....	4
1.2. Пользовательские требования .....	7
1.3. Функциональные требования .....	10
1.4. Матрица отслеживания требований.....	12
<b>Глава 2. Проектирование приложения.....</b>	<b>15</b>
2.1. Способы проектирования.....	16
2.2. Ключевые бизнес-процессы в моделях «AS-IS» и «TO-BE» .....	19
2.3. Архитектура данных разрабатываемой системы.....	23
2.4. Описание разрабатываемого приложения.....	25
<b>Глава 3. Разработка программ .....</b>	<b>28</b>
3.1. Методы разработки.....	28
3.2. Реализация процесса регистрации нового пациента .....	28
3.3. Реализация процесса полного анамнеза пациентов .....	30
3.4. Реализация процесса поиска по выбранному параметру.....	30
<b>Глава 4. Тестирование разработанного приложения .....</b>	<b>32</b>
4.1. Методы тестирования.....	32
4.2. Функциональное тестирование .....	33
4.2. Интеграционное тестирование .....	35
4.3. Нагрузочное тестирование .....	36
4.4. Экономическая целесообразность.....	37
<b>Заключение.....</b>	<b>42</b>
<b>Список литературы .....</b>	<b>43</b>

## Введение

В последнее время всё больше и больше медицинских учреждений подвергаются объединению. Реформы здравоохранения нацелены на централизацию медучреждений и создание единых крупных административно – экономических механизмов управления.

В связи с этим, администрация тех больниц и клиник, к которым присоединяют другие лечебные заведения, сталкивается с серьёзной проблемой – объединение данных и создание общей базы пациентов.

Данная проблема осложняется тем, что подавляющее большинство учреждений в качестве базы данных использует либо свои собственные программы, либо таблицы MS Excel. Несомненно, для маленьких и отдалённых больниц подобное ведение дел может быть вполне приемлемым. Но для крупных центров – категорически нет. Это обусловлено следующими факторами:

- затруднённый поиск информации о пациенте;
- невозможность вывода на экран информации только по 1 искомому пациенту (обилие прочих данных повышает вероятность ошибиться, особенно во время экстренных ситуаций);
- высокая вероятность ошибиться и занести данные по одному пациенту в графу к другому.

Для создания приложения по автоматизации работы более крупных медицинских учреждений целесообразным будет использование системы управления базами данных (СУБД), например, Microsoft (MS) Access. Преимущества данной системы заключаются в следующем:

- простота освоения и возможность использования непрофессиональным программистом (в большинстве случаев базу данных заполняет медицинский персонал);

- возможность работы как на автономном компьютере, так и в локальной сети;
- работа под управлением операционной системы (ОС) Microsoft Windows, которая и установлена во всех медучреждениях;
- возможен импорт файлов из MS Excel.

Таким образом, все вышеперечисленные факты говорят о целесообразности выполнения данной работы. Целью данной работы является разработка автоматизированной системы, обрабатывающей информацию с использованием программного обеспечения персонального компьютера и упрощающей создание отчётов, что должно значительно облегчить работу любого учреждения. В процессе реализации цели должны быть реализованы следующие задачи:

- проанализированы требования, предъявляемые к приложению;
- спроектирована структура программного приложения;
- разработана программа в среде MS Access;
- тестирование разработанного приложения.

## Глава 1. Анализ требований

Анализ требований – это важнейшая часть процесса разработки программного обеспечения (ПО), которая задаёт вектор всей остальной работе. Она включает в себя следующие подпроцессы:

- сбор требований к разрабатываемому ПО;
- систематизацию выявленных требований;
- поиск взаимосвязей и их рационализация;
- документирование полученной информации.

Во время сбора требований необходимо учитывать вероятность возникновения противоречий между тремя заинтересованными сторонами: заказчик, разработчик и конечный пользователь. Успех всего проекта напрямую зависит от качества сбора и анализа требований. Требования к разрабатываемому ПО должны отвечать следующим критериям:

- документируемость (в письменном и/или электронном виде);
- выполнимость;
- тестируемость;
- достаточный (для текущего проекта) уровень детализации.

Анализ требований – это достаточно длительный и трудоёмкий процесс, который состоит из трёх основных этапов:

- сбор информации:
  - общение с заказчиком и конечным пользователем;
  - изучение предметной области;
- анализ собранной информации:
  - систематизация информации;
  - поиск противоречий и неточностей;
  - в случае наличия данных проблем, их решение;

- поиск взаимосвязей;
- документирование информации (описание, сценарии, спецификация процессов и т.д.).

Основная задача анализа требований – это получение списка не дублируемых требований к разрабатываемому ПО [1]. Корректная группировка требований позволит определить минимальное и достаточное количество необходимых функций, которые в свою очередь смогут удовлетворить максимально большое количество целей. Таким образом, грамотный подход к этой задаче поможет обозначить чёткие рамки проекта, что сэкономит бюджет и облегчит процесс разработки ПО.

### ***1.1. Способы анализа требований***

Во время подготовки данной работы для наиболее полного сбора и анализа всех возможных требований следовало воспользоваться пятью способами, описанными далее.

Опрос – это один из самых часто используемых методов получения информации. Суть данного метода заключается в том, что респонденту (опрашиваемому) задают заранее подготовленные вопросы, ответы на которые позволяют исследователю получить необходимые данные [2]. Опросы бывают следующих видов:

- анкетирование;
- метод лестницы;
- свободный;
- устный;
- письменный;
- стандартизированный.

Также опросы можно разделить по способу взаимодействия исследователя с респондентом:

- личный (прямой контакт);
- дистанционный (опрос по телефону, по сети Интернет и т.д.).

Таким образом, при использовании данного способа анализа требований, возможно узнать о недочётах текущего ПО и об ожиданиях от разрабатываемой программы напрямую от людей, непосредственно работающих с ней.

Наблюдение за выполнением операций, как и опрос, позволяет собрать довольно полную информацию о работе пользователя с текущим ПО. Преимуществом данного способа анализа является тот факт, что наблюдатель может заметить такие требования, о которых пользователь может забыть упомянуть при опросе. Но ожидания конечного пользователя от разрабатываемого ПО данный способ выявить не способен.

Учитывая все особенности этих двух методов, логичнее всего использовать их сообща. Такой подход также поможет выявить, насколько правильно пользователь работает со старым ПО – бывают случаи, когда неумение работать с программой списывается на недочёты самой программы. В таком случае зачастую скорее нужна не разработка нового обеспечения, а обучение персонала работе с уже имеющимся.

Понятие «документооборот» определяется как «движение документов в организации с момента их создания или получения до завершения исполнения или отправки» [3]. Понятие «электронный документооборот» определяется как «документооборот с использованием автоматизированной информационной системы (системы электронного документооборота)». Следовательно, анализ электронного и/или обычного документооборота поможет узнать степень загруженности как отдельных сотрудников, так и организационно – методического отдела в целом, а также наметить основные направления рационализации документооборота, что, несомненно, окажется полезным при разработке нового ПО.



Управленческую документацию принято подразделять на три основных блока, которые указаны на рисунке 1.1. Основные цели создания управленческой документации:

- информирование (доведение до сведения сотрудников информации);
- подтверждение соответствия (подтверждение выполнения ранее запланированного действия);
- фиксация знаний (сохранение полученного опыта, передача его новому персоналу при обучении и т.п.).



**Рисунок 1.1** – Структура управленческой документации

Таким образом, данный способ анализа позволяет оценить эффективность управления в целом. Составляется иерархическая классификационная схема, наглядно демонстрирующая разделение управленческих функций, деятельности и отдельные операции, которые сопровождаются документами.

Разработчик ПО во время анализа требований, разумеется, может опираться также и на свой собственный опыт, полученный во время работы. Вероятно, некоторые требования не будут названы ни заказчиком, ни конечным пользователем, но всё равно должны быть учтены при создании новой программы. Также при возникновении спорных ситуаций между заказчиком и пользователем решить их разработчик сможет, опираясь только на собственный опыт.

В доступе к управленческой документации было отказано. Следовательно, и анализ документооборота был невозможен. Также было запрещено наблюдение за выполнением операций. Данные запреты обусловлены пунктом 1 статьи 13 главы 2 Федерального закона РФ [4]. В связи с этим остаются доступными только два способа анализа требований к разрабатываемому ПО: проведение опроса, использование знаний.

### ***1.2. Пользовательские требования***

Пользовательские требования – это требования, описывающие задачи, которые должны выполняться пользователями при помощи разрабатываемого ПО. При проведении свободного, устного, личного опроса были зафиксированы следующие требования:

- база данных (БД) должна хранить данные:
  - о пациенте (ФИО; дата рождения; возраст; пол; вес; контрольная группа);
  - об анамнезе пациента (сопутствующие заболевания сердечно – сосудистой системы (ССС); сопутствующие заболевания дыхательной системы (дых.); прочие сопутствующие заболевания; номер класса по шкале American Association of Anaesthetists (шкала ASA применяется для оценки степени

- операционно-анестезиологического риска, оценивающая физическое состояние пациента);
- об операции (поставленный диагноз, по которому проводилась операция; дата проведения операции; длительность операции; длительность анестезии; ФИО хирурга, проводившего операцию);
  - о контроле состояния пациента во время операции (систолическое (сист.) артериальное давление; диастолическое (диаст.) артериальное давление; среднее (средн.) артериальное давление; частота сердечных сокращений (ЧСС); степень насыщения крови кислородом ( $SpO_2$ ); номер контроля (за операцию их в среднем проводится три: в начале (до вмешательства), в середине (в процессе инвазивного вмешательства) и в конце);
  - о реабилитации пациента после операции (время пробуждения (после общего наркоза); время экстубации (извлечение интубационной трубки после наркоза, при восстановлении самостоятельного адекватного дыхания больного); время ориентации (приход пациента в сознание); время перевода в палату (перевод из реанимации в обычную палату);
- вывод данных на экран для ознакомления:
- полный анамнез зарегистрированного пациента;
  - полная информация о зарегистрированном пациенте;
  - выборочная информация о зарегистрированном пациенте:
    - ФИО – дата рождения;
    - ФИО – сопутствующие заболевания;
    - ФИО – дата операции;

- ФИО – состояние во время операции;
- список ФИО пациентов:
  - по году рождения;
  - по сопутствующему(им) заболеванию(ям) ССС;
  - по сопутствующему(им) заболеванию(ям) дых.;
  - по проч. сопутствующему(им) заболеванию(ям);
  - по ФИО хирурга;
  - по дате операции (по конкретному дню);
  - по дате операции (по конкретному месяцу);
  - по длительности операции;
- список с полной информацией о пациентах из выбранной контрольной группы.

Также необходимо включить следующие три требования:

- возможность вносить данные:
  - регистрация нового пациента;
  - изменение и/или добавление данных об уже зарегистрированном пациенте;
- возможность удалить данные:
  - выборочно удалить ошибочно введённые или неактуальные данные об уже зарегистрированном пациенте;
  - полностью удалить данные о пациенте;
- возможность работы с приложением на любом компьютере медицинского учреждения;
- обеспечение лёгкости работы с приложением для всего медицинского персонала учреждения;
- возможность вывода информации на бумажный носитель;
- хранение информации.

Хоть они и не были выявлены в процессе опроса, но без удовлетворения данных требований работа с приложением представляется невозможной. Итого было зафиксировано 46 пользовательских требований.

### ***1.3. Функциональные требования***

Функциональные требования – это требования, объясняющие, что должно быть в разрабатываемом ПО, а также какие действия должны выполняться системой. Значит, пользовательские требования будут иметь следующий вид:

- БД должна содержать таблицы:
  - «Пациент», куда входит информация о ФИО больного, его дате рождения, возрасте, поле, весе и контрольной группе;
  - «Анамнез», куда входит информация о наличии или отсутствии сопутствующих заболеваний ССС, дых. и проч., а также значение класса по шкале ASA;
  - «Операция», куда входит информация о поставленном диагнозе, дате проведения операции, длительности операции и анестезии, а также ФИО хирурга;
  - «Контроль», куда входит информация о артериальном давлении (сист., диаст. и средн.), ЧСС, SpO2 и номер контроля;
  - «Реабилитация», куда входит информация о времени пробуждения, экстубации, ориентации и перевода в палату;
- вывод на экран данных:
  - ФИО из таблицы «Пациент» и все данные из таблицы «Анамнез»;
  - всех заполненных ранее полей из таблиц «Пациент», «Анамнез», «Операция», «Контроль» и «Реабилитация»;
  - поля «Фамилия», «Имя», «Отчество», «Дата рождения» из таблицы «Пациент»;

- таблицы «Пациент» (поля «Фамилия», «Имя», «Отчество») и «Анамнез» (поля «Сопутствующие заболевания (ССС)», «Сопутствующие заболевания (дых.)», «Сопутствующие заболевания (проч.)»);
- таблицы «Пациент» (поля «Фамилия», «Имя», «Отчество») и «Операция» (поле «Дата операции»);
- таблицы «Пациент» (поля «Фамилия», «Имя», «Отчество») и все данные таблицы «Контроль»;
- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением года рождения в поле «Дата рождения»;
- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением в поле «Сопутствующие заболевания (ССС)» из таблицы «Анамнез»;
- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением в поле «Сопутствующие заболевания (дых.)» из таблицы «Анамнез»;
- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением в поле «Сопутствующие заболевания (проч.)» из таблицы «Анамнез»;
- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением в поле «ФИО хирурга» из таблицы «Операция»;
- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением дня в поле «Дата операции» из таблицы «Операция»;

- список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением месяца в поле «Дата операции» из таблицы «Операция»;
  - список с полями «Фамилия», «Имя», «Отчество» из таблицы «Пациент» с одинаковым значением в поле «Длительность операции» из таблицы «Операция»;
  - все имеющиеся данные тех зарегистрированных пациентов, у которых стоит выбранное одинаковое значение в поле «Контрольная группа» из таблицы «Пациент»;
- на экран выводится окно с анкетой пациента, в которой можно ввести данные о новом пациенте или отредактировать данные о зарегистрированном ранее больном;
  - есть возможность удалить частично или полностью данные о зарегистрированном пациенте;
  - программа должна корректно работать на всех компьютерах медучреждения;
  - программа должна быть интуитивно понятна всему медперсоналу;
  - любая хранящаяся в приложении информация должна иметь возможность вывода при помощи принтера;
  - данные хранятся непосредственно в создаваемом приложении.

#### ***1.4. Матрица отслеживания требований***

Матрица отслеживания требований — это таблица, которая связывает требования к разрабатываемому продукту. Иными словами, в ней сопоставляются требования пользовательские и функциональные, а также описывается программный компонент будущей системы, которым будет покрываться каждое из требований.

Составление и применение данной матрицы облегчает отслеживание требований на протяжении всего жизненного цикла проекта, что помогает удостовериться в том, что одобренные требования выполнены в конце проекта. Матрица отслеживания требований разрабатываемого приложения для автоматизации городской больницы представлена в таблице 1.1.

**Таблица 1.1 – Матрица отслеживания требований**

Пользовательское требование	Функциональное требование	Программный компонент
Наличие данных о пациенте	Ведение таблицы данных «Пациент», содержащей информацию о пациенте, полученную путём опроса или из медкарты	Программа по ведению данных пациента
Наличие данных о анамнезе пациента	Ведение таблицы данных «Анамнез», содержащей информацию о сопутствующих заболеваниях, полученную путём опроса или из медкарты	Программа по ведению данных о анамнезе пациента
Наличие данных о проведённой операции	Ведение таблицы данных «Операция», содержащей информацию о поставленном диагнозе и проведённой операции	Программа по ведению данных об операции
Наличие данных о контроле состояния пациента во время операции	Ведение таблицы данных «Контроль», содержащей информацию о состоянии пациента, полученную путём снятия показаний с приборов	Программа по ведению данных о контроле состояния пациента
Наличие данных о реабилитации пациента после проведённой операции	Ведение таблицы данных «Реабилитация», содержащей информацию о скорости восстановления пациента после проведённой операции	Программа по ведению данных о реабилитации пациента
Вывод на экран полного анамнеза пациента	Таблица с соответствующими данными из таблиц «Пациент» и «Анамнез»	Программа по выводу на экран запрашиваемых данных
Вывод на экран полной информации о пациенте	Таблица с соответствующими имеющимися данными	Программа по выводу на экран запрашиваемых данных
Вывод на экран выборочной информации о зарегистрированном пациенте	Таблица с соответствующими запрашиваемыми данными	Программа по выводу на экран запрашиваемых данных



Пользовательское требование	Функциональное требование	Программный компонент
Вывод на экран списка ФИО пациентов, отобранных по определённому критерию	Таблица с соответствующими запрашиваемыми данными	Программа по выводу на экран запрашиваемых данных
Полная информация о пациентах выбранной контрольной группы	Таблица с соответствующими запрашиваемыми данными	Программа по выводу на экран запрашиваемых данных
Внесение новых данных или редакция уже имеющейся информации	Возможность ввода информации	Средство для внесения информации
Частичное или полное удаление информации о пациенте	Возможность удаления данных	Средство для удаления информации
Возможность установки на любой компьютер учреждения	Корректная работа на всех персональных компьютерах медучреждения	Разрабатываемое ПО
Доступность всем работникам учреждения	Доступность всем работникам учреждения	Разрабатываемое ПО
Возможность вывода информации на бумажный носитель	Возможность вывода данных	Средство для вывода информации
Возможность хранения данных	Возможность хранения данных	Разрабатываемое ПО

## Глава 2. Проектирование приложения

Проектирование – это процесс, в результате которого составляются описания нового или модернизируемого технического объекта, необходимые и достаточные для создания объекта в заданных условиях. Проектирование СУБД – это определение архитектуры, а также компонентов, интерфейсов и других характеристик системы или её части. Основные задачи данного процесса:

- обеспечение хранения в СУБД всей необходимой информации;
- обеспечение возможности получения данных по всем заявленным заказчиком и/или пользователем запросам;
- сокращение избыточности и дублирования данных (нормализация);
- обеспечение целостности базы данных.

Этап проектирования – это условно выделенная часть процесса проектирования, которая сводится к выполнению одной или нескольких процедур. Процедуры объединяются или же разбиваются в зависимости от того, к какому иерархическому уровню и/или аспекту описания относятся получаемые при их выполнении проектные решения. Проектирование СУБД принято разделять на три этапа:

- концептуальное проектирование, данный этап заключается в построении семантической модели предметной области. Она имеет самый высокий уровень абстракции.
- логическое проектирование, второй этап включает в себя создание схемы на основе конкретной модели данных. Для реляционной модели данных логическая модель — это набор схем отношений с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.
- Физическое проектирование, на последнем этапе создаётся схема для конкретной СУБД (MS Access в данном случае). Специфика выбранной

СУБД может включать в себя, например, ограничения на именование объектов или на поддерживаемые типы данных. Также данный подэтап включает в себя выбор решений касательно физической среды хранения данных (например, разделение по файлам и устройствам), создание индексов и проч.

### ***2.1. Способы проектирования***

Как было сказано выше, проектирование — это процесс создания схемы и определения необходимых ограничений. Результатом этого процесса является совокупность моделей, свойств или характеристик, описанных в форме, пригодной для реализации системы – проект. По выполнению данного этапа, должны быть достигнуты следующие цели:

- обеспечение понимания структуры организации, а также ключевых происходящих в ней процессов;
- выявление наличия или отсутствия проблем и путей их решения;
- удостовериться в том, что заказчики, разработчики и конечные пользователи одинаково понимают цели и задачи.

Также важно отметить, что модели бизнес-процессов – это не промежуточный результат, который используется для написания каких – либо рекомендаций и заключений. Это самостоятельный имеющий важное практическое значение результат. Модель должна показывать наглядные ответы на следующие вопросы:

- Какие действия должны быть выполнены для достижения выбранного конечного результата?
- В каком порядке выполняются данные действия?
- Какие присутствуют механизмы контроля и управления?
- Кто выполняет данные действия?
- Какие присутствуют входящие/выходящие документы/информация?


- Какие параметры характеризуют выполнение процедур и процесса в целом?


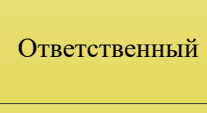
Одной из методологий бизнес-моделирования, получившей широкое распространение в РФ, является методология Architecture of Integrated Information Systems (ARIS) [5], которая была разработана профессором компании IDS Scheer AG (Германия) Августом – Вильгельмом Шеером, а на данный момент принадлежит компании Software AG. Данная нотация имеет ряд преимуществ, которые делают выбор именно данного способа проектирования целесообразным:

- наглядность;
- возможность изучить деятельность выбранной компании с разных точек зрения;
- разнообразие методов моделирования.

Существует несколько нотаций проектирования, которые подходят для разных уровней описания. Нотация ARIS Value Added Chain Diagram (ARIS VACD) используется для верхнеуровневого (1 уровень) описания. На этих уровнях описываются ключевые процессы, выполняемые в организации. ARIS VACD подходит для изучения организации в целом. Несмотря на то, что для выявления возможных проблем этой нотации может быть недостаточно, она помогает определить те процессы, которые и нуждаются в изменениях и доработке. Элементы, используемые в данной нотации, приведены в таблице 2.1.

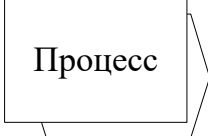

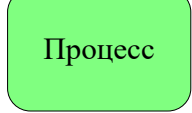

**Таблица 2.1 – Графические элементы ARIS VACD**

Наименование	Описание	Графическое представление
Процесс	Элемент отражает процессы, выполняемые в организации	

Наименование	Описание	Графическое представление
Входящий или исходящий объект	Элемент отражает реальные носители информации	
Ответственный	Элемент, отражающий различные организационные звенья организации, ответственный за исполнение процесса	

Нотация ARIS extended Event Driven Process Chain (ARIS eEPC) подходит для более низкоуровневого (2 – 8 уровень) описания процессов. По сути это расширенная версия нотации ARIS VACD, которая позволяет более детально изучить протекающие в организации процессы. Как правило, именно на данном этапе проектирования становятся очевидны проблемы, мешающие развитию организации. Элементы, используемые в данной нотации, приведены в таблице 2.2.

**Таблица 2.2 – Графические элементы ARIS eEPC**

Наименование	Описание	Графическое представление
Иницирующий или последующий процесс	Элемент обозначает внешний процесс или функцию, имеющие влияние на текущий этап	
Иницирующее или последующее событие	Элемент обозначает состояние, которое является существенным и оказывает влияние или контролирует дальнейшее развитие одного или более бизнес-процессов	
Процесс	Данный элемент отражает выполняемые процессы	
Ответственный	Элемент отражает звенья организации (например, отдел), ответственный за исполнение процесса	

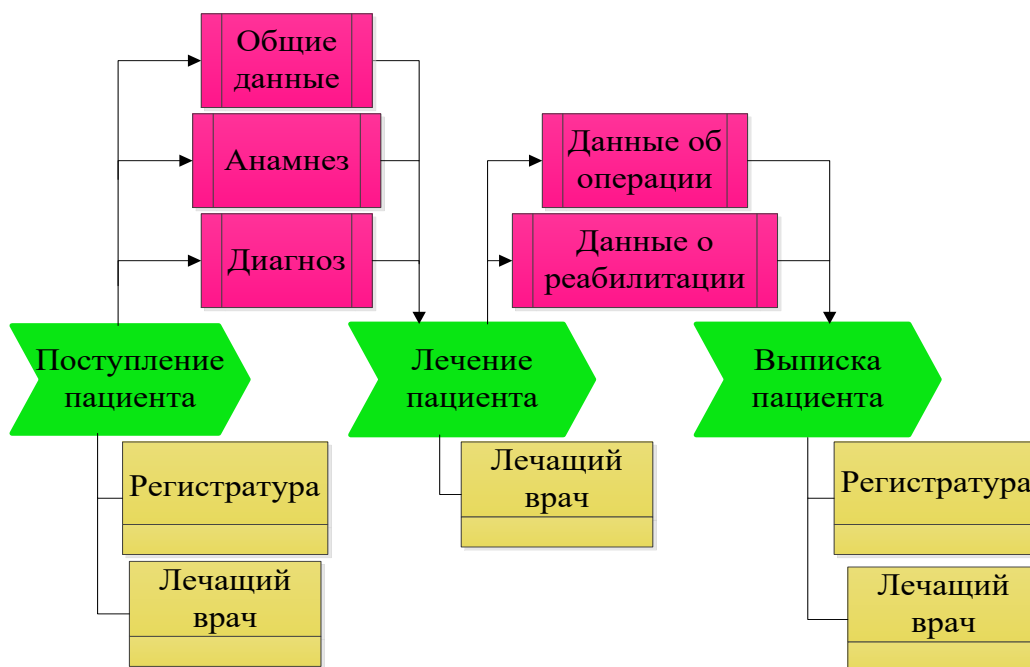
Наименование	Описание	Графическое представление
Документ	Элемент отображает бумажные документы, сопровождающих выполнение функции	
Прикладная система	Элемент отображает программу ИС, которая реализует процесс	
Оператор «И»	Оператор ставится в случае, когда начало/завершение выполнения функции должно инициировать одновременно несколько событий	
Оператор «ИЛИ»	Оператор ставится, когда начало/завершение выполнения функции может инициировать несколько событий	
Исключающий оператор «ИЛИ»	Оператор ставится, когда начало/завершение выполнения функции может инициировать только одно из событий в зависимости от условия	

Для проектирования структуры приложения используется диаграмма классов, демонстрирующая классы системы, их атрибуты, методы и взаимосвязи между ними. На диаграмме классы представлены в рамках, содержащие следующие компоненты: имя класса, поля (атрибуты) класса.

## 2.2. Ключевые бизнес-процессы в моделях «AS-IS» и «TO-BE»

Если рассматривать первый уровень описания процессов (рисунок 2.1) в больнице, то ключевых процессов будет 3: поступление пациента в больницу (по направлению из поликлиники или через бригаду скорой медицинской помощи), непосредственно лечение пациента, а также выписка после завершения лечения. При поступлении пациента в регистратуре и лечащим врачом принимаются общие данные о поступившем больном, его анамнез и диагноз (в случае поступления по направлению из поликлиники). Далее пациент поступает на лечение, в процессе которого собираются данные об операции и реабилитации, после чего его выписывают из больницы. Все

собранные данные заносятся в медкарту больного, а также в таблицу MS Excel для более быстрого поиска при создании отчётов.



**Рисунок 2.1** – Первый уровень описания процессов

Далее необходимо перейти ко второму уровню описания каждого из процессов. В качестве наиболее показательного примера будет достаточно рассмотреть второй уровень описания процесса «Лечение пациента» последовательно в начале в модели «AS-IS», а затем в модели «TO-BE». Данный процесс был выбран, поскольку именно на этом этапе идёт самое активное взаимодействие с данными о пациенте.

Процесс «Лечение пациента» состоит из пяти основных процессов: обследование для уточнения диагноза, подготовка пациента к операции и её проведение, перевод в реанимацию, перевод в палату. Во время первых четырёх процессов доктор постоянно обращается к общим данным о пациенте, его анамнезу, а также информации о диагнозе. Во время перевода в палату лечащий врач обращается только к данным о процессе реабилитации пациента.

Графические представления второго уровня процесса «Лечение пациента» в моделях «AS-IS» и «TO-BE» представлены на рисунках 2.2 и 2.3 соответственно.

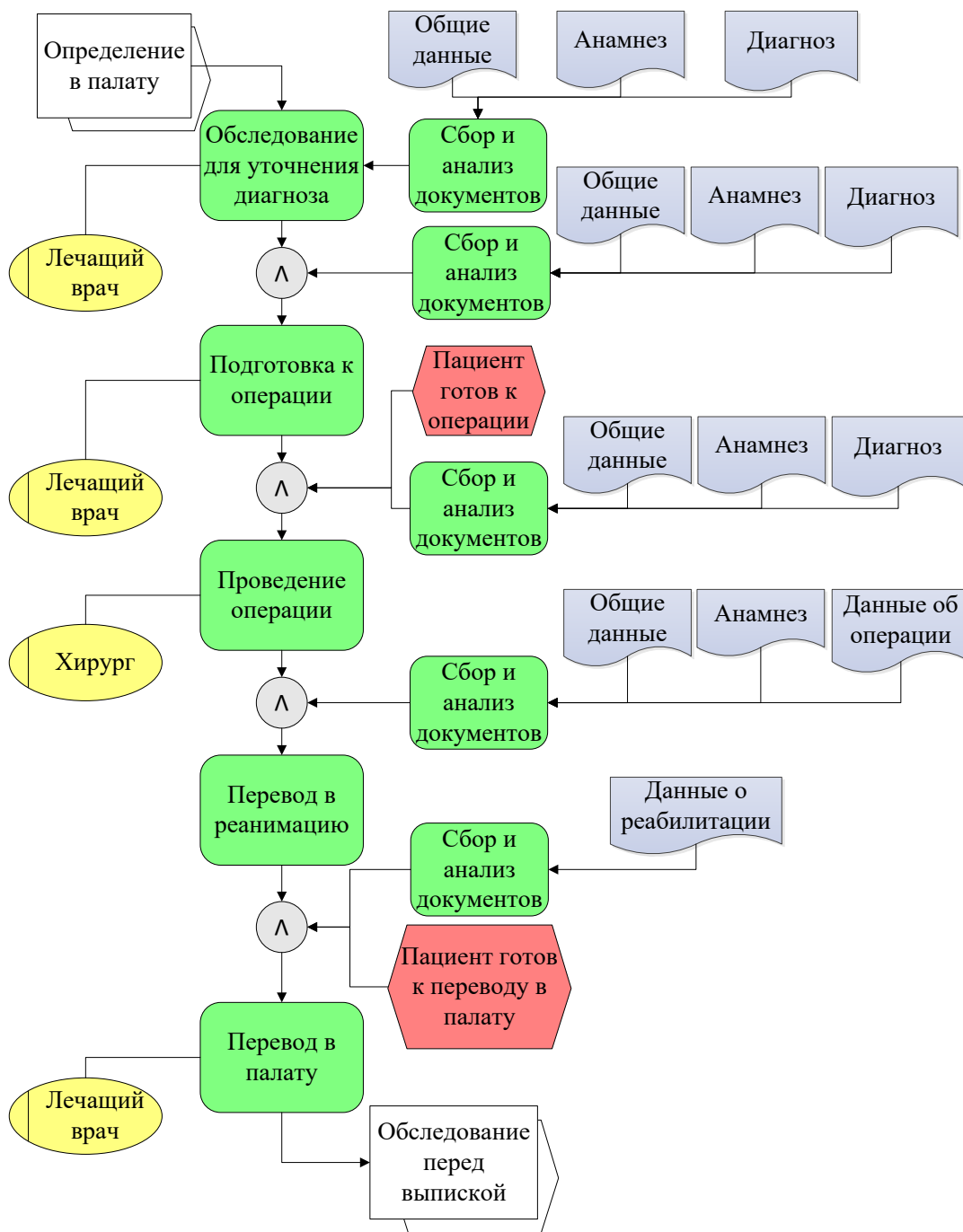
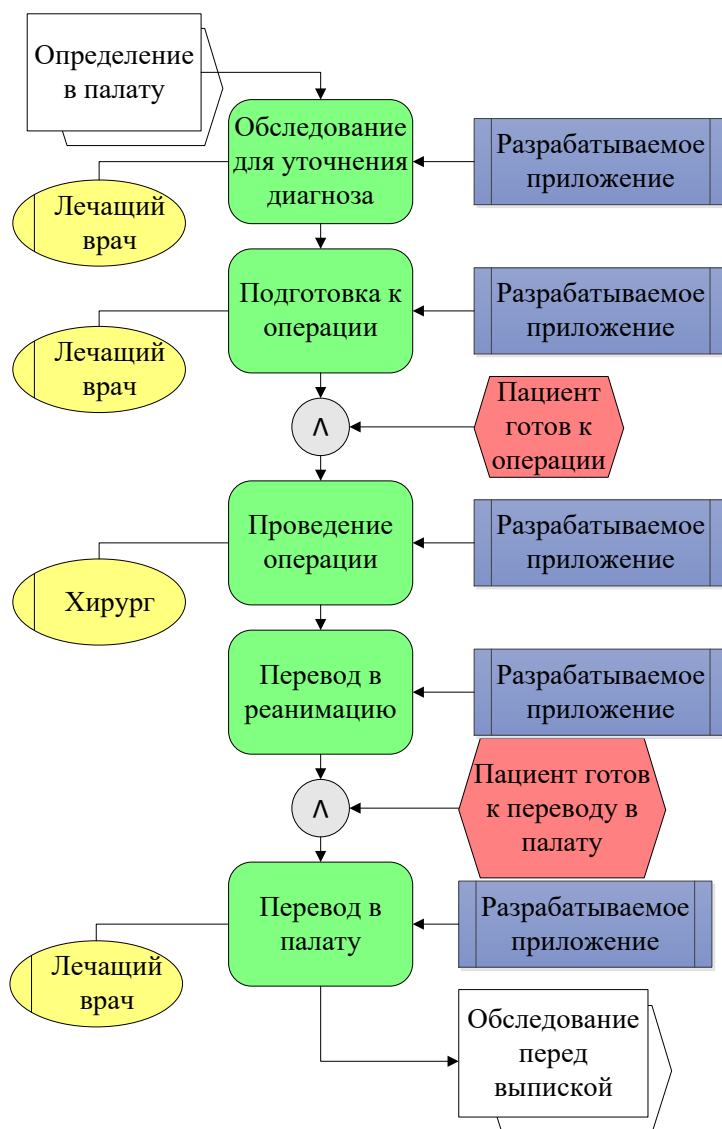


Рисунок 2.2 – Процесс «Лечение пациента» в модели «AS-IS»





**Рисунок 2.3** – Процесс «Лечение пациента» в модели «ТО-ВЕ»

Как видно из сравнения двух моделей, разрабатываемое приложение действительно упрощает работу медицинского персонала. Сейчас необходимые данные приходится искать в медицинской карте пациента или в таблице MS Excel и распечатывать. Поскольку карты хранятся в регистратуре, а таблицы – на компьютере у некоторых докторов, общий доступ в любой момент к необходимой информации отсутствует. А в любом медицинском учреждении не редки ситуации, когда счёт идёт даже не на минуты, а на секунды – именно

поэтому необходимо обеспечить всеобщий доступ в любой момент ко всей необходимой информации.

### 2.3. Архитектура данных разрабатываемой системы

На основе порядка использования данных, приведённого выше, всю информацию можно разбить на классы, приведённые в таблице 2.3.

**Таблица 2.3 – Классы данных**

Название класса	Данные
Пациент	ФИО
	Дата рождения
	Возраст
	Пол
	Вес
	Контрольная группа
Анамнез	Сопутствующие заболевания (ССС)
	Сопутствующие заболевания (дых.)
	Сопутствующие заболевания (проч.)
	ASA
Операция	Диагноз
	Дата операции
	Длительность операции
	Длительность анестезии
	ФИО хирурга
Контроль	Артериальное давление (сист.)
	Артериальное давление (диаст.)
	Артериальное давление (средн.)
	ЧСС
	SpO <sub>2</sub>
	Номер контроля
Реабилитация	Время пробуждения
	Время экстубации
	Время ориентации
	Время перевода в палату

Далее необходимо провести процедуру нормализации данных [6]. Стандартно все данные приводят к третьей нормальной форме (НФ), потому как нормализации такого уровня бывает достаточно. Согласно 1НФ, в одном поле таблицы должен храниться только один атрибут. Таким образом в классе

«Пациент» данные «ФИО» необходимо разбить на три отдельные строки: «Фамилия», «Имя», «Отчество». В классе «Операция» данные «ФИО хирурга» необходимо вынести в отдельный класс.

Согласно 2НФ, должны соблюдаться условия 1НФ и каждый не ключевой атрибут должен зависеть от ключа. Следовательно, в классы «Пациент», «Анамнез» и «Операция» добавляется графа «Код пациента», а в таблицы «Операция», «Контроль» и «Реабилитация» - «Код операции». Также из класса «Пациент» убирается графа «Возраст», потому как она зависит от графы «Дата рождения», что противоречит условию 2НФ.



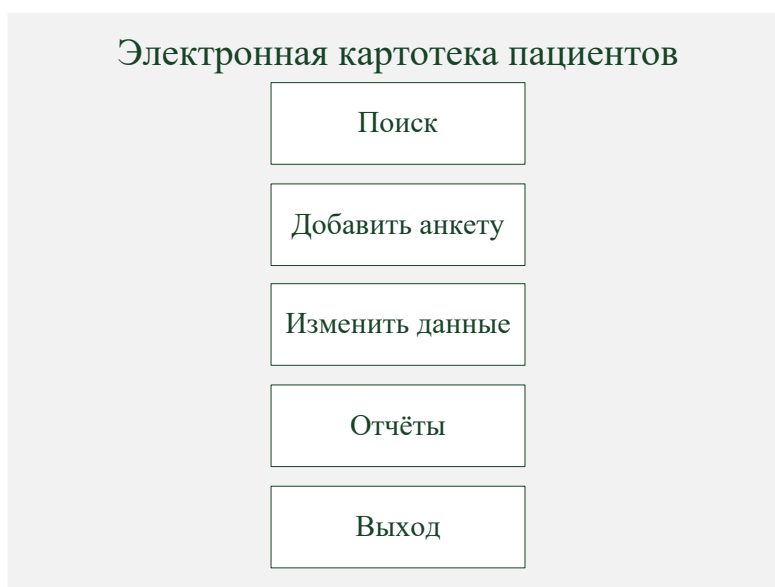
**Рисунок 2.4** – Архитектура данных разрабатываемого приложения

Согласно 3НФ, должны соблюдаться условия 2НФ и все не ключевые поля, содержимое которых может относиться к нескольким записям таблицы, должно выноситься в отдельные таблицы. Таким образом, после разделения

всех данных на классы и проведения процедуры нормализации, получается архитектура разрабатываемого приложения, приведённая на рисунке 2.4.

#### ***2.4. Описание разрабатываемого приложения***

Для наиболее удобной работы конечного пользователя с программой необходимо создать максимально понятные пользовательские интерфейсы программы. Когда пользователь только открывает программу, ему предстоит выбрать, нужно ли внести данные о новом пациенте или найти данные по уже зарегистрированному больному. Первый интерфейс представлен на рисунке 2.5.



**Рисунок 2.5 – Первый интерфейс**

Допустим, пользователь решил занести данные о новом пациенте. В таком случае, ему откроется следующее диалоговое окно (рисунок 2.6), в котором он выбирает «Пациент».



Добавить анкету

Пациент

Анамнез

Операция

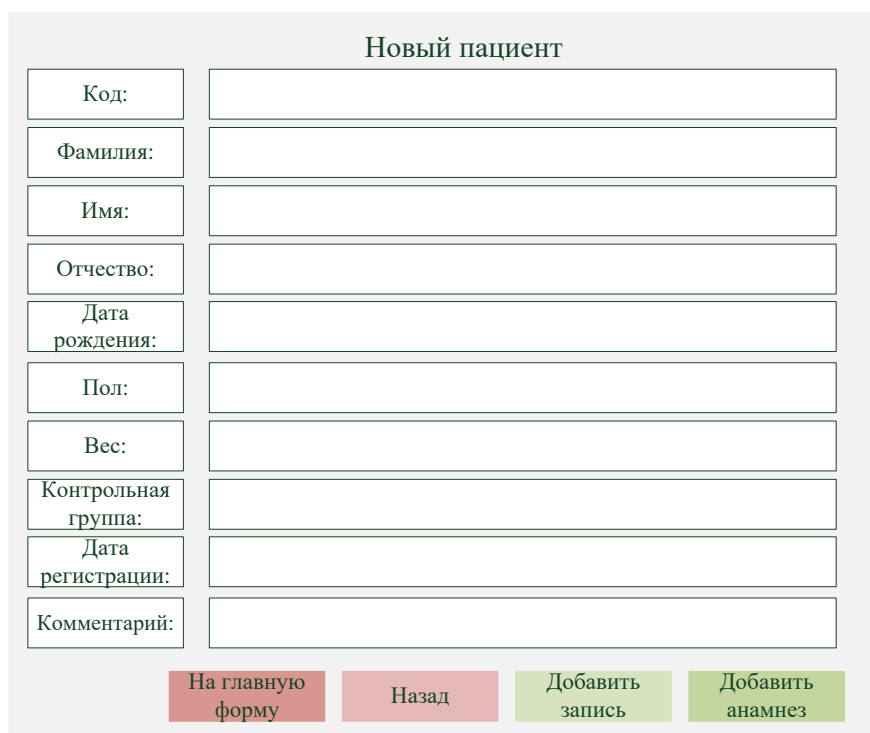
Реабилитация

Контроль

Назад

Рисунок 2.6 – Интерфейс «Добавить анкету»

Далее на экране появляется интерфейс «Пациент» (рисунок 2.7). Данные, внесённые в графы, заполняют соответствующие строки таблицы «Пациент». Таким же образом заполняются прочие данные.



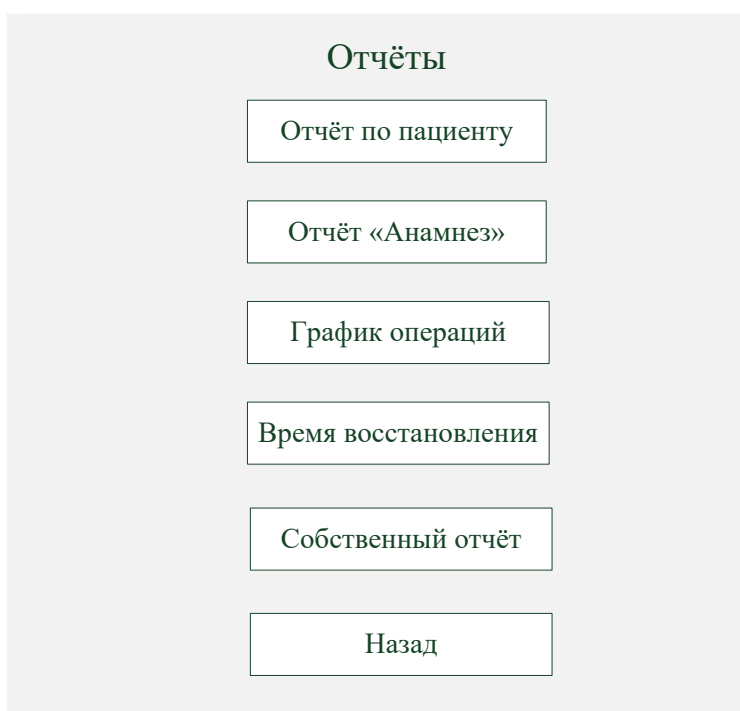
Новый пациент

Код:	
Фамилия:	
Имя:	
Отчество:	
Дата рождения:	
Пол:	
Вес:	
Контрольная группа:	
Дата регистрации:	
Комментарий:	

На главную форму    Назад    Добавить запись    Добавить анамнез

Рисунок 2.7 – Интерфейс «Новый пациент»

Если пользователю необходимо редактировать данные, то он точно так же выбирает конкретный блок, который необходимо изменить. Интерфейсы для редакции информации выглядят точно так же, как и соответствующие интерфейсы внесения данных. В том случае, если пользователю необходимо вывести на экран определённую информацию или составить отчёт, необходимо нажать на кнопку «Отчёт» и выбрать либо из предложенных форм отчётов (рисунок 2.8).



**Рисунок 2.8** – интерфейс «Отчёты»

Если пользователю необходимо составить отчёт, шаблон которого не предусмотрен программой, то необходимо выбрать «Собственный отчёт». После этой процедуры он сможет в окне «Поиск» отобразить требуемые таблицы и данные в них.

## Глава 3. Разработка программ

### *3.1. Методы разработки*

Частотный принцип основан на выделении наиболее часто используемых функций и групп. Для наиболее востребованных действий создаются условия более быстрого доступа, а также стараются сделать их более короткими. Принцип модульности основа на выделении отдельных элементов рассматриваемой системы, создавая для них отдельное и законченное оформление, а также средства сопряжения с подобными элементами. Принцип функциональной избирательности является логическим продолжением первых двух принципов. Его суть заключается в выделении некоторых наиболее важных модулей, к которым обеспечивается наиболее быстрый доступ.


Принцип генерируемости заключается в настройке на определённый перечень решаемых проблем и условия работы конечного пользователя. Принцип функциональной избыточности предоставляет возможность проведения одной и той же операции различными средствами, учитывая психологические различия пользователей в восприятии информации. В данном случае наиболее целесообразным будет выбрать путь функциональной избирательности – ряд операций конечные пользователи программы проводят значительно чаще остальных, значит, и доступ к ним должен быть более простым и быстрым.

### *3.2. Реализация процесса регистрации нового пациента*

Когда пациент только поступил в больницу, необходимо внести его данные в общую базу. Для этого пользователь выбирает последовательно «Добавить анкету», «Пациент» и открывает окно «Новый пациент», приведённое на рисунке 3.1. Далее заполняет соответствующие графы информацией о больном, полученной в результате опроса.

## Пациент

Код пациента:	<input type="text"/>
Фамилия:	<input type="text"/>
Имя:	<input type="text"/>
Отчество:	<input type="text"/>
Дата рождения:	<input type="text"/>
Пол:	<input type="text"/> ▼
Вес:	<input type="text"/>
Контрольная группа:	<input type="text"/> ▼
Дата регистрации:	<input type="text"/>
Комментарий:	<input type="text"/>




**Рисунок 3.1** – Внесение первичных данных о пациенте

После этого доктор приступает к сбору анамнеза. Полученные либо при опросе, либо из медицинской книги данные пользователь вносит в окно «Анамнез» (рисунок 3.2).

## Анамнез

Код пациента	<input type="text"/>
Сопутствующие заболевания	<input type="text"/>
ASA	<input type="text"/> ▼



**Рисунок 3.2** – Внесение данных об анамнезе пациента



### 3.3. Реализация процесса полного анамнеза пациентов

Когда врачу необходимо изучить анамнеза пациента, нужно выбрать последовательно «Отчёты» и «Отчёт анамнез» (рисунок 3.3).



Рисунок 3.3 – Интерфейс «Отчёты»

После этого пользователю будет предложено ввести код пациента, чьи данные необходимо получить. Далее пользователь увидит на экране полный анамнез зарегистрированного пациента, как на рисунке 3.4.

#### Полный анамнез

Код пациента	Сопутствующие заболевания	Фамилия	Имя	Отчество	ASA
5297	СД 2	Анифов	Егор	Яковлевич	3
5297	Бр астма	Анифов	Егор	Яковлевич	3
5297	Стенокардия напряжения 2 Ф	Анифов	Егор	Яковлевич	3

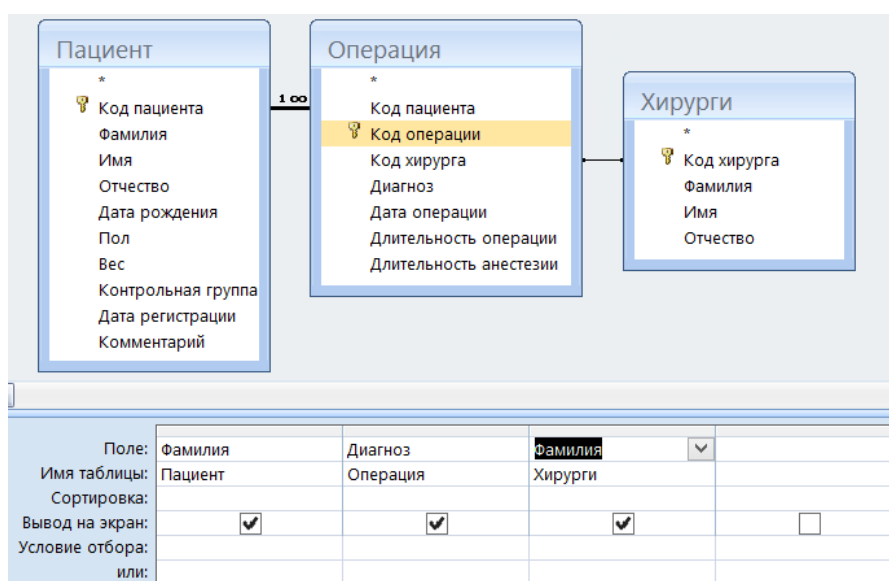
9 июня 2017 г.

Рисунок 3.4 – Анамнез пациента

### 3.4. Реализация процесса поиска по выбранному параметру

В том случае, когда пользователю необходимо найти информацию, быстрой вывод которой не рассчитан ни одним из расчётов, он должен самостоятельно выбрать интересующие таблицы и данные в них. Допустим,

необходимо составить список фамилий пациентов с диагнозом «хронический калькулёзный холецистит», а также узнать фамилии оперировавших хирургов. Для этого пользователь выводит необходимые таблицы, выбирает требуемые данные и устанавливает ограничения. Пример такого запрос можно увидеть на рисунке 3.5.



**Рисунок 3.5** – Поиск по выбранному параметру

Далее пользователь увидит таблицу (рисунок 3.6), которая будет содержать только запрашиваемые данные.

Пациент.Фа	Диагноз	Хирурги.Фамил
Татарина	ХКХ	Ивашко
Борзов	Аппендицит	Сидоров
Байчурова	Аппендицит	Сидоров
Калакутская	Полипы ЖКТ	Орлов
Анифов	ХКХ	Наилев
Круглова	ХКХ	Ивашко
Воротинова	Полипы ЖКТ	Орлов
Иванченко	ХКХ	Наилев
Орлова	Аппендицит	Ивашко

**Рисунок 3.6** – Таблица с выбранными данными

## Глава 4. Тестирование разработанного приложения

### 4.1. Методы тестирования

V-Model – модель проектирования, которая призвана сделать более простой разработку системы. Структура данного метода представлена на рисунке 4.1.

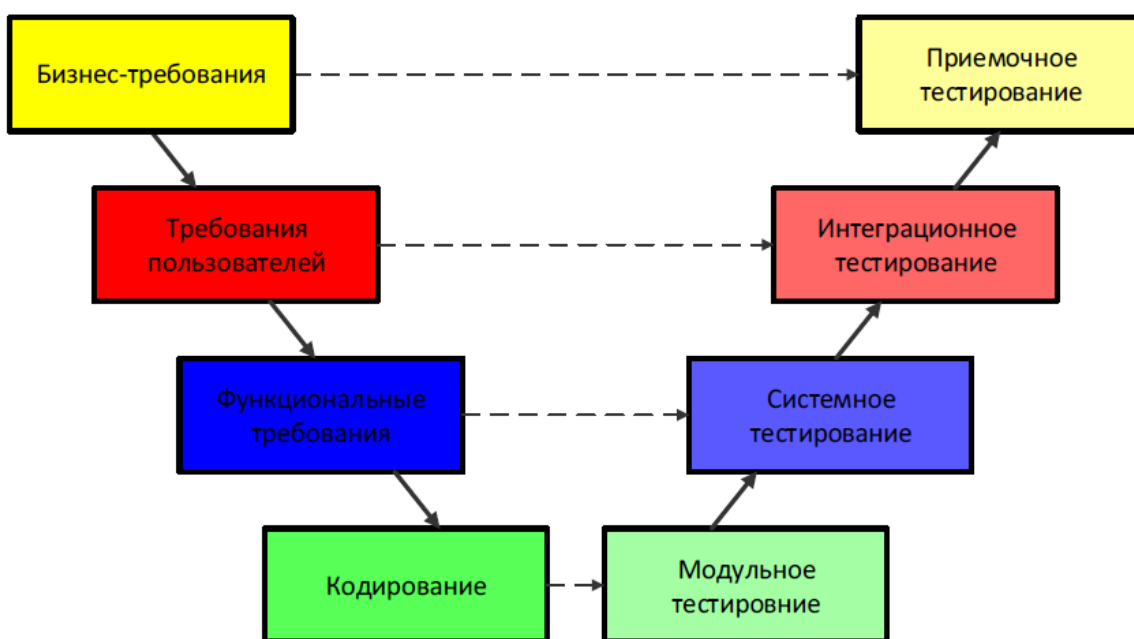


Рисунок 4.1 – Структура V-модели



V-образная модель часто применяется при разработке систем, которым особенно важно бесперебойное функционирование. Например, прикладные программы в медицинских учреждениях для наблюдения за состоянием пациентов. Особенностью данной модели можно назвать то, что она направлена на тщательную проверку и тестирование продукта, находящегося уже на первоначальных стадиях проектирования. Стадия тестирования проводится одновременно с соответствующей стадией разработки. Для проверки










работоспособности разработанного ПО достаточно провести три тестирования: функциональное, интеграционное и нагрузочное.

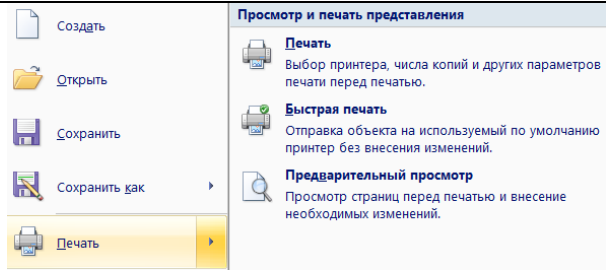
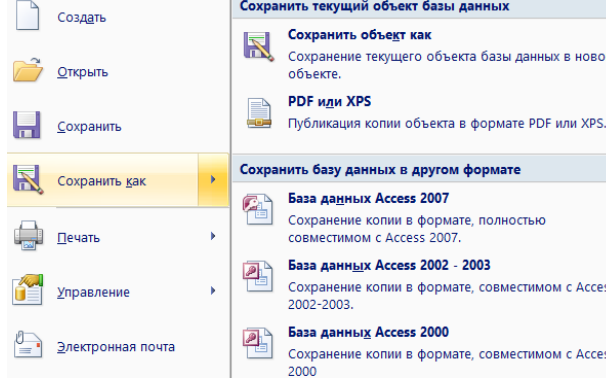
#### 4.2. Функциональное тестирование

Функциональное тестирование — это тестирование ПО в целях проверки реализуемости функциональных требований. То есть на данном этапе проверяется, способен ли разработанный продукт ПО в определённых условиях решать задачи, нужные пользователям. Для этого необходимо сверить соответствие функциональных требований из главы 1 с имеющимися программными компонентами разработанной программы. Данные тестирования занесены в таблицу 4.1.

**Таблица 4.1 – Функциональное тестирование программы**

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)
Ведение таблицы данных «Пациент», содержащей информацию о пациенте, полученную путём опроса или из медкарты	Программа по вводу данных пациента	<p>Пациент</p> <p>Код пациента: <input type="text"/></p> <p>Фамилия: <input type="text"/></p> <p>Имя: <input type="text"/></p> <p>Отчество: <input type="text"/></p> <p>Дата рождения: <input type="text"/></p> <p>Пол: <input type="text"/></p> <p>Вес: <input type="text"/></p> <p>Контрольная группа: <input type="text"/></p> <p>Дата регистрации: <input type="text"/></p> <p>Комментарий: <input type="text"/></p> <p> <input type="button" value="Назад"/> <input type="button" value="Добавить запись"/> <input type="button" value="Заполнить анамнез"/></p>
Ведение таблицы данных «Анамнез», содержащей информацию о сопутствующих заболеваниях, полученную путём опроса или из медкарты	Программа по вводу данных о анамнезе пациента	<p>Анамнез</p> <p>Код пациента <input type="text"/></p> <p>Сопутствующие заболевания <input type="text"/></p> <p>ASA <input type="text"/></p> <p> <input type="button" value="Назад"/> <input type="button" value="Добавить запись"/></p>

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)																														
Ведение таблицы данных «Операция», содержащей информацию о поставленном диагнозе и проведённой операции	Программа по вводу данных об операции	<p>Операция</p> <p>Код пациента: <input type="text"/></p> <p>Код операции: (№) <input type="text"/></p> <p>Код хирурга: <input type="text"/></p> <p>Диагноз: <input type="text"/></p> <p>Дата операции: <input type="text"/></p> <p>Длительность операции: <input type="text"/></p> <p>Длительность анестезии: <input type="text"/></p> <p>  </p>																														
Ведение таблицы данных «Контроль», содержащей информацию о состоянии пациента, полученную путём снятия показаний с приборов	Программа по вводу данных о контроле состояния пациента	<p>Контроль</p> <p>Код операции: <input type="text"/></p> <p>Номер контроля: <input type="text"/></p> <p>Артериальное давление(сист): <input type="text"/></p> <p>Артериальное давление(диаст): <input type="text"/></p> <p>Артериальное давление(средн): <input type="text"/></p> <p>ЧСС: <input type="text"/></p> <p>SpO2: <input type="text"/></p> <p>  </p>																														
Ведение таблицы данных «Реабилитация», содержащей информацию о скорости восстановления пациента после проведённой операции	Программа по вводу данных о реабилитации пациента	<p>Реабилитация</p> <p>Код операции: <input type="text"/></p> <p>Время пробуждения: <input type="text"/></p> <p>Время экстубации: <input type="text"/></p> <p>Время ориентации: <input type="text"/></p> <p>Время перевода в палату: <input type="text"/></p> <p>  </p>																														
Таблица с соответствующими имеющимися данными	Программа по выводу на экран запрашиваемых данных	<table> <tr> <th>Пациент.Фа</th><th>Диагноз</th><th>Хирурги.Фамил</th></tr> <tr><td>Татарина</td><td>ХКХ</td><td>Ивашко</td></tr> <tr><td>Борзов</td><td>Аппендицит</td><td>Сидоров</td></tr> <tr><td>Байчурова</td><td>Аппендицит</td><td>Сидоров</td></tr> <tr><td>Калакутская</td><td>Полипы ЖКТ</td><td>Орлов</td></tr> <tr><td>Анифов</td><td>ХКХ</td><td>Наилев</td></tr> <tr><td>Круглова</td><td>ХКХ</td><td>Ивашко</td></tr> <tr><td>Воротынова</td><td>Полипы ЖКТ</td><td>Орлов</td></tr> <tr><td>Иванченко</td><td>ХКХ</td><td>Наилев</td></tr> <tr><td>Орлова</td><td>Аппендицит</td><td>Ивашко</td></tr> </table>	Пациент.Фа	Диагноз	Хирурги.Фамил	Татарина	ХКХ	Ивашко	Борзов	Аппендицит	Сидоров	Байчурова	Аппендицит	Сидоров	Калакутская	Полипы ЖКТ	Орлов	Анифов	ХКХ	Наилев	Круглова	ХКХ	Ивашко	Воротынова	Полипы ЖКТ	Орлов	Иванченко	ХКХ	Наилев	Орлова	Аппендицит	Ивашко
Пациент.Фа	Диагноз	Хирурги.Фамил																														
Татарина	ХКХ	Ивашко																														
Борзов	Аппендицит	Сидоров																														
Байчурова	Аппендицит	Сидоров																														
Калакутская	Полипы ЖКТ	Орлов																														
Анифов	ХКХ	Наилев																														
Круглова	ХКХ	Ивашко																														
Воротынова	Полипы ЖКТ	Орлов																														
Иванченко	ХКХ	Наилев																														
Орлова	Аппендицит	Ивашко																														

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)
Возможность ввода информации	Средство для внесения информации	-
Возможность удаления данных	Средство для удаления информации	-
Корректная работа на всех персональных компьютерах медучреждения	Разрабатываемое ПО	-
Доступность всем работникам учреждения	Разрабатываемое ПО	-
Возможность вывода данных	Средство для вывода информации	
Возможность хранения данных	Разрабатываемое ПО	

Таким образом, можно сказать, что функциональные требования были реализованы и функциональное тестирование прошло успешно.

#### 4.2. Интеграционное тестирование

Интеграционное тестирование проводится с целью проверки корректной совокупной работы модулей. Для данного тестирования возможно выполнить поиск информации по любым данным из каждой таблицы. После отображения всех имеющихся модулей на экране, выбираются любые данные из каждого

класса. В итоге на экран выводится таблица, в которой содержатся все требуемые данные (рисунок 4.2).

Пациент.Ф:	ASA	Диагноз	Хирурги	Время проб	Номер контроля	Артериальное $\mu$	Артериальное	Артериальное д	ЧСС	SpO2
Татарина	2	ХКХ	Ивашко	1	1	118	74	78	77	97
Татарина	2	ХКХ	Ивашко	1	1	118	74	78	77	97
Татарина	2	ХКХ	Ивашко	1	2	94	56	70	82	99
Татарина	2	ХКХ	Ивашко	1	2	94	56	70	82	99
Борзов	3	Аппендицит	Сидоров	1	1	128	79	98	79	99
Борзов	3	Аппендицит	Сидоров	1	2	166	93	122	80	99
Борзов	3	Аппендицит	Сидоров	1	3	117	89	109	92	98
Байчурова	2	Аппендицит	Сидоров	1	1	152	105	135	94	99
Байчурова	2	Аппендицит	Сидоров	1	2	126	80	99	68	97
Калакутская	3	Полипы ЖКТ	Орлов	3	1	152	89	124	65	97
Калакутская	3	Полипы ЖКТ	Орлов	3	1	152	89	124	65	97
Калакутская	3	Полипы ЖКТ	Орлов	3	2	110	86	104	78	99
Калакутская	3	Полипы ЖКТ	Орлов	3	2	110	86	104	78	99
Анифов	3	ХКХ	Наилев	2	1	154	70	96	81	99
Анифов	3	ХКХ	Наилев	2	1	154	70	96	81	99
Анифов	3	ХКХ	Наилев	2	1	154	70	96	81	99
Анифов	3	ХКХ	Наилев	2	2	125	92	108	78	97
Анифов	3	ХКХ	Наилев	2	2	125	92	108	78	97
Анифов	3	ХКХ	Наилев	2	2	125	92	108	78	97
Анифов	3	ХКХ	Наилев	2	3	96	55	73	89	97
Анифов	3	ХКХ	Наилев	2	3	96	55	73	89	97
Анифов	3	ХКХ	Наилев	2	3	96	55	73	89	97
Круглова	3	ХКХ	Ивашко	1	1	145	91	103	86	99
Круглова	3	ХКХ	Ивашко	1	2	124	84	100	85	99
Круглова	3	ХКХ	Ивашко	1	3	120	70	105	99	98
Воротынова	2	Полипы ЖКТ	Орлов	2	1	98	66	75	90	97
Воротынова	2	Полипы ЖКТ	Орлов	2	2	181	130	138	66	99
Иванченко	3	ХКХ	Наилев	1	1	122	104	115	72	98
Иванченко	3	ХКХ	Наилев	1	2	143	88	71	89	99
Иванченко	3	ХКХ	Наилев	1	3	155	83	115	124	98

**Рисунок 4.2** – Таблица с запрошенными данными

На основе полученных данных можно сказать, что программа работает корректно и интеграционное тестирование прошло успешно.

### 4.3. Нагрузочное тестирование

Нагрузочное тестирование – это способ тестирования производительности, сбор показателей и определение производительности и времени отклика программно-технической системы или устройства в ответ на внешний запрос. Для тестирования было выбрано следующее количество записей: 1, 30 и 50 (примерное количество пациентов за 2 недели), 100 (примерное количество пациентов за месяц). Данные по нагрузочному тестированию приведены в таблице 4.2.

Таблица 4.2 – Результаты нагрузочного тестирования

Количество записей	Действие	Время отклика
1	Регистрация нового пациента	мгновенно
	Вывод на экран всей имеющейся информации	мгновенно
30	Регистрация нового пациента	мгновенно
	Вывод на экран всей имеющейся информации	мгновенно
50	Регистрация нового пациента	мгновенно
	Вывод на экран всей имеющейся информации	мгновенно
100	Регистрация нового пациента	мгновенно
	Вывод на экран всей имеющейся информации	1,5 секунды

На основе полученных результатов можно сказать, что нагрузочное тестирование прошло успешно.

#### 4.4. Экономическая целесообразность

На разработку отводится 60 рабочих дней. Этапы разработки, сроки выполнения, а также организация работ представлена в таблице 4.3.

Таблица 4.3 – Этапы, сроки и организация работ

№	Название этапа	Исполнитель	Трудоемкость, чел/дни	Продолжительность работ, дни
1	Разработка и утверждение технического задания	Руководитель	3	3
2	Технические предложения	Руководитель	2	3
		Разработчик	1	
3	Эскизный проект:		7	7
3.1	Анализ данных и требований	Разработчик	2	2
3.2	Постановка задачи	Руководитель	2	2
3.3	Разработка общего описания алгоритма функционирования	Руководитель	1	3
		Разработчик	2	
4	Технический проект:		17	17
4.1	Определение формы представления входных и выходных данных	Руководитель	2	6
		Разработчик	4	



№	Название этапа	Исполнитель	Трудоемкость, чел/дни	Продолжительность работ, дни
4.2	Разработка структуры программы и логической структуры базы данных	Руководитель	2	11
		Разработчик	9	
5	Рабочий проект:		30	30
5.1	Программирование и отладка программы	Разработчик	15	15
5.2	Испытание программы	Разработчик	4	4
5.3	Корректировка программы по результатам испытаний	Разработчик	4	4
5.4	Подготовка технической документации на программный продукт	Руководитель	1	2
		Разработчик	1	
5.5	Сдача готового продукта и внедрение	Руководитель	1	5
		Разработчик	4	
Итого			60	60

Статья «Материалы, покупные изделия и полуфабрикаты» включает в себя (кроме стоимости приобретаемых материалов и изделий) транспортно-заготовительные расходы, которые составляют около 15% стоимости затрат по статье. Дополнительно включаются затраты на оформление комплекта документов. Расчёт затрат по первой статье приведён в таблице 4.4.

**Таблица 4.4 – Расчёт затрат по статье**  
«Материалы, покупные изделия, полуфабрикаты»

№	Наименование материалов	Единицы измерения	Количество	Цена за единицу (руб.)	Стоимость (руб.)
1	Диск CD-R 700 Мб	шт	2	39	78
2	USB накопитель для ПК 4Гб	шт	1	599	599
3	Набор картриджей	шт	1	3977	3977
4	Ручка	шт	5	11	55
5	Карандаш	шт	5	9	45
6	Резинка стирательная	шт	2	7	14
7	Бумага А4	пачка	1	175	175
<b>Транспортно-заготовительные расходы</b>					<b>742</b>
<b>Итого</b>					<b>5 685</b>

Расчёт по статье «Основная заработная плата» приведён в таблице 4.5. Оплата за день рассчитана делением месячного оклада на 22 дня.

**Таблица 4.5 – Расчёт основной заработной платы**

№	Наименование этапа	Исполнитель (должность)	Мес. оклад (руб.)	Трудоемкость (чел/дни)	Оплата за день (руб.)	Оплата за этап (руб.)
1	ТЗ	Руководитель	20 000	3	909	2 727
2	ТП	Руководитель	20 000	2	909	1 818
		Разработчик	9 000	1	409	409
3	Эскизный проект	Руководитель	20 000	3	909	2 727
		Разработчик	9 000	4	409	1 636
4	Технический проект	Руководитель	20 000	4	909	3 272
		Разработчик	9 000	13	409	5 317
5	Рабочий проект	Руководитель	20 000	2	909	1 818
		Разработчик	9 000	28	409	11 452
Итого						31 176

Расход по статье «Дополнительная заработная плата» (ДЗП) составляет 25% от суммы основной заработной платы (ОЗП) и рассчитывается по формуле (4.1).

$$\text{ОЗП} \times 0,25 = \text{ДЗП} \quad (4.1)$$

Дополнительная заработная плата научного и производственного персонала составляет по проекту 7 794 руб. Расход по статье «Страховые отчисления» (СО) составляет 30% от фонда оплаты труда (ФОТ), который состоит из основной и дополнительной заработной платы, и рассчитывается по формуле (4.2).

$$(\text{ОЗП} + \text{ДЗП}) \times 0,3 = \text{СО} \quad (4.2)$$

Таким образом, страховые отчисления по текущему проекту составляют 11 691 руб. Статья «Накладные расходы» (НР) составляет 250% от суммы ОЗП научного и производственного персонала и рассчитываются по формуле (4.3).

$$\text{ОЗП} \times 2,5 = \text{НР} \quad (4.3)$$

Накладные расходы текущего проекта составляют 77 940 руб. При разработке, отладке и тестировании программного продукта использовался один компьютер, за которым было проведено 60 рабочих дней по 8 часов. Исходя из расчета оплаты 10 рублей за 1 час машинного времени, сумма составит 4 800 руб. Полная себестоимость (ПС) проекта приведена в таблице 4.6.

**Таблица 4.6 – Полная себестоимость проекта**

№	Номенклатура статей расходов	Затраты (руб.)
1	Материалы, покупные изделия и полуфабрикаты	5 685
2	Специальное оборудование для научных работ	–
3	ОЗП научного и производственного персонала	31 176
4	ДЗП научного и производственного персонала	7 794
5	Страховые взносы в социальные фонды	11 691
6	Расходы на научные и производственные командировки	–
7	Оплата работ, выполненных сторонними организациями	–
8	Прочие прямые расходы	4 800
9	Накладные расходы	77 940
<b>Итого</b>		<b>139 086</b>

Норма прибыли (НП) рассчитывается по формуле (4.4).

$$\text{ПС} \times 0,3 = \text{НП} \quad (4.4)$$

Таким образом, норма прибыли по текущему проекту составляет 41 726 руб. НДС рассчитывается по формуле (4.5).

$$(ПС + НП) \times 0,18 = НДС \quad (4.5)$$

НДС текущего проекта составляет 32 546 руб. Договорная цена будет рассчитываться по формуле (4.6).

$$ПС + НДС = ДЦ \quad (4.6)$$

Договорная цена текущего проекта составляет 171 632 руб., что позволяет утверждать о экономической целесообразности разработки.

## Заключение

В процессе работы были собраны и проанализированы пользовательские требования. На их основе были составлены функциональные требования, которым должно удовлетворять разрабатываемое приложение. Также была составлена матрица отслеживания требований, которая помогала в процессе проектирования и разработки отслеживать соответствие программных компонентов ожиданиям конечных пользователей.

На этапе проектирования благодаря описанию некоторых бизнес – процессов в моделях «AS-IS» и «TO-BE» в очередной раз подтвердилась необходимость создания приложения, а также удалось составить наиболее удобную для конечного пользователя структуру разрабатываемого ПО.

Далее на основе всех собранных данных была разработана программа в среде MS Access. После разработки было проведено тестирование приложения на предмет соответствия всем требованиям, качества работы всех элементов и скорости выполнения некоторых задач. Также была доказана экономическая целесообразность работы.

Таким образом, можно утверждать, что поставленная цель работы была достигнута, а задачи – полностью выполнены. В процессе подготовки пояснительной записки для повышения наглядности было составлено 10 таблиц, приведено 17 рисунков (моделей, проектов программы и пользовательских интерфейсов, снимков с экрана), а также использовано 6 расчётных формул для экономического раздела.

## Список литературы

1. Химонин Ю. Сбор и анализ требований к программному продукту. Версия 1.03 – 2009 – 51 с.
2. Никандров В. В. Экспериментальная психология. Учебное пособие. – СПб.: «Речь» – 2003. – 480 с.
3. ГОСТ Р 7.0.8-2013. Национальный стандарт Российской Федерации. Система стандартов по информации, библиотечному и издательскому делу. Делопроизводство и архивное дело. Термины и определения" (утв. Приказом Росстандарта от 17.10.2013 N 1185-ст).
4. Федеральный закон «Об основах охраны здоровья граждан Российской Федерации» (принят Государственной Думой 01.10.2011, одобрен Советом Федераций 09.10.2011).
5. Шеер А. – В. ARIS - моделирование бизнес-процессов / пер. с англ. и ред. А. А. Рыбьянец – 3-е изд. – М.: Вильямс – 2009. – 223 с.
6. Дейт К. Дж. Введение в системы баз данных / пер. с англ. и ред. К. А. Птицына – 8-е изд. – М.: Вильямс – 2016. – 1327 с.
7. Компьютерная справочная правовая система «КонсультантПлюс»  
<http://www.consultant.ru>
8. Карчевский Е. М., Филиппов И. Е., Филиппова И.А. Access 2010 в примерах. Учебное пособие – К.: Казанский университет – 2012. – 140 с.
9. Фуллер Л. У., Кук К., Кауфельд Д. Microsoft Office Access 2007 для “чайников” / пер. с англ. и ред. А. Г. Сысоян – М.: Вильямс – 2007. – 384 с.
10. Одиноккина С. В. Разработка баз данных в Microsoft Access 2010 – СПб.: НИУ ИТМО – 2012. – 83 с.
11. Бекаревич Ю. Б., Пушкина Н. В. Самоучитель Microsoft Access 2013 – СПб.: БХВ-Петербург – 2014. – 464 с.

12. Официальный сайт Российской Государственной Библиотеки  
<http://www.rsl.ru>

13. INTERNATIONAL STANDARD ISO/IEC/IEEE 24765. Systems and software engineering — Vocabulary – 2010. – 410 с.

## **ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ**

на выпускную квалификационную работу студентки  
Физико-технологического института МИРЭА группы ТББО-03-13  
Орешкиной А.М. на тему «Анализ, проектирование, разработка и тестирование  
приложения для автоматизации городской больницы»

Автоматизация работы предприятия позволяет ускорить выполнение рутинных операций, тем самым обеспечивая сотрудникам возможность заниматься целевой деятельностью. Именно поэтому тематика проектирования и разработки информационных систем является весьма актуальной и востребованной, что доказывает целесообразность работы Орешкиной А.М.

Квалификационная работа состоит из четырех глав, введения, заключения и списка литературных источников. Первая глава описывает процесс идентификации требований, предъявляемых к проектируемой программной системе городской больницы. Вторая глава посвящена моделированию процессов, данных и приложений больницы, в то время как третья – реализации программ в среде MS Access. Завершает работу четвертая глава, демонстрирующая результаты тестирования разработанного программного приложения.

Дипломная работа Орешкиной А.М. структурирована, логична и наглядна. Содержание работы оформлено доступным даже для не технического специалиста языком. Разработанное в рамках квалификационной работы приложение обладает общностью, удобством и простотой использования. Заявленные цель и задачи работы реализованы в полном объеме.

Незначительными недостатками работы являются: не соблюдение нотации моделирования ARIS eEPC (рис.2.2, название операций в форме существительного, отсутствие элемента событий), поверхностное документирование разработанной программы (стр.33-36), а также скудный обзор литературных источников (стр.48-49, 13 источников).

Выпускная квалификационная работа Орешкиной А.М. выполнена в соответствии с предъявляемыми требованиями, рекомендована к защите на соискание степени бакалавра по направлению подготовки «12.03.04 – Биотехнические системы и технологии» и заслуживает высокой оценки.

Кафедра оптических и биотехнических систем и технологий,  
Физико-технологический институт МИРЭА,  
к.т.н., доц. МИРЭА  
Д.Ю. Степанов



## **РЕЦЕНЗИЯ**

на выпускную квалификационную работу по теме  
**«Анализ, проектирование, разработка и тестирование приложения для  
автоматизации городской больницы»**  
студентки 4 курса Физико-технологического института МТУ (МИРЭА)  
группы ТББО-03-13 Орешкиной А. М.

Предоставленная на рецензию выпускная квалификационная работа посвящена разработке и тестированию приложения, которое позволит автоматизировать некоторые области работы медицинского персонала в городской клинической больнице. Актуальность выбранной темы обусловлена необходимостью ускорить регистрацию новых пациентов и поиск необходимой информации в условиях ограниченного времени.

Выпускная квалификационная работа Орешкиной А. М. содержит 49 страниц пояснительной записки, которая включает в себя четыре главы, введение, заключение и список литературы. Автором проведён глубокий и всесторонний анализ выбранной темы, кроме того, на каждом этапе производился грамотный выбор путей и средств решения поставленных задач.

Работа выполнена на высоком техническом уровне и оформлена согласно стандартам и рекомендациям МИРЭА. Орешкина А. М. продемонстрировала профессиональный подход к сбору, структуризации и анализу разнородной информации, а также значительные навыки проектирования приложений в среде MS Access.

К основным достоинствам работы можно отнести получение программного продукта, готового к вводу в эксплуатацию. Недостатком данной работы является визуальная составляющая пользовательских интерфейсов. Однако выявленные недоработки не влияют на качество проделанной работы, а существенных недостатков в дипломе не выявлено.

В целом выпускная квалификационная работа является актуальной научно – практической работой и выполнена на высоком инженерно – техническом уровне. Рецензируемый проект свидетельствует о высокой подготовке и хороших практических навыках разработчика в предметной области, демонстрирует творческий подход к работе.

Таким образом, выпускная квалификационная работа Орешкиной Анны Михайловны выполнена полностью в соответствии с предъявляемыми требованиями, рекомендована к защите на соискание степени бакалавра по направлению подготовки «12.03.04 – Биотехнические системы и технологии» и заслуживает высокой оценки.

Рецензент:

заведующий кафедрой анестезиологии и реаниматологии ФУВ  
ГБУЗ МО МОНИКИ им. М.Ф. Владимирского, д.м.н., доцент

Овезов А. М.