

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Г. М. Цибульский

подпись

« ____ » _____ 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 — Информационные системы и технологии

Разработка прототипа информационной системы удаленного
консультирования в медицинском учреждении

Руководитель ст. преп. каф. СИИ _____
подпись, дата

М. А. Аникьева

Выпускник _____
подпись, дата

Д. С. Широкова

Нормоконтролер _____
подпись, дата

М. А. Аникьева

Красноярск 2019

Продолжение титульного листа бакалаврской работы по теме
«Разработка прототипа информационной системы удаленного
консультирования в медицинском учреждении».

Консультант:

Маглинец Ю. А. канд. техн. наук, доцент каф. СИИ

подпись, дата

Нормоконтролер

подпись, дата

М. А. Аникьева

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ

Заведующий кафедрой

_____ Г. М. Цибульский

подпись

«_____» _____ 2019 г.

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

студенту Широковой Дарье Сергеевне
группы КИ15-11Б, направления 09.03.02 «Информационные системы и технологии», профиль 09.03.02.04 «Информационные системы и технологии в медиаиндустрии».

Тема выпускной квалификационной работы: «Разработка прототипа информационной системы удаленного консультирования в медицинском учреждении».

Утверждена приказом по университету №6523/с от 16 мая 2019г.

Руководитель ВКР М. А. Аникьева, старший преподаватель кафедры систем искусственного интеллекта ИКИТ СФУ.

Исходные данные для ВКР: задание на бакалаврскую работу.

Перечень разделов ВКР:

- введение;
- обзор предметной области;
- выводы по главе 1;
- проектирование информационной системы;
- выводы по главе 2;
- реализация прототипа информационной системы;
- выводы по главе 3;
- заключение;
- список сокращений;
- список использованных источников;
- приложения.

Перечень графического материала: презентация «Разработка прототипа информационной системы удаленного консультирования в медицинском учреждении».

Руководитель ВКР

М. А. Аникьева

подпись

Задание принял к исполнению

Д. С. Широкова

подпись

«_____» _____ 2019 г.

График

выполнения выпускной квалификационной работы студентом направления 09.03.02 «Информационные системы и технологии», профиля 09.03.02.04 «Информационные системы в медиаиндустрии».

График выполнения выпускной квалификационной работы приведен в таблице 1.

Таблица 1 — График выполнения выпускной квалификационной работы

Наименование этапа	Срок выполнения	Результат выполнения этапа	Примечание руководителя (отметка о выполнении этапа)
Определение темы работы	28.01.19 — 10.02.19	Краткое эссе по теме ВКР	Выполнено
Сбор литературных источников	11.02.19 — 18.02.19	Список источников литературы	Выполнено
Анализ собранных литературных источников	19.02.19 — 5.03.19	Реферат о проблемно-предметной области	Выполнено
Решение первой задачи ВКР	6.03.19 — 30.03.19	Доклад и презентация по первой задаче ВКР	Выполнено
Решение второй задачи ВКР	31.03.19 — 11.05.19	Доклад и презентация по второй задаче ВКР	Выполнено
Уточнение и обоснование актуальности, цели и задач ВКР	12.05.19 — 13.05.19	Окончательная формулировка цели и задач ВКР	Выполнено
Решение третьей задачи ВКР	14.05.19 — 12.06.19	Доклад и презентация по третьей задаче ВКР	Выполнено
Подготовка доклада и презентации по теме ВКР	13.06.19 — 14.06.19	Доклад с презентацией по теме ВКР	Выполнено
Компоновка отчета по результатам решения задач ВКР	15.06.19 — 16.06.19	Отчет по результатам решения задач ВКР	Выполнено
Первичный нормоконтроль	17.06.19 — 18.06.19	Пояснительная записка, презентация к ВКР	Выполнено
Предварительная защита результатов ВКР	19.06.19 — 20.06.19	Доклад и презентация по проделанной работе	Выполнено

Окончание таблицы 1

Наименование этапа	Срок выполнения	Результат выполнения этапа	Примечание руководителя (отметка о выполнении этапа)
Вторичный нормоконтроль	21.06.19 — 24.06.19	Пояснительная записка, презентация к ВКР	Выполнено
Итоговый нормоконтроль	25.06.19 — 30.06.19	Пояснительная записка, презентация к ВКР	Выполнено
Защита ВКР	1.07.19 — 2.07.19	Доклад и презентация по результатам бакалаврской работы	Выполнено

Руководитель

подпись, дата

М. А. Аникьева

Студент группы КИ15-116

подпись, дата

Д. С. Широкова

Федеральное государственное автономное образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

Отзыв

на выпускную квалификационную работу
в форме бакалаврской работы
студента группы КИ15-11Б Широковой Дарьи Сергеевны
на тему: «Разработка прототипа информационной системы удаленного
консультирования в медицинском учреждении»

Бакалаврская работа Широковой Д. С. посвящена разработке прототипа информационной системы удаленного консультирования в медицинском учреждении. Актуальность данной разработки обусловлена необходимостью как совершенствования методов консультирования и дальнейшей терапии, так и автоматизации способов хранения медицинских данных и больничных карт пациентов.

Бакалаврская работа Широковой Д. С. представляет собой самостоятельное законченное исследование. В ней объективно рассмотрены программные продукты для сферы профессионального образования, выявлены недостатки информационных систем, ставшие основой для написания работы. Бакалаврская работа выполнена на высоком теоретико-практическом уровне, соблюден научный стиль изложения работы и грамотное оформление. Состоит из введения, трех глав, заключения, списка сокращений, списка использованных источников и приложений. План работы отражает содержание заявленной темы. Бакалаврская работа сопровождается необходимыми таблицами, диаграммами и рисунками.

Широкова Дарья Сергеевна продемонстрировала хорошие аналитические способности, умение анализировать и систематизировать собранную информацию, а также делать самостоятельные выводы, предложения и обобщения. Поставленные в бакалаврской работе задачи выполнены. Существенных недостатков результаты работы не имеют. Считаю, что бакалаврская работа заслуживает оценки «отлично», а её автор заслуживает присвоения квалификации «бакалавр» по направлению 09.03.02 — Информационные системы и технологии, профилю подготовки 09.03.02.04 «Информационные системы в медиаиндустрии».

Руководитель ВКР ст. преп. каф. СИИ _____

подпись, дата

М. А. Аникьева

РЕФЕРАТ

Выпускная квалификационная работа на тему: «Разработка прототипа информационной системы удаленного консультирования в медицинском учреждении» содержит 52 страницы текстового документа, 34 иллюстрации, 1 таблицу, 3 приложения, 12 использованных источников.

МЕДИЦИНСКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, MVT, ПРОГРАММИРОВАНИЕ, PYTHON, DJANGO FRAMEWORK, СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ, API

Целью данной выпускной квалификационной работы является создание прототипа информационной системы удаленного консультирования в медицинском учреждении.

Цель выпускной квалификационной работы достигается решением следующих задач:

1. Обзор и сравнение существующих решений, выявление требований к информационной системе удаленного консультирования;
2. Проектирование информационной системы удаленного консультирования;
3. Разработка прототипа информационной системы удаленного консультирования.

Результатом выпускной квалификационной работы является прототип информационной системы удаленного консультирования.

СОДЕРЖАНИЕ

Введение	5
1 Обзор предметной области	7
1.1 Описание предприятия	7
1.2 Обзор существующих решений	9
1.2.1 Описание критериев	9
1.2.2 Сравнение существующих решений.....	10
1.3 Пользовательские требования к ИС.....	12
1.4 Функциональные требования к ИС.....	12
Выводы по первой главе	16
2 Проектирование информационной системы	17
2.1 Модель бизнес-процессов.....	17
2.2 Диаграмма переходов	19
2.3 Архитектура ИС	21
2.4 Выбор системы управления базой данных.....	23
2.5 Среда и инструменты разработки	26
Выводы по второй главе	28
3 Реализация прототипа информационной системы.....	29
3.1 Разработка прототипа серверной части	29
3.2 Разработка прототипа клиентской части	32
3.3 Панель администратора.....	36
Выводы по третьей главе	37
Заключение	38
Список сокращений.....	39
Список использованных источников	40
Приложение А Макеты	42
Приложение Б Интерфейс администратора.....	44
Приложение В Плакаты презентации	46

ВВЕДЕНИЕ

Телемедицина является одним из наиболее быстрорастущих направлений развития сферы оказания медицинских услуг. Под телемедициной понимается подход, при котором лечебно-диагностические, превентивные и организационно-управленческие процессы реализуются посредством использования информационных и телекоммуникационных технологий [1]. Это позволяет с одной стороны расширить возможности доступа пациентов к медицинским услугам, а с другой обеспечить централизованность хранения данных о пациенте с целью последующего увеличения точности диагностики.

В открытых источниках не удалось обнаружить распространенных и готовых решений. Существуют либо медицинские информационные системы, ориентированные на организацию электронного документооборота и реализующие функции онлайн-регистратуры, либо агрегаторы, выступающие в роли связующего звена между пациентом и различными клиниками, либо специалистами напрямую.

Проект «Информационная система удаленного консультирования в медицинском учреждении» является инструментом организации работы персонала частного психотерапевтического медицинского учреждения и его клиентов. Организация работы в общем случае предполагает составление графика работы врача и пациентов, ведение электронных медицинских карт пациентов, оплаты консультаций пациентом, возможность ведения личных записей и переписки врача и пациента. Использование предложенной разработки на предприятии позволит решить с одной стороны задачи экономической выгоды — сокращение времени личного приема дает возможность снизить затраты на аренду, привлечь удаленно работающих консультантов и является способом привлечения большего количества пациентов, с другой — наличие подобного сервиса значительно упрощает процесс коммуникации врача и пациента [2].

Цель — разработка прототипа информационной системы удаленного консультирования в медицинском учреждении.

Задачи:

- Обзор и сравнение существующих решений, выявление требований к информационной системе удаленного консультирования;
- Проектирование информационной системы удаленного консультирования;
- Разработка прототипа информационной системы удаленного консультирования.

1 Обзор предметной области

В настоящее время основным условием стабильного функционирования на рынке любого предприятия является совершенствование методов управления. Один из инструментов совершенствования систем управления предприятием — внедрение информационных систем (ИС), которое позволяет организовать эффективное планирование всей хозяйственной и финансовой деятельности и, как следствие, повысить рентабельность предприятия на 5–15 %; снизить риски за счет оперативного получения информации обо всех бизнес-процессах предприятия, за счет своевременного принятия решений [3]. В соответствии со стандартом ГОСТ РВ 51987 «Информационная технология. Комплекс стандартов на автоматизированные системы. Требования и показатели качества функционирования информационных систем», информационная система может быть определена как автоматизированная система, результатом функционирования которой является представление выходной информации для последующего использования.

Особенный интерес представляют медицинские информационные системы (МИС) — комплексные информационные системы, предназначенные для автоматизации различных процессов, протекающих в медицинском учреждении. Функциональные возможности таких систем, в первую очередь, связаны с организацией документооборота: сбор, структуризация и документирование данных, статистический анализ данных, контроль эффективности и качества оказания медицинской помощи, анализ и контроль работы учреждения, управление ресурсами учреждения [4].

1.1 Описание предприятия

Для примера рассмотрим работу частного психоневрологического медучреждения. Любое лечение всегда начинается с диагностики

и постановки верного заключения, а обследование, в свою очередь, начинается с установления симптомов недуга. Чем правдивее и объемнее будут ответы пациента, тем проще доктору отыскать причины болезни.

В некоторых случаях для постановки диагноза и последующего лечения достаточно только беседы врача и пациента. Если у специалиста остались сомнения касательно диагноза после первичного осмотра и опроса, назначается обследование с использованием специального оборудования — МРТ, рентген, электроэнцефалография, реография или миография.

После постановки диагноза назначается амбулаторное лечение, в ходе которого пациенту необходимо с некоторой периодичностью консультироваться с лечащим врачом для контроля и своевременной коррекции терапии.

В настоящий момент необходимость очного первичного приема существенно ограничивает возможность привлечения новых клиентов и затрудняет ведение терапии с постоянными клиентами. И со стороны пациента, и со стороны лечащего врача требуется большое количество ресурсов на сторонние процессы, связанные с затратами времени, средств и т. д.

Также возникают сложности с организацией централизованного хранения медицинских данных и больничных карт пациентов. Особенности психотерапевтической поддержки предполагают, что пациент ведет большое количество личных ассоциированных с историей болезни записей, которые также должны быть доступны врачу для анализа, а также организовано храниться для дальнейшего уточнения диагностики и терапии.

Учитывая сложности, возникающие в процессе работы предприятия и отсутствие готовых решений подобной тематики, было принято решение создать проект «Информационная система удаленного консультирования в медицинском учреждении», который является инструментом организации

работы персонала частного психотерапевтического медицинского учреждения и его клиентов.

1.2 Обзор существующих решений

В открытых источниках не удалось обнаружить распространенных и готовых решений подобной тематики. Существуют либо медицинские информационные системы, ориентированные на организацию электронного документооборота и реализующие функции онлайн-регистратуры, либо агрегаторы, выступающие в роли связующего звена между пациентом и различными клиниками либо специалистами напрямую. Однако существуют как описательные материалы и теоретический базис по данной проблеме, так и подобные примеры решения подобных задач в иных сферах применения.

1.2.1 Описание критериев

Критерии для сравнения существующих инструментов были выбраны исходя из предположений о том, какие функции позволят оптимизировать процессы хранения медицинских данных, организации консультирования и дальнейшей терапии.

1. Личный кабинет пациента — отсутствие посредников при обращении позволяет повысить уровень обслуживания за счет уменьшения времени реакции на запрос пациента;

2. Личный кабинет врача — увеличение эффективности работы врача может быть достигнута путем уменьшения времени простоя и предоставления возможности оперативного уточнения вторичных данных в неприемное время;

3. Возможность уникальной конфигурации — внесение изменений в работу системы, учитывающих специфику работы конкретного предприятия, увеличивает эффективность работы предприятия;

4. Ведение медицинской карты — увеличение скорости доступа к информации о пациенте и ее количества и позволяет повысить уровень организации хранения данных.

1.2.2 Сравнение существующих решений

Для сравнения были выбраны следующие существующие решения:

1. Медицинские информационные системы — комплексные информационные системы, предназначенные для автоматизации различных процессов, протекающих в медицинском учреждении. Функциональные возможности таких систем в первую очередь связаны с организацией документооборота: сбор, структуризация и документирование данных, статистический анализ данных, контроль эффективности и качества оказания медицинской помощи, анализ и контроль работы учреждения, управление ресурсами учреждения. Примерами подобных медицинских информационных систем являются «Медохват», «Клиентикс», «MedWork» и многие другие.

2. Сервисы онлайн-консультаций — веб-приложения, функциональные возможности которых заключаются в предоставлении возможности пользователю получить консультацию квалифицированного врача вне зависимости от своего физического местоположения. Технологии подобных сервисов позволяют передать заявку на консультацию первому освободившемуся специалисту, что минимизирует время ожидания ответа. Также подобные сервисы сотрудничают с медучреждениями и аптеками, предоставляют возможность покупки, бронирования и доставки медикаментов, предоставляет справочник о режиме работ учреждений. Представителем подобного сервиса в России является веб-сервис Яндекс.Здоровье.

3. АИС телемедицинского консультирования — проект, разработанный в Томском государственном университете в 2009 году.

Во многом аналогичный информационной системе удаленного консультирования, проект так и не получил окончательной реализации.

Сравнение производится по описанным критериям и представлено в таблице 1.

Таблица 1 — Сравнительная таблица существующих решений

	Медицинские информационные системы	Яндекс.Здоровье	АИС Телемедицинского консультирования
Личный кабинет пациента	-	±	+
Личный кабинет врача	+	+	+
Возможность уникальной конфигурации	+	-	-
Ведение медицинской карты	+	±	+

После проведенного сравнения были сформулированы ключевые требования к информационной системе удаленного консультирования. Очевидно, что принципиальным отличием от существующих инструментов будет активное участие пользователя в системе. Именно отсутствие посредников при обращении позволяет как повысить уровень обслуживания за счет уменьшения времени реакции на запрос пациента, так и увеличить эффективность работы врача путем уменьшения времени простоя и предоставления возможности оперативного уточнения вторичных данных в неприемное время. Также оптимизация хранения и доступа к данным увеличивает удобство работы врачей с системой, повышает эффективность работы предприятия.

1.3 Пользовательские требования к ИС

На основе проведенного сравнения существующих решений были сформулированы ключевые требования к информационной системе удаленного консультирования:

1. Возможность активного участия пациента в информационной системе;
2. Отсутствие посредника и необходимости личного присутствия при обращении пациента в клинику;
3. Организация централизованного хранения медицинских данных.

Организация работы в общем случае предполагает составление графика работы врача и пациентов, ведение электронных медицинских карт пациентов, оплату консультаций пациентом, возможность ведения личных записей и переписки врача и пациента.

1.4 Функциональные требования к ИС

Возможности, предоставляемые пользователю при работе с системой, определяются его ролью. В системе предусмотрены следующие роли:

- Гость — незарегистрированный пользователь, который не имеет возможности совершения операций в системе;
- Пациент — зарегистрированный пользователь, который имеет доступ в систему для совершения операций;
- Врач — зарегистрированный пользователь, который имеет доступ в систему для совершения операций;
- Администратор — пользователь, имеющий неограниченные привилегии управления информационной системой.

Пользователи могут авторизоваться на специально созданной странице портала, ссылка на которую располагается в шапке сайта, с помощью специальной формы авторизации.

После авторизации и в зависимости от роли пользователям могут быть доступны следующие разделы:

1. Личный кабинет пациента.

В данном разделе пациенту доступна информация о своих личных данных. Изменение информации данного раздела производится путём заполнения данных формы, состоящей из полей: имя пользователя, фамилия пользователя, отчество пользователя, пароль, номер телефона, e-mail адрес, город проживания и описание.

- Администратор — просмотр личных данных пациентов;
- Пациент — просмотр и редактирование собственных личных данных;
- Врач — просмотр личных данных пациента открытых для чтения.

После авторизации пациента в личном кабинете будут доступны переходы в следующие разделы: просмотр и редактирование профиля, переписка, календарь, личные записи, запись.

2. Личный кабинет врача.

В данном разделе врачу доступна информация о своих личных данных. Изменение информации данного раздела производится путём заполнения данных формы, состоящей из полей: имя пользователя, фамилия пользователя, отчество пользователя, пароль, должность, номер телефона, e-mail адрес, город проживания.

- Администратор — просмотр личных данных врачей;
- Пациент — просмотр данных открытых для чтения;
- Врач — просмотр и редактирование собственных личных данных.

После авторизации врача в личном кабинете будут доступны переходы в следующие разделы: переписка, календарь, терапевтические материалы, медицинские карты.

3. Переписка.

В данном разделе пользователи могут связаться друг с другом.

- Пациент — просмотр истории переписки с врачом, создание и редактирование нового сообщения, отправка сообщения врачу.

- Врач — просмотр истории переписки, создание и редактирование нового сообщения, отправка сообщений всем своим пациентам.

4. Календарь.

В данном разделе пользователь может ознакомиться с полным списком событий, закрепленных за ним.

Создание, либо редактирование события со стороны врача, или администратора осуществляется путем заполнения формы со следующими полями: id врача, id пациента, описание события, дата и время проведения события, тип события (встреча либо терапия), статус события (активно либо неактивно). Событие будет доступно тем пользователям, id которых указаны в первых двух полях. При выборе типа события «встреча» после проведения события необходимо указать следующую информацию: место проведения встречи (номер кабинета либо информационная система), жалобы пациента, диагноз, рекомендации. При выборе типа события «терапия» после проведения события необходимо указать следующую информацию: представленные материалы, отчет о их выполнении.

- Администратор — просмотр, создание, редактирование и удаление события.

- Пациент — просмотр событий, закрепленных за ним, создание события, получение уведомления о событии.

- Врач — просмотр событий, закрепленных за ним, создание, редактирование и удаление события, получение уведомления о событии.

5. Личные записи.

В данном разделе пациент размещает личные заметки, отчеты о прохождении терапии.

- Администратор — просмотр материалов пользователя.

- Пациент — просмотр, создание, редактирование и удаление записей дневникового типа, присвоение любой из записей статуса «доступно» либо «недоступно для чтения».

- Врач — просмотр открытых для чтения материалов пользователя.

6. Терапевтические материалы.

В данном разделе врач размещает различные материалы, необходимые ему для работы. Это могут быть тексты, тесты, графические материалы, ссылки на сторонние ресурсы.

- Администратор — просмотр материалов.

- Пациент — просмотр открытых для чтения терапевтических материалов.

- Врач — просмотр, создание, редактирование и удаление терапевтических материалов, назначение пациенту для ознакомления записей.

7. Медицинские карты.

В данном разделе находятся записи электронных медицинских карт. После первичного приема администратор вносит информацию о новом пользователе. В дальнейшем возможна выборка всех медицинских записей по уникальному идентификатору пользователя, в совокупности эти записи и представляют собой электронную медицинскую карту.

- Администратор — просмотр, создание, редактирование, удаление медицинских карт.

- Пациент — просмотр личной медкарты.

- Врач — просмотр, создание, редактирование записей в медицинской карте пациента.

Диаграмма вариантов использования (usecase) предназначена для наглядной демонстрации возможных сценариев взаимодействия с системой [5]. Актером (субъектом, осуществляющим взаимодействие с системой) является зарегистрированный пользователь, сценарием использования, в свою очередь, является набор действий, совершение которых

в информационной системе доступно зарегистрированному пользователю. Диаграмма вариантов использования представлена на рисунке 1.

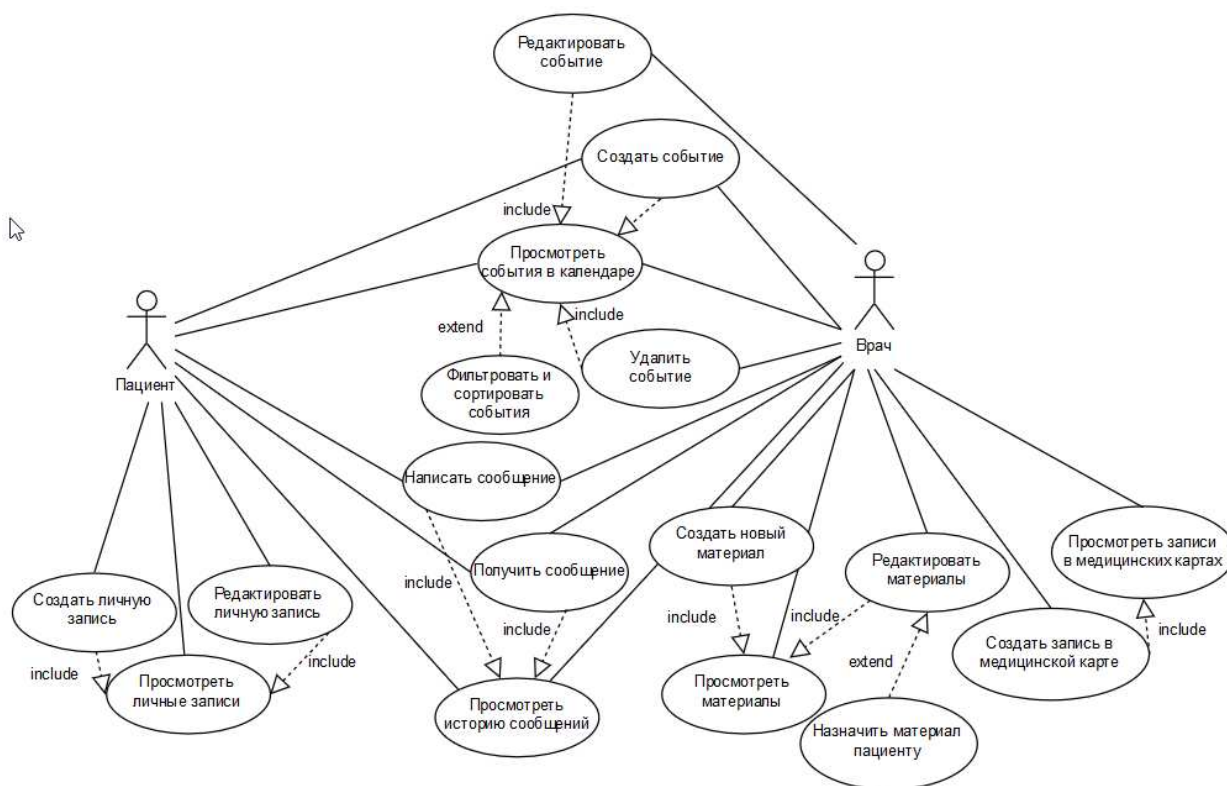


Рисунок 1 — Диаграмма вариантов использования системы

Выводы по первой главе

Были сформулированы пользовательские и функциональные требования к информационной системе, описаны роли пользователей системы и доступные им действия. Возможности и поведение системы наглядно отражены на диаграмме вариантов использования. Установлены типы отношений между вариантами использования.

В результате была выполнена первая задача — обзор существующих решений, выявление требований к информационной системе удаленного консультирования.

2 Проектирование информационной системы

Проектирование — процесс определения архитектуры, компонентов, интерфейсов и других характеристик системы или ее части. Результатом проектирования является проект — целостная совокупность моделей, свойств или характеристик, описанных в форме, пригодной для реализации системы.

2.1 Модель бизнес-процессов

IDEF0 (Integration Definition for Function Modeling) — методология функционального моделирования для описания функций предприятия [6]. Модель IDEF0 представляет собой графическое описание информационной системы в виде набора нескольких иерархически связанных IDEF0-диаграмм, которые описывают функции системы.

Контекстная диаграмма — это IDEF0-диаграмма, расположенная на вершине иерархии диаграмм, представляющая собой самое общее описание системы. Она состоит из одного блока, описывающего функцию верхнего уровня — получение консультации — и представлена на рисунке 2.

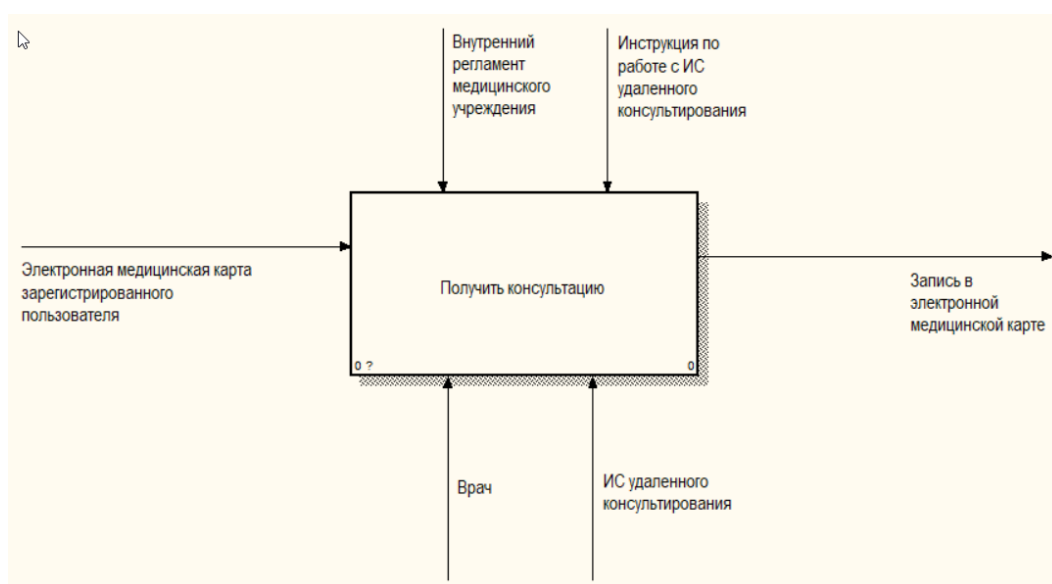


Рисунок 2 — Контекстная диаграмма процесса «Получить консультацию» А-0

В данном случае функция верхнего уровня — получить консультацию. Входными данными фактически являются записи в электронной медицинской карте пациента, а результатом процесса, соответственно, новая электронная медицинская запись. Управление процессом происходит согласно внутреннему регламенту работы медицинского учреждения и документации по работе с информационной системой удаленного консультирования. Механизмами, выполняющими этот процесс, являются врачи и информационная система.

После построения контекстной диаграммы происходит декомпозиция процесса верхнего уровня — разбиение его на несколько блоков для детализации процесса. Результат декомпозиции исследуемого процесса представлен на рисунке 3.

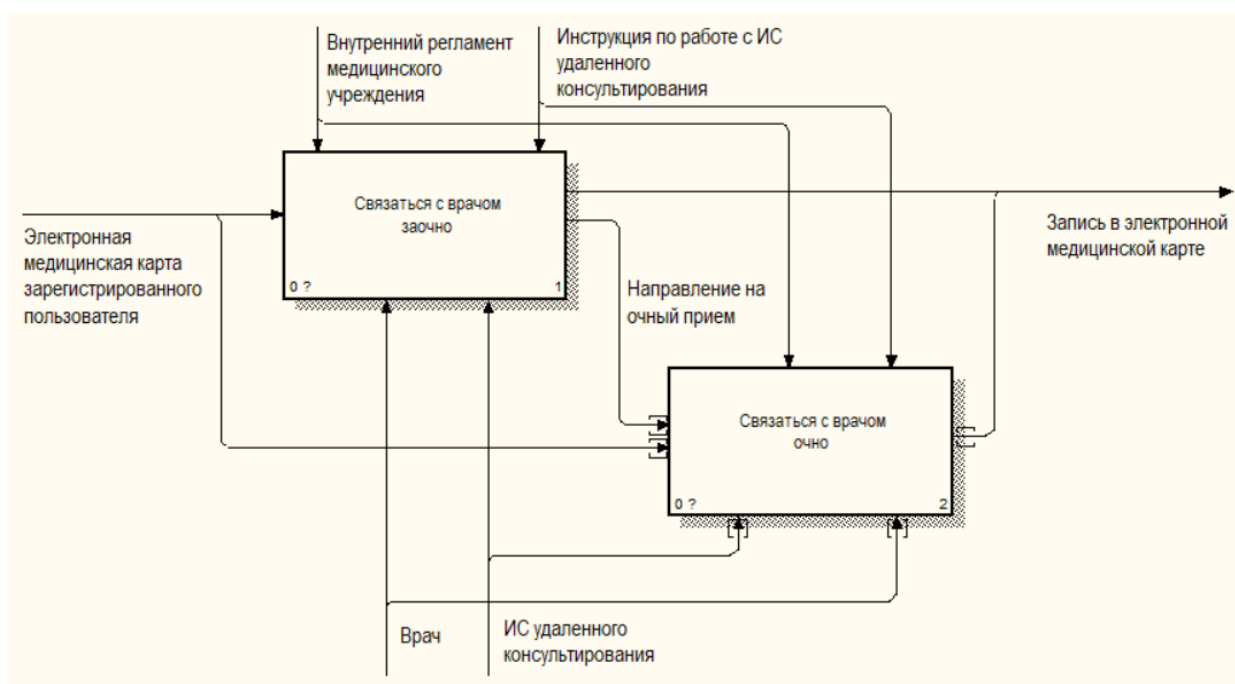


Рисунок 3 — Диаграмма декомпозиции процесса «Получить консультацию» IDEF0 A0

Процесс получения консультации разбит на следующие подпроцессы: связаться с врачом очно, либо заочно. На первом этапе происходит определение, необходим ли очный прием. В случае, если это так, входными данными для варианта «связаться очно» становится документ «направление на

прием». В случае, если в очном приеме нет необходимости, после удаленной консультации врач сразу создает новую электронную медицинскую запись. Декомпозиция процесса заочного обращения представлена на рисунке 4.

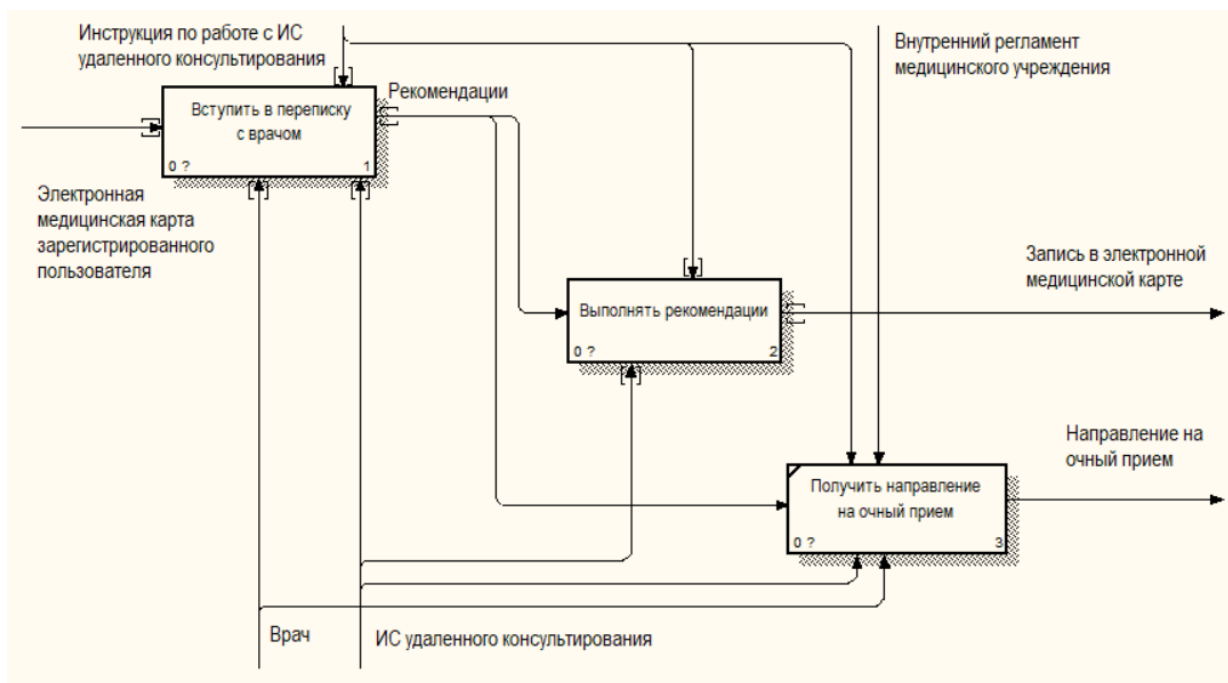


Рисунок 4 — Диаграмма декомпозиции процесса «Связаться с врачом заочно» IDEF0 A1

При заочном обращении первое, что необходимо сделать — написать сообщение лечащему врачу и получить рекомендации. Необходимо отметить, что, если это необходимо, уже на этом этапе врач может дать направление на очный прием. В иных случаях пациент выполняет рекомендации врача в течении определенного времени, отчитывается о них в личном кабинете, и на основании полученных данных врач вторично может как внести информацию в медицинскую карту, так и отправить пациента на очный прием.

2.2 Диаграмма переходов

При первом обращении в клинику пациент проходит процедуру регистрации и после первичной консультации получает данные для дальнейшей работы с системой — логин и пароль от личного кабинета. Врач

в свою очередь создает электронную карту пациента, заполняет ее и вносит информацию о первичном приеме. Любое дальнейшее взаимодействие врача и пациента в системе считается «Событием», либо «Перепиской», причем тип события может быть очным или заочным приемом, либо терапией, которая включает в себя выполнение лечебно-диагностических мероприятий и ведение отчета о них. Итогом приема становится электронная медицинская персональная запись, которая отображается на странице «Медицинские карты» [7].

На странице «Терапевтические материалы» врач может размещать необходимую ему для работы тематическую литературу, статьи, тесты, заготовки для проведения консультаций. Пациент, в свою очередь, может вносить личные материалы, отчеты о выполнении терапевтических упражнений, заметки на странице «Личные записи». Также пользователь может записаться на прием на странице «Запись». Общая структурная диаграмма системы, представленные на рисунке 5.

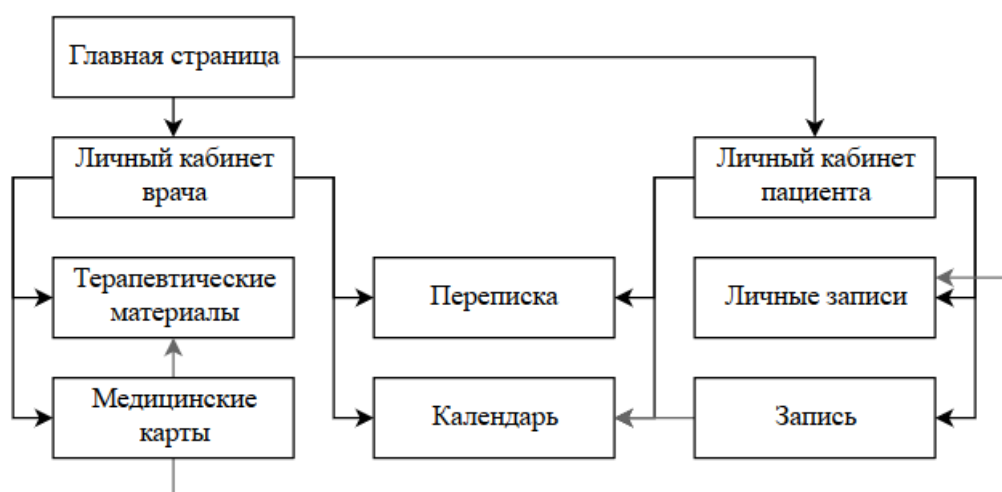


Рисунок 5 — Структурная диаграмма информационной системы

2.3 Архитектура ИС

Реализация данной информационной системы предполагает формат веб-приложения, современным промышленным стандартом для которого является архитектура клиент/сервер [8]. Ее ключевым преимуществом является централизованный доступ к данным. Хранение всех данных в памяти одной вычислительной машины (сервера) является как более безопасным способом хранения данных, так и более удобным для администрирования доступа к данным.

Использование такого типа архитектуры дает возможность уменьшить размер и сложность программ, перенести наиболее трудоемкие операции на сервер и уменьшить объем информации, передаваемой по сети. Все это улучшает производительность системы.

Тем не менее, традиционная архитектура клиент/сервер имеет множество недостатков, включая тесное связывание данных и бизнес-логики приложения на сервере, что может иметь негативное влияние на расширяемость и масштабируемость системы, а также зависимость от центрального сервера, что негативно сказывается на надежности системы. Для решения этих проблем архитектурный стиль клиент/сервер был развит в более универсальный многоуровневый, в котором устранены некоторые недостатки, свойственные 2-х уровневой архитектуре клиент/сервер, и обеспечиваются дополнительные преимущества:

1. Удобство поддержки. Уровни не зависят друг от друга, что позволяет выполнять обновления или изменения, не оказывая влияния на приложение в целом.
2. Масштабируемость. Уровни организовываются на основании развертывания слоев, поэтому масштабировать приложение довольно просто.
3. Гибкость. Управление и масштабирование каждого уровня может выполняться независимо, что обеспечивает повышение гибкости.

4. Доступность. Приложения могут использовать модульную архитектуру, которая позволяет использовать в системе легко масштабируемые компоненты, что повышает доступность.

Исходя из этого, был выбран многоуровневый архитектурный стиль проектирования.

Одной из разновидностей многоуровневого стиля проектирования является архитектурный паттерн MVT (Model-View-Template), изображенный на рисунке 7.

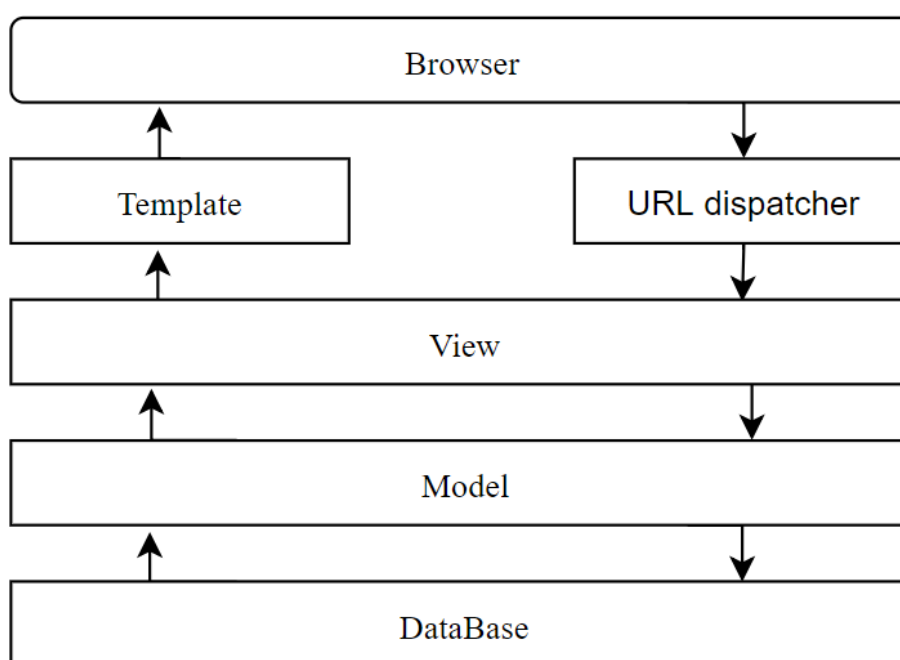


Рисунок 6 — Архитектурный паттерн Model-View-Template

Компонент *URL dispatcher* определяет обращенный к приложению запрос и направляет его к выбранной функции *View*, которая, в свою очередь, интерпретирует запрос согласно своей внутренней логике, а также контролирует передачу данных от пользователя системе и обратно, взаимодействуя для этого с уровнем представления данных *Model*. Результатом обработки запроса является сгенерированный код, предоставляемый конечному пользователю — *Template*.

Ключевой задачей уровня *Model* является описание используемых данных — как правило, все используемые в приложении сущности выделяют в отдельные модели, которые, в свою очередь, описывают структуру и логику каждой сущности.

2.4 Выбор системы управления базой данных

Выбор системы управления базой данных (СУБД) представляет собой сложную многопараметрическую задачу. При попытке создать реляционную модель данных стало очевидным, что данный подход имеет не является оптимальным для отражения данных предметной области — таблицы вынуждены были содержать множество избыточных полей, как, например, параметры, которые необходимо отслеживать при разных заболеваниях. На основании этого было принято решение использовать нереляционную базу данных.

Поскольку при моделировании бизнес-процессов (см. пункт 2.1) было принято, что итогом получения консультации становится запись в электронной медицинской карте, была выбрана документо-ориентированная СУБД MongoDB.

MongoDB — документо-ориентированная СУБД, информация в которой хранится в виде JSON-документов [9].

К ключевым преимуществам использования нереляционной базы данных MongoDB относятся:

1. Возможность использования смешанного подхода — СУБД MongoDB не обязывает использовать ссылочную связность для всех объектов, не предоставляет такую возможность для разработчика;
2. Динамическая структура данных — СУБД MongoDB предоставляет возможность создавать документы, не зная их структуру заранее, что позволяет создавать объекты только с необходимой и достаточной информацией о них;

3. Горизонтальная масштабируемость — возможность при необходимости архивировать данные в виде документов.

Использование нереляционной базы данных требует уточнения следующего аспекта: данные в документо-ориентированных базах данных могут быть дублированы. Хранение одинаковых данных в разных коллекциях снижает затраты времени на доступ к этим данным, так как нужно обратиться только к одному документу вместо формирования запроса по нескольким документам с помощью механизма ссылочной связи. Но также это усложняет операции записи по отношению к этим данным: при обновлении либо удалении данных изменения необходимо внести в каждый документ, который содержит эти данные. Решение о применении дублирования информации принимается в зависимости от частоты выполнения запросов на чтение и запись, а также критичности задачи обеспечения согласованности. Например, информация о запланированном приеме хранится как в документе «Врач», так и в документе «Пациент». Эти данные являются дублирующими, но это оправданно в связи с тем, что отмена либо перенос консультации является куда более редким запросом и в этом случае допустимо снижение производительности, а при нарушении согласованности будет возможно восстановить данные из второго источника. Также необходимо отметить, что несмотря на то, что использование нереляционной базы данных не предполагает обеспечение целостности и связности данных, эти задачи выполняются на уровне приложения и технология ORM.

Таким образом база данных хранит коллекцию из двух документов — «Врач» и «Пациент». Каждый из этих документов хранит произвольную информацию и может содержать вложенные документы как содержимое своих полей. На рисунке 6 представлена схема данной базы. В документе «Доктор» содержится информация о медицинских сотрудниках учреждения — ФИО, занимаемая должность, контакты, образование, стаж работы. Также в нем хранятся документы:

- «Календарь приемов» — содержит информацию о событиях, очных либо заочных приемах;
- «Терапевтические материалы» — данный документ содержит информацию о различных объектах (графических, текстовых), необходимых врачу для работы.

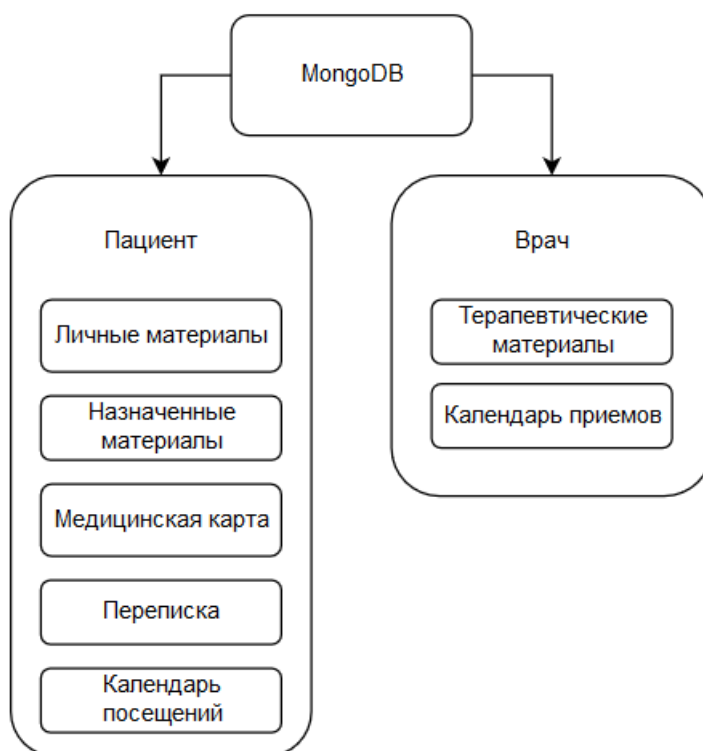


Рисунок 7 — Схема документо-ориентированной базы данных

В документе «Пациент» содержится информация о пациентах, обратившихся в медицинское учреждение — ФИО, пол, возраст, контакты, лечащий врач и т. д. Также в нем хранятся документы:

1. «Переписка» — поскольку пациент может написать только своему врачу, то любое сообщение будет либо «к пациенту», либо «от пациента». В данном документе хранится текст входящих и исходящих сообщений;
2. «Календарь посещений» — содержит информацию о событиях, очных либо заочных приемах, частично дублирует информацию из «Календаря приемов». Каждое событие содержит информацию о дате и времени, в которое оно проводится, формат очный либо заочный и т. д.

3. «Медицинская карта» — документ, содержащий массив документов — электронных медицинских записей. Электронная медицинская запись является итогом приема и содержит информацию о жалобах пациента, рекомендациях врача, комментариях, диагнозе.

4. «Личные материалы» — данный документ содержит информацию о внесении пользователем личных записей, выполнении им терапевтических заданий, назначенных врачом.

5. «Назначенные материалы» — данный документ содержит информацию о назначенных пациенту врачом материалов, которую частично дублирует из «Терапевтических материалов». Дублирование информации в этом случае обусловлено тем, что перед назначением материала пациенту врач может внести индивидуальные правки; также хранение результата работы с материалами также должно храниться в этом документе.

2.5 Среда и инструменты разработки

Язык программирования — Python

Python — высокоуровневый язык программирования общего назначения, обладает такими архитектурными чертами как динамическая типизация, автоматическое управление памятью, механизм обработки исключений, поддержка многопоточных вычислений, высокоуровневые структуры данных [10]. Поддерживается разбиение программ на модули, которые, в свою очередь, могут объединяться в пакеты.

1. Язык, и платформа являются продуктами с открытым исходным кодом, допустимо неограниченно использовать их в веб-приложениях в коммерческих целях.

2. Язык программирования Python — структурирован, имеет пакетную дистрибуцию, что делает возможным использование только необходимых пакетов для решения поставленных задач.

3. Наличие библиотек и готовых приложений на Python позволяет сократить время написания программ в несколько раз.

4. Эталонный интерпретатор CPython имеет совместимость с большинством существующих платформ, код Python выполняется практически под всеми современными операционными системами и используется для широкого круга задач.

5. Относительная простота Python и наличие документации, в том числе и на русском языке обеспечивает успешное проведение обучения, что очень важно для модернизации проекта и дальнейшего его сопровождения.

Платформа — Django Framework

Django Framework 2.0.7 — программная платформа для разработки веб-приложений на языке программирования Python 3.6.1.

Основными преимуществами Django Framework являются:

- Русскоязычная документация, открытый исходный код;
- Встроенная технология ORM;
- Автоматически генерируемая панель администратора;
- Поддержка шаблона MTV (Model-Template-View);
- Высокая скорость работы, имеет встроенные возможности

кэширования и распределения нагрузки.

Среда разработки — repl

Для разработки информационной системы используется интерактивная среда repl.it [11]. Данный сервис позволяет разрабатывать проект непосредственно в браузере, а также предоставляет хостинг и инструменты отладки кода. В основе лежит подход REPL (read-eval-print-loop) — форма организации простой интерактивной среды программирования в рамках средств интерфейса командной строки.

Главными преимуществами использования данного сервиса являются:

1. Удобство проверки и демонстрации — сервис предоставляет хостинг для разрабатываемого проекта,

2. Удобство редактирования — проект доступен для редактирования с любого устройства, существует возможность совместного одновременного редактирования.

3. Удобство тестирования и отладки — изменения кода мгновенно доступны для проверки и тестирования на постоянном адресе, размещенном публично в сети интернет.

4. Возможность контроля версий — встроенная в среду разработки git-подобная система контроля версий, которая обеспечивает возможность просмотреть либо восстановить предыдущее состояние файлов проекта.

5. Надежность — разрабатываемый проект размещен на облачном хостинге сторонней компании, что снижает риск недоступности сервиса по причине инфраструктурных проблем (отключение питания, проблемы с доступом в интернет, и прочие технические проблемы).

Выводы по второй главе

Система была графически описана в виде нескольких иерархических связанных диаграмм в нотации IDEF0. Возможности и поведение системы отражены на диаграмме вариантов использования. Определены действующие лица приложения — пользователи. Установлены типы отношений между вариантами использования. Также была выбрана СУБД и описана модель документо-ориентированной базы данных.

Описана трехзвенная архитектура разрабатываемой информационной системы. Был выбран паттерн проектирования MVP, а также обозначены средства разработки — язык программирования Python, Django Framework, база данных MongoDB.

Задача проектирования информационной системы удаленного консультирования выполнена в полной мере.

3 Реализация прототипа информационной системы

Создание проекта происходит в несколько этапов. Первоначально создаётся чистый Python-проект, в котором разворачивают фреймворк Django. Далее в конфигурационном файле проекта указывается используемая база данных. Фреймворк Django содержит стандартные варианты — MySQL, Oracle, SQLite и PostgreSQL, но, так как было решено использовать нереляционную базу данных, необходимо установить библиотеку, позволяющую фреймворку Django взаимодействовать с выбранной СУБД MongoDB — Django [12]. Данная библиотека находится в открытом доступе и делает возможным работу встроенной технологии Django ORM с нереляционной базой данных MongoDB. Одним из преимуществ использования технологии Django ORM является поддержка миграций. При изменении структуры данных в файле `models.py`, внесение изменений в структуру базы данных можно произвести последовательным запуском двух команд — «`manage.py makemigrations`», которая подготовит миграцию, и «`manage.py migrate`», которая непосредственно применяет миграцию к базе данных.

После того, как созданы базовые настройки проекта — создание файла адресации, создание и регистрация приложений и создания администратора для использования автоматической административной панели платформы Django — начинается разработка непосредственно серверной и клиентской части проекта.

3.1 Разработка прототипа серверной части

В настоящий момент успех реализации во многом зависит от особенностей используемого инструмента. Для решения стандартных задач хорошим тоном считается использование проверенного кода, покрытого тестами. Главная задача разработчика в подобном контексте состоит в

корректной компоновке элементов, создания инфраструктуры и описания алгоритма создания необходимых функций.

Сущности, с которыми работает система, описаны в файле `models.py` методом класса. Класс содержит методы, переменные и их параметры. На рисунке 8 приведен фрагмент кода класса «Врач», в полях которого содержится информация о враче (имя, должность, город и другие), а также два метода — `str`, определяющий, в каком виде при вызове экземпляра класса будет отображаться информация о нем, и `get_absolute_url`, возвращающий ссылку на страницу с более подробным описанием экземпляра класса.

```
class Doctor(models.Model):
    """Model representing an doctor."""
    first_name = models.CharField(max_length=100)
    last_name = models.CharField(max_length=100)
    position = models.CharField(max_length=100, null=True)
    city = models.CharField(max_length=100, null=True)
    phone = models.CharField(max_length=11, null=True)
    summary = models.TextField(max_length=1000, help_text="Enter a brief about doctor", null=True)

    def __str__(self):
        """String for representing the Model object."""
        return '{0}, {1}'.format(self.last_name, self.first_name)

    def get_absolute_url(self):
        """Returns the url to access a particular doctor instance."""
        return reverse('doctor-detail', args=[str(self.id)])
```

Рисунок 8 — Фрагмент кода класса "Врач"

Также платформа Django Framework имеет встроенный способ описания функций `view` в виде классов — `class-based views (CBV)`. Это необходимо в тех случаях, когда существуют часто используемые функции (чаще всего CRUD-операции), и, для оптимизации кода, подобные функции описаны непосредственно в коде фреймворка.

В качестве выходных данных функции необходим какой-либо специфический параметр, такой как тип объекта или имени шаблона. Для увеличения числа передаваемых функции параметров, функцию представляют

в виде класса, в котором основные параметры представлены полями, а составная функция декомпозирована на элементарные функции.

Технология CBV широко применяется при, например, настройке адресации. При переопределении всего несколько простых атрибутов класса становится возможным просто передать эти значения как аргументы метода `as_view`. На рисунке 9 приведен фрагмент кода файла `urls.py`, в котором при помощи представления функции в виде класса `urlpatterns` описана диспетчеризация HTTP-запросов для редактирования информации о пациенте.

```
# Add URLConf to create, update, and delete clients
urlpatterns += [
    path('client/create/', views.ClientCreate.as_view(), name='client_create'),
    path('client/<int:pk>/update/', views.ClientUpdate.as_view(), name='client_update'),
    path('client/<int:pk>/delete/', views.ClientDelete.as_view(), name='client_delete'),
]
```

Рисунок 9 — Фрагмент кода файла `urls.py`

Также значительным преимуществом использования технологии CBV является использование их в качестве родительских классов. Каждый класс CBV можно указать в качестве родительского для пользовательского класса CBV и переопределить в теле класса-потомка значения атрибутов или методы, определив тем самым специфику его работы. На рисунке 10 представлены классы CBV для вывода на страницу информации о врачах и пациентах как списком, так и отдельно каждого экземпляра класса.

Также язык программирования Python имеет специализированную библиотеку для проведения тестирования приложений — `unittest`. Данный модуль определяет тесты, используя подход на основе классов. Так, например, при запуске собственных тестов модуль запускает поиск файлов проекта, имя которых начинается с `test`, автоматически создает набор тестов из этих тестовых случаев и запускает этот набор.

```

from django.views import generic
from .models import Doctor, Client

class ClientListView(generic.ListView):
    """Generic class-based view for a list of client."""
    model = Client
    paginate_by = 10

class ClientDetailView(generic.DetailView):
    """Generic class-based detail view for a client."""
    model = Client

class DoctorListView(generic.ListView):
    """Generic class-based list view for a list of doctor."""
    model = Doctor
    paginate_by = 10

class DoctorDetailView(generic.DetailView):
    """Generic class-based detail view for an doctor."""
    model = Doctor

```

Рисунок 10 — Фрагмент кода файла views.py

3.2 Разработка прототипа клиентской части

Отличительной особенностью динамического генерирования HTML-кода Django является использование встроенного шаблонизатора. В качестве языка для создания шаблонов допустимо использовать как встроенный язык шаблонов Django Template Language (DTL), так и сторонние системы шаблонов, такие как, например, Jinja2. К прочим сторонним системам шаблонов фреймворка Django предоставляет стандартный API загрузки и рендеринга, что делает возможным их использование независимо от используемой серверной логики. Загрузка включает в себя поиск шаблона для заданного идентификатора и его предварительную обработку — компиляцию его представления в память. Рендеринг представляет собой функцию передачи шаблону данных контекста и возврат строки с результатом.

Главной особенностью использования шаблонов является наследование. Каждый следующий шаблон может расширять либо уточнять родительский

шаблон, и, в свою очередь, любому участку шаблона можно присвоить имя и обернуть в блочный тег.

Шаблоны хранятся в директории `templates`, где ключевым файлом является `base.html` — именно этот шаблон является родительским для всех остальных. Структура папки шаблонов проекта представлена на рисунке 11.

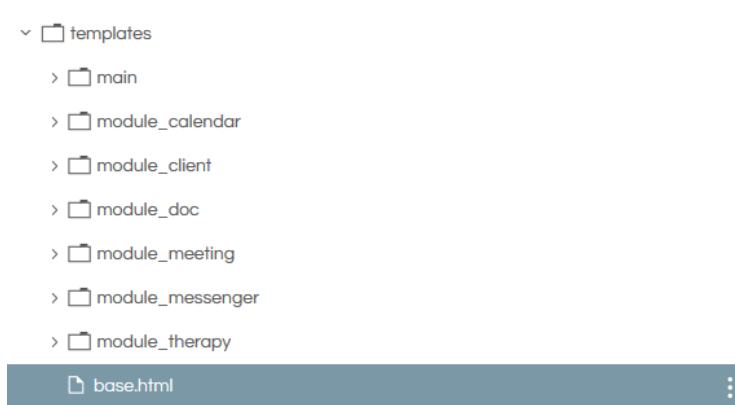


Рисунок 11 — Структура папки шаблонов

Шаблон содержит статические части HTML-кода, а также некоторый специальный синтаксис, описывающий, как будет вставляться динамический контент. В шаблоне `base.html` статическим контентом является верстка «скелета» страницы — `head`, содержащий информацию о используемых кодировках, таблицах стилей, метаданные, необходимых для корректного отображения страниц на различных устройствах и `body`, содержащий информацию о сквозных структурных элементах, которые остаются неизменными на любой из страниц сайта; также описано расположение блоков, заполняемых контентом. Тег `{% block content %}` указывает на место в документе, куда будет подгружаться динамическая информация. Фрагмент кода представлен на рисунке 12.

```
templates/base.html  saved
1  {% load staticfiles %}
2  | | | | <!DOCTYPE html>
3
4  <html lang="en">
5  <head>
6  | | <meta charset="UTF-8">
7  | | <title>Hello Django</title>
8  | | <meta name="viewport" content="width=device-width, initial-scale=1.0">
9  | | <meta name="viewport" content="width=device-width, initial-scale=1"/>
10 | | <link rel="stylesheet" href="{% static "css/all.css" %}">
11 | | <link rel="stylesheet" href="{% static "css/bootstrap.css" %}">
12 | | <link rel="stylesheet" href="{% static "css/style.css" %}">
13 </head>
14 <body>
15 | <div class="navbar navbar-inverse navbar-fixed-top">...
38 </div>
39 | <div class="container">...
58 </div>
59 | {% block content %}{% endblock content %}
60 | | <div class="footer">...
79 </div>
80
81 | <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.mir
82 | <script src="{% static "js/bootstrap.js" %}"></script>
83 </body>
84 </html>
```

Рисунок 12 — Код файла base.html

Папки, расположенные в директории, содержат шаблоны для остальных страниц. Так, например, в папке `module_doc` содержатся шаблоны для отображения личного кабинета врача. При создании файлов первой строкой указывается родительский шаблон с помощью тега `{% extends "base.html" %}`, а содержимое каждого из этих шаблонов также обернуто в тег `{% block content %}`. Также, например, при генерации страницы редактирования личной информации о враче происходит обращение к переменным, содержащимся в переданном в момент загрузки шаблона контексте. Фрагмент кода представлен на рисунке 13.

```
templates/module_doc/edit.html  saving...
1  {% extends "base.html" %}
2
3  {% block content %}
4      <form method="POST">
5          {% csrf_token %}
6          <p>
7              <label>Введите имя</label><br>
8              <input type="text" name="name" value="{{doctor.name_doctor}}" />
9          </p>
10         <p>
11             <label>Введите фамилию</label><br>
12             <input type="text" name="lastname" value="{{doctor.lastname_doctor}}" />
13         </p>
14         <p>
15             <label>Введите отчество</label><br>
16             <input type="text" name="patronimyc" value="{{doctor.patronimyc_doctor}}" />
17         </p>
18         <p>
19             <label>Введите должность</label><br>
20             <input type="text" name="position" value="{{doctor.position_doctor}}" />
21         </p>
22         <p>
23             <label>Введите город</label><br>
24             <input type="text" name="city" value="{{doctor.city_doctor}}" />
25         </p>
26         <p>
27             <label>Введите город</label><br>
28             <input type="text" name="city" value="{{doctor.city_doctor}}" />
29         </p>
30         <input type="submit" value="Сохранить" />
31     </form>
32 {% endblock content %}
```

Рисунок 13 — Код шаблона страницы редактирования личной информации о враче

Помимо специфических тегов, язык шаблонов DTL имеет специальный синтаксис для работы с переменными. При рендеринге шаблона переменные заменяются на свое значение, вычисленное в контексте вызова. Так, например, для выведения списка всех пациентов определенного врача на экран используется код, фрагмент которого представлен на рисунке 14. Значение переменной `{{doctor}}` будет определено из переданного сервером контекста, значение переменной `{{client}}` в свою очередь будет определено в контексте вызова цикла `{% for client in doctor.client_set.all %}`.


```

1  {% extends "doctor_detail.html" %}
2
3  {% block content %}
4
5  <h1>Doctor: {{ doctor }} </h1>
6
7  <h4>Пациенты:</h4>
8
9  {% for client in doctor.client_set.all %}
10 |   <dt><a href="{% url 'client-detail' client.pk %}">{{client}}</a></dt>
11 |   <dd>{{client.summary}}</dd>
12 |   <br>
13 | {% endfor %}
14
15 | {% endblock %}

```

Рисунок 14 — Фрагмент кода шаблона списка пациентов врача

Результат рендеринга данного шаблона представлен на рисунке 15.

Doctor: Домоседов, Григорий

Пациенты:

Самсонова, Екатерина

Большое тревожное расстройство. Предпочтительно связываться в формате переписки. Предполагается продолжительная терапия.

Шапелова, Екатерина

Проживает в Екатеринбурге, предпочтительны удаленные консультации. Расстройство депрессивного спектра. Направление к неврологу.

Белый, Егор

Предпочтительны очные встречи.

Рисунок 15 — Представление списка пациентов врача

3.3 Панель администратора

Одним из главных преимуществ фреймворка Django является автоматически генерируемый интерфейс администратора. Это страница, доступная администратору, и предоставляющая интерфейс для работы с содержимым сайта. Поля для редактирования содержимого сайта создаются с использованием мета-данных моделей.

Чтобы создать инструмент редактирования какой-либо сущности, необходимо зарегистрировать в файле `admin.py` соответствующую сущности

модель и в описании модели указать, какие ее поля доступны для редактирования. Интерфейс администратора представлен в приложении Б.

Выводы по третьей главе

В данной главе описаны структурные нюансы развертывания проекта, такие как регистрация приложений, создание моделей и их регистрация в базе данных, принцип маршрутизации. Предложен алгоритм добавления новых функций, моделей и шаблонов, а также редактирования панели администратора информационной системы.

Также представлен результат работы создания клиентской части информационной системы. Описан механизм работы встроенного инструмента шаблонов Django Template Language, представлен интерфейс пользователя.

ЗАКЛЮЧЕНИЕ

Обзор существующих медицинских информационных систем на предмет возможности непосредственного активного участия пациента в системе позволил выявить пользовательские и функциональные требования, необходимые для полноценной работы информационной системы.

При помощи визуального моделирования были созданы такие представления системы, как диаграмма вариантов использования информационной системы, диаграмма бизнес-процессов клиники, схема документно-ориентированной базы данных.

Для создания прототипа информационной системы удаленного консультирования определены следующие технические средства и инструменты, такие как: язык программирования Python, платформа Django Framework, СУБД MongoDB и компоненты связи с БД.

Представлен результат работы создания прототипа клиентской части информационной системы. Описан механизм работы встроенного инструмента шаблонов Django Template Language, представлены макеты интерфейса пользователя. Также описаны структурные нюансы развертывания проекта, предложен алгоритм добавления новых функций, моделей и шаблонов, а также редактирования панели администратора информационной системы.

Таким образом, полностью выполнены задачи — обзор существующих решений, выявление требований к информационной системе, проектирование информационной системы разработка прототипа информационной системы удаленного консультирования, а цель — достигнута.

Разработанная система может быть расширена и дополнена следующими модулями: онлайн-оплата консультации, видео-чат с врачом, возможность выгрузки медицинских документов в необходимом пользователю формате, а также разработка мобильного приложения.

СПИСОК СОКРАЩЕНИЙ

API — Application programming interface;

CBV — Class based views;

CRUD — Create read update delete;

DB — Data base;

DTL — Django Template Language

IDEF0 — Integration Definition for Function Modeling;

JSON — JavaScript Object Notation;

MVT — Model view template;

ORM — Object relational mapper;

REPL — Read eval print loop;

URL — Uniform Resource Locator;

АИС — Автоматизированная информационная система;

ИС — Информационная система;

МИС — Медицинская информационная система;

СУБД — Система управления баз данных.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Лебедев Г.С., Шадеркин И.А., Фомина И.В., Лисненко А.А., Рябков И.В., Качковский С.В., Мелаев Д.В. Эволюция интернет-технологий в системе здравоохранения // Журнал телемедицины и электронного здравоохранения. 2017. №2 (4).
2. Владзимирский А.В. Первичная телемедицинская консультация «Пациент-врач»: первая систематизация методологии // Журнал телемедицины и электронного здравоохранения. 2017. №2 (4).
3. Бунова Елена Вячеславовна, Буслаева Ольга Станиславовна Оценка эффективности внедрения информационных систем // Вестник АГТУ. Серия: Управление, вычислительная техника и информатика. 2012. №1.
4. Г. И. Назаренко, Я. И. Гулиев, Д. Е. Ермаков Медицинские информационные системы: теория и практика Москва: ФИЗМАТЛИТ, 2005. - 320 с.
5. Роберт, А. Максимчук UML для простых смертных / Роберт А. Максимчук, Эрик Дж. Нейбург. - Москва: СИНТЕГ, 2014. - 272 с.
6. Дворников А. IDEF0 как инструмент моделирования процессов // Авант Партнер. 2005. № 22 (79).
7. Приказ Минздрава России от 11.11.2013 г. № 18-1/1010 «Основные разделы электронной медицинской карты»
8. Эрик Фримен, Элизабет Фримен, Кэтти Сьерра, Берт Бейтс Head First. Паттерны проектирования. Обновленное юбилейное издание Санкт-Петербург: Питер, 2018. - 656 с.
9. Официальный сайт продукта MongoDB [Электронный ресурс] // <https://www.mongodb.com> Компания производитель «MongoDB, Inc.». – Режим доступа: <https://www.mongodb.com/what-is-mongodb/>.
10. Официальный сайт продукта Python [Электронный ресурс] // <https://www.python.org> Компания производитель «Python Software Foundation». – Режим доступа: <https://www.python.org/doc/>.

11. IT-портал [Электронный ресурс] // <https://www.repl.it> Компания производитель «Neoreason, Inc.». – Режим доступа: <https://repl.it/talk/>.

12. IT-портал [Электронный ресурс] // <https://github.io> Компания производитель «GitHub, Inc.». – Режим доступа: <https://github.com/nesdis/djongo>

ПРИЛОЖЕНИЕ А

Макеты

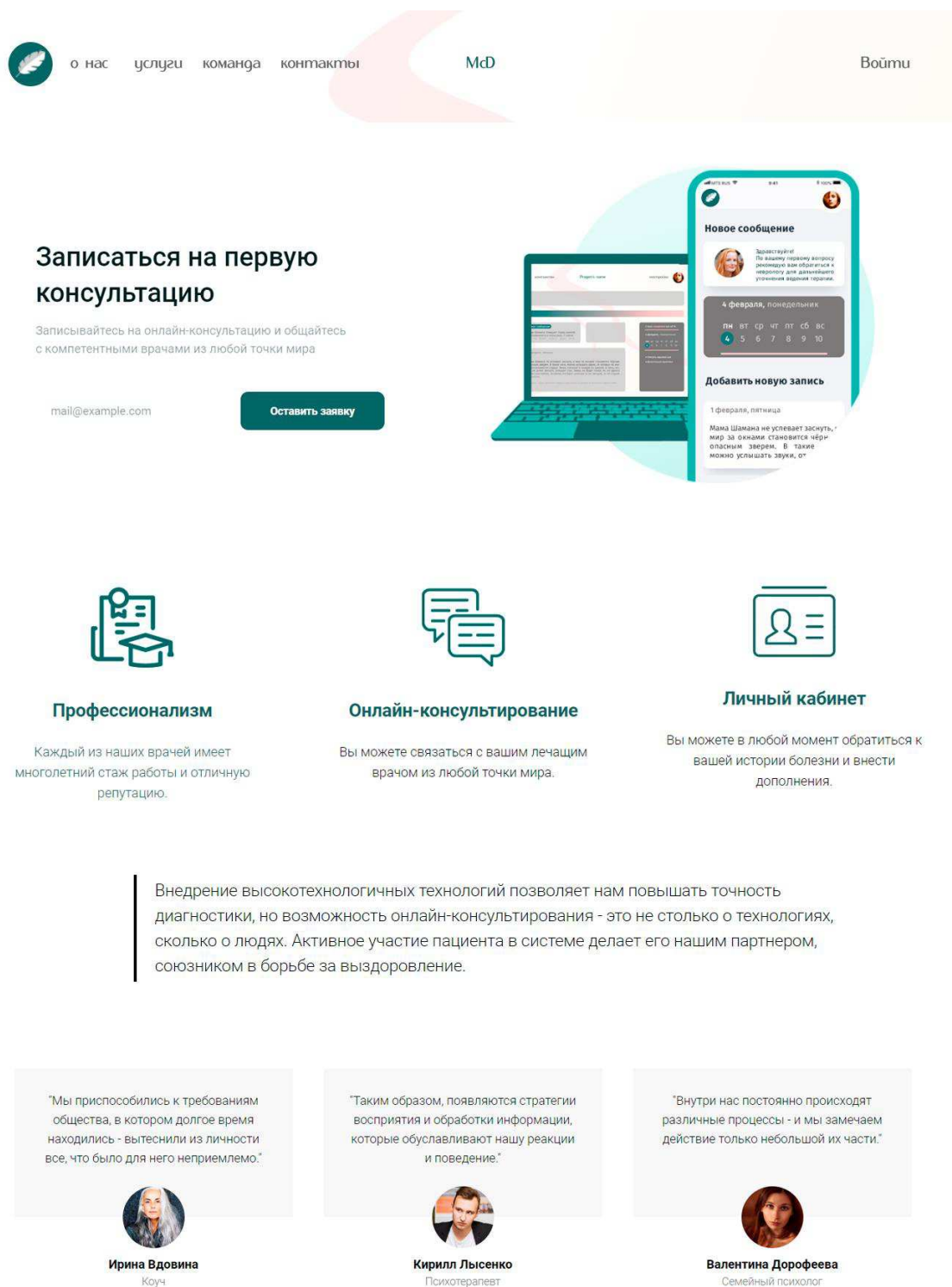


Рисунок А.1 — Макет стартовой страницы

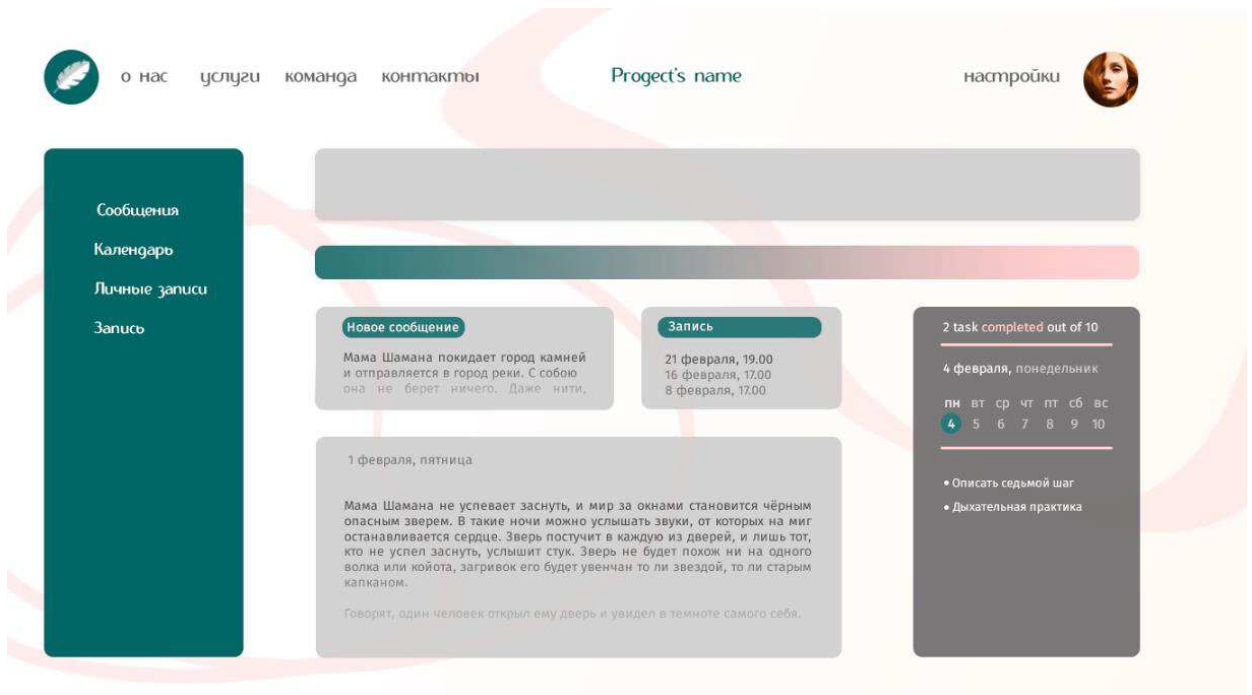


Рисунок А.2 — Макет личного кабинета пациента

ПРИЛОЖЕНИЕ Б

Интерфейс администратора

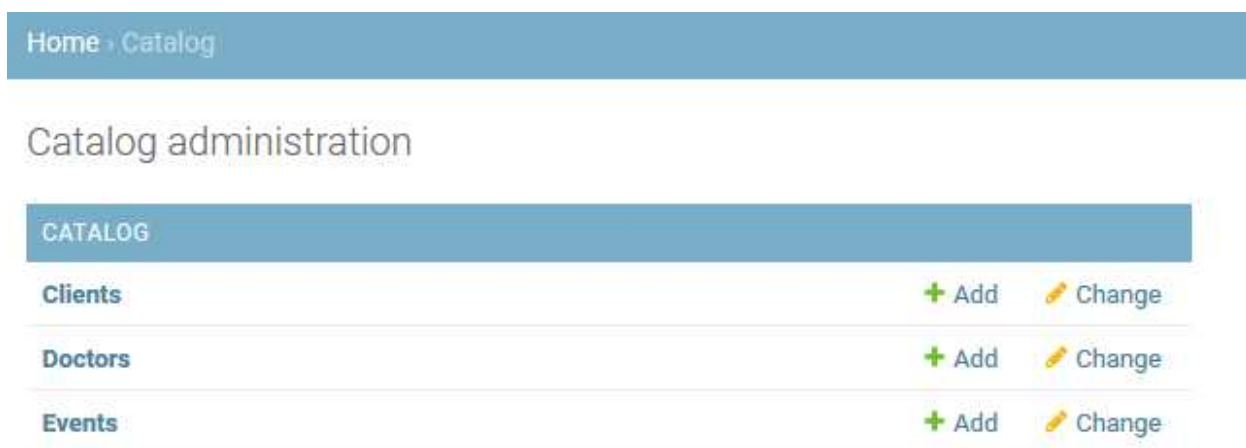


Рисунок Б. 1 — Список редактируемых сущностей

The screenshot shows a 'Change client' form. It has several input fields: 'First name' (Екатерина), 'Last name' (Самсонова), 'Date of birth' (1982-06-11) with a calendar icon and 'Today' link, 'City' (Москва), 'Phone' (98235865724), and 'Doctor' (Домоседов, Григорий) with a dropdown arrow and edit/delete icons. A 'Summary' field contains the text: 'Большое тревожное расстройство. Предпочтительно связываться в формате переписки. Предполагается продолжительная терапия.'

Change client

First name:

Last name:

Date of birth: Today

City:

Phone:

Doctor:

Summary:

Большое тревожное расстройство. Предпочтительно связываться в формате переписки. Предполагается продолжительная терапия.

Рисунок Б. 2 — Форма редактирования информации о пациенте

Change doctor

First name:	<input type="text" value="Кирилл"/>
Last name:	<input type="text" value="Лысенко"/>
Position:	<input type="text" value="Психолог, семейный психотерапевт"/>
City:	<input type="text" value="Москва"/>
Phone:	<input type="text" value="89068068968"/>
Summary:	<div><p>Прошел профессиональную переподготовку в педагогическом институте. Специализируюсь на работе с семейными парами.</p></div>

Рисунок Б. 3 — Форма редактирования информации о враче

Change event











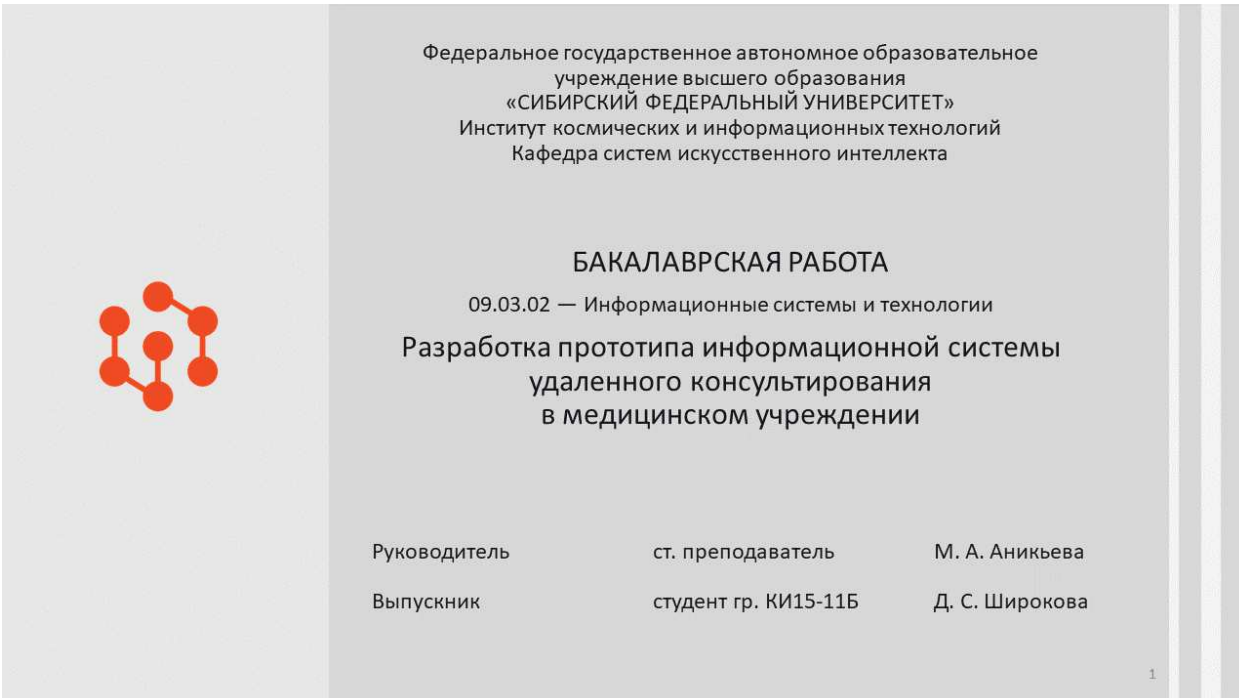
Client:	<input type="text" value="Самсонова, Екатерина"/>   
Doctor:	<input type="text" value="Домоседов, Григорий"/>   
Date:	<input type="text" value="2019-06-25"/> Today 
Time:	<input type="text" value="16:33:32"/> Now 
Note:	<input type="text" value="Первичная консультация."/> <small>Note about event</small>
Status:	<input type="text" value="Успешно"/>  <small>event status</small>
Type:	<input type="text" value="Очная консультация"/>  <small>event type</small>

Рисунок Б. 4 — Форма редактирования информации о событии

ПРИЛОЖЕНИЕ В

Плакаты презентации



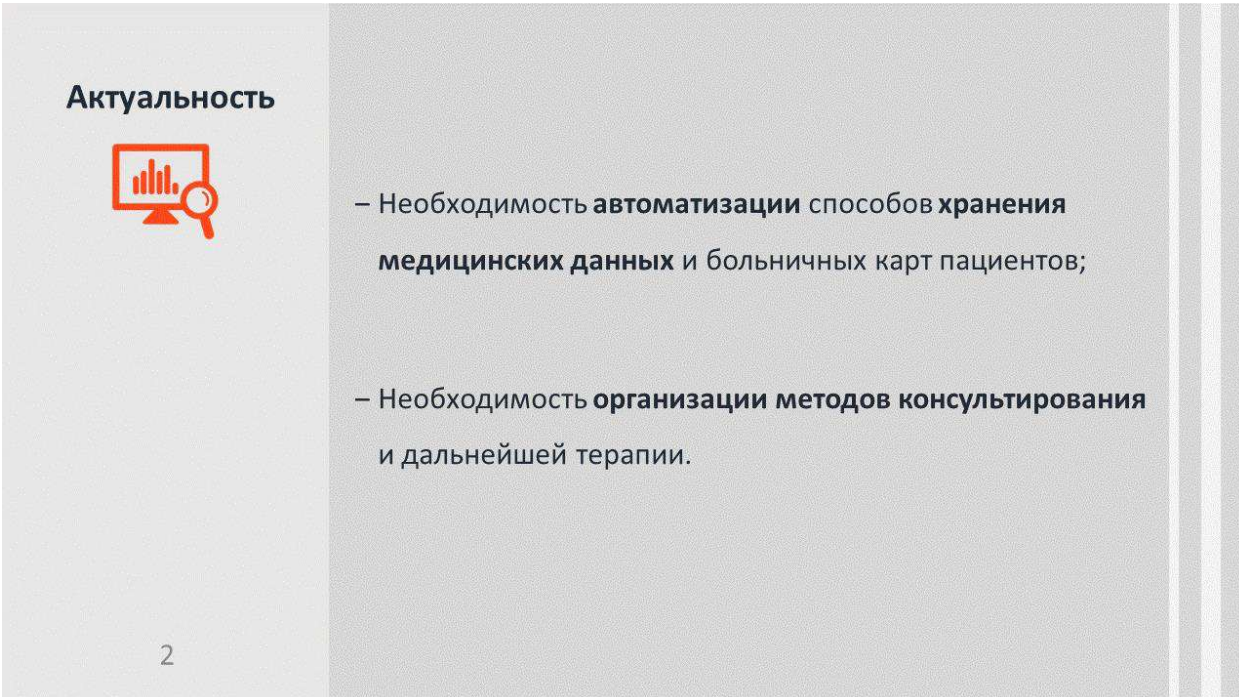
Федеральное государственное автономное образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра систем искусственного интеллекта

БАКАЛАВРСКАЯ РАБОТА
09.03.02 — Информационные системы и технологии
Разработка прототипа информационной системы
удаленного консультирования
в медицинском учреждении


Руководитель ст. преподаватель М. А. Аникьева
Выпускник студент гр. КИ15-11Б Д. С. Широкова

1

Рисунок В. 1 — Слайд презентации №1



Актуальность



– Необходимость **автоматизации** способов хранения **медицинских данных** и больничных карт пациентов;

– Необходимость **организации методов консультирования** и дальнейшей терапии.

2

Рисунок В. 2 — Слайд презентации №2

Цель и задачи



Цель – разработка прототипа информационной системы удаленного консультирования в медицинском учреждении.

Задачи:

- Обзор существующих решений, выявление требований к информационной системе;
- Проектирование информационной системы;
- Разработка прототипа информационной системы.

3

Рисунок В. 3 — Слайд презентации №3

Существующие решения



- Медицинские информационные системы;
- Сервисы онлайн-консультаций;
- АИС Телемедицинского консультирования.

4

Рисунок В. 4 — Слайд презентации №4

Пользовательские требования к ИС



1. Возможность **активного участия пациента** в информационной системе;
2. **Отсутствие посредника** и необходимости **личного** присутствия при обращении пациента в клинику;
3. **Централизованное** хранение медицинских данных.

5

Рисунок В. 5 — Слайд презентации №5

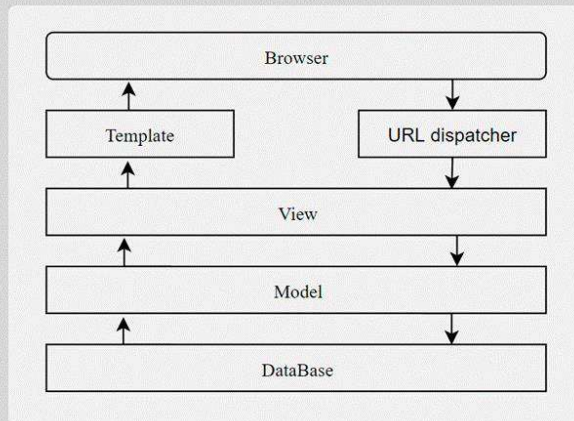
Диаграмма переходов по страницам



6

Рисунок В. 6 — Слайд презентации №6

Паттерн MVТ



7

Рисунок В. 7 — Слайд презентации №7

Документо-ориентированная база данных



8

Рисунок В. 8 — Слайд презентации №8

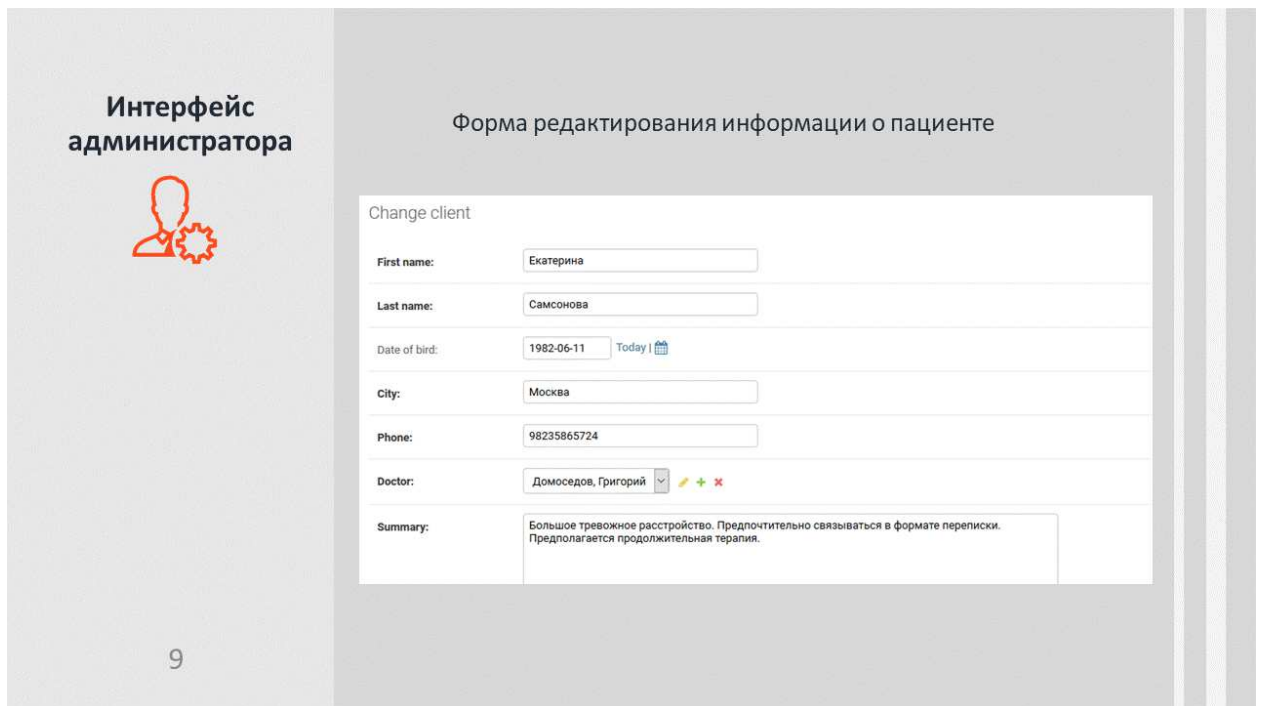


Рисунок В. 9 — Слайд презентации №9

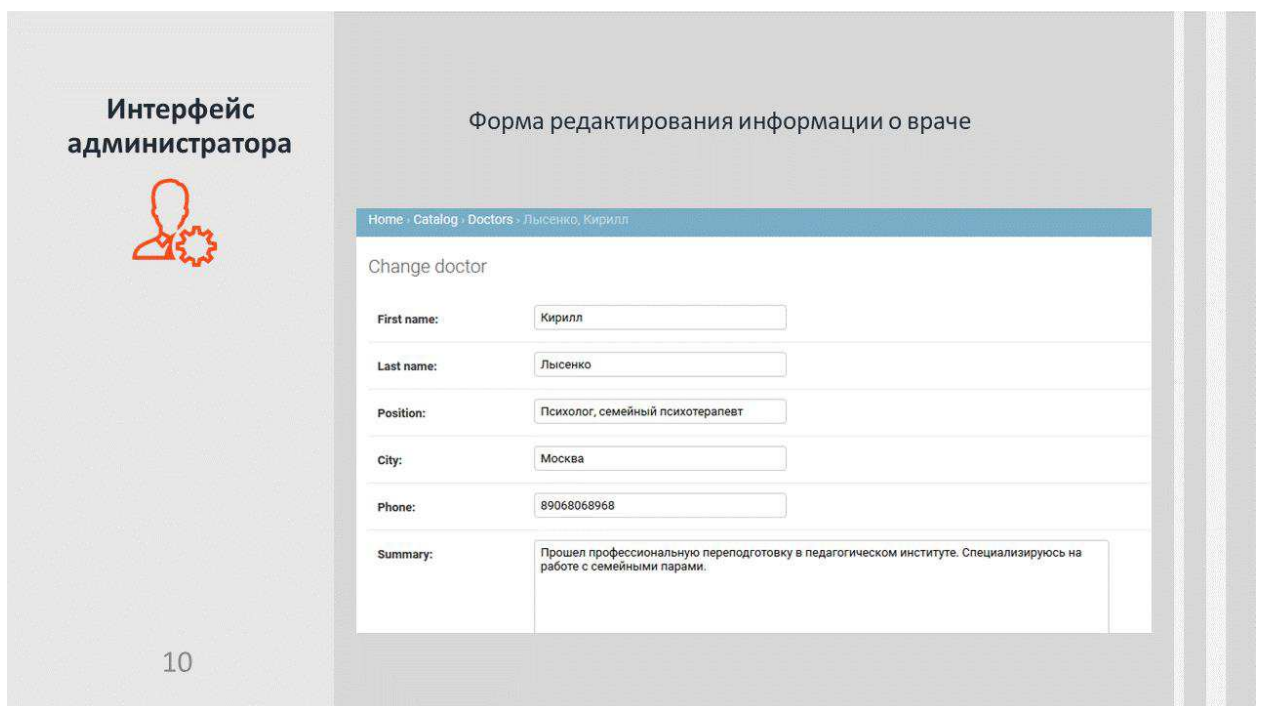
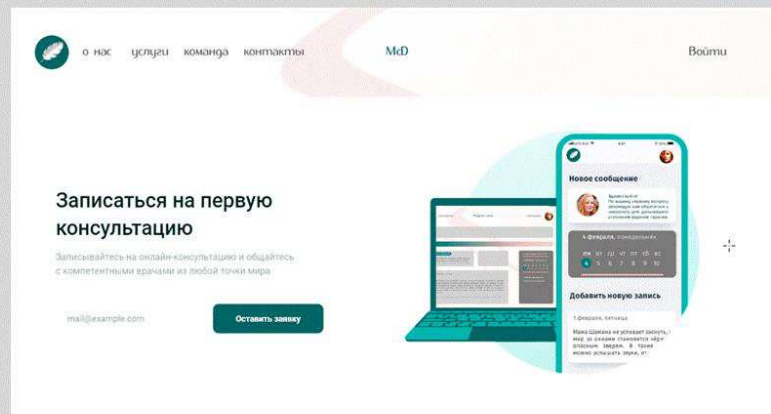


Рисунок В. 10 — Слайд презентации №10

Клиентская часть



Макет стартовой страницы



11

Рисунок В. 11 — Слайд презентации №11

Клиентская часть



Макет личного кабинета пациента



12

Рисунок В. 12 — Слайд презентации №12

Заклучение



13

- Выполнен обзор с сравнение существующих решений;
- Представлены функциональные требования к информационной системе;
- Представлена диаграмма вариантов использования;
- Определены технические средства разработки информационной системы;
- Описана структура базы данных;
- Представлена разработка прототипа информационной системы.

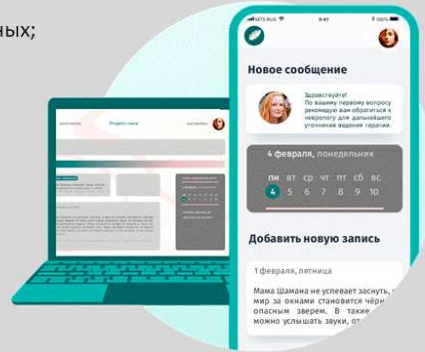


Рисунок В. 13 — Слайд презентации №13

Федеральное государственное автономное образовательное учреждение
высшего образования

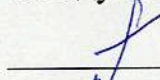
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра систем искусственного интеллекта

УТВЕРЖДАЮ

Заведующий кафедрой


Г. М. Цибульский

подпись

« 28 » июня 2019 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.02 — Информационные системы и технологии

Разработка прототипа информационной системы удаленного
консультирования в медицинском учреждении

Руководитель ст. преп. каф. СИИ


подпись, дата

28.06.19

М. А. Аникьева

Выпускник


подпись, дата

28.06.2019

Д. С. Широкова

Нормоконтролер


подпись, дата

28.06.19

М. А. Аникьева

Красноярск 2019