



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:

**«АВТОМАТИЗАЦИЯ КЛЮЧЕВЫХ БИЗНЕС-ПРОЦЕССОВ
ГОРОДСКОЙ БОЛЬНИЦЫ НА ОСНОВЕ КАСКАДНОЙ МОДЕЛИ
ВНЕДРЕНИЯ»**

Студент:

Катасонова Н.С.

Научный руководитель:

к.т.н., доц. МИРЭА Степанов Д.Ю.

Москва – 2018

Оглавление

Введение	3
Раздел 1. Методология внедрения по каскадной схеме	6
1.1. Описание методологии внедрения ИС по каскадной схеме	6
1.2. Этапы каскадной модели	9
1.3. Применение каскадной модели в работе	9
Раздел 2. Сбор, анализ и приоритизация требований	12
2.1. Способы сбора требований.....	12
2.2. Пользовательские требования.....	13
2.3. Функциональные требования	15
2.4. Матрица отслеживания требований	16
Раздел 3. Проектирование ключевых бизнес-процессов в моделях «AS-IS» и «TO-BE».....	21
3.1. Способы проектирования	23
3.2. Описание ключевых бизнес-процессов.....	27
3.3. Архитектура данных разрабатываемой системы	31
3.4. Описание разрабатываемой системы	33
Раздел 4. Разработка и тестирование ключевых бизнес-процессов с использованием PHP и MySQL	35
4.1. Принципы разработки программ	35
4.2. Реализация основных функций системы	36
4.3. Функциональное и системное тестирование	44
4.4. Нагрузочное тестирование	48
Заключение.....	2
Список использованных литературных источников	4
Приложение А	7

Введение

Темпы современной жизни, как и население планеты, возрастают с каждым годом. В связи с этим повышается объем работ с информацией для персонала различных государственных учреждений. При этом имеется необходимость в следующих аспектах:

- быстрое и качественное обслуживание;
- быстрая работа с данными;
- увеличение персонала и числа учреждений, соответствующее росту населения;
- доступность данных.

Следовательно, будет целесообразно создать систему, которая позволит персоналу какого-либо заведения упростить работу. Наличие баз данных в городской больнице позволит решить многие вопросы, такие как:

- скорость обслуживания – отсутствие бумажной волокиты и наличие электронного устройства с доступом в Интернет позволит увеличить скорость работы с данными и избавиться от очередей;
- улучшение качества работы врачей – в связи с тем, что отпадет необходимость в заполнении лишних бумаг у врачей будет больше возможностей уделять пациентам;
- доступность данных – при походе в другое учреждение пациенту не нужно будет забирать свою медицинскую карту из больницы, достаточно просто показать ее на сайте; врачи и медсестры смогут работать и следить за состоянием пациентов дистанционно;
- удобство работы с данными – наличие связанных между собой баз данных позволит легко их дополнять или редактировать;

- сохранность данных – срок хранения существенно возрастает по сравнению с бумажными носителями информации.

Целью данной работы будет создание системы для автоматизации ключевых бизнес-процессов городской больницы города. При выполнении будут использованы системы управления базами данных (СУБД), например, MySQL [1], а также средства html и php. Преимущества данной системы заключаются в следующем:

- простота освоения и использования технологии (для этого достаточно иметь минимальные навыки пользования ПК или мобильным устройством и наличие сети Internet);
- доступность (просмотр или редакция доступны из любого места с любого устройства, при наличии интернета);
- удобность администрирования (сайт и базу данных можно хранить на локальном сервере, расположенном в больнице);
- наглядность;
- обновление в реальном времени (если в карту внесены изменения, они тут же отобразятся у пациента, нет необходимости в постоянном ее получении);
- отсутствие очередей в регистратуру.

Также для удобства работы с системой необходим простой интерфейс для создания и просмотра медицинской карты пациента, а также базы данных для хранения этих сведений. В конечном итоге, целью данной работы будет автоматизация ключевых бизнес-процессов городской больницы при помощи каскадной модели внедрения. В процессе ее реализации необходимо выполнить следующие задачи:

- описать и применить методологию внедрения ИС по каскадной схеме;
- проанализировать требования, предъявляемые к системе;

- спроектировать процессы, данные и программы;
- разработать систему;
- провести тестирование.

Изучению проблем автоматизации информационных систем посвящены труды Маглинец Ю.А. [2-3], в которых рассматриваются различные понятия, связанные с ИС, различные стандарты и требования к системам. Моделирование бизнес-процессов в своих работах [4], [5] описывает специалист по менеджменту и информационным технологиям для организаций, а также руководитель компании IDS Scheer – производителя системы ARIS, А.В. Шеер. Решением подобных задач в своей выпускной квалификационной работе [6] занималась Орешкина А.М. Суммируя все вышесказанное, можно с уверенностью заявить, что разработка данной системы будет весьма актуальной и поможет объединить и улучшить предыдущие достижения в этой области. При этом, всегда будет иметься возможность легко совершенствовать функционал системы без существенных изменений в ее структуре (например, система для записи пациентов на прием к врачу и т.д.).

Раздел 1. Методология внедрения по каскадной схеме

1.1. Описание методологии внедрения ИС по каскадной схеме

Каскадная модель (англ. waterfall model, иногда переводят как модель «Водопад») – модель процесса разработки программного обеспечения, основной характеристикой которой является разбиение всей разработки на этапы, при этом переход на следующий этап происходит только после полного завершения работ на текущем. Позволяет быстро создавать систему, без дополнительных накладных расходов на организацию процесса разработки [7].



Рисунок 1.1 –Каскадная схема внедрения информационных систем

В эту категорию попадают сложные расчетные системы, системы реального времени и другие подобные задачи. Однако, в процессе использования этого подхода обнаружился ряд его недостатков, вызванных прежде всего тем, что реальный процесс создания ПО никогда полностью не укладывался в такую жесткую схему. В процессе создания ПО постоянно

возникала потребность в возврате к предыдущим этапам и уточнении или пересмотре ранее принятых решений. У каскадного метода внедрения ИС (информационной системы) имеются такие преимущества, как:

- стабильность требований в течение всего жизненного цикла разработки;
- возможность последовательного устранения возникающих сложностей;
- определенность и понятность шагов модели, и простота ее применения;
- упрощение возможности осуществления планирования, контроля и управления проектом;
- доступность для понимания заказчиками;
- эффективность для проектов с четкими и понятными, но трудно реализуемыми требованиями;
- эффективность для проектов с высокими требованиями к качеству при отсутствии жестких ограничений затрат и графика работ.

Основным недостатком каскадного подхода является существенное запаздывание с получением результатов. Согласование результатов с пользователями производится только в точках, планируемых после завершения каждого этапа работ, требования к ИС «заморожены» в виде технического задания на все время ее создания. Таким образом, пользователи могут внести свои замечания только после того, как работа над системой будет полностью завершена. В случае неточного изложения требований или их изменения в течение длительного периода создания ПО (программного обеспечения), пользователи получают систему, не удовлетворяющую их потребностям. Модели (как функциональные, так и информационные) автоматизируемого объекта могут устареть одновременно с их утверждением. К недостаткам каскадного метода также стоит отнести:

- сложность четкого формулирования требований в начале жизненного цикла и невозможность их динамического изменения на его протяжении;
- последовательность линейной структуры процесса разработки, в результате возврат к предыдущим шагам для решения возникающих проблем приводит к увеличению затрат и нарушению графика работ;
- непригодность промежуточного продукта для использования;
- невозможность гибкого моделирования систем, не имеющих аналогов;
- позднее обнаружение проблем, связанных со сборкой, в связи с одновременной интеграцией всех результатов в конце разработки;
- недостаточное участие пользователя в создании системы – только в самом начале (при разработке требований) и в конце (во время приемочных испытаний);
- невозможность предварительной оценки качества системы пользователем;
- проблемность финансирования проекта, связанная со сложностью единовременного распределения больших денежных средств.

Отсутствие обратной связи также негативно влияет на качество конечного продукта, так как у разработчика не имеется возможности вернуться к предыдущему этапу [8]. Ограничение области применения каскадной модели определяется ее недостатками. Ее использование наиболее эффективно в следующих случаях:

- при разработке проектов с четкими, неизменяемыми в течение ЖЦ (жизненного цикла) требованиями, понятными реализацией и техническими методиками;

- при разработке проекта, ориентированного на построение системы или продукта такого же типа, как уже разрабатывались разработчиками ранее;
- при разработке проекта, связанного с созданием и выпуском новой версии уже существующего продукта или системы;
- при разработке проекта, связанного с переносом уже существующего продукта на новую платформу;
- при выполнении больших проектов, в которых задействовано несколько больших команд разработчиков.

1.2. Этапы каскадной модели

Исходя из выбранной модели внедрения, работа над системой будет также разделена на 7 этапов:

- подготовка проекта – выбор концепции реализации проекта, определение предварительного объема проекта;
- проектирование – анализ требований, проектирование и утверждение решений, формирование списка доработок системы;
- реализация – настройка и доработка системы, системное тестирование, документирование системы;
- подготовка к ОПЭ (опытно-промышленная эксплуатация) /ПЭ (промышленная эксплуатация) – регистрация тестовых данных;
- ОПЭ/ОЭ (опытная эксплуатация) – регистрация и устранение дефектов.
- переход к ПЭ (промышленная эксплуатация) – техническая подготовка системы;
- ПЭ – обновление документации, передача в поддержку.

1.3. Применение каскадной модели в работе

На первом этапе работы, исходя из темы дипломной работы, была выбрана каскадная модель внедрения проектируемой системы.

Проанализированы ее преимущества и недостатки. Оценены возможности применения готового программного продукта в городской больнице. Определен объем выполняемых работ. Результат: применение каскадной модели рационально. Система при небольших доработках под конкретное учреждение будет готова к применению. Объем работ: 40 часов на выполнение каждого этапа модели. Переход на следующий этап.

Второй этап включает в себя общение с заказчиком, работу с заявленными и выявленными требованиями, поиск пути решения проблем по разработке системы. Результат: в ходе общения определены требования и способы их выполнения. Переход на следующий этап.

На третьем этапе необходимо сосредоточиться на реализации программных компонентов для выполнения требований заказчика. Результат: готова промежуточная версия системы. Проведены необходимые тесты, система функционирует должным образом. Переход на следующий этап.

На четвертом этапе необходимо собрать и проанализировать данные с предыдущих тестов. Результат: готова информация о работе промежуточной версии системы.

Пятый этап подразумевает тестовые запуски системы в выбранном учреждении, обнаружение и устранение дефектов. Появление готового программного продукта. Переход на следующий этап. Результат: в ходе тестовых запусков система работает нормально, дефекты устранены. Переход на следующий этап.

На шестом этапе проводятся заключительные приготовления перед окончательным внедрением системы. Последние тесты и устранение оставшихся дефектов. Результат: система готова к работе в городской больнице. Переход на следующий этап.

На заключительном, седьмом, этапе происходит установка на рабочий сервер, написание инструкций и документации, обучение пользователей, техподдержка. Результат: система функционирует согласно требованиям заказчика, работа выполнена.

Раздел 2. Сбор, анализ и приоритизация требований

Анализ требований – часть процесса разработки программного обеспечения, включающая в себя сбор требований к ПО, их систематизацию, выявление взаимосвязей, а также документирование [2]. В процессе сбора требований важно принимать во внимание возможные противоречия требований различных заинтересованных лиц, таких как заказчики, разработчики или пользователи.

Полнота и качество анализа требований играют ключевую роль в успехе всего проекта. Требования к ПО должны быть документируемые, выполнимые, тестируемые, с уровнем детализации, достаточным для проектирования системы. Требования могут быть функциональными и нефункциональными. Анализ требований включает три типа деятельности:

- сбор требований – общение с клиентами и пользователями, чтобы определить, каковы их требования; анализ предметной области;
- анализ требований – определение, являются ли собранные требования неясными, неполными, неоднозначными или противоречащими; решение этих проблем; выявление взаимосвязи требований;
- документирование требований – требования могут быть задокументированы в различных формах, таких как простое описание, сценарии использования, пользовательские истории, или спецификации процессов [9].

2.1. Способы сбора требований

Для того, чтобы получить сведения, необходимые для выполнения задания, можно использовать один или несколько следующих методов:

- анкетирование;
- интервью;

- автозапись;
- изучение существующей документации;
- повторное использование спецификации;
- представитель заказчика в компании разработчика;
- работа «в поле»;
- обучение;
- мозговой штурм;
- совещание;
- use case (сценарий использования).

Конечно, лучшим вариантом будет использовать более одного из представленных способов, что позволит взглянуть на решение данного задания с разных сторон, учесть мнение заказчика, персонала и пользователей [10]. В этой главе необходимо сформулировать пользовательские и функциональные требования к системе, полученные из интервью и посредством изучения уже существующей документации для того, чтобы получить полное представление о том, какой эта система будет.

2.2. Пользовательские требования

Пользовательские требования – это требования, формулируемые пользователями к конечному продукту. На основе выбранных методов были выявлены следующие требования:

- возможность вывода данных о пациенте;
- возможность просмотра информации о персонале;
- возможность для заведения медицинской карты;
- возможность для удаления медицинской карты;
- возможность просмотра и редактирования данных из БД «Пациенты»;
- возможность просмотра и редактирования данных из БД «Эпикриз»;
- возможность просмотра и редактирования данных из БД «Анамнез»;

- возможность просмотра и редактирования данных из БД «Температура»;
- возможность просмотра и редактирования данных из БД «Назначения»;
- возможность просмотра и редактирования данных из БД «Дневник»;
- возможность просмотра и редактирования данных из БД «Персонал»;
- возможность печати карты при выписке;
- возможность поиска медицинской карты по ее номеру или фамилии пациента;
- возможность работы на любом устройстве;
- наличие базы данных для температурных листов пациентов;
- наличие базы данных для учета лечебных назначений;
- наличие базы данных для хранения врачебных заключений;
- наличие базы данных для хранения данных о пациенте после его поступления;
- наличие базы данных для хранения информации о медицинском персонале;
- наличие базы данных для хранения медицинских карт пациентов;
- наличие базы данных для эпикриза пациентов;
- невозможность подмены пациентом данных о себе;
- работа системы в любом месте;
- удобный интерфейс на мобильных устройствах;
- конфиденциальность персональных данных;
- доступность (легкость восприятия);
- легкость в использовании (простота);
- возможность хранения данных.

2.3. 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:

- вывод информации из базы данных «Пациенты»;
- страница, посвященная медицинскому персоналу;
- добавление данных о пациенте;
- удаление медицинской карты пациента;
- просмотр и редактирование данных из БД «Пациенты»;
- просмотр и редактирование данных из БД «Эпикриз»;
- просмотр и редактирование данных из БД «Анамнез»;
- просмотр и редактирование данных из БД «Температура»;
- просмотр и редактирование данных из БД «Назначения»;
- просмотр и редактирование данных из БД «Дневник»;
- просмотр и редактирование данных из БД «Персонал»;
- печать карты пациента при его выписке;
- поиск медицинской карты пациента;
- корректная работа на любом устройстве;
- хранение данных;
- доступность работникам и пользователям;
- простота разрабатываемого интерфейса;
- ведение температурных листов пациентов;
- ведение базы учета лечебных назначений;
- ведение базы данных с врачебными заключениями;

- ведение «Дневника» пациента, путем последующего добавления сведений;
- ведение базы данных «Персонал», содержащей информацию о персонале, полученную путём регистрации персонала;
- ведение базы данных «Пациенты», содержащей информацию о пациенте, полученную путём опроса или из медкарты;
- ведение эпикриза пациентов;
- ведение базы данных «Анамнез», содержащей информацию о сопутствующих заболеваниях, полученную путём опроса или из медкарты;
- разграничение доступа между пациентом и персоналом;
- сервер для размещения сайта;
- мобильная версия сайта;
- частичное шифрование личных данных.

2.4. Матрица отслеживания требований

Матрица отслеживания требований представляет собой таблицу, которая связывает требования с их происхождением и отслеживает требования на протяжении жизненного цикла проекта. Применение матрицы отслеживания требований помогает удостовериться, что каждое требование увеличивает ценность бизнеса, связывая его с целями бизнеса и проекта. Это позволяет отслеживать требования на протяжении жизненного цикла проекта, что помогает удостовериться в том, что требования, одобренные в документах по требованиям, выполнены в конце проекта. Наконец, матрица отслеживания требований обеспечивает структуру для управления изменениями содержания продукта [11].

Параметры, связанные с каждым требованием, могут быть записаны в матрице отслеживания требований. Данные параметры помогают определить

ключевую информацию относительно требований. Типичные параметры, используемые в матрице отслеживания требований, могут включать в себя [12]:

- уникальный идентификатор;
- текстовое описание требования;
- обоснование включения в список требований;
- владельца;
- источник;
- приоритет;
- версию;
- текущий статус:
 - активный;
 - отменен;
 - отложен;
 - добавлен;
 - одобрен;
- дату выполнения.

Дополнительные параметры, позволяющие удостовериться, что требование удовлетворяет заинтересованные стороны проекта, могут включать также стабильность, сложность и критерии приемки. Матрица отслеживания требований разрабатываемой системы для автоматизации ключевых бизнес-процессов городской больницы представлена в табл.2.1.

Таблица 2.1 – Матрица отслеживания требований

Пользовательское требование	Функциональное требование	Программный компонент	Приоритет
Для пациентов:			

Пользовательское требование	Функциональное требование	Программный компонент	Приоритет
Возможность вывода данных о пациенте	Вывод информации из базы данных «Пациенты»	Форма запроса по базе данных «Пациенты»	Высокий
Возможность просмотра информации о персонале	Страница, посвященная мед. персоналу	Страница, посвященная мед. персоналу	Низкий
Для персонала:			
Возможность для заведения медицинской карты	Добавление данных о пациенте	Форма для регистрации новых пациентов	Высокий
Возможность для удаления медицинской карты	Удаление медицинской карты пациента	Форма удаления данных о пациентах	Средний
Возможность просмотра и редактирования данных из БД «Пациенты»	Просмотр и редактирование данных из БД «Пациенты»	Формы просмотра и редактирования данных	Средний
Возможность просмотра и редактирования данных из БД «Эпикриз»	Просмотр и редактирование данных из БД «Эпикриз»	Формы просмотра и редактирования данных	Средний
Возможность просмотра и редактирования данных из БД «Анамнез»	Просмотр и редактирование данных из БД «Анамнез»	Формы просмотра и редактирования данных	Средний
Возможность просмотра и редактирования данных из БД «Температура»	Просмотр и редактирование данных из БД «Температура»	Формы просмотра и редактирования данных	Средний
Возможность просмотра и редактирования данных из БД «Назначения»	Просмотр и редактирование данных из БД «Назначения»	Формы просмотра и редактирования данных	Средний
Возможность просмотра и редактирования данных из БД «Дневник»	Просмотр и редактирование данных из БД «Дневник»	Формы просмотра и редактирования данных	Средний

Пользовательское требование	Функциональное требование	Программный компонент	Приоритет
Возможность просмотра и редактирования данных из БД «Персонал»	Просмотр и редактирование данных из БД «Персонал»	Формы просмотра и редактирования данных	Средний
Возможность печати карты при выписке	Печать карты пациента при его выписке	Форма вызова данных из баз на печать	Средний
Общее:			
Возможность поиска медицинской карты по ее номеру или фамилии пациента	Поиск медицинской карты пациента	Форма поиска карты	Высокий
Возможность работы на любом устройстве	Корректная работа на любом устройстве	Разрабатываемая система	Высокий
Возможность хранения данных	Хранение данных	Разрабатываемая система	Высокий
Доступность	Доступность работникам и пользователям	Разрабатываемая система	Высокий
Легкость в использовании	Простота разрабатываемого интерфейса	Разрабатываемая система	Высокий
Наличие базы данных для температурных листов пациентов	Ведение температурных листов пациентов	База данных «Температура»	Средний
Наличие базы данных для учета лечебных назначений	Ведение базы учета лечебных назначений	База данных «Назначения»	Средний
Наличие базы данных для хранения врачебных заключений	Ведение базы данных с врачебными заключениями	База данных «Заключения»	Средний
Наличие базы данных для хранения данных о пациенте после его поступления	Ведение «Дневника» пациента, полученную путем последующего добавления сведений	База данных «Дневник»	Средний
Наличие базы данных для хранения информации о медицинском персонале	Ведение базы данных «Персонал», содержащей информацию о персонале, полученную путём регистрации персонала	База данных «Персонал»	Высокий

Пользовательское требование	Функциональное требование	Программный компонент	Приоритет
Наличие базы данных для хранения медицинских карт пациентов	Ведение базы данных «Пациенты», содержащей информацию о пациенте, полученную путём опроса или из медкарты	База данных «Пациенты»	Высокий
Наличие базы данных для эпикриза пациентов	Ведение эпикриза пациентов	База данных «Эпикриз»	Средний
Наличие данных об анамнезе пациента	Ведение базы данных «Анамнез», содержащей информацию о сопутствующих заболеваниях, полученную путём опроса или из медкарты	База данных «Анамнез»	Средний
Невозможность подмены пациентом данных о себе	Разграничение доступа между пациентом и персоналом	Форма регистрации и авторизации персонала	Высокий
Работа системы в любом месте	Сервер для размещения сайта	Разрабатываемая система	Высокий
Удобный интерфейс на мобильных устройствах	Мобильная версия сайта	Разрабатываемая система	Низкий
Конфиденциальность персональных данных	Частичное шифрование личных данных	Разрабатываемая система	Низкий

Раздел 3. Проектирование ключевых бизнес-процессов в моделях «AS-IS» и «TO-BE»

Модель AS-IS («как есть») позволит отразить текущую ситуацию, систематизировать протекающие в организации процессы и информационные потоки в рамках этих процессов. На основе данной модели выявляются проблемные (узкие) места при выполнении и взаимодействии процессов, определяется необходимость внесения тех или иных изменений. В рамках моделирования бизнес-процессов осуществляется анализ их текущего состояния и способы оптимизации. Для выявления неэффективности в действующих процессах, как правило, проводится их декомпозиция вплоть до элементарных операций. Таким образом могут быть выявлены детали процессов, которые могут препятствовать их эффективному выполнению:

- избыточные или дублирующиеся функции;
- нерегламентированные или слабо регламентированные операции, которые разные участники могут выполнять по-своему;
- документооборот не соответствует выполняемому бизнес-процессу (требуемый документ не оказывается в нужном месте в нужное время);
- отсутствие обратных связей по управлению (на проведение функции не оказывает влияния ее результат);
- отсутствие обратной связи по входным данным (объекты или информация используются неправильно или нерационально) и т.д.

Детализация процессов по модели «AS-IS» («как есть») позволяет выявить недостатки в исследуемой области деятельности, которые будут учитываться при создании модели «TO-BE» («как должно быть») – модели новой (усовершенствованной) организации процессов [13].

Найденные в модели «AS-IS» недостатки исправляются путем создания модели «TO-BE» (как будет), т.е. модели новой организации процессов на

предприятию. Создание и внедрение ИС приводит к изменению условий выполнения отдельных операций, структуры процессов и предприятия в целом. Это приводит к необходимости изменения системы правил, используемых на предприятии, модификации должностных инструкций сотрудников.

Функциональная модель «ТО-ВЕ» позволяет уже на стадии проектирования будущей ИС определить эти изменения. Применение функциональной модели «ТО-ВЕ» позволяет не только сократить сроки внедрения информационной системы, но также снизить риски, связанные с невосприимчивостью персонала к информационным технологиям. Модель «ТО-ВЕ» нужна для анализа альтернативных (лучших) путей выполнения функции и документирования того, как компания будет делать бизнес в будущем.

Общепринятая технология проектирования ИС подразумевает сначала создание модели «AS-IS», затем на основе ее анализа определяются направления улучшения процессов, т.е. создание модели «ТО-ВЕ». Только на основе разработанной модели «ТО-ВЕ» в дальнейшем происходит построение модели данных, прототипа и затем окончательного варианта ИС. Если в основу автоматизации предприятия будет изначально заложена модель «AS-IS», то создаваемая «новая» ИС будет выполняться по принципу «все оставить, как есть», и вместо информатизации предприятия на основе новых ИТ, произойдет (в лучшем случае) простая компьютеризация несовершенных процессов. В результате внедрение и эксплуатация такой «новой» ИС приведет к дополнительным издержкам на закупку оборудования, создание программного обеспечения и их сопровождение.

3.1. Способы проектирования

Моделирование бизнес-процессов является важной составной частью проектов по созданию крупномасштабных систем ПО. Отсутствие таких моделей является одной из главных причин неудач многих проектов. Назначением будущего ПО является, в первую очередь, решение проблем бизнеса. Требования к ПО формируются на основе бизнес-модели, а критерии проектирования систем прежде всего основываются на наиболее полном их удовлетворении. Модели бизнес-процессов являются не просто промежуточным результатом, используемым консультантом для выработки каких-либо рекомендаций и заключений. Они представляют собой самостоятельный результат, имеющий большое практическое значение.

На сегодняшний день в моделировании бизнес-процессов преобладает процессный подход. Его основной принцип заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. Модель, основанная на организационно-штатной структуре, может продемонстрировать лишь хаос, царящий в организации (о котором в принципе руководству и так известно, иначе оно бы не инициировало соответствующие работы), на ее основе можно только внести предложения об изменении этой структуры. С другой стороны, модель, основанная на бизнес-процессах, содержит в себе и организационно-штатную структуру предприятия.

Процессный подход может использовать любые из перечисленных выше средств моделирования. Однако, в настоящее время наблюдается тенденция интеграции разнообразных методов моделирования и анализа систем, проявляющаяся в форме создания интегрированных средств моделирования. Одним из таких средств является продукт, носящий название ARIS (Architecture of Integrated Information System), разработанный германской фирмой IDS Scheer

[14]. Система ARIS представляет собой комплекс средств анализа и моделирования деятельности предприятия. Ее методическую основу составляет совокупность различных методов моделирования, отражающих разные взгляды на исследуемую систему. Одна и та же модель может разрабатываться с использованием нескольких методов, что позволяет использовать ARIS специалистам с различными теоретическими знаниями и настраивать его на работу с системами, имеющими свою специфику. ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы:

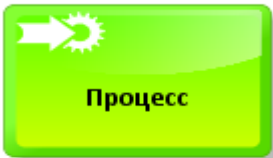
- организационные модели, представляющие структуру системы - иерархию организационных подразделений, должностей и конкретных лиц, связи между ними, а также территориальную привязку структурных подразделений;
- функциональные модели, содержащие иерархию целей, стоящих перед аппаратом управления, с совокупностью деревьев функций, необходимых для достижения поставленных целей;
- информационные модели, отражающие структуру информации, необходимой для реализации всей совокупности функций системы;
- модели управления, представляющие комплексный взгляд на реализацию бизнес-процессов в рамках системы.

В процессе моделирования каждый аспект деятельности предприятия сначала рассматривается отдельно, а после детальной проработки всех аспектов строится интегрированная модель, отражающая все связи между различными аспектами [15].

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. При описании бизнес-процессов воспользуемся предлагаемым разработчиком для их составления приложением ARIS Express [16]. Элементы, используемые в данной нотации, приведены в табл.3.1.

Таблица 3.1 – Графические элементы ARIS VACD (0-1 уровень)

Наименование	Описание	Графическое представление
Процесс	Элемент отражает процессы, выполняемые в организации	
Входящий или исходящий объект	Элемент отражает реальные носители информации	
Организационная единица	Элемент, отражающий различные организационные звенья организации, ответственный за исполнение процесса	

Модели в ARIS представляют собой диаграммы, элементами которых являются разнообразные объекты – «функция», «событие», «структурное подразделение», «документ» и т.п. Между объектами устанавливаются разнообразные связи. Каждому объекту соответствует определенный набор атрибутов, которые позволяют ввести дополнительную информацию о

конкретном объекте. Значения атрибутов могут использоваться при имитационном моделировании или для проведения стоимостного анализа.

Основная бизнес-модель ARIS – eEPC (extended Event Driven Process Chain) – расширенная модель цепочки процессов, управляемых событиями). Подходит для более низкоуровневого (2-8 уровень) описания процессов. Бизнес-процесс в нотации eEPC представляет собой поток последовательно выполняемых работ (процедур, функций), расположенных в порядке их выполнения. Используемые при построении модели символы логики позволяют отразить ветвление и слияние бизнес-процесса [14]. Элементы, используемые в данной нотации, приведены в табл.3.2.

Таблица 3.2 – Графические элементы ARIS eEPC (2-8 уровень)

Наименование	Описание	Графическое представление
Процесс	Объект "Процесс" служит для описания функций (процедур, работ), выполняемых подразделениями/сотрудниками предприятия	
Событие	Объект "Событие" служит для описания реальных состояний системы, влияющих и управляющих выполнением функции	
Организационная единица	Объект, отражающий реальные различные организационные звенья предприятия (например, управление или отдел)	

Наименование	Описание	Графическое представление
Документ	Объект, отражающий реальные носители информации, например, бумажный документ	
Прикладная система	Объект отражает реальную прикладную систему, используемую в рамках технологии выполнения функции	
Стрелка связи между объектами	Объект описывает тип отношений между другими объектами (например, активацию выполнения функции некоторым событием)	
Логическое «И»	Оператор ставится в случае, когда начало/завершение выполнения функции должно инициировать одновременно несколько событий	
Логическое «ИЛИ»	Оператор ставится, когда начало/завершение выполнения функции может инициировать несколько событий	
Логическое исключаящее «ИЛИ»	Оператор ставится, когда начало/завершение выполнения функции может инициировать только одно из событий в зависимости от условия	

3.2. Описание ключевых бизнес-процессов

При рассмотрении первого уровня описания процессов (рис.3.1) в больнице можно выделить 3 ключевых бизнес-процесса: поступление пациента в больницу, лечение пациента, а также выписка после завершения лечения [13]. При поступлении пациента в регистратуре принимаются общие данные о поступившем больном и его анамнез. Далее пациент поступает на лечение, в

процессе которого собираются данные о лечении, заносятся в дневник пациента, температурный лист и лист врачебных назначений, после чего составляется эпикриз и пациента выписывают из больницы.

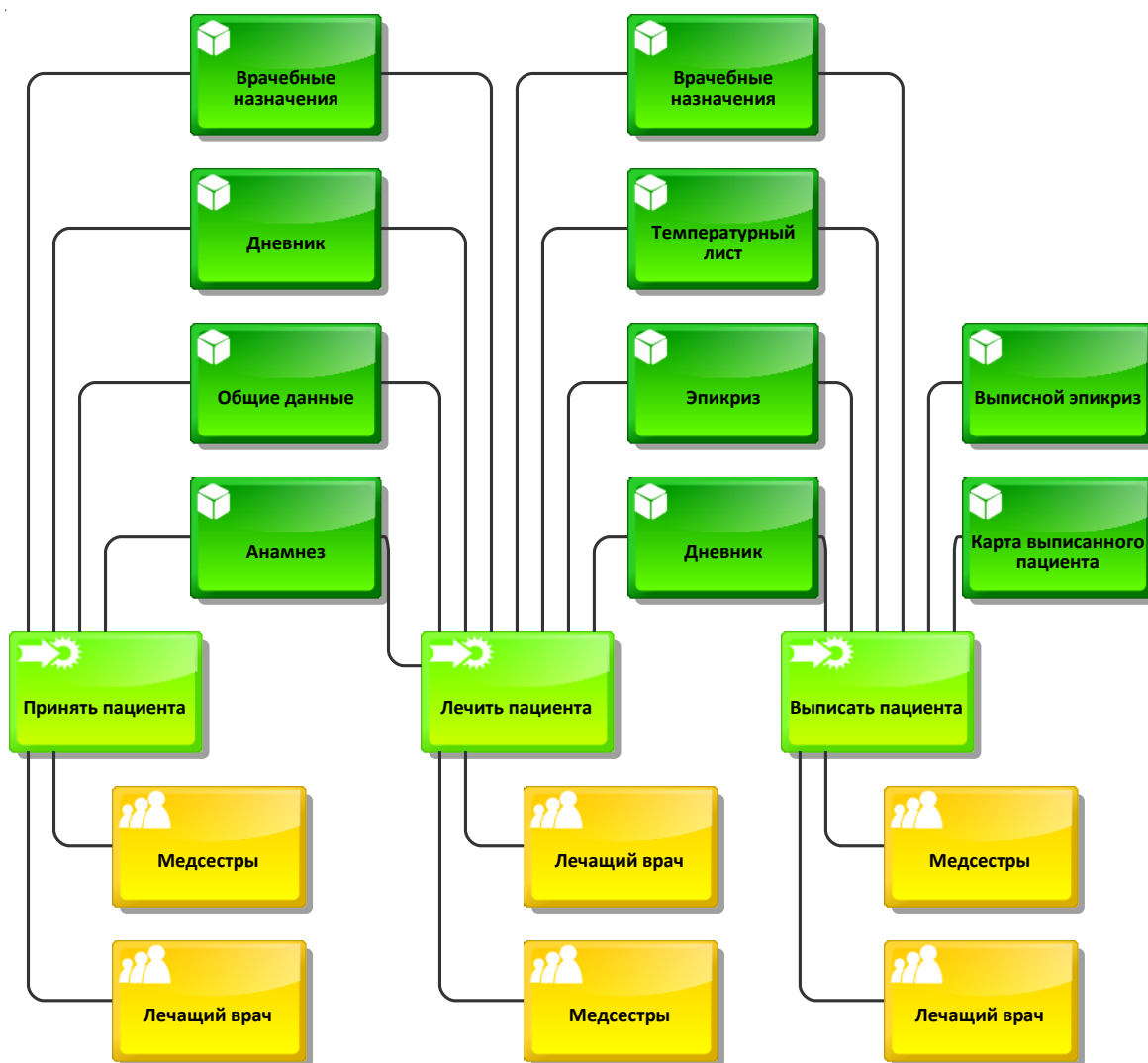


Рисунок 3.1 – Первый уровень описания процессов ARIS VACD в модели «AS-IS»

3.2.1. Описание процессов в модели «AS-IS»

Для того, чтобы понять, какие именно функции возьмет на себя разрабатываемая система и какие изменения в работе городской больницы произойдут после этого, необходимо рассмотреть каждый из основных бизнес-процессов более подробно.

3.2.1.1. Бизнес-процесс «Принять пациента»

Начнем рассмотрение с основополагающего процесса – «Принять пациента», который состоит из 5 основных этапов [17]:

- прием и регистрация пациентов;
- осмотр, первичное обследование пациентов, диагностика;
- санитарно-гигиеническая обработка вновь поступивших пациентов;
- оказание квалифицированной медицинской помощи;
- транспортировка пациентов в лечебные отделения больницы.

Вид второго уровня процесса «Принять пациента» в модели «AS-IS» представлен в Приложении А на рис.1-2, третьего – на рис.3-4.

3.2.1.2. Бизнес-процесс «Лечить пациента»

Отправлением пациента в лечебное отделение больницы заканчивается первый бизнес-процесс и начинается второй – «Лечить пациента», который включает в себя [18]:

- предоставление койки в лечебном отделении;
- курирование лечащего врача;
- консультирование;
- диагностика;
- лечение.

Вид второго уровня процесса «Лечить пациента» в модели «AS-IS» представлен в Приложении А на рис.5-6, третьего – на рис.7-8.

3.2.1.3. Бизнес-процесс «Выписать пациента»

После окончания лечения пациента начинается последний бизнес-процесс – «Выписать пациента», который включает в себя следующее:

- оформление выписного эпикриза;
- беседа с пациентом;
- подписание выписного эпикриза;

- оформление карты выписанного пациента.

Вид второго уровня процесса «Выписать пациента» в модели «AS-IS» представлен в Приложении А (рис.9). Как можно заметить, данный процесс является достаточно простым с точки зрения его организации, то есть он, по сути, состоит исключительно из работы с документами, связанными с выпиской. По причине несложности внутренних процессов и наглядности происходящих изменений уже на втором уровне не имеет смысла дальнейшее углубление в структуру бизнес-процесса «Выписать пациента».

3.2.2. Описание процессов в модели «ТО-BE»

На рис.3.2 представлены изменения, которые внесет разрабатываемая система в ключевые бизнес-процессы городской больницы на 1-м уровне их описания. Далее проведем аналогичное углубление в структуру каждого из ключевых бизнес-процессов, но в модели «ТО-BE» для того, чтобы показать какие изменения привнесет в работу создаваемая система. Их вид описан в Приложении А на рис.10-17.

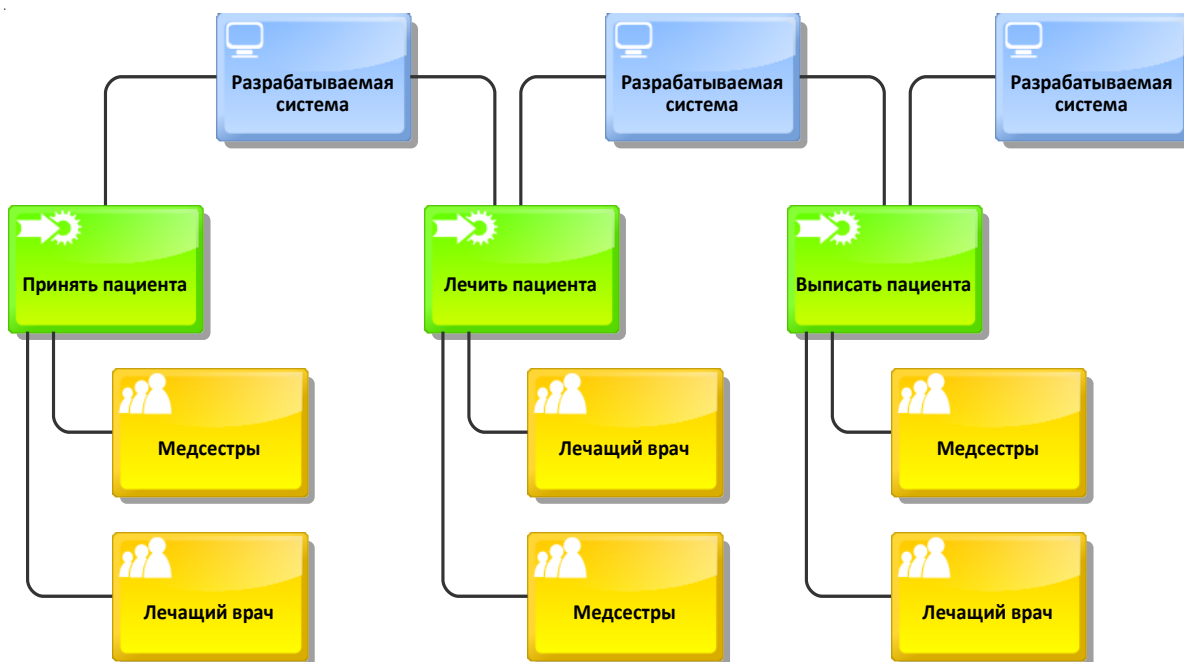


Рисунок 3.2 – Первый уровень описания процессов ARIS VACD в модели «ТО-BE»

3.3. Архитектура данных разрабатываемой системы

На основе порядка использования данных, приведённых выше, всю информацию можно разбить на классы (приведены в табл.3.3).

Таблица 3.3 – Классы данных

Название класса	Атрибуты
Пациенты	🔑 Номер карты пациента
	Фамилия
	Имя
	Отчество
	Пол
	Возраст
	Место жительства
	Место работы, профессия или должность
Персонал	🔑 Идентификатор сотрудника
	Фамилия
	Имя
	Отчество
	Должность
Анамнез	🔑 Номер карты пациента
	🔑 Идентификатор сотрудника
	Анамнез пациента
Эпикриз	🔑 Номер карты пациента
	🔑 Идентификатор сотрудника
	Эпикриз пациента
Дневник	🔑 Номер карты пациента
	🔑 Идентификатор сотрудника
	Дневник пациента
Температура	🔑 Номер карты пациента
	🔑 Идентификатор сотрудника
	Дата
	Время
	Дыхание
	Температура
Назначения	🔑 Номер карты пациента

Название класса	Атрибуты
	Идентификатор сотрудника
	Дата
	Назначение

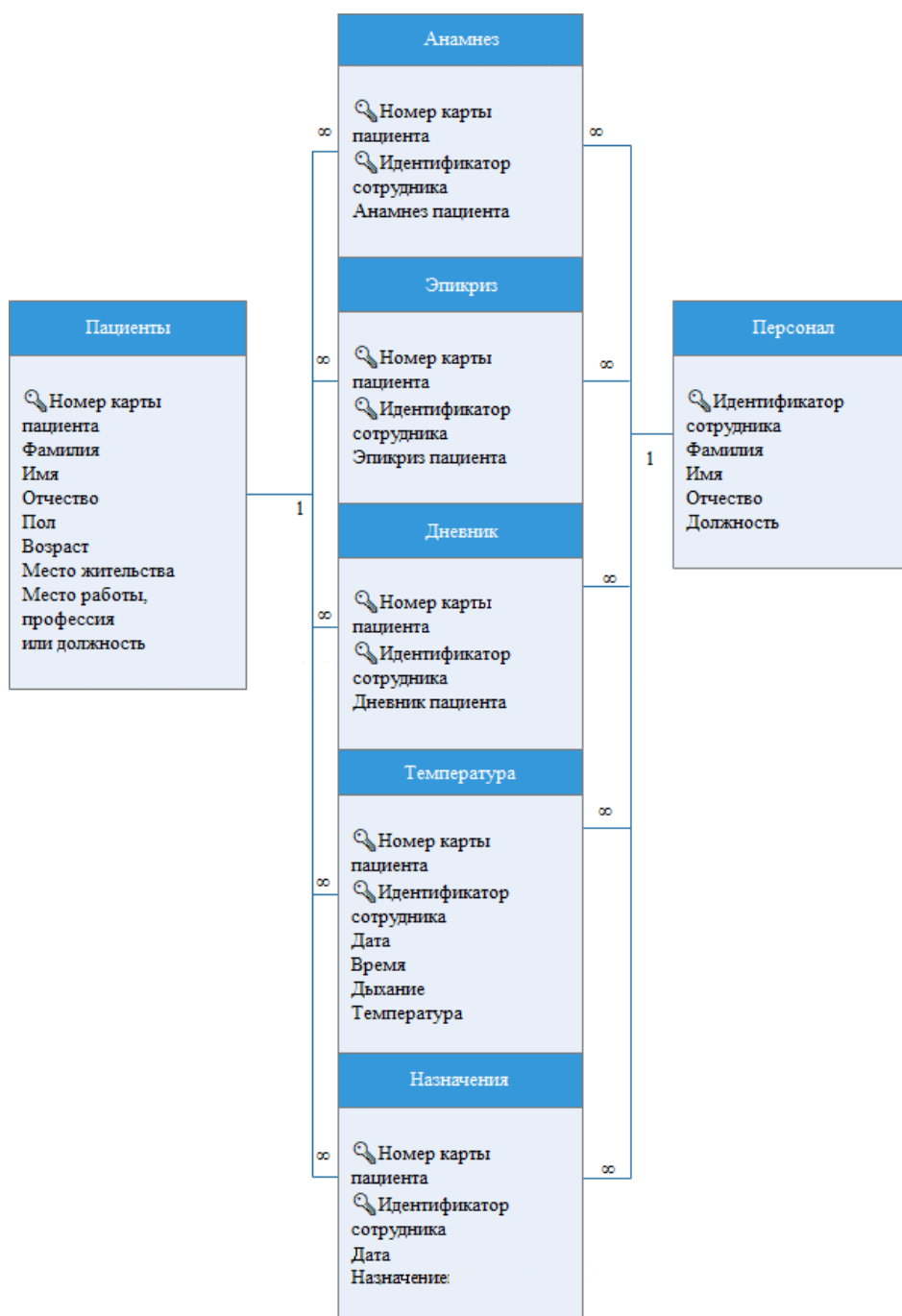


Рисунок 3.3 – Архитектура данных разрабатываемой систем

3.4. Описание разрабатываемой системы

Для удобства использования системы как пациентами, так и медицинским персоналом необходимо разработать доступный для понимания новыми пользователями интерфейс. Для создания схем воспользуемся программой MS Visio. На рис.3.4 приведена схема главной страницы разрабатываемого сайта городской больницы.

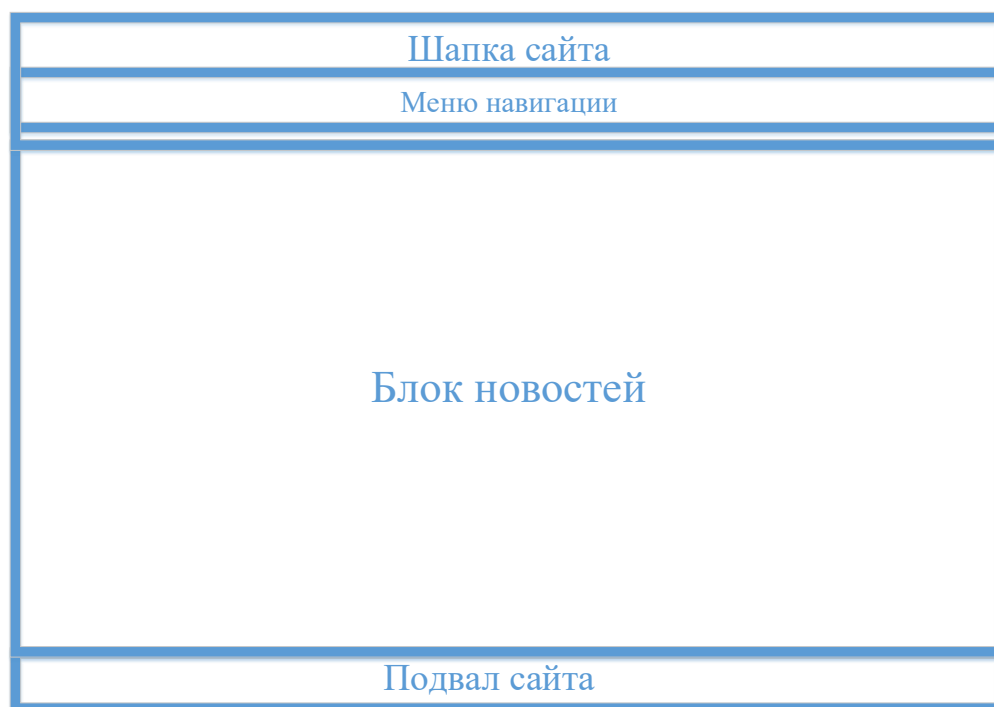


Рисунок 3.4 – Схема главной страницы разрабатываемого сайта

Остальные страницы сайта должны иметь структуру, схожую с приведенной выше, за исключением того, что блок новостей будет заменен на блок вывода информации, принадлежащей к конкретной ссылке. Примерный вид формы регистрации сотрудника представлен на рис.3.5, а на рис.3.6 представлен вид формы авторизации персонала.

Форма регистрации

Имя

Фамилия

Отчество

Должность

E-mail

Пароль

Рисунок 3.5 – Схема формы регистрации сотрудника

Форма авторизации

E-mail

Пароль

Введите проверочный код

Рисунок 3.6 – Схема формы авторизации персонала

Раздел 4. Разработка и тестирование ключевых бизнес-процессов с использованием PHP и MySQL

4.1. Принципы разработки программ

Программы различаются по назначению, выполняемым функциям, формам реализации. При создании и развитии ПО рекомендуется применять следующие общесистемные принципы [20, 21]:

- принцип включения, который предусматривает, что требования к созданию, функционированию и развитию ПО определяются со стороны более сложной, включающей его в свой состав системы;
- принцип системного единства, который состоит в том, что на всех стадиях создания, функционирования и развития ПО его целостность будет обеспечиваться связями между подсистемами, а также функционированием подсистемы управления;
- принцип развития, который предусматривает в ПО возможность его наращивания и совершенствования компонентов и связей между ними;
- принцип комплексности, который заключается в том, что ПО обеспечивает связность обработки информации как отдельных элементов, так и для всего объема данных в целом на всех стадиях обработки;
- принцип информационного единства, т. е. во всех подсистемах, средствах обеспечения и компонентах ПО используются единые термины, символы, условные обозначения и способы представления;
- принцип совместимости, который состоит в том, что язык, символы, коды и средства обеспечения ПО согласованы, обеспечивают совместное функционирование всех его подсистем и сохраняют открытой структуру системы в целом;

- принцип инвариантности, который предопределяет, что подсистемы и компоненты ПО инвариантны к обрабатываемой информации, т. е. являются универсальными или типовыми.

4.2. Реализация основных функций системы

Для выполнения поставленных задач воспользуемся:

- пакетом denwer – для локального хранения разрабатываемого сайта, работы с базами данных MySQL;
- редактором Notepad++ – для написания кода страниц сайта.

В первую очередь, необходимо зайти на ресурс <http://localhost>, и во вкладке phpmyadmin производится создание новой БД под названием registration (рис. 4.1).

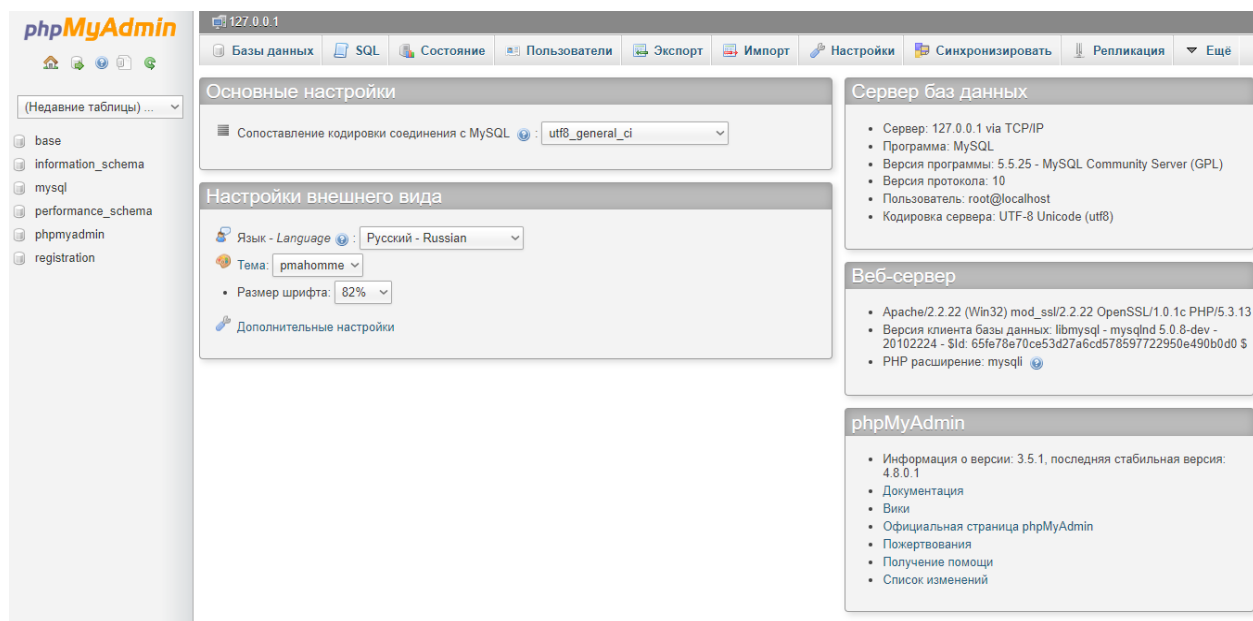


Рисунок 4.1 – Главная страница *phpmyadmin*

Внутри данной базы данных создаем 7 таблиц (Пациенты, Персонал, Эпикриз, Анамнез, Назначения, Температурный лист, Дневник) с полями, соответствующими ранее заявленным, и создаем связи между ключевыми полями. В итоге получаем базу данных, представленную на рис.4.2.

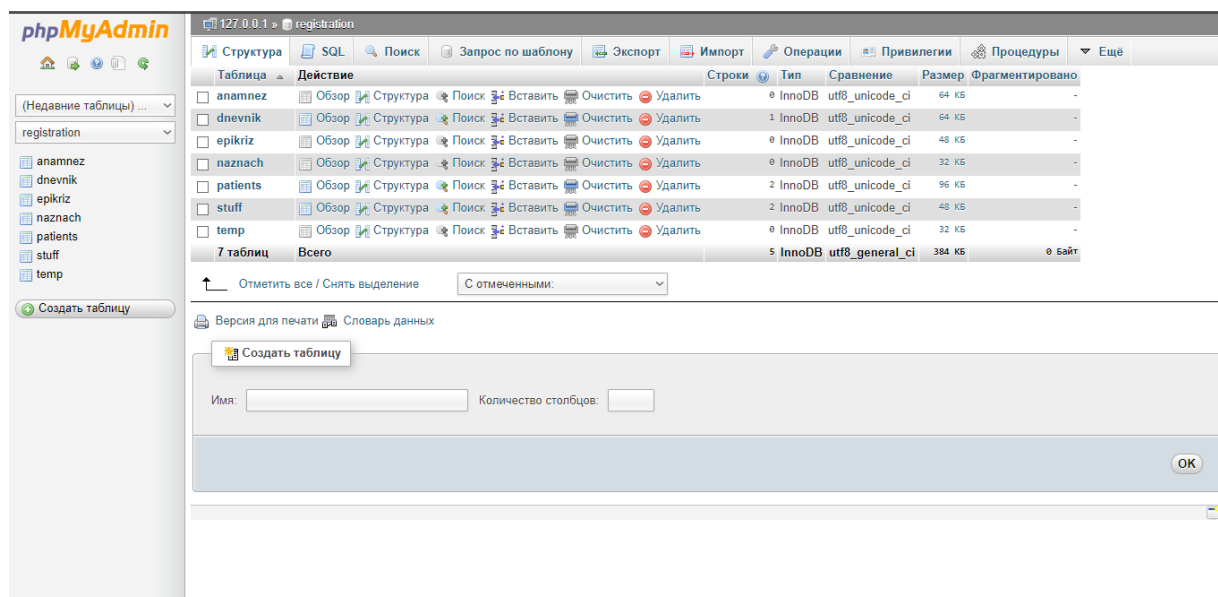


Рисунок 4.2 – Итоговый вид базы данных

Далее следует разработка сайта в соответствии с ранее заявленными требованиями в php и html (см. Приложение Б) [22-30] и просмотр результатов работы в веб-браузере. Вид главной страницы приведен на рис.4.3.

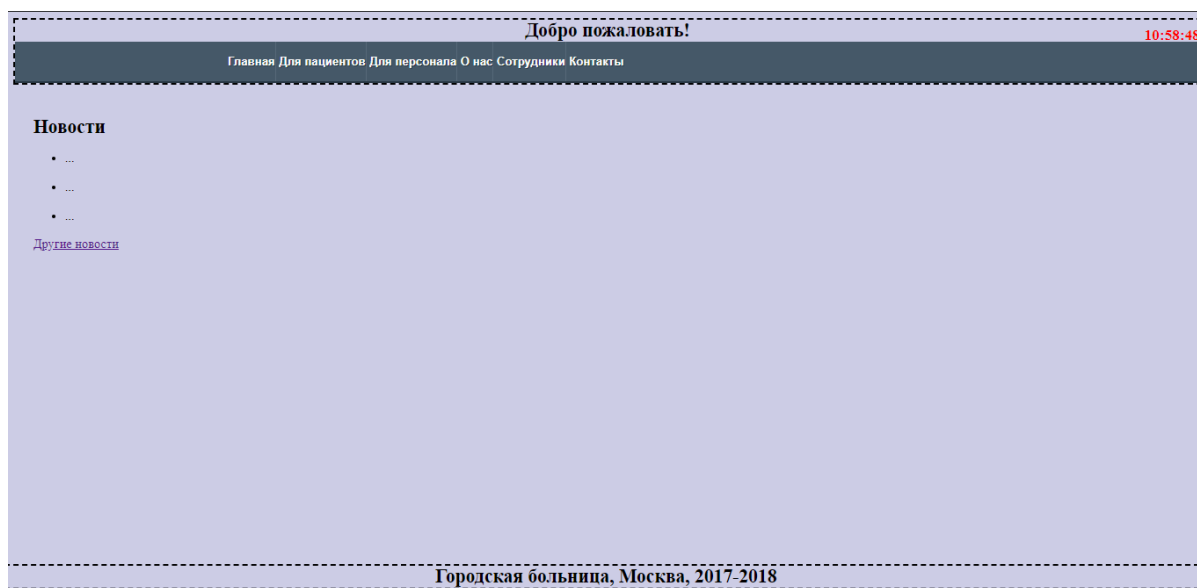


Рисунок 4.3 – Главная страница разрабатываемого сайта

На ней располагается блок навигации (в шапке), а также в основной части находится блок для новостей. Кнопка «Главная» возвращает на главную страницу из любой части сайта. После нажатия «Для пациентов» пользователей направляет на форму поиска медицинской карты (рис.4.4), результат поиска представлен на рис.4.5.

Ищем по:
Номеру карты
Фамилии
Что ищем:
Поиск

Рисунок 4.4 – Форма поиска медицинской карты пациента

Номер карты 05
Фамилия: Иванов
Имя: Игорь
Отчество: Петрович
Пол: Мужской

Возраст: 13

ПМЖ: Россия

Место работы: школа

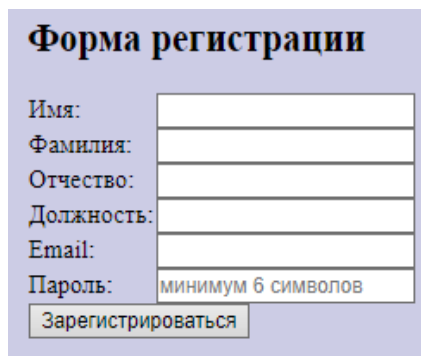
Рисунок 4.5 – Результат поиска

«Для персонала» отправляет пользователей на форму авторизации персонала (рис.4.6).

Форма авторизации
Email:
Пароль: минимум 6 символов
Введите проверочный код: 4043
Проверочный код
Войти
Еще не зарегистрированы?

Рисунок 4.6 – Форма авторизации персонала

Кнопка «Еще не зарегистрированы?» переводит сотрудника на форму регистрации персонала (рис.4.7).

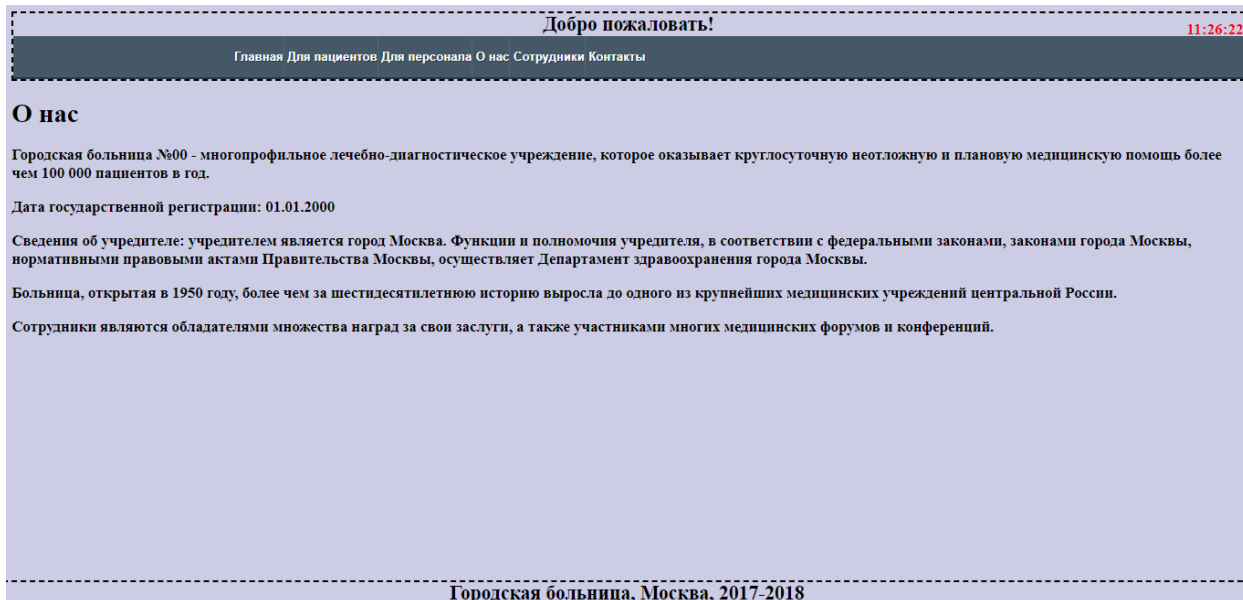


Форма регистрации

Имя:	<input type="text"/>
Фамилия:	<input type="text"/>
Отчество:	<input type="text"/>
Должность:	<input type="text"/>
Email:	<input type="text"/>
Пароль:	<input type="password"/> минимум 6 символов
<input type="button" value="Зарегистрироваться"/>	

Рисунок 4.7 – Форма регистрации персонала

Вкладка «О нас» в шапке сайта переводит пользователя на страницу, посвященную информации о данном учреждении (рис.4.8).



Добро пожаловать! 11:26:22

Главная Для пациентов Для персонала О нас Сотрудники Контакты

О нас

Городская больница №00 - многопрофильное лечебно-диагностическое учреждение, которое оказывает круглосуточную неотложную и плановую медицинскую помощь более чем 100 000 пациентов в год.

Дата государственной регистрации: 01.01.2000

Сведения об учредителе: учредителем является город Москва. Функции и полномочия учредителя, в соответствии с федеральными законами, законами города Москвы, нормативными правовыми актами Правительства Москвы, осуществляет Департамент здравоохранения города Москвы.

Больница, открытая в 1950 году, более чем за шестидесятилетнюю историю выросла до одного из крупнейших медицинских учреждений центральной России.

Сотрудники являются обладателями множества наград за свои заслуги, а также участниками многих медицинских форумов и конференций.

Городская больница, Москва, 2017-2018

Рисунок 4.8 – Вкладка «О нас»

Вкладка «Сотрудники» демонстрирует пользователям таблицу, содержащую информацию о сотрудниках больницы (рис.4.9).

Фамилия	Имя	Отчество	Должность
Иван	Иванов	Иванович	Терапевт
Наталья	Катасонова	Сергеевна	медсестра

Рисунок 4.9 – Информация о сотрудниках больницы

Вкладка «Контакты» (рис.4.10) хранит контактную информацию об учреждении (телефоны, факс, e-mail, физический адрес), схему проезда, время работы и тэги, облегчающие поиск больницы.

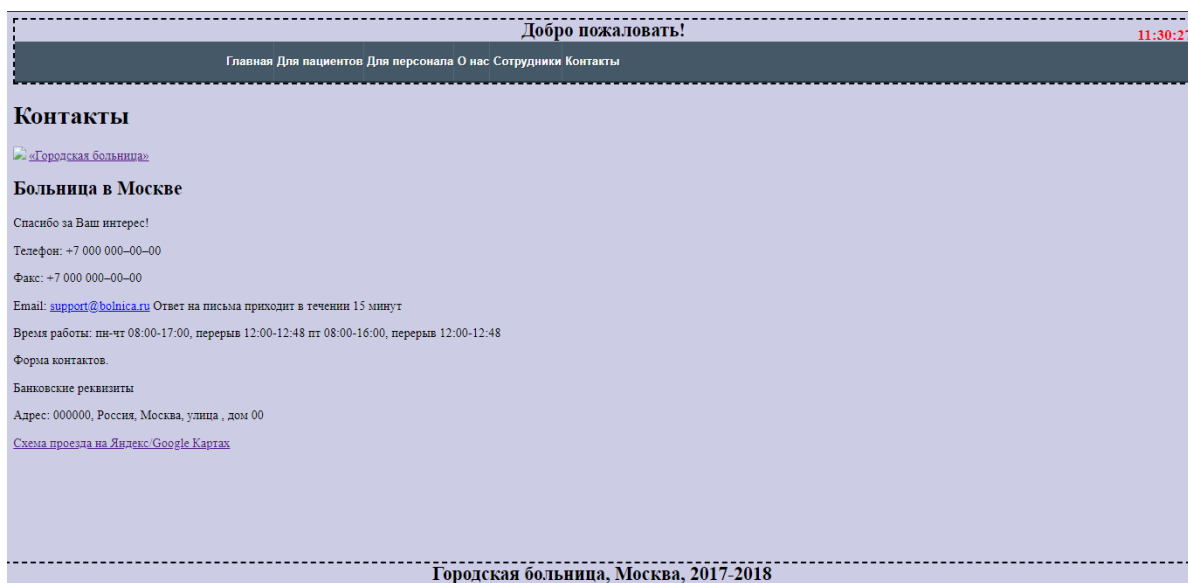


Рисунок 4.10 – Вкладка «Контакты»

После прохождения авторизации персонала, главной страница принимает вид, представленный на рис.4.11.

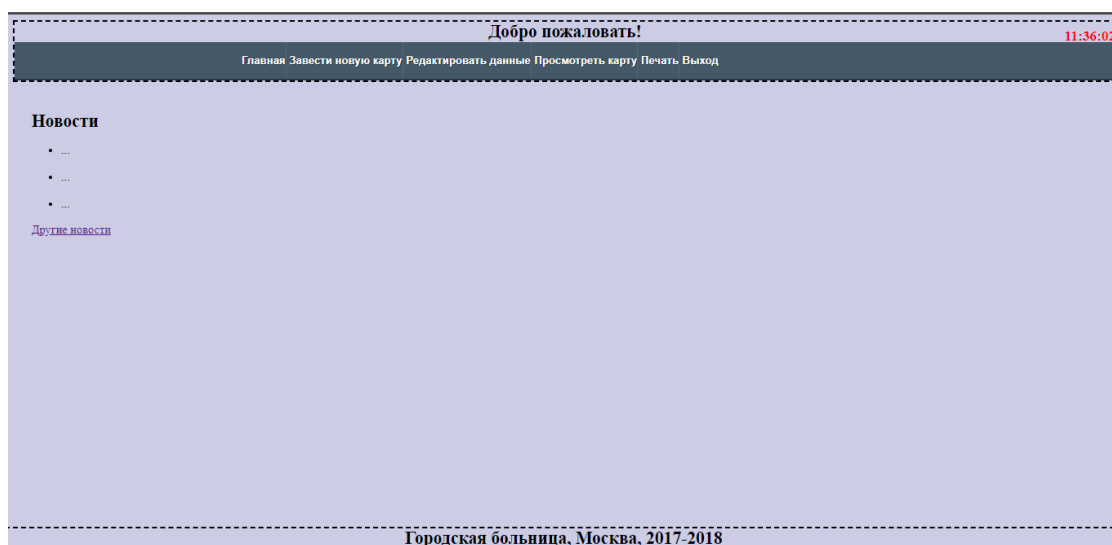


Рисунок 4.11 – Вид главной страницы после авторизации

Вкладка «Главная» выполняет все те же функции. «Завести новую карту» позволяет сотруднику перейти к форме регистрации карты пациента (рис.4.12).

Рисунок 4.12 – Форма регистрации пациента

При наведении на «Редактировать данные» появляется выпадающий список с предложенными для изменения таблицами (рис.4.13). При переходе на каждую из них предлагается форма для изменения данных (для таблицы «Дневник» – рис.4.14). Оставшиеся формы устроены аналогично.

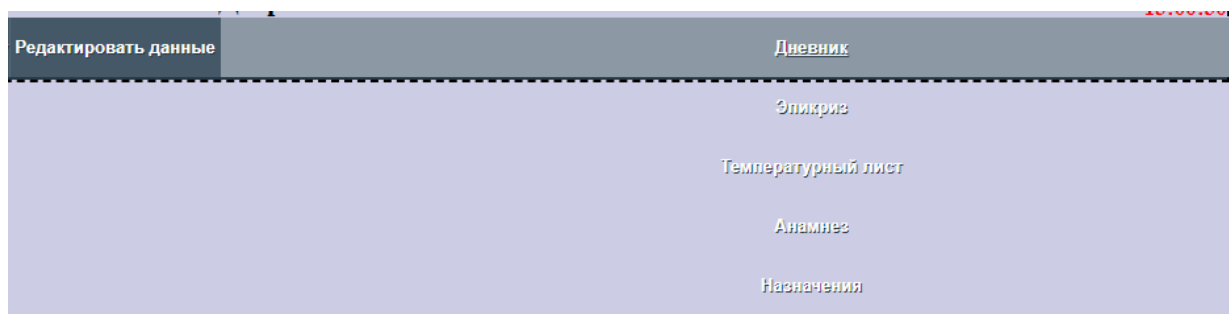


Рисунок 4.13 – Список таблиц для изменения

Номер карты	Идентификатор сотрудника	Записи	
4	1	11223344	Удалить Редактировать

Номер карты:
 Идентификатор сотрудника:
 Запись:

Рисунок 4.14 – Форма изменения данных таблицы «Дневник»

Вкладка «Просмотр карты» выполняет функции, аналогичные «Для пациентов» и направляет на форму, аналогичную представленной на рис.4.4, но при поиске выполняется проверка авторизации пользователя и на экран выдается информация, связанная с пациентом, из всех таблиц (рис.4.15).

Номер карты 04 Фамилия: Сидоров Имя: Петр Отчество: Федорович Пол: Мужской Возраст: 63 ПМЖ: Москва Место работы: Работа Анамнез Записи: Поступил больным Дневник Записи: 11223344	Эпикриз Записи: Выздоровливает Назначения Дата: 2018-04-11 Назначения: Анальгин Температурный лист Дата: 2018-04-11 Время: 14:00:05 Дыхание: 60 Температура: 37
---	---

Рисунок 4.15 – Результат поиска

И для печати карты при выписке имеется кнопка «Печать», которая выводит пользователя на форму поиска карты по номеру (рис.4.16), затем выводит выборку из нескольких таблиц по нему (рис.4.17), а после нажатия кнопки «Распечатать» выводит данные с экрана на печать (рис.4.18). Кнопка «Выход» позволяет персоналу выйти из рабочих аккаунтов.

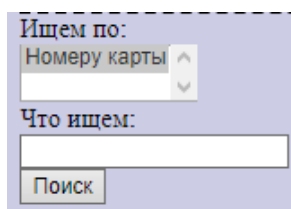


Рисунок 4.16 – Форма поиска данных из всех таблиц

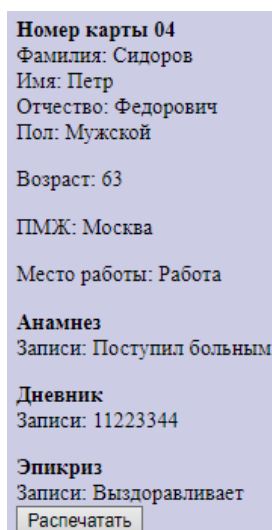


Рисунок 4.17 – Результаты поиска

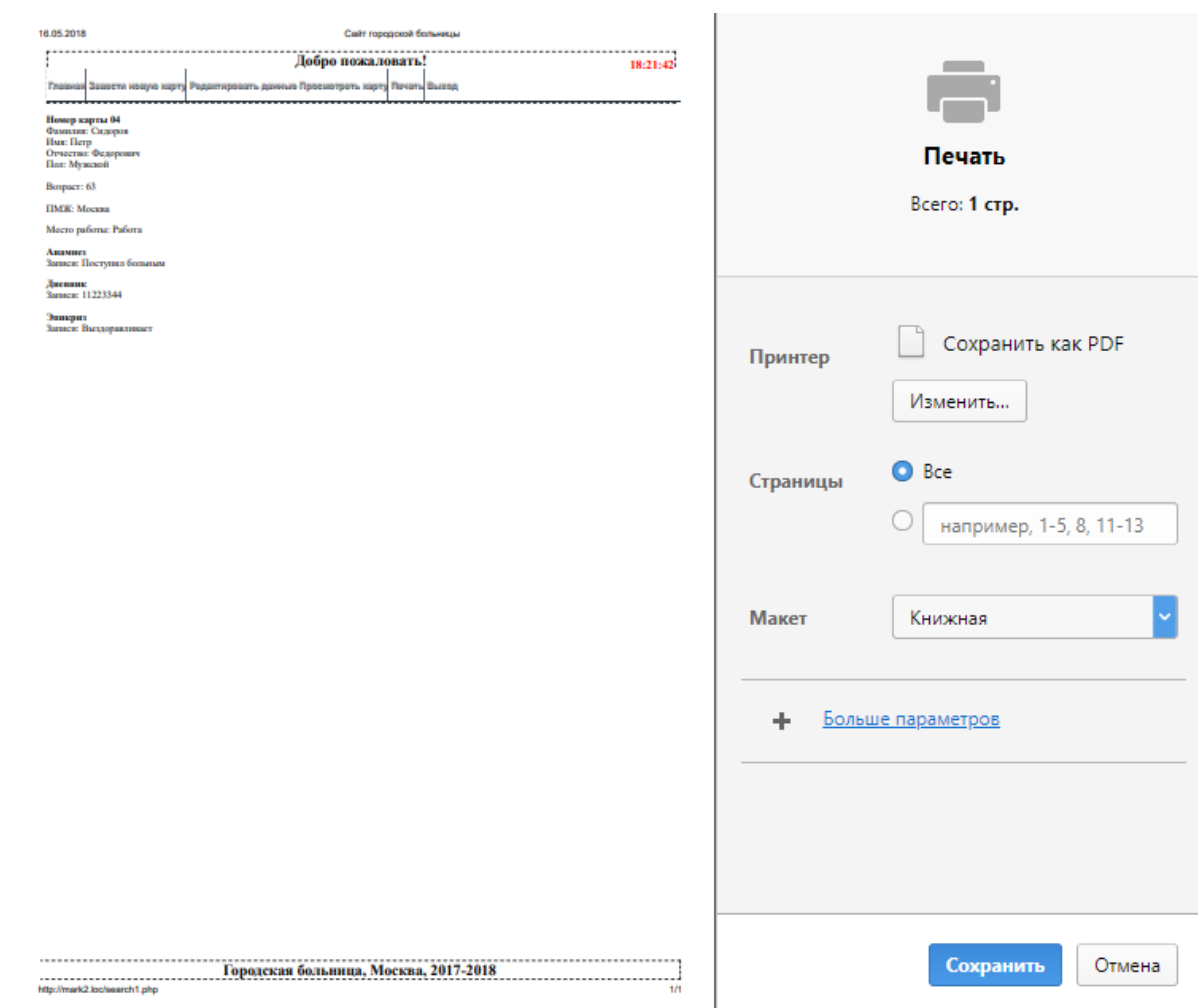


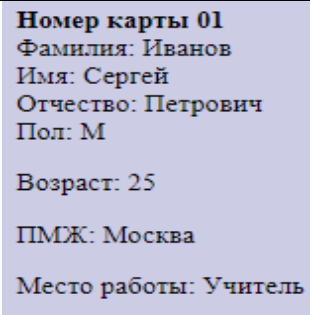
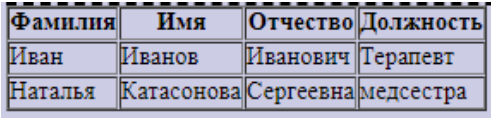
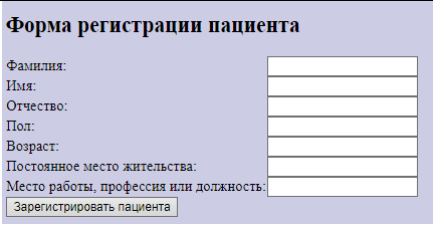
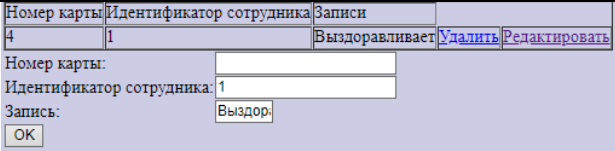
Рисунок 4.18 – Результат, выводимый на печать

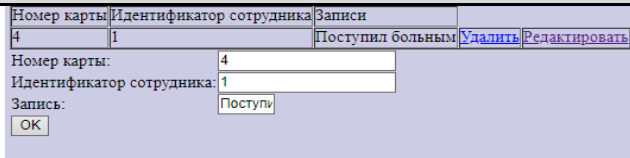
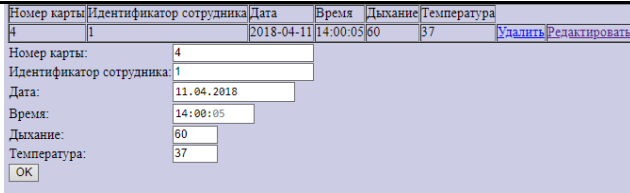
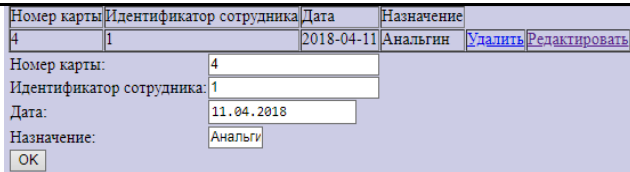
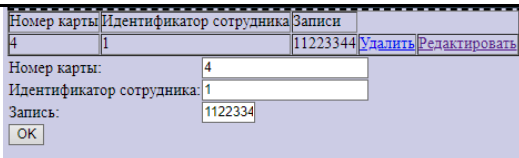
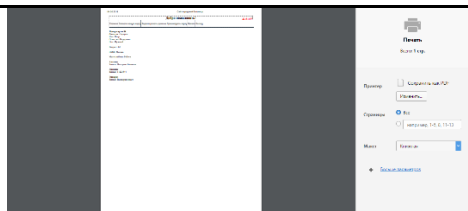
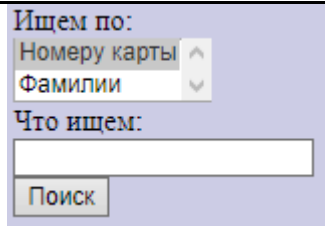
4.3. Функциональное и системное тестирование

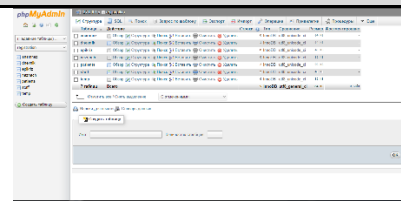
Согласно заявленной модели проведем первое из возможных тестирований системы. Функциональное тестирование является одним из ключевых видов тестирования, задача которого – установить соответствие разработанного программного обеспечения (ПО) исходным функциональным требованиям заказчика. То есть, проведение функционального тестирования позволяет проверить способность информационной системы в определенных условиях решать задачи, нужные пользователям.

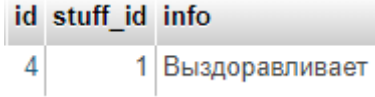
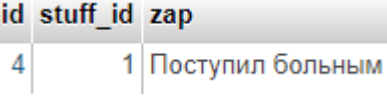
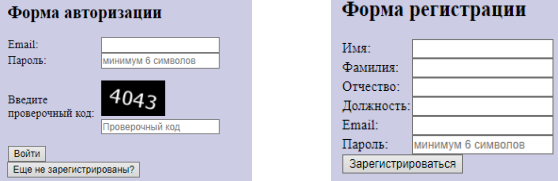
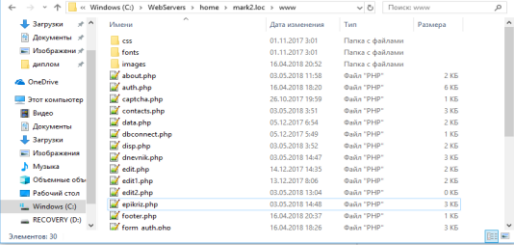
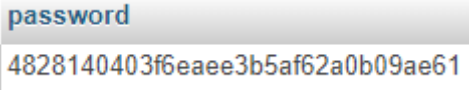
Для проведения тестирования необходимы данные из матрицы отслеживания требований, а также результаты использования системы, которые для наглядности будут занесены в табл.4.1.

Таблица 4.1 – Функциональное тестирование системы

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)
Вывод информации из базы данных с соответствующей информацией	Форма запроса по базе данных «Пациенты»	
Страница, посвященная мед. персоналу	Страница, посвященная мед. персоналу	
Добавление данных о пациенте	Форма для регистрации новых пациентов	
Удаление медицинской карты пациента	Форма удаления данных о пациентах	-
Просмотр и редактирование данных из БД «Пациенты»	Формы просмотра и редактирования данных	-
Просмотр и редактирование данных из БД «Эпикриз»	Формы просмотра и редактирования данных	

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)
Просмотр и редактирование данных из БД «Анамнез»	Формы просмотра и редактирования данных	
Просмотр и редактирование данных из БД «Температура»	Формы просмотра и редактирования данных	
Просмотр и редактирование данных из БД «Назначения»	Формы просмотра и редактирования данных	
Просмотр и редактирование данных из БД «Дневник»	Формы просмотра и редактирования данных	
Просмотр и редактирование данных из БД «Персонал»	Формы просмотра и редактирования данных	-
Печать карты пациента при его выписке	Форма вызова данных из баз на печать	
Поиск медицинской карты пациента	Форма поиска карты	
Корректная работа на любом устройстве	Разрабатываемая система	-

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)																																
Хранение данных	Разрабатываемая система																																	
Доступность работникам и пользователям	Разрабатываемая система	-																																
Простота разрабатываемого интерфейса	Разрабатываемая система	-																																
Ведение температурных листов пациентов	База данных «Температура»	<table><tr><th>id</th><th>stuff_id</th><th>date</th><th>time</th><th>dyx</th><th>temp</th></tr><tr><td>4</td><td>1</td><td>2018-04-11</td><td>14:00:05</td><td>60</td><td>37</td></tr></table>	id	stuff_id	date	time	dyx	temp	4	1	2018-04-11	14:00:05	60	37																				
id	stuff_id	date	time	dyx	temp																													
4	1	2018-04-11	14:00:05	60	37																													
Ведение базы учета лечебных назначений	База данных «Назначения»	<table><tr><th>id</th><th>stuff_id</th><th>date</th><th>naznach</th></tr><tr><td>4</td><td>1</td><td>2018-04-11</td><td>Анальгин</td></tr></table>	id	stuff_id	date	naznach	4	1	2018-04-11	Анальгин																								
id	stuff_id	date	naznach																															
4	1	2018-04-11	Анальгин																															
Ведение «Дневника» пациента, полученную путем последующего добавления сведений	База данных «Дневник»	<table><tr><th>id</th><th>stuff_id</th><th>zap</th></tr><tr><td>4</td><td>1</td><td>11223344</td></tr></table>	id	stuff_id	zap	4	1	11223344																										
id	stuff_id	zap																																
4	1	11223344																																
Ведение базы данных «Персонал», содержащей информацию о персонале, полученную путём регистрации персонала	База данных «Персонал»	<table><tr><th>stuff_id</th><th>first_name</th><th>last_name</th><th>name</th><th>dolg</th><th>email</th><th>password</th></tr><tr><td>1</td><td>Иван</td><td>Иванов</td><td>Иванович</td><td>Терапевт</td><td>123@mail.ru</td><td>4828140403f6eae3b5af62a0b09ae61</td></tr><tr><td>2</td><td>Наталья</td><td>Катасонова</td><td>Сергеевна</td><td>медсестра</td><td>1234@mail.ru</td><td>4828140403f6eae3b5af62a0b09ae61</td></tr></table>	stuff_id	first_name	last_name	name	dolg	email	password	1	Иван	Иванов	Иванович	Терапевт	123@mail.ru	4828140403f6eae3b5af62a0b09ae61	2	Наталья	Катасонова	Сергеевна	медсестра	1234@mail.ru	4828140403f6eae3b5af62a0b09ae61											
stuff_id	first_name	last_name	name	dolg	email	password																												
1	Иван	Иванов	Иванович	Терапевт	123@mail.ru	4828140403f6eae3b5af62a0b09ae61																												
2	Наталья	Катасонова	Сергеевна	медсестра	1234@mail.ru	4828140403f6eae3b5af62a0b09ae61																												
Ведение базы данных «Пациенты», содержащей информацию о пациенте, полученную путём опроса или из медкарты	База данных «Пациенты»	<table><tr><th>id</th><th>first_name</th><th>last_name</th><th>name</th><th>gender</th><th>age</th><th>pmg</th><th>job</th></tr><tr><td>1</td><td>Сергей</td><td>Иванов</td><td>Петрович</td><td>М</td><td>25</td><td>Москва</td><td>Учитель</td></tr><tr><td>4</td><td>Петр</td><td>Сидоров</td><td>Федорович</td><td>Мужской</td><td>63</td><td>Москва</td><td>Работа</td></tr><tr><td>5</td><td>Игорь</td><td>Иванов</td><td>Петрович</td><td>Мужской</td><td>13</td><td>Россия</td><td>школа</td></tr></table>	id	first_name	last_name	name	gender	age	pmg	job	1	Сергей	Иванов	Петрович	М	25	Москва	Учитель	4	Петр	Сидоров	Федорович	Мужской	63	Москва	Работа	5	Игорь	Иванов	Петрович	Мужской	13	Россия	школа
id	first_name	last_name	name	gender	age	pmg	job																											
1	Сергей	Иванов	Петрович	М	25	Москва	Учитель																											
4	Петр	Сидоров	Федорович	Мужской	63	Москва	Работа																											
5	Игорь	Иванов	Петрович	Мужской	13	Россия	школа																											

Функциональное требование	Программный компонент	Графическое представление (в случае наличия)
Ведение эпикриза пациентов	База данных «Эпикриз»	
Ведение базы данных «Анамнез», содержащей информацию о сопутствующих заболеваниях, полученную путём опроса или из медкарты	База данных «Анамнез»	
Разграничение доступа между пациентом и персоналом	Форма регистрации и авторизации персонала	
Сервер для размещения сайта	Разрабатываемая система	
Мобильная версия сайта	Разрабатываемая система	-
Частичное шифрование личных данных	Разрабатываемая система	

В результате описания разрабатываемой системы (гл.3.4) и проведения функционального тестирования каждого элемента (гл.4.3) установлено, что система целиком функционирует без ошибок и соответствует всем ранее предъявленным к ней требованиям.

4.4. Нагрузочное тестирование

Нагрузочное тестирование – подвид тестирования производительности, сбор показателей и определение производительности и времени отклика

программно-технической системы или устройства в ответ на внешний запрос с целью установления соответствия требованиям, предъявляемым к данной системе (устройству). Для его проведения рассмотрим время отклика разработанной системы при работе с разным числом записей (1, 10, 25, 50, 100), занесем данные в табл.4.2 и произведем расчеты с помощью MS Excel.

В начале 5 раз проводится проверка отклика системы на различные действия, ее результаты вносятся в соответствующие строки 3-7 столбцов. Далее рассчитывается среднее арифметическое всех измерений по формуле:

$t_{\text{ср.ариф.}} = \frac{\sum_{i=1}^n t_i}{n}$, а результат заносится в 8 столбец таблицы. В 9 столбце находятся соответствующие средние квадратические отклонения, рассчитанные

по формуле: $\sigma = \sqrt{\frac{1}{n} * \sum_{i=1}^n (t_i - t_{\text{ср.ариф.}})^2}$. Погрешность измерений $\Delta t = \sqrt{\left(\frac{\sigma}{\sqrt{n}} * t_{\alpha(n-1)}\right)^2 + (A)^2}$ в 10 столбце, где n – число измерений, $t_{\alpha(n-1)}$ – доверительный коэффициент Стьюдента, равный 0.95, A – абсолютная погрешность прибора (в данном случае – электронного секундомера, равная 0.005); в 11 столбце – итоговое время отклика $t_{\text{отк}} = t_{\text{ср.ариф.}} \pm \Delta t$.

Таблица 4.2 – Результаты нагрузочного тестирования

Количество записей	Действие	t ₁ , с	t ₂ , с	t ₃ , с	t ₄ , с	t ₅ , с	Среднее время отклика, с	Средн. квадр. отклон., с	Погрешность измерений, с	Время отклика, с
1	Запись	0,11	0,13	0,12	0,09	0,1	0,1100	0,0141	0,0184	0,110±0,018
	Поиск	0,1	0,09	0,09	0,11	0,12	0,1020	0,0117	0,0154	0,102±0,015
10	Запись	0,1	0,12	0,11	0,15	0,13	0,1220	0,0172	0,0221	0,122±0,022
	Поиск	0,09	0,11	0,14	0,11	0,11	0,1120	0,0160	0,0206	0,112±0,021
25	Запись	0,15	0,16	0,15	0,14	0,17	0,1540	0,0102	0,0137	0,154±0,014
	Поиск	0,14	0,13	0,15	0,13	0,14	0,1380	0,0075	0,0106	0,138±0,011
50	Запись	0,2	0,19	0,2	0,18	0,18	0,1900	0,0089	0,0123	0,190±0,012
	Поиск	0,17	0,2	0,18	0,2	0,18	0,1860	0,0120	0,0158	0,186±0,016
100	Запись	0,3	0,27	0,29	0,29	0,28	0,2860	0,0102	0,0137	0,286±0,014
	Поиск	0,27	0,25	0,25	0,29	0,27	0,2660	0,0150	0,0194	0,266±0,019

Заключение

В рамках данной работы изучен каскадный метод внедрения ИС, его достоинства и недостатки и доказана целесообразность его применения в разработке системы для автоматизации ключевых бизнес-процессов городской больницы; произведен сбор требований с помощью опроса персонала, изучение существующей документации и анализ полученных данных, на основе которого были сформулированы пользовательские и функциональные требования к разрабатываемой системе и определен их приоритет в разработке.

На основе рассмотренных материалов в среде ARIS Express смоделированы ключевые бизнес-процессы учреждения, каждый из которых рассмотрен с углублением до 3 уровня, а затем составлены бизнес-процессы в модели «ТО-ВЕ» для того, чтобы показать, какие изменения принесет разрабатываемая система на каждом из уровней организации. По причине того, что создание системы предполагало работу с БД, проведена классификация классов данных, их нормализация и спроектирована архитектура данных.

Далее на базе собранных и проанализированных данных созданы соответствующие таблицы с использованием БД MySQL, установлены связи между соответствующими ключевыми полями, после чего начата разработка непосредственно сайта и его верстка. После его создания, продемонстрирован интерфейс и произведено его функциональное и системное тестирование, показавшее работоспособность и готовность каждого элемента в отдельности и всей системы вообще.

Таким образом, можно заключить, что все поставленные задачи выполнены. В процессе подготовки отчета для наглядности в нем приведены 6 таблиц и 41 рисунок/снимок экрана. Безусловно, у разработанной системы имеются возможности для улучшения, такие как:

- добавление возможности записи к врачу;
- добавление личного кабинета пациента;
- улучшение возможностей работы мобильной версии сайта;

- улучшение дизайна, посредством работы со стилями CSS;
- добавление дополнительных полей в таблицы для заполнения;
- ведение страницы новостей;
- разделение информации о персонале по отделениям и т.д.

Подытоживая, данная система позволила объединить в себе работы, посвященные работе с базами данных, разработке сайтов, проектированию бизнес-процессов и предыдущими вариантами их автоматизации, что позволит решать уже существующие проблемы, так и при несущественной доработке – будущие.

Список использованных литературных источников

1. Самоучитель по работе с php и MySQL – <http://www.php-s.ru/MySQL>
2. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам: учебное пособие – Москва: Интуит НОУ, 2016. – 192 с.
3. Маглинец Ю.А. Разработка информационных систем. Часть 1, Структурные методы. – Красноярск: Кларитеанум, 2004. – 120 с.
4. Шеер А.-В. Бизнес-процессы: основные понятия, теории, методы. – М.: Просветитель, 1999.
5. Шеер А.В. Моделирование бизнес-процессов – М.: Серебряные нити, 2014. – 219 с.
6. Орешкина А.М. Анализ, проектирование, разработка и тестирование приложения для автоматизации городской больницы: диплом бакалавра / МИРЭА. – М., 2017. – 49 с.
7. Модели жизненного цикла: учеб. пособие / Д.Б. Берг, Е.А. Ульянова, П.В. Добряк. – Екатеринбург: Издательство Уральского университета, 2014. – 74 с.
8. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: теория и практика // Российский технологический журнал, 2015. – т.8, №3. – URL: http://stepanovd.com/article_2015_2_erpthpr.html?lang=RU
9. Илья Корнипаев. Требования для программного обеспечения: рекомендации по сбору и документированию – М.: Издательство «Книга по требованию», 2013. – 118 с.
10. Интернет портал – <http://habrhabr.ru>

11. Степанов Д.Ю. Анализ, проектирование и разработка корпоративных информационных систем: основные термины и определения. – URL: http://stepanovd.com/training_erp_1-1_ru.pdf
12. Леффингуэлл Дин, Уидриг Дон. Принципы работы с требованиями к программному обеспечению. Унифицированный подход. М.: Вильямс, 2002. – 448 с.
13. Шматалюк А. и др. Моделирование бизнеса. Методология ARIS. Практическое руководство. – М.: Серебряные нити, 2001.
14. Шеер А.В. ARIS – моделирование бизнес-процессов / пер. с англ. и ред. Рыбьянец А.А. – 3-е изд. – М.: Вильямс, 2009.
15. Репин В. В. Бизнес-процессы компании: построение, анализ, регламентация. – М.: Стандарты и качество, 2007.
16. Учебное пособие по работе со средствами ARIS Express - <http://library.miit.ru/methodics/29.09.17/Уч-мет.ARIS.pdf>
17. Мухина С.А., Тарновская И.И. Практическое руководство к предмету «Основы сестринского дела»: учебник. – 2-е изд., исправл. и доп. – М.: ГЭОТАР-Медиа, 2013 – 512 с.
18. Гулиев Я.И., Бельшев Д.В., Михеев А.Е. Моделирование бизнес-процессов медицинской организации: классификация процессов: научная статья – (Институт программных систем им. А.К. Айламазяна РАН, Переславль-Залесский, Россия, ГБУ «Инфогород», Москва, Россия).
19. Дейт К.Дж. Введение в системы баз данных / пер. с англ. и ред. Птицына К.А. – 8-е изд. – М.: Вильямс, 2016.
20. Иан Соммервилл. Инженерия программного обеспечения. – Москва: Вильямс, 2002. – с. 642.
21. Голицына О.Л., Партыка Т.Л., Попов И.И. Языки программирования. Учебное пособие – ФОРУМ, ИНФРА-М, 2008.

22. Мишель Е. Дэвис Изучаем PHP и MySQL. – М.: Символ-плюс, 2008.
23. Линн Бейли, Майкл Моррисон Изучаем PHP и MySQL – Эксмо, 2010.
24. Робин Никсон - Создаем веб-сайты с помощью PHP, MySQL и JS – O'Reilly, 2011.
25. Ларри Ульман: Основы программирования на PHP. – М.: ДМК-Пресс, 2001.
26. Фримен Эрик, Фримен Элизабет Изучаем HTML, XHTML и CSS. – СПб.: Питер, 2012.
27. Гладкий А. - Веб-Самоделкин. Как самому создать сайт быстро и профессионально – Литрес, 2012.
28. Ташков П. А. Веб-мастеринг. HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка. – СПб.: Питер, 2010.
29. Влад Мержевич - Вёрстка веб-страниц – HTMLBOOKS, 2011.
30. Филиппов С.А. - Основы современного веб-программирования – НИЯУ МИФИ, 2011.

Приложение А

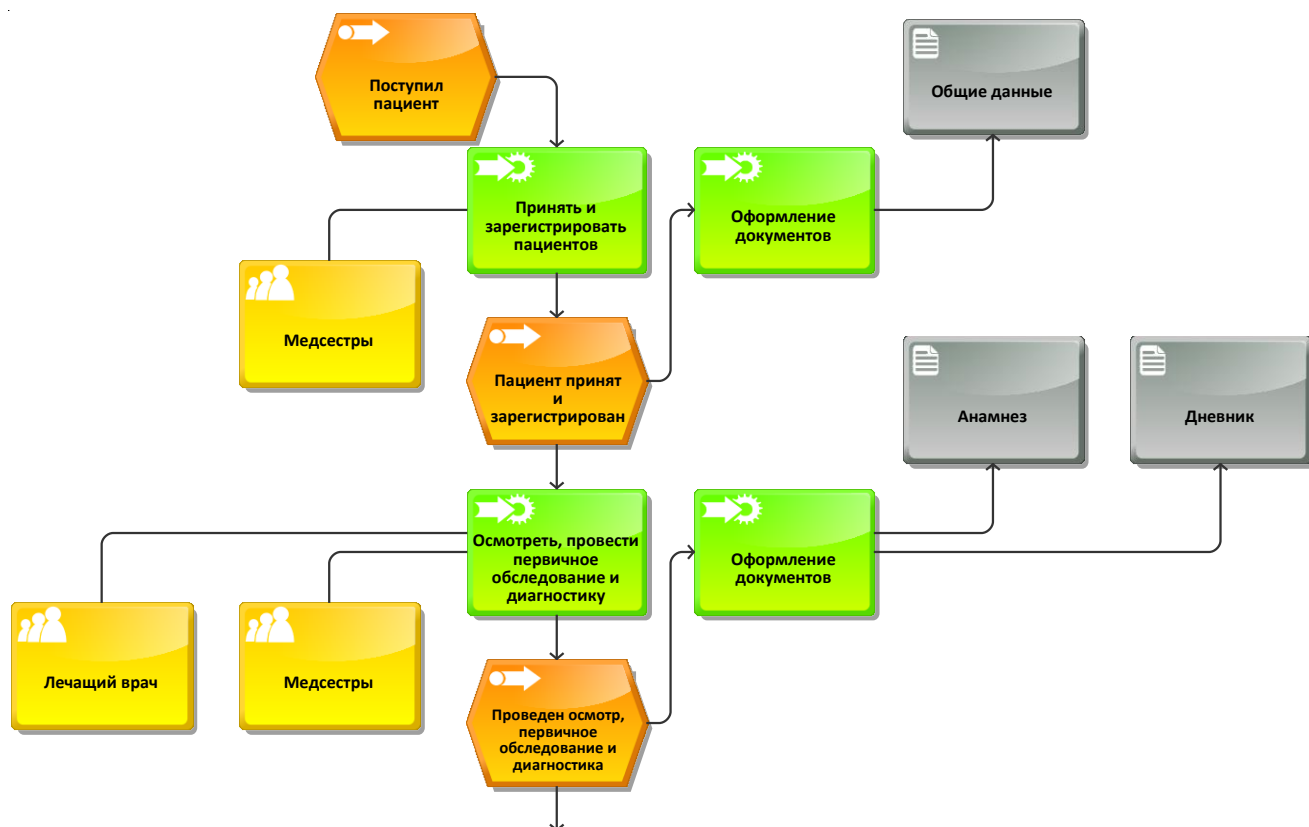


Рис.1 – Процесс «Принять пациента» в модели ARIS eEPC «AS-IS» ч.1
(2 уровень)

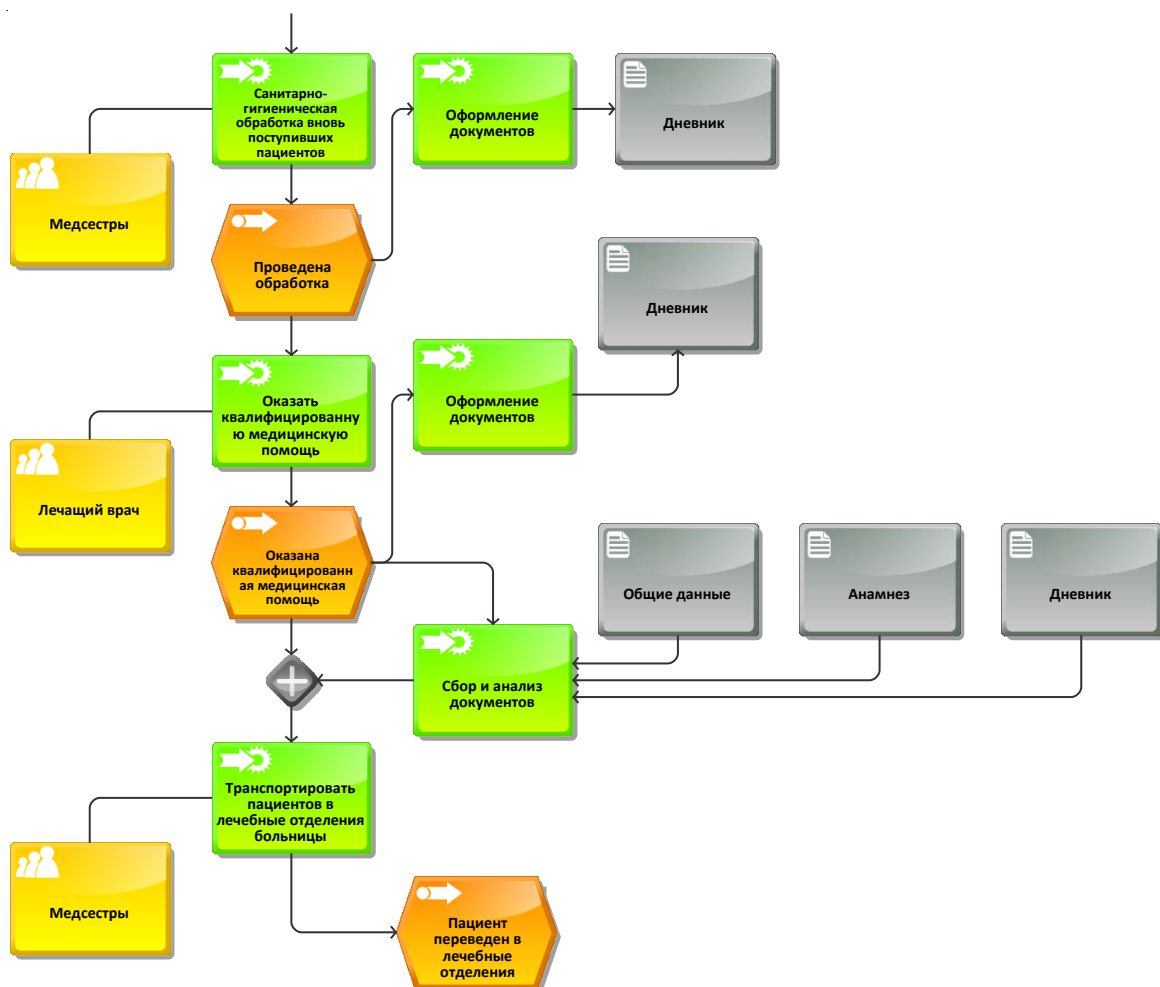


Рис.2 – Процесс «Принять пациента» в модели ARIS eEPC «AS-IS» ч.2
(2 уровень)

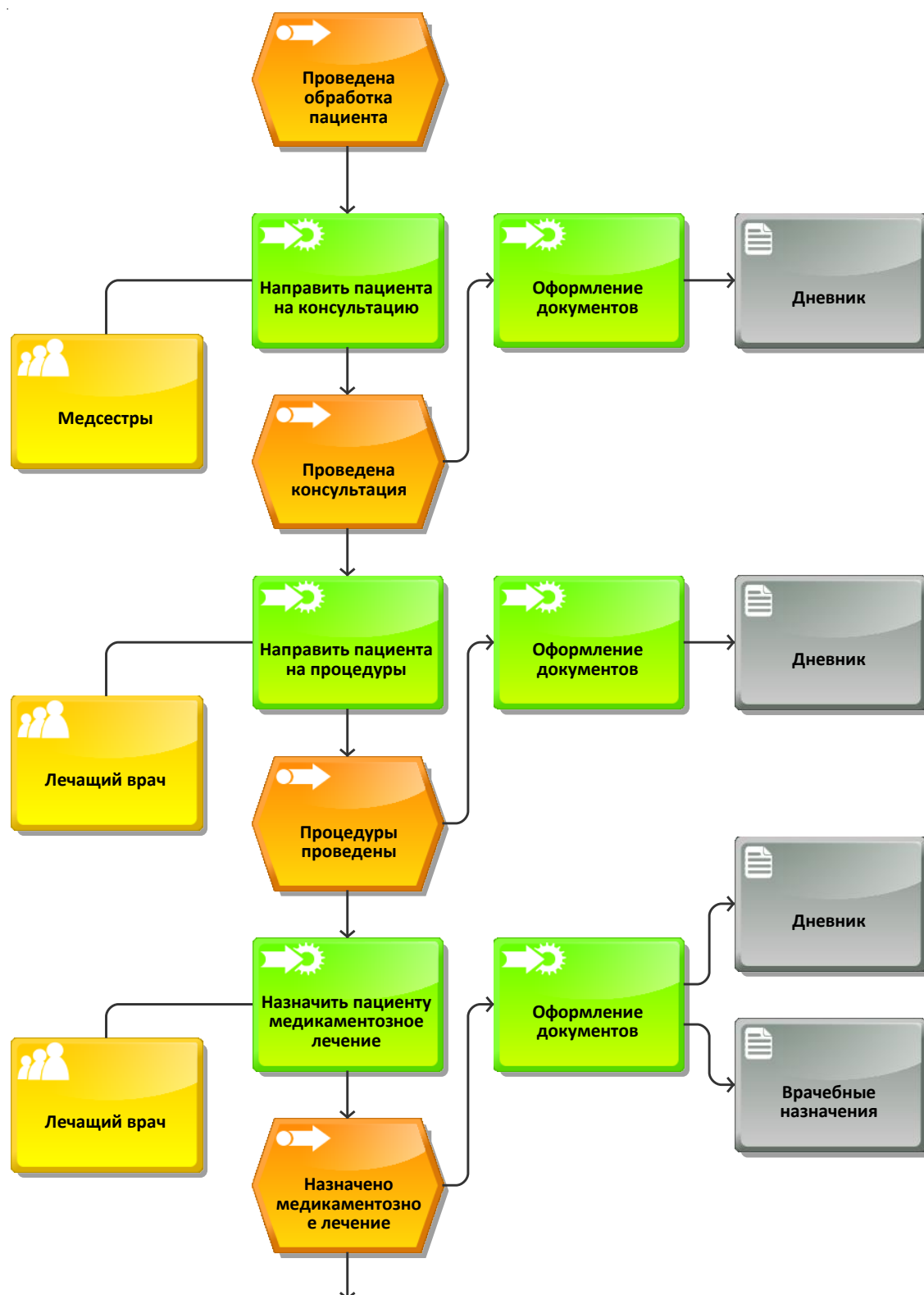


Рис.3 – Процесс «Оказать квалифицированную медицинскую помощь» в модели ARIS eEPC «AS-IS» ч.1 (3 уровень)

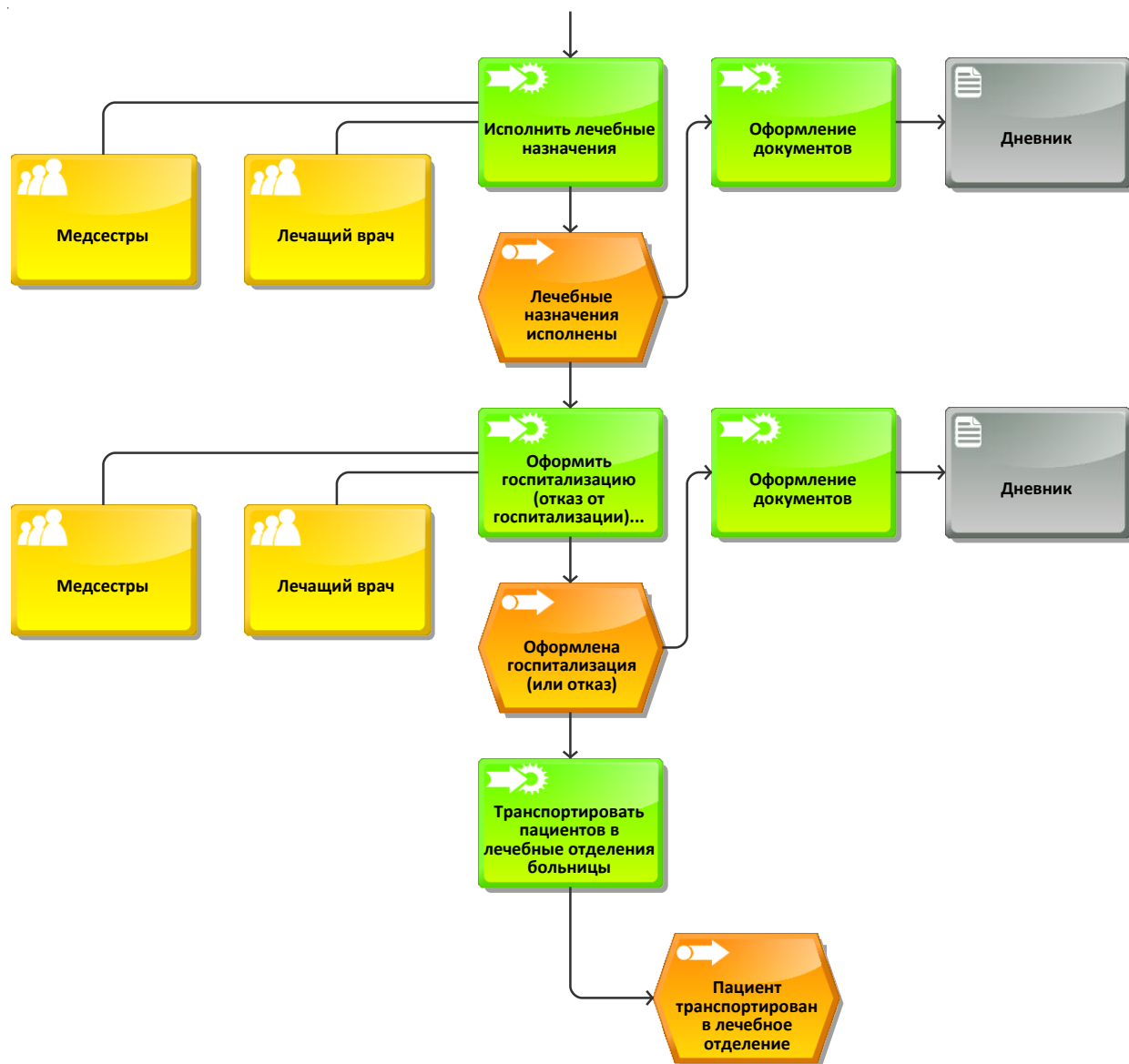


Рис.4 – Процесс «Оказать квалифицированную медицинскую помощь» в модели ARIS eEPC «AS-IS» ч.2 (3 уровень)

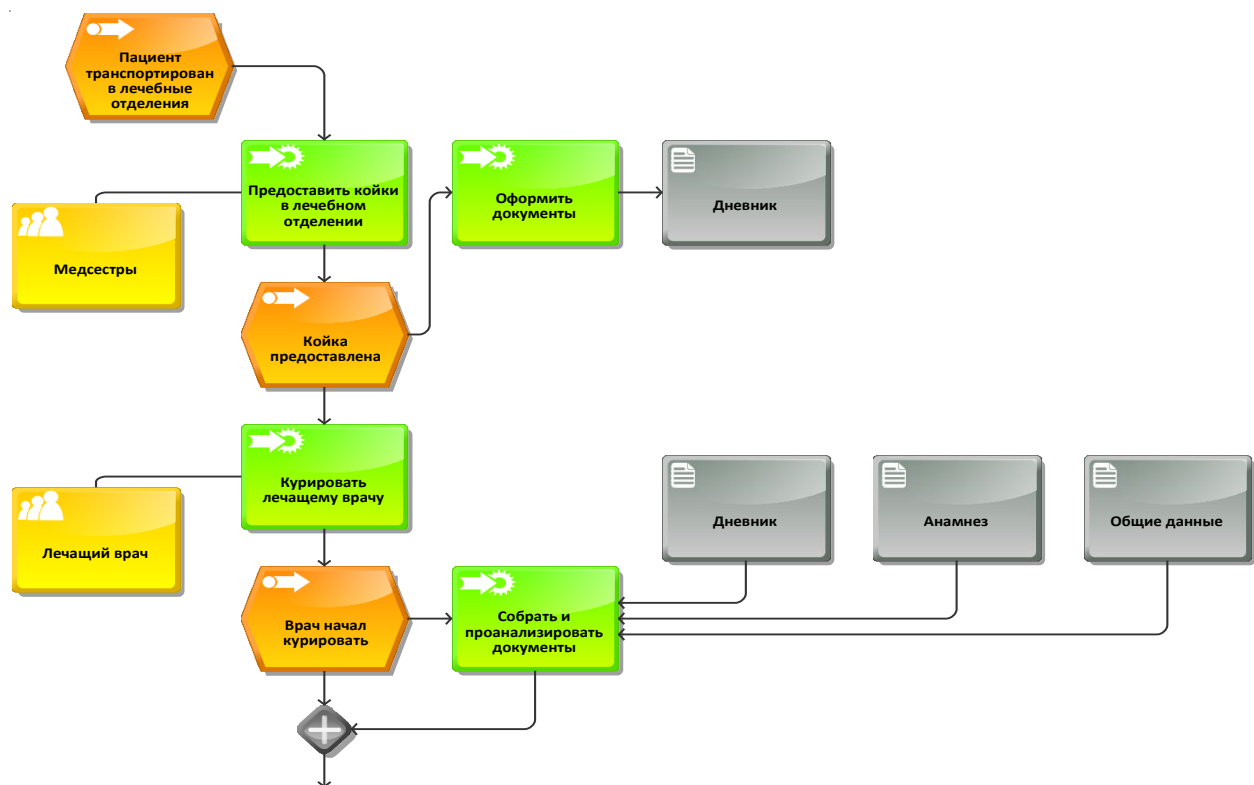


Рис.5 – Процесс «Лечить пациента» в модели ARIS eEPC «AS-IS» ч.1
(2 уровень)

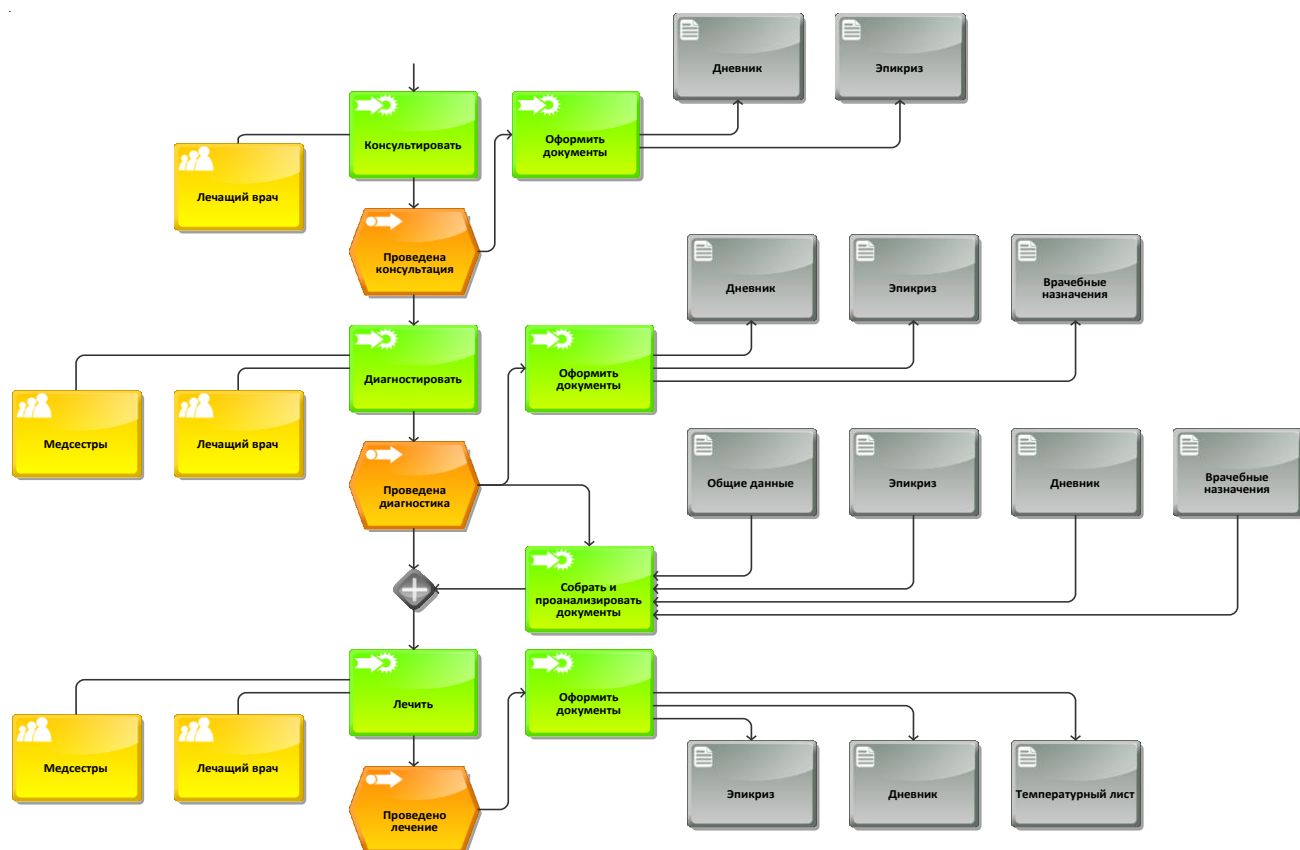


Рис.6 – Процесс «Лечить пациента» в модели ARIS eEPC «AS-IS» ч.2
(2 уровень)

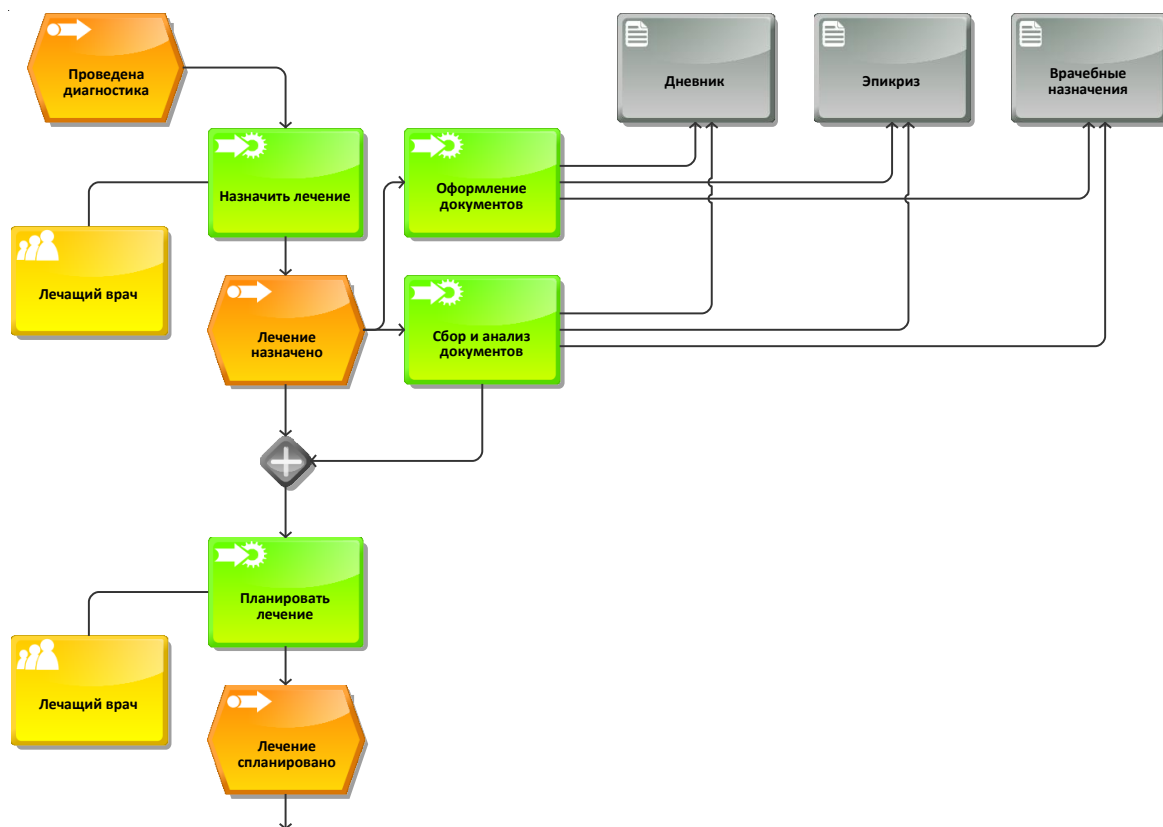


Рис.7 – Процесс «Лечить» в модели ARIS eEPC «AS-IS» ч.1 (3 уровень)

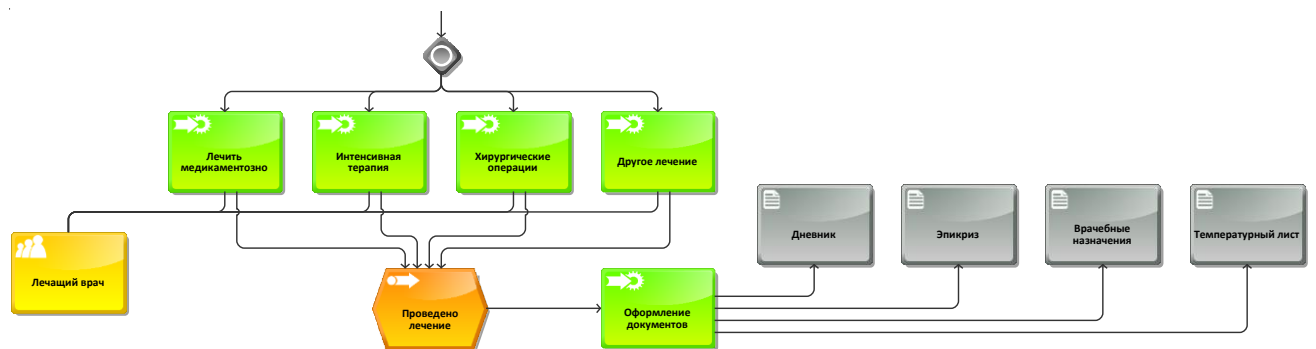


Рис.8 – Процесс «Лечить» в модели ARIS eEPC «AS-IS» ч.2 (3 уровень)

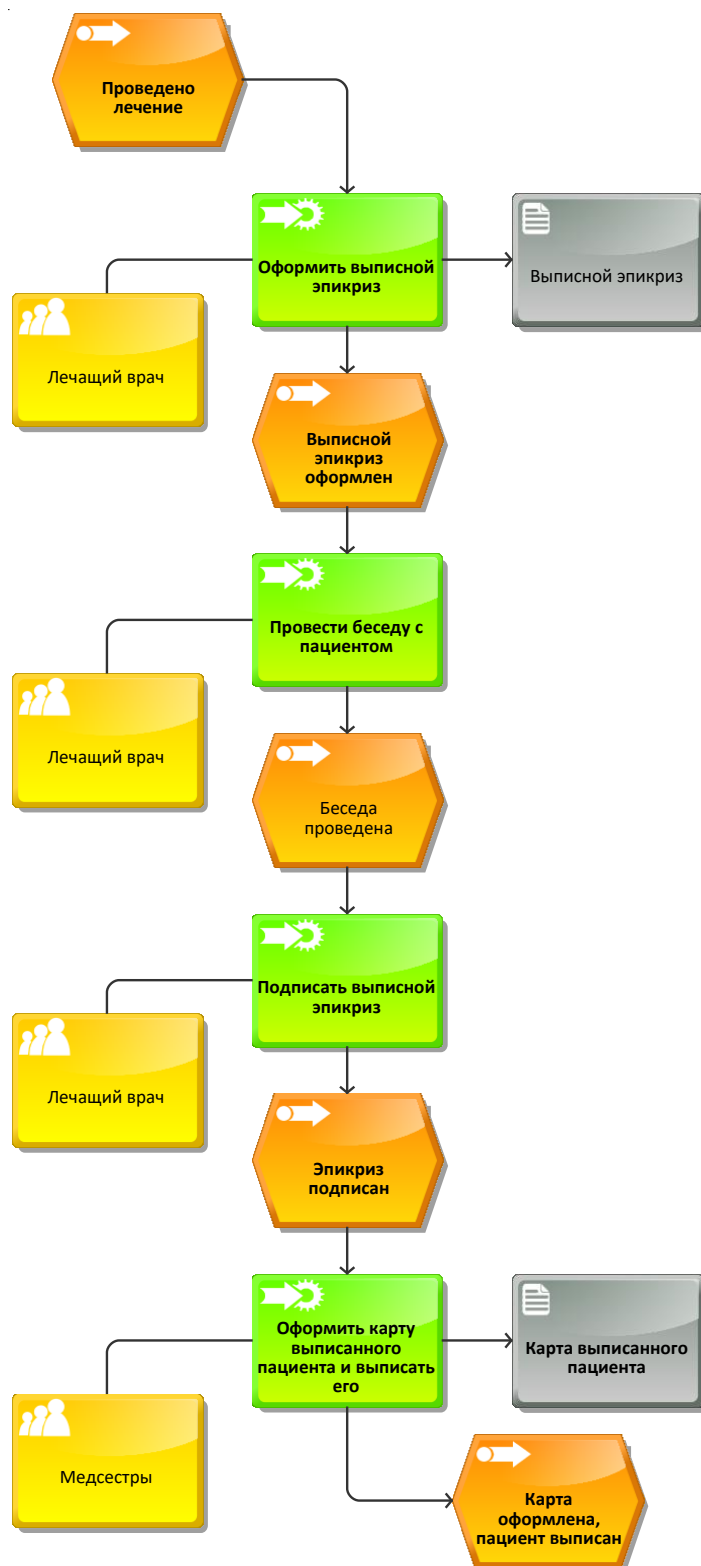


Рис.9 – Процесс «Выписать пациента» в модели ARIS eEPC «AS-IS»
(2 уровень)

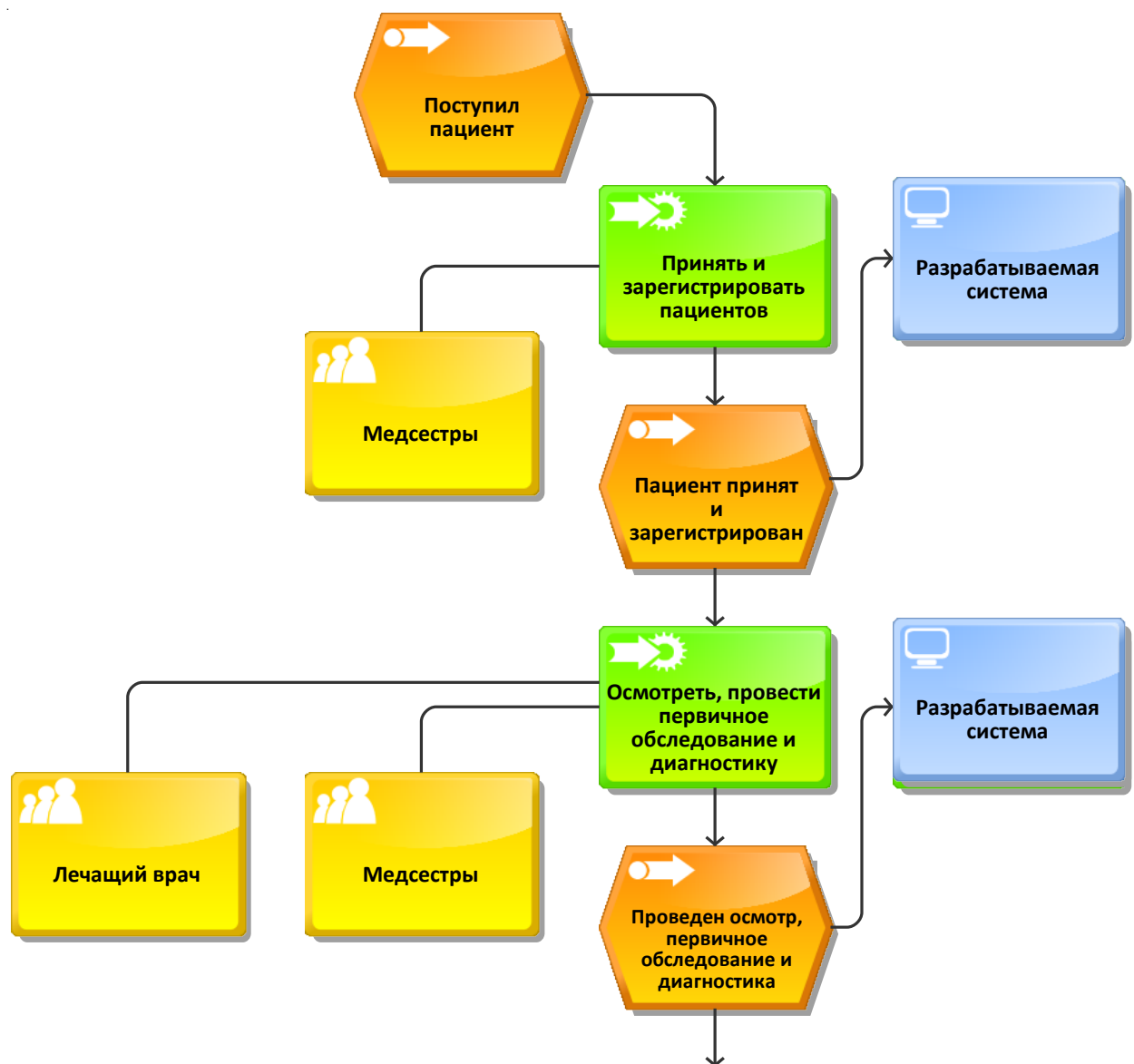


Рис.10 – Процесс «Принять пациента» в модели ARIS eEPC «TO-BE» ч.1
(2 уровень)

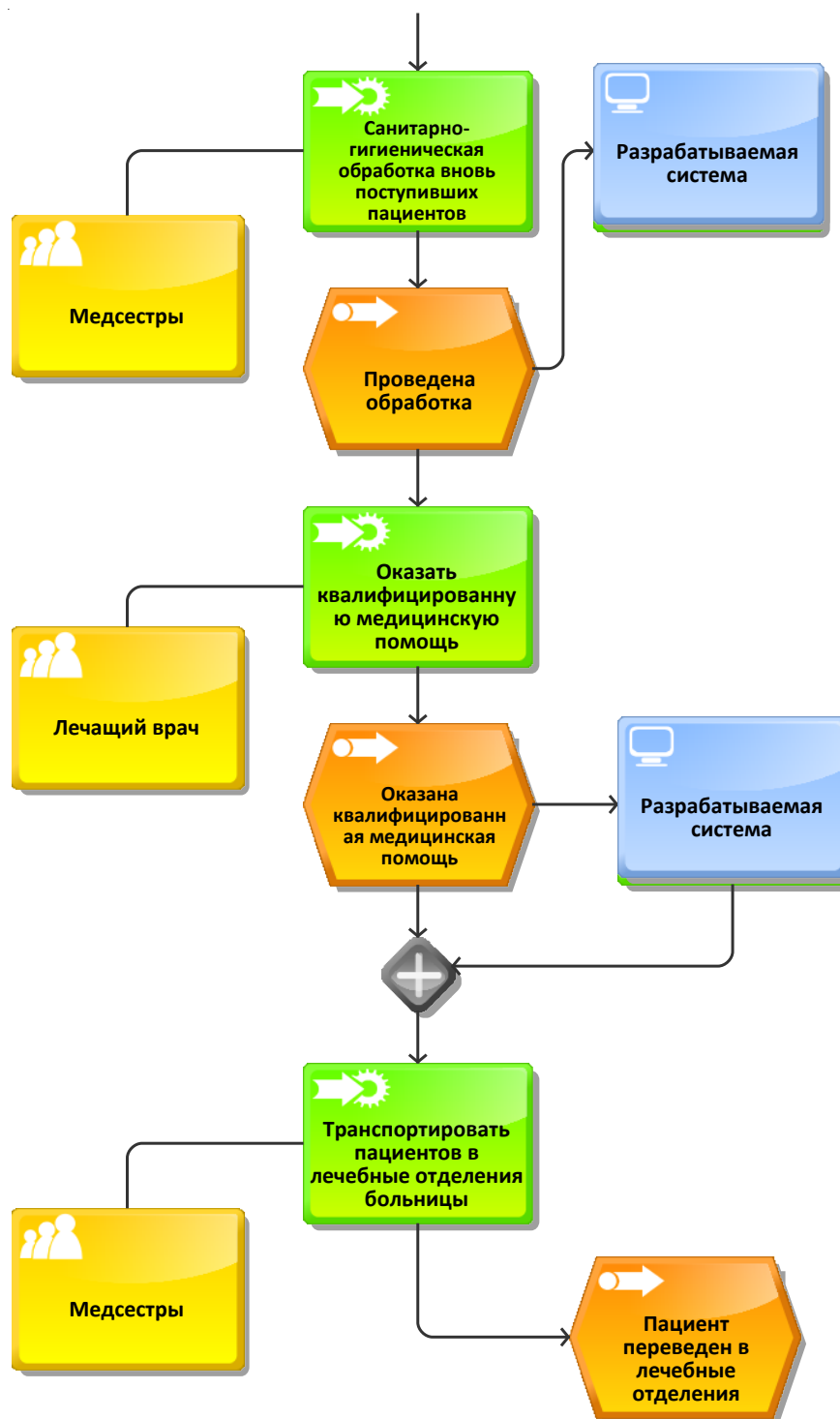


Рис.11 – Процесс «Принять пациента» в модели ARIS eEPC «TO-BE» ч.1 (2 уровень)

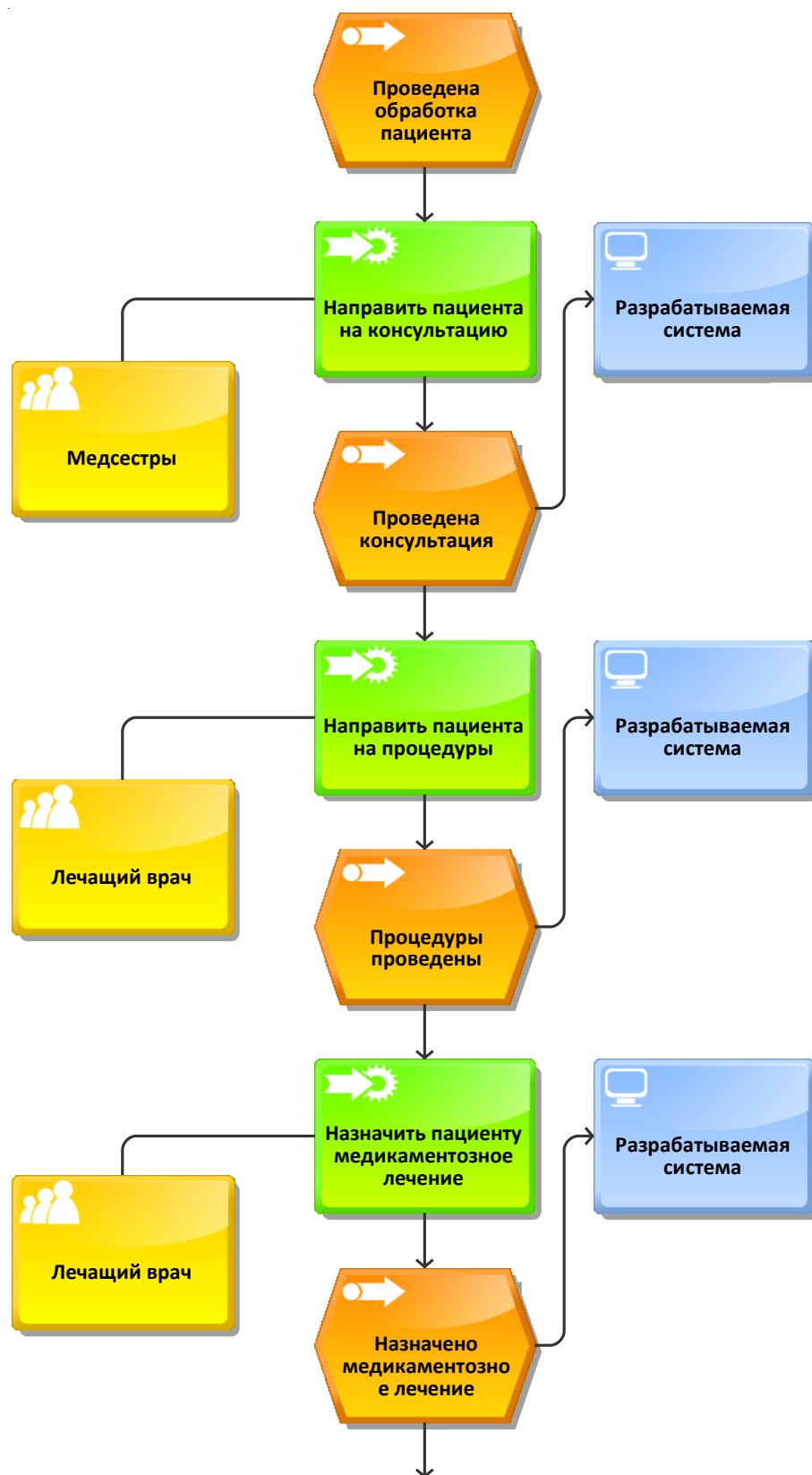


Рис.12 – Процесс «Оказать квалифицированную медицинскую помощь» в модели ARIS eEPC «TO-BE» ч.1 (3 уровень)

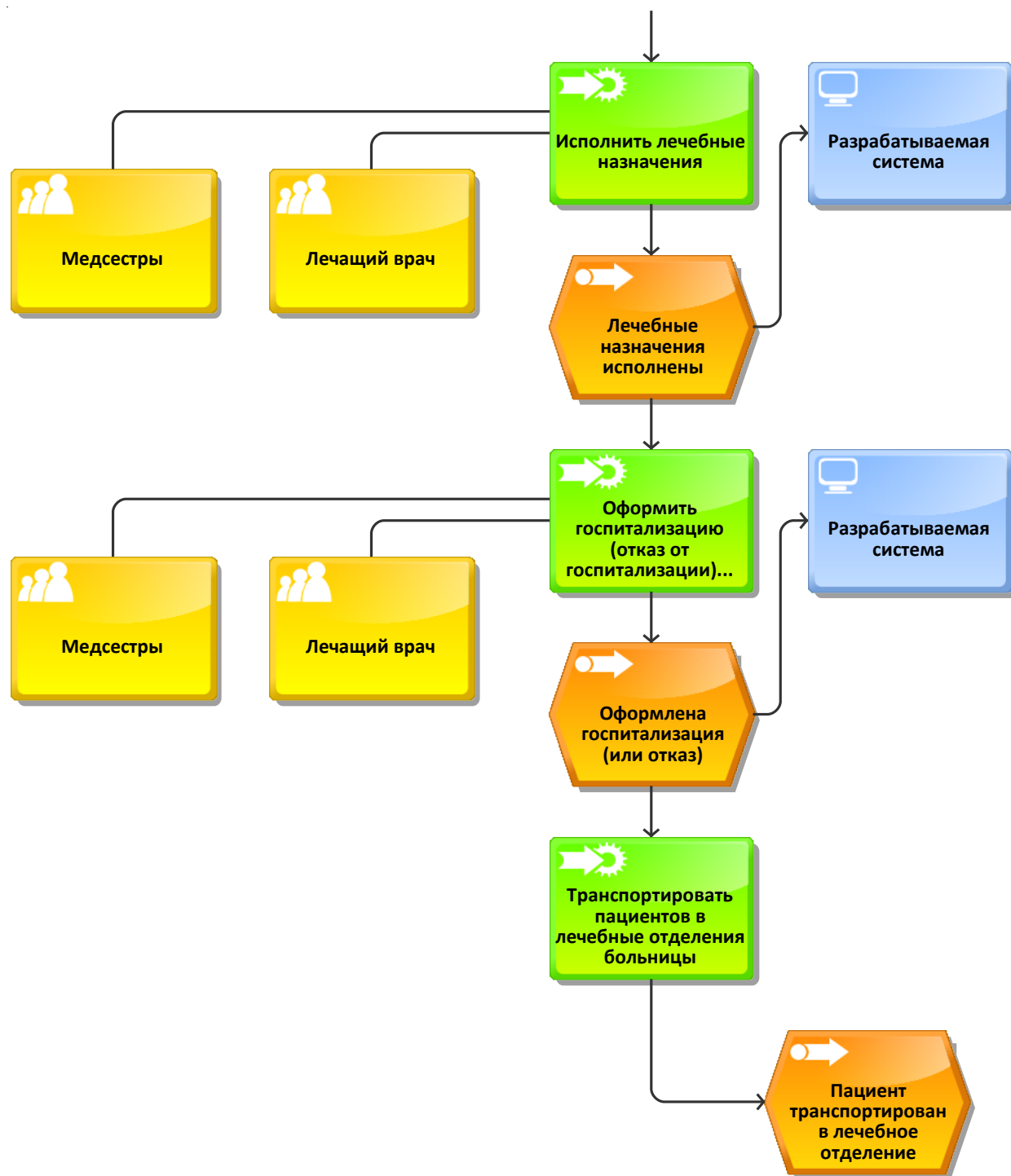


Рис.13 – Процесс «Оказать квалифицированную медицинскую помощь» в модели ARIS eEPC «ТО-ВЕ» ч.2 (3 уровень)

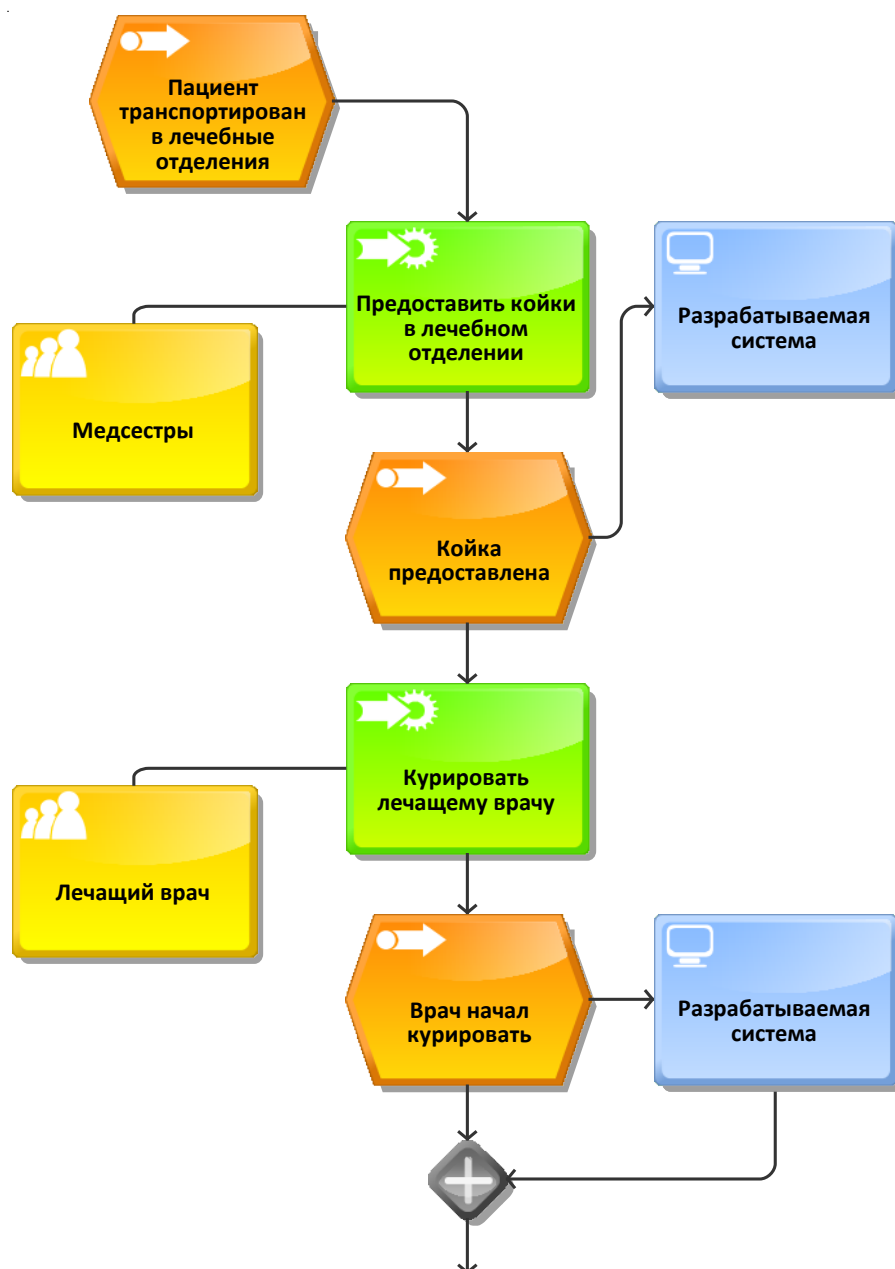


Рис.14 – Процесс «Лечить пациента» в модели ARIS eEPC «ТО-ВЕ» ч.1
(2 уровень)

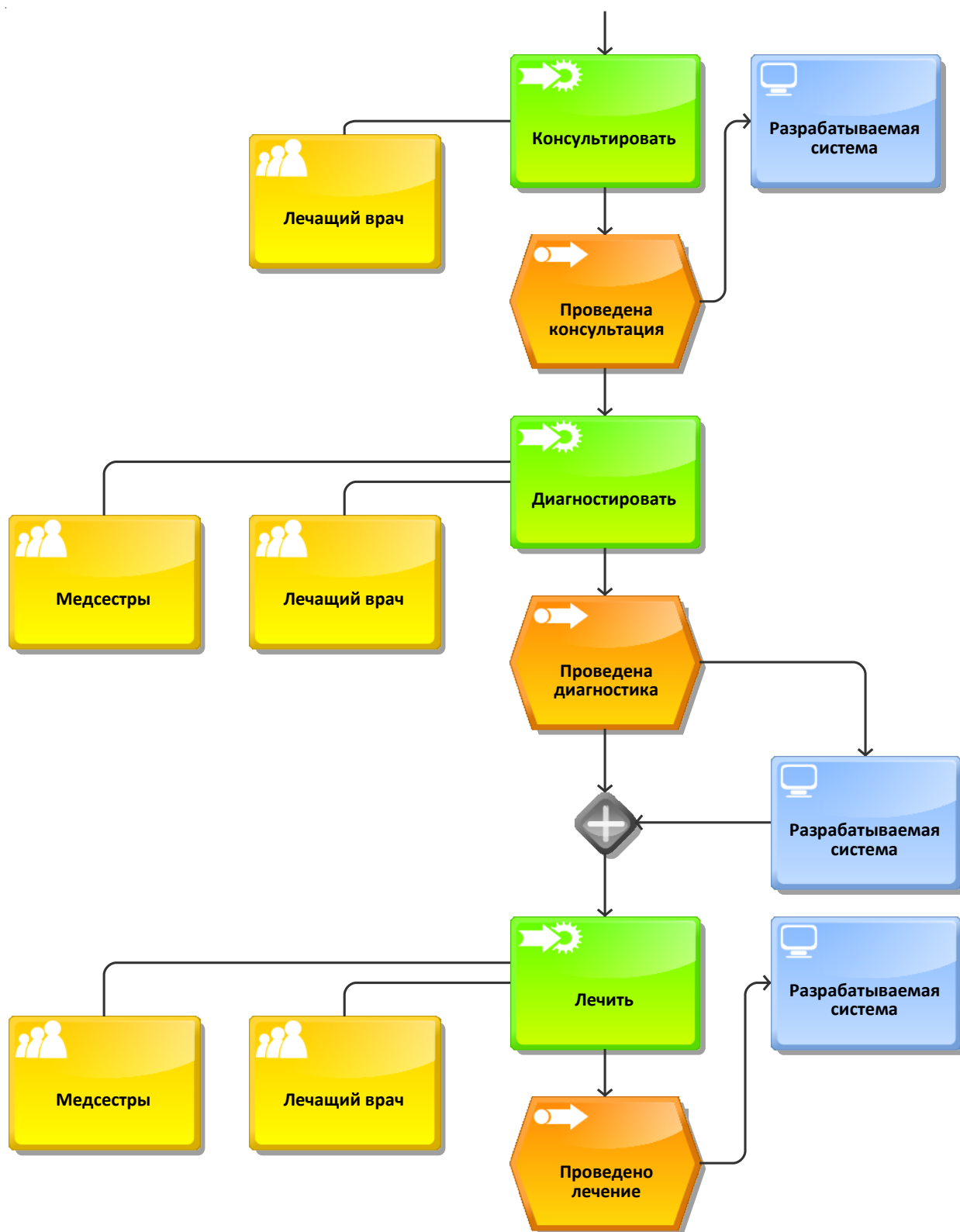


Рис.15 – Процесс «Лечить пациента» в модели ARIS eEPC «TO-BE» ч.2
(2 уровень)

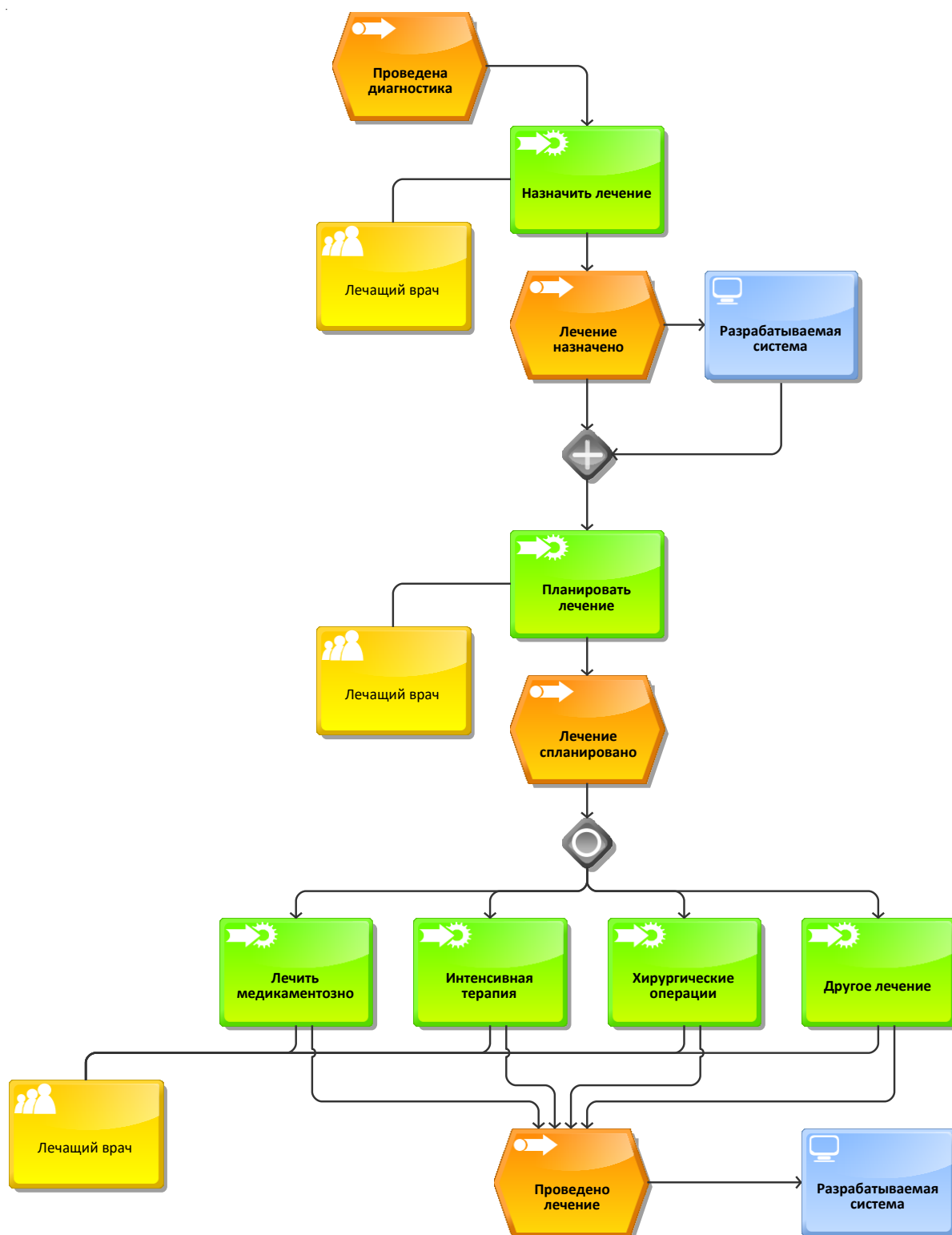


Рис.16 – Процесс «Лечить» в модели ARIS eEPC «TO-BE» (3 уровень)

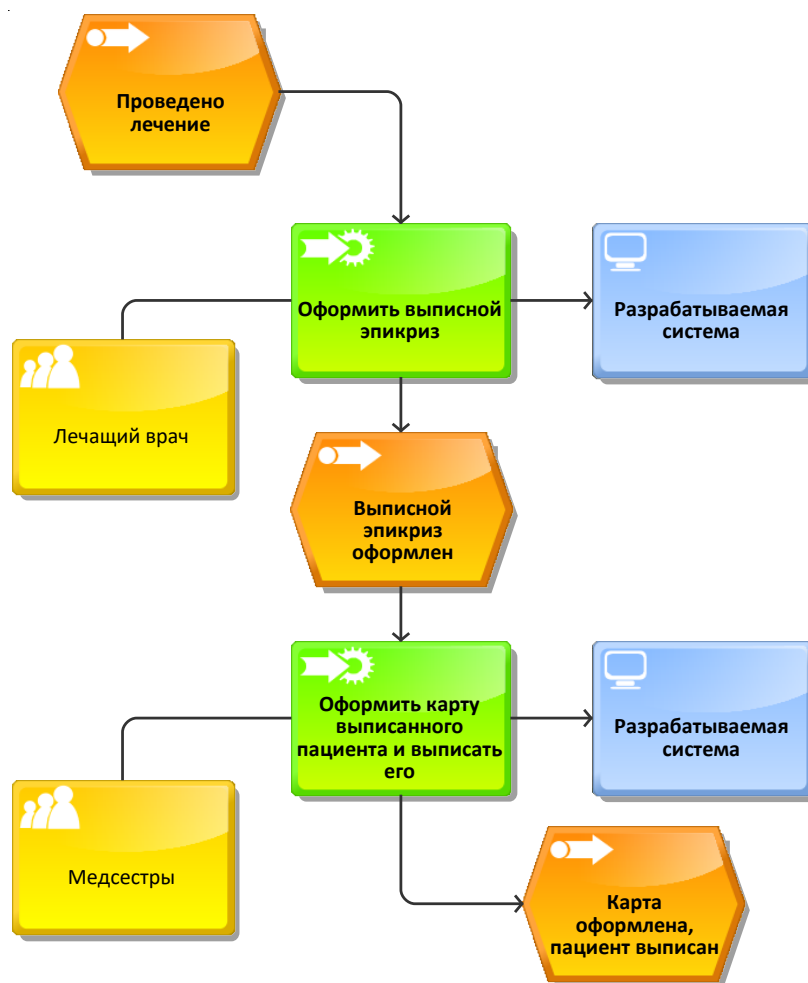


Рис.17 – Процесс «Выписать пациента» в модели ARIS eEPC «ТО-ВЕ»
(2 уровень)