



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования «Уральский федеральный университет имени
первого Президента России Б. Н. Ельцина» (УрФУ)
Институт радиоэлектроники и информационных технологий – РТФ

ОТЧЕТ
о проектной работе
Разработка системы помощи принятия решений при управлении
подразделением клиники

по дисциплине: Проектный практикум

Студенты команды RedRock

Елькин С.А.
Линьков А.С.
Курочкина Д.Н.
Сумарев Е.Л.
Цибин Н.Е.

Екатеринбург 2025

СОДЕРЖАНИЕ

Введение.....	3
1 Команда.....	4
2 Определение проблемы.....	5
3 Анализ аналогов.....	6
4 Подходы к решению проблемы.....	8
5 Требования к MVP.....	9
6 Планирование работ.....	11
7 Отчет о работе команды.....	12
8 Отчет о работе тимлида.....	14
9 Отчет о работе аналитика.....	16
10 Отчет о работе фронтенд-разработчика.....	18
11 Отчет о работе дизайнера.....	20
12 Отчет о работе бекенд-разработчика.....	22
Заключение.....	24

ВВЕДЕНИЕ

По данным Росстата, количество частных клиник растёт с каждым годом, появляются новые сети филиалов, что, в свою очередь, усложняет процессы управления и организации работы. Пациенты ценят удобство цифровых сервисов, а работники всё активнее используют средства автоматизации труда. Однако сейчас зарубежные сервисы, которые удовлетворяли эти запросы, уходят с российского рынка, что создаёт дополнительные трудности для клиник. Ввиду этих причин, команда поставила себе цель создать сервис для автоматизации процессов управления персоналом и приемами в медицинских клиниках.

В рамках этого семестра команда планирует доработать систему фильтрации с прошлого семестра и реализовать функционал страницы задач, которая позволит администраторам и врачам отслеживать изменение статусов приема и не забывать про актуальные задачи. В планах создание страницы конфигурации клиники, где директор сможет указать сотрудников и выдать им полномочия в системе. Для самостоятельной записи пациента на прием команда планирует создать встраиваемую форму, которая будет размещаться на сайте клиники и направлять данные в систему

Наш продукт ориентирован на работу с медицинскими клиниками, и в будущем может быть использован как в сети клиник, так и в отдельном филиале. Так как область применения ограничена медицинской сферой, приложение может учитывать особенности, такие как специализация кабинетов, ведение отчетности или особенности трудового законодательства для медицинских работников.

После создания продукта команда планирует протестировать его в знакомой клинике, которая как раз сталкивается с решаемой проблемой.

1 Команда

- Елькин Степан Андреевич РИ-320945 - тимлид
- Курочкина Дарья Николаевна РИ-320947 - аналитик
- Линьков Алексей Сергеевич РИ-320935 – frontend- разработчик
- Сумарев Евгений Леонидович РИ-320910 - дизайнер
- Цибин Никита Евгеньевич РИ-320932 – backend-разработчик

2 Определение проблемы

Целевой аудиторией являются владельцы клиник и руководители филиалов, которые могут быть заинтересованы в том, чтобы внедрить наш продукт. Он поможет повысить эффективность работы клиники и оптимизировать затраты.

К пользователям относятся владельцы и руководители, пациенты, которые будут записываться на приём, администраторы, отвечающие за организацию записи пациентов, и врачи, которые смогут отслеживать своё расписание с помощью нашего сервиса.

Сейчас пользователи сталкиваются с рядом трудностей:

- В настоящее время для решения проблем в этой сфере клиники либо используют CRM-системы, которые не предназначены для медицинского сектора, либо работают с данными вручную, например, в таблицах Excel.
- Для выполнения основных задач применяется сразу несколько сервисов с ограниченным функционалом, что требует постоянного ручного переноса данных из одного сервиса в другой. Поскольку не все сервисы являются облачными, доступ к ним может быть не всегда удобным.
- Зарубежные сервисы, которые использовались ранее, постепенно становятся недоступными в России. Клиники начинают испытывать трудности с доступом в систему.
- Сложность интерфейса требует дополнительного времени на обучение администраторов, что замедляет обслуживание и может привести к ошибкам при заполнении карточки приема.
- Высокие цены при использовании в маленьких клиниках или филиалах.

3 Анализ аналогов

На данный момент в интернете можно найти CRM системы, предлагающие решения для клиник. Эти сервисы решают ту же проблему, что и наш проект, поэтому они были рассмотрены в качестве конкурентов для выявления их преимуществ и недостатков. Было проанализировано 12 сервисов, результат сравнения был представлен в виде таблицы. Рассмотрим основных представителей.

DocaCRM

(Оценка приложения: 5)

Преимущества: Пробный период, облачная реализация, большой функционал, сопровождение при внедрении, наличие обучения, понятный интерфейс.

Недостатки: Цена для малых клиник может быть высокой, нет бесплатной версии, нет автоматического составления расписания врачей

Medesk

(Оценка приложения: 4.2)

Преимущества: Простота использования, хорошая аналитика, интеграции с сервисами онлайн-записи, облачная реализация.

Недостатки: Цена для малых клиник может быть высокой, каждая опция продается отдельно, неудобный интерфейс, пользователи отмечают ошибки при работе, сырая программа.

Youclients

(Оценка приложения: 4.3)

Преимущества: Облачная реализация, наличие онлайн-записи, программы лояльности для клиентов и онлайн-предоплата

Недостатки: Ошибки в автоматических отчетах, ограничения по использованию без стабильного интернета

Renovatio

(Оценка приложения: 4.5)

Преимущества: Продвинутая аналитика, поддержка мобильного приложения, большой функционал

Недостатки: Из-за обширного функционала внедрение и адаптация системы могут занять время и потребовать дополнительного обучения персонала.

В результате анализа конкурентных приложений мы выявили их основные минусы, влияющие на мнение целевой аудитории:

- Ограниченная автоматизация работы
- Высокая стоимость для внедрения в небольшие клиники
- Сложность интерфейса для освоения рядовыми работниками клиники

Были отмечены и преимущества, которые команда учитывала при проектировании и разработке своего решения:

- Облачная реализация обеспечивает доступ без скачивания приложения
- Интеграция с сервисами онлайн-записи

4 Подходы к решению проблемы

Учитывая приведенные проблемы и достоинства/недостатки аналогов при пришли к нескольким важным решениям для нашего продукта:

- Приложение должно быть ориентировано на работу с клиниками и должно хранить все необходимые данные пациентов и врачей в единой системе.
- Приложение должно быть облачным, чтобы обеспечить доступ к нему без установки с любого устройства, и для реализации интеграции с сервисами онлайн-записи в будущем.
- Интерфейс должен быть понятным для пользователей без дополнительного обучения.
- Приложение должно облегчать работу сотрудников за счет автоматизации рутинных процессов, и уменьшать количество возможных ошибок благодаря автоматическому отслеживанию приемов.

5 Требования к MVP

На основе проведенной аналитики и исходя из представлений команды были предложены требования к продукту на этот семестр:

- 1) Страница задач: На одной из страниц приложения администраторы и врачи смогут управлять своими задачами и получать уведомления.
- 2) Уведомление о приемах: Для напоминания о ближайших приемах на страницу задач врачу и администратору должны приходить уведомления.
- 3) Встраиваемая форма для записи на прием: Должна быть возможность добавить форму на сайт клиники для отправки запроса записи на прием.
- 4) Фильтрация приёмов в календаре: Приёмы в календаре должны быть фильтруемыми по нескольким параметрам для удобства поиска.
- 5) Настройки параметров клиники: На одной из страниц приложения должна быть возможность указать сотрудников и выдать им права в системе

Ожидаемые результаты:

После реализации этих требований, администратор сможет более эффективно управлять записями пациентов на приём, а также быстро находить необходимые приёмы в календаре благодаря фильтрации по различным параметрам. Уведомления о ближайших приёмах помогут администратору быть в курсе расписания и своевременно подготавливаться к приёму пациентов. Задачи позволят вовремя уведомлять пациентов, подтверждать или отменять приемы.

Небольшие клиники смогут без лишних затрат разместить форму для записи на прием на своей странице в интернете.

Такой функционал позволит использовать наш сервис уже сейчас, так как он будет уже удобнее ведения таблиц Excel и бумажных журналов. Небольшие клиники смогут цифровизировать ведение записей о приемах и облегчить работу сотрудников.

6 Планирование работ

Исходя из поставленной цели, перед командой были поставлены следующие задачи:

- 6) Распределить роли в команде на основе компетенций каждого участника.
- 7) Изучить аналоги, целевую аудиторию и собрать требования к продукту на основе полученных данных.
- 8) Составить календарный план и распределить зоны ответственности.
- 9) Разработать MVP опираясь на поставленные требования.
- 10) Провести тестирование MVP и обсудить результаты работы.

Работа в команде была организована по системе Waterfall. Каждую неделю проводились встречи для обсуждения результатов работы. Задачи каждого участника отмечались на доске Yougile, где их можно удобно распределять по статусам: «планируется», «в работе», «ревью» и «готово».

Для отслеживания прогресса параллельно заполнялась диаграмма Ганта, куда заносились выполненные задачи каждой недели. В ходе работы эта диаграмма позволяла оценить трудозатраты отдельных участников и продвижение проекта в целом.

7 Отчет о работе команды

В рамках первой контрольной точки команда занималась планированием работ и проектированием решения. Была организована встреча для обсуждения функционала для дальнейшей реализации и общего курса проекта. Решено было реализовать в первую очередь страницу задач для администратора и врачей. После того как роли были распределены, тимлид и аналитик составили техническое задание для разработчиков с подробным описанием работы функционала, интерфейса и кейсов использования. Часть информации об организации работы клиники команда получила от знакомой клиники Екатеринбурга, которая сталкивается с изучаемыми проблемами и может помочь в тестировании продукта в будущем. Фронтенд разработчик и дизайнер создали лендинг-страницу для продвижения продукта в будущем. В данный момент она информирует о функциональных возможностях продукта и позволяет связаться с командой для сотрудничества.

В рамках второй контрольной точки шла разработка минимального жизнеспособного продукта. Работа аналитика была выполнена и лишь дополнялась на основании обратной связи от разработчиков. Дизайнер приступил к созданию макетов основных страниц и форм. Были созданы макеты страницы задач и настроек, модальных окон для фильтрации приемов. Перед началом реализации разработчикам удалось созвониться с представителями сервиса DocaCRM, который решает ту же проблему, что и наш проект. Команда уточнила, как лучше организовать базу данных, при условии большого количества записей, и детали разработки интерфейса. Ко второй контрольной точке фронтенд разработчик сверстал основную страницу задач и интерфейс добавления и удаления уведомлений в списке, а также дооботал интерфейс фильтрации. Бекенд разработчик создал архитектуру базы данных и разместил её на сервере, в данный момент она

хранит записи о пациентах , врачах и активных приемах , но в будущем должна расширяться карточкой посещений пациента , кабинетами и сотрудниками. Запущен интерфейс для управления данными сервиса. В админ-панели предусмотрены разделы для работы с пациентами, событиями, настройками и расписанием. Реализована фильтрация и поиск для удобства работы с большими объемами данных.

В ходе третьей контрольной точки аналитик и дизайнер начали работу над материалами для следующего семестра, так как команда планирует продолжать проект в будущем. Фронтенд разработчик вносил изменения в страницу календаря, так как обнаружились ошибки отображения при некоторых ситуациях использования, а так же дорабатывал фильтры приемов , с которыми возникли проблемы из-за большой нагрузки на единственного фронтендера. Бекенд разработчик протестировал возможность перехода логики продукта на язык 1С, который имеет готовые функции и модули для работы в медицинской сфере и ведения документации , но из-за необходимости дополнительного обучения и сжатых сроков от этой идеи пока отказались.

Были подведены итоги работы в этом семестре. Разработанное решение позволяет вести запись пациентов на приемы, отслеживать их и искать в календаре, администратор и врач могут получать уведомления и задачи. Поставленные задачи были выполнены, кроме страницы настроек на которую не хватило времени у фронтенд разработчика. Возникли сложности при разработке интерфейса, так как один фронтенд разработчик с трудом справлялся с нагрузкой, поэтому команда рассмотрит возможность приглашения ещё одного разработчика. В других вопросах команда следовала составленному плану, который доказал свою эффективность, поэтому система планирования в команде останется такой и в будущем. Мы остались довольны сделанной работой и надеемся, что продукт окажется полезен и востребован.

8 Отчет о работе тимлида

Перед началом работы над проектом была организована встреча для обсуждения проблемы проекта и компетенций участников команды. По результатам встречи был составлен план работы над проектом, позволивший организовать параллельную разработку аналитики, макетов и самого продукта. Тимлидом были созданы командные хранилища материалов, доски для постановки задач, было определено время для еженедельных собраний.

В рамках первой контрольной точки тимлид вместе с аналитиком занимался составлением технического задания для разработчиков и дизайнера. Оно позволило разработчикам создавать приложение по пунктам плана с подробным описанием нюансов работы, что снизило количество уточнений от разработчиков и ускорило процесс создания. Разработчики дали положительную обратную связь после использования технического задания, поэтому команда будет использовать его и дальше.

После начала второго этапа работы тимлид понял, что поставленная ранее задача слишком велика для команды, поэтому на очередном собрании вновь обсудили свои возможности и требования к минимальному жизнеспособному продукту были скорректированы до реализации записи пациента и поиска приемов. В будущем оказалось, что решение было правильным, так как единственный фронтенд разработчик не справился бы с большей нагрузкой, чем та, которая была предложена тимлидом.

На третьем этапе тимлид начал подводить итоги проекта. Были выявлены направления, в которых проект отстает от первоначальных планов, тимлид обратил на это внимание разработчиков. Дополнительно с аналитиком и дизайнером была проведена подготовка работы на следующий семестр. В конце семестра тимлид занимался составлением отчетных документов команды, сбором артефактов и командных хранилищ материалов.

Тимлид считает свою работу выполненной в полном объеме, так как работа над проектом прошла по плану и критических трудностей не возникало. Для продолжения проекта тимлид рассмотрит возможность приглашения в команду второго фронтенд разработчика.

9 Отчет о работе аналитика

За прошедший семестр аналитиком была проделана работа по улучшению и формулированию требований к проекту. Основные задачи включали анализ требований, написание технического задания, а также разработку спецификаций.

Для работы были использованы следующие инструменты:

- Доска Miro
- Google Таблицы
- Google Документы

Благодаря конкурентному анализу аналитику удалось определить конкурентную среду, изучить сильные и слабые стороны других сервисов, выявить рыночные тренды и разработать рекомендации для повышения конкурентоспособности нашего продукта.

В предыдущем семестре были рассмотрены 12 различных CRM-систем, анализ проводился по 20 параметрам. В данном семестре аналитиком проверена актуальность собранных данных, а также обновлены основные выводы о конкурентах.

Целью составления UseCase было формализовать сценарии взаимодействия пользователей с системой, определить требования к функциональности и обеспечить прозрачность и структурированность процессов.

Аналитиком были отредактированы основные, альтернативные и крайние случаи определённых сценариев работы системы. Изменения обусловлены корректировкой тенденции развития проекта и обновлением основных требований к системе.

Аналитик пересмотрел работу и содержание всех основных разделов сервиса, это помогло создать более четкую и понятную структуру продукта,

формализовать ключевые функциональные модули и обеспечить команду разработки и дизайна необходимой информацией для реализации системы.

Основываясь на требованиях к системе, аналитиком было составлено техническое задание на разработку страницы задач. Описаны все ключевые функции страницы, их логика работы, а также ограничения.

Техническое задание — это ключевой документ, который формализует требования к системе, описывает её функционал, архитектуру и критерии приемки. Оно служит основой для разработки, тестирования и контроля качества продукта.

Планы на дальнейшую работу:

Составление ТЗ на разработку дополнительных страниц системы

Сопровождение команды разработки.

Проведение ревью разработанных функциональных модулей.

Провести тестирование системы с участием конечных пользователей для выявления скрытых проблем.

Постепенное расширение функционала.

Сбор обратной связи от пользователей для последующих улучшений.

10 Отчет о работе фронтенд-разработчика

В рамках текущей итерации основной задачей фронтенд-разработчика было улучшение архитектуры и реализация новых пользовательских сценариев, связанных с функциональностью напоминаний.

Для реализации задач использовался следующий стек технологий:

- React – для построения интерфейса.
- TypeScript – для строгой типизации.
- MobX – для управления состоянием приложения.
- swagger-typescript-api – для генерации контрактов API.
- Playwright и Jest – для тестирования.
- GitHub Actions – для автоматизации процессов проверки кода.
- Husky – для автоматической проверки eslint перед коммитами.
- Vite – для сборки проекта.
- Архитектура FSD (Feature-Sliced Design) – для структурирования приложения.

Прежде всего, был выполнен масштабный рефакторинг существующего кода. Весь фронтенд был приведён к архитектуре FSD (Feature-Sliced Design), что позволило структурировать проект, упростить навигацию по коду и обеспечить более удобную масштабируемость в дальнейшем. Это также повысило читаемость и позволило упростить внесение изменений без риска затронуть другие части приложения.

Кроме того, был реализован роутинг, что стало важным шагом к построению полноценного SPA. Теперь переходы между страницами осуществляются без полной перезагрузки, что улучшает пользовательский опыт и делает интерфейс более отзывчивым.

Одной из ключевых задач текущей итерации стала разработка страницы с напоминаниями. Была реализована полноценная страница, на

которой пользователь может просматривать напоминания, фильтровать их по различным параметрам, а также работать с ними через новые модальные окна, которые позволяют создавать, редактировать и удалять напоминания в удобной форме.

В рамках работы над страницей напоминаний были добавлены фильтры, позволяющие пользователям быстро находить нужные напоминания по заданным критериям. Это значительно повысило удобство взаимодействия с системой.

Также была выполнена интеграция с веб-сокетами и сервис-воркерами. Это позволило организовать работу в реальном времени — данные обновляются без необходимости ручной перезагрузки страницы, что особенно важно для работы с уведомлениями и изменениями, происходящими на сервере.

В целом, итерация прошла продуктивно. Удалось не только улучшить структуру проекта, но и добавить важный пользовательский функционал, повысив стабильность и интерактивность интерфейса.

11 Отчет о работе дизайнера

В рамках работы над проектом CRM-системы для эффективного управления клиникой UI/UX-дизайнером была проведена разработка ключевых экранов интерфейса системы.

За прошедший семестр дизайнером была проведена работа по развитию ключевых модулей системы. Основное внимание уделялось созданию интерфейсов для управления персоналом и рабочими задачами, а также оптимизации проектных материалов. Работа строилась на основе ранее проведенного анализа требований и бенчмаркинга.

Спроектированы интерфейсы для управления рабочими задачами медицинского персонала с учетом специфики ролей. Создана система организации задач с различными представлениями для врачей и администраторов. Обеспечена возможность назначения, отслеживания и контроля выполнения задач. Реализована система приоритетов и категоризации задач в зависимости от должностных обязанностей пользователей.

Проведен анализ пользовательских флоу конкурирующих решений в области управления задачами и персоналом. Изучены подходы к организации управления задачами в медицинских CRM-системах. Проанализированы лучшие практики в области управления персоналом. Выявлены оптимальные паттерны взаимодействия для данного типа систем. Определены лучшие практики и недостатки существующих решений.

Выполнена работа по оптимизации файла фигмы и используемой дизайн-системы в целом. Исключены неиспользуемые компоненты и стили из дизайн-системы.

Следующим этапом развития системы станет создание интерфейса для самостоятельной записи пациентов на прием. Планируется разработать удобную и интуитивно понятную форму, которая позволит пациентам:

- Самостоятельно выбирать врача и удобное время приема
- Указывать тип необходимой медицинской услуги
- Заполнять контактную информацию
- Получать подтверждение записи

Данный функционал значительно снизит нагрузку на администраторов клиники и повысит удобство для пациентов, обеспечив возможность записи в любое время.

12 Отчет о работе бекенд-разработчика

За прошедший период была выполнена разработка и развертывание серверной части приложения с использованием фреймворка Django. Работа включала проектирование архитектуры приложения, реализацию основных функций и размещение проекта на хостинге.

Используемые технологии:

- Язык программирования: Python
- Фреймворк: Django
- Средства контейнеризации: Docker и Docker Compose
- СУБД: PostgreSQL
- Хостинг: [Cloud.ru](https://cloud.ru)
- Веб-сервер: Nginx

Основные достижения:

Разработан API для взаимодействия с календарем, позволяющий просматривать расписание с запланированными приемами и фильтровать приемы по разным параметрам (дата, врач, кабинет, статус). Внедрена функциональность отслеживания статусов приема. Каждый прием может автоматически менять статус (запланирован, в процессе, завершен)

Добавлена возможность регистрации клиники в системе, что позволяет настроить её для эффективного управления. Реализована функция добавления врачей, кабинетов и их характеристик для учета при составлении расписания. Создан функционал добавления информации о врачах, включая их контактные данные для упрощения записи и дальнейшего взаимодействия.

Встроена система уведомлений и задач, которые появляются на странице задач в зависимости от изменения статуса приема и внутренних событий.

Серверная часть размещена на платформе Cloud.ru. Настроены Docker и PostgreSQL для стабильной работы и изоляции окружения. Настроен веб-сервер Nginx для обработки запросов и шифрования трафика через HTTPS.

Настроены процессы CI/CD для автоматизации тестирования, сборки и развертывания приложения. Это позволяет ускорить процесс разработки, минимизировать ошибки при развертывании и обеспечить непрерывную доставку обновлений. Тесты автоматически прогоняются через интеграцию с GitHub, обеспечивая стабильность и качество кода перед его деплоем.

Планы на будущее:

- Усовершенствование фильтров в календаре для более точного поиска записей.
- Автоматизация дополнительных процессов, включая учет изменений в статусе врачей.
- Оптимизация API для улучшения производительности при высокой нагрузке.
- Работа над проектом принесла значительный опыт в использовании Django, PostgreSQL и интеграции систем.

ЗАКЛЮЧЕНИЕ

Рассмотренные вопросы в рамках данной проектной работы, продолжают развитие проекта, начатого в прошлом семестре. Приведены определенные правила и принципы, которые позволят в дальнейшем правильно сопровождать, поддерживать, и грамотно разрабатывать новый функционал для данной разработки.

Основные технические решения, которые были предложены в данной проектной работе, включают в себя следующие аспекты:

- Python, FastAPI, Docker, Nginx, PostgreSQL, React, GitHub – для разработки и хранения кода.
- Playwrite и Jest для тестирования кода. Swagger-typescript-api - для генераций контрактов.
- Github actions - для автоматического запуска тестов на Github.
- Miro – для создания доски задач, пользовательских сценариев и историй.
- Figma – для разработки макетов и дизайна приложения

В итоге работы в течение семестра представлен MVP разрабатываемого продукта, позволяющий работникам:

- 1) Отслеживать актуальные задачи в системе.
- 2) Оперировать задачами работника
- 3) Принимать заявки на прием от пациентов через форму
- 4) Получать уведомления о ближайших приемах
- 5) Указать перечень работников и выдать им права в системе

Это дополняет достижения прошлого семестра:

- 1) Просматривать на странице в календаре запланированные приемы
- 2) Заполнять информацию о пациенте при создании нового приема
- 3) Отслеживать статус приема с помощью цветовой индикации
- 4) Искать прием с помощью изменения интервала отображаемых дат

5) Искать прием с помощью фильтров

К недостаткам можно отнести то, что от части первоначальных идей команда отказалась из-за нехватки времени. Большим плюсом будет тестирование продукта в настоящей клинике в следующем семестре.

Команда осталась довольна разработанным продуктом, сам процесс создания прошел по плану, благодаря чему и получилось выполнить все основные задачи. Команда планирует продолжать работу над проектом из-за его актуальности в связи с уходом зарубежных сервисов. Надеемся, что внедрение нашей системы управления клиникой в будущем позволит повысить эффективность работы, улучшить качество обслуживания и оптимизировать ресурсы.