

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Уральский федеральный университет
имени первого Президента России Б.Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РТФ
Школа бакалавриата

ОТЧЕТ

По проекту
«Разработка цифрового помощника для адаптации студентов»
по дисциплине «Проектный практикум»

Заказчик: Куклин Илья Эдуардович

Куратор: Хрушков Артем Евгеньевич
ученая степень, ученое звание, должность

Студенты команды _____

Исаков Владимир Александрович

Крохалев Дмитрий Константинович

Прошина Евгения Васильевна

Худоерко Богдан Романович

Шукшин Иван Дмитриевич

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Основная часть	5
1.1 Участники проекта и их вклад.....	5
1.1.1 Работа тимлида.....	5
1.1.2 Работа аналитика.....	5
1.1.3 Работы backend-разработчик	10
1.1.4 Работа frontend разработчика.....	14
1.1.5 Работа дизайнера.....	16
1.2 Требования заказчика и пользователей. План действий.....	21
1.2.1 Требования заказчика	21
1.2.2 Требования пользователей	22
1.2.3 Backlog проекта	22
1.3 Анализ и сопоставление аналогов.....	23
1.4 Методология разработки и процесс реализации.....	23
1.5 Промежуточное тестирование и выявленные ошибки.....	24
1.6 Планирование деятельности и распределение задач.....	25
1.6.1 Подход к планированию.....	25
1.6.2 Пример распределения задач по направлениям:	25
1.6.3 Используемые инструменты планирования	26
1.6.4 Результаты планирования	27
2 ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30

ВВЕДЕНИЕ

Целью проекта является разработка цифрового помощника (цифрового тьютора) для адаптации студентов Уральского федерального университета (УРФУ), который упрощает ориентацию в образовательной среде. Веб-приложение должно предоставлять студентам доступ к персонализированной информации о расписании, учебных материалах, расположении корпусов, а также предлагать ответы на частые вопросы и рекомендации по адаптации.

Основные задачи проекта:

- Разработка чат-бота на основе искусственного интеллекта для диалога с пользователями;
- Создание функции персонального расписания с синхронизацией университетского графика;
- Интеграция навигации по ресурсам УРФУ: курсы, преподаватели, аудитории, сервисы;
- Формирование базы адаптационных рекомендаций: советы по учёбе, тайм-менеджменту, социальной интеграции;
- Реализация административной панели для управления пользователями и контентом;
- Разработка аналитических инструментов для оценки эффективности работы помощника.

Актуальность и важность проекта

Адаптация студентов — один из ключевых факторов успешного старта в университете. Первокурсники особенно остро нуждаются в поддержке, но ресурсы учебных заведений ограничены, и система живых тьюторов требует значительных человеческих и организационных затрат.

Разработка цифрового помощника способна частично или полностью заменить необходимость в живых тьюторах, обеспечивая при этом

постоянную доступность, масштабируемость, единый стандарт предоставления информации и мгновенный отклик. Это особенно важно для УРФУ как крупного университета, где численность обучающихся и масштабы кампуса затрудняют индивидуальный подход к каждому студенту.

Таким образом, проект отвечает вызовам времени, способствует цифровой трансформации образовательной среды и способствует более эффективному использованию ресурсов университета.

Область применения программного продукта

Цифровой тьютор ориентирован на студентов УРФУ, особенно на первокурсников и обучающихся, недавно поступивших на образовательную программу. Веб-приложение может использоваться:

- как внутренняя адаптационная платформа университета;
- как часть сопровождения абитуриентов и новых студентов;
- как дополнение к системе дистанционного и смешанного обучения.

В перспективе приложение может быть масштабировано на другие факультеты, институты или даже вузы с аналогичной структурой и потребностями.

Ожидаемые результаты и планируемые достижения

В результате выполнения проекта планируется создать:

- полнофункциональное веб-приложение с ИИ-чат-ботом;
- удобную систему персонального расписания;
- навигатор по образовательным и административным ресурсам УРФУ;
- систему рекомендаций по адаптации;
- админ-панель для управления и анализа.

В перспективе цифровой помощник может не только повысить качество адаптации, но и частично заменить тьюторов, снизив административную нагрузку и улучшив цифровую инфраструктуру университета.

1 Основная часть

1.1 Участники проекта и их вклад

Проект по разработке цифрового помощника для адаптации студентов реализуется командой из пяти человек, каждый из которых отвечает за отдельное направление работ:

1.1.1 Работа тимлида

Шукшин Иван Дмитриевич (тимлид)

Руководил постановкой задач и контролем сроков. На этапе инициализации проекта была разработана ролевая модель взаимодействия участников, а также сформирована структурная схема (flow), отражающая основные процессы внутри цифрового помощника. Особое внимание уделялось формализации целей и масштабов проекта, выстраиванию логики взаимодействия между компонентами (чат-бот, интерфейс, база данных) и согласованию архитектурных решений с участниками команды.

1.1.2 Работа аналитика

Прошина Евгения Васильевна (аналитик)

Выполняла анализ конкурентов, сбор требований и анализ целевой аудитории, участвовала в формулировке функциональных требований и сборе, и предварительной разметке датасета. Также отвечает за часть тестирования и подготовку итогового отчета.

В рамках работы аналитиком выполняла анализ конкурентов, сбор требований и анализ целевой аудитории, участвовала в формулировке функциональных требований, сборе и предварительной разметке датасета. Также в бязанности входила часть тестирования.

Основные пользователи — активные студенты УрФУ, участвующие в жизни

университета (например, члены Союза студентов). Именно они чаще всего совмещают учебу с мероприятиями и нуждаются в удобном инструменте для организации времени.

Функциональность

Сервис включает следующие ключевые модули:

- 1) Регистрация и профиль
 - Вход через email или телефон с обязательной верификацией.
 - Заполнение базовой информации (ФИО, институт, направление) с возможностью редактирования.
- 2) Расписание
 - Автоматическая загрузка учебного расписания из Модеуса.
 - Ручное добавление внеучебных мероприятий.
 - Гибкая настройка отображения (цвета, метки, категории).
 - Экспорт расписания в удобный формат (PDF, изображение).
- 3) Календарь мероприятий
 - Просмотр анонсов событий университета.
 - Возможность регистрации и уведомлений о предстоящих мероприятиях.
- 4) Чат-бот на основе ИИ
 - Ответы на частые вопросы (расписание, расположение аудиторий, правила университета).
 - Интерактивное взаимодействие в реальном времени.
- 5) Информационный раздел
 - База знаний об университете (история, структура, контакты).
 - FAQ с поиском.
- 6) Обратная связь
 - Анкетирование по завершении курсов.
 - Модерация отзывов для улучшения образовательного процесса.

Роли пользователей

Студент — основной пользователь, работающий с расписанием, мероприятиями и чат-ботом.

Администратор — управляет контентом, мероприятиями, пользователями и аналитикой.

Сценарии использования

– Для студента:

- 1) Регистрация → заполнение профиля.
- 2) Настройка расписания: автоматическая загрузка пар + добавление мероприятий.
- 3) Использование чат-бота для быстрых справок.
- 4) Участие в мероприятиях через календарь.
- 5) Оценка курсов через систему обратной связи.

– Для администратора:

- 1) Управление пользователями (добавление, блокировка).
- 2) Публикация мероприятий и обновление информации.
- 3) Модерация контента (FAQ, отзывы, чат-бот).
- 4) Анализ активности студентов и качества образовательных процессов.

Графический User Flow

Также совместно с тимлидом составили Графический User Flow на основе функциональных требований и сценариев использования.

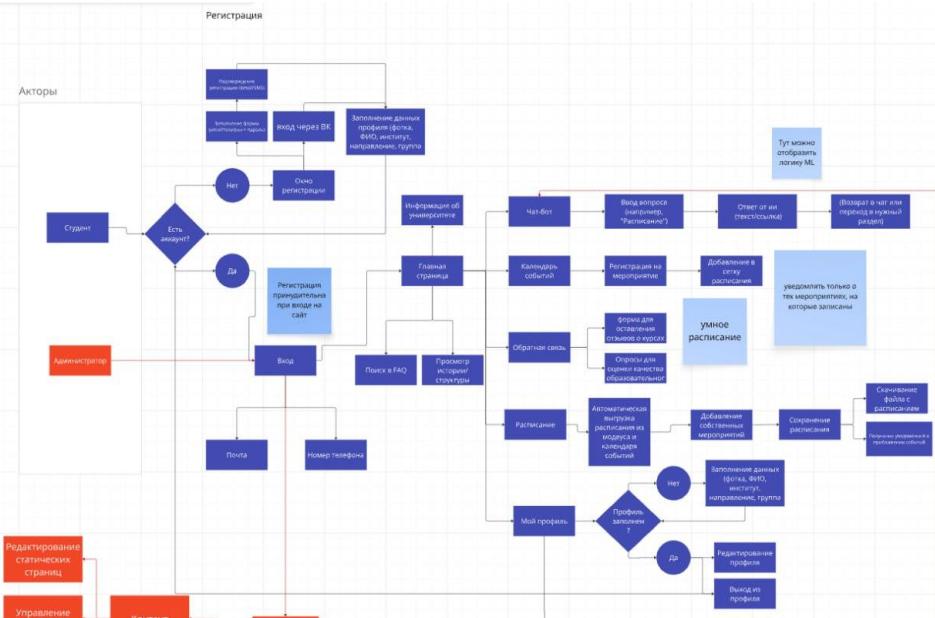


Рисунок 1 –

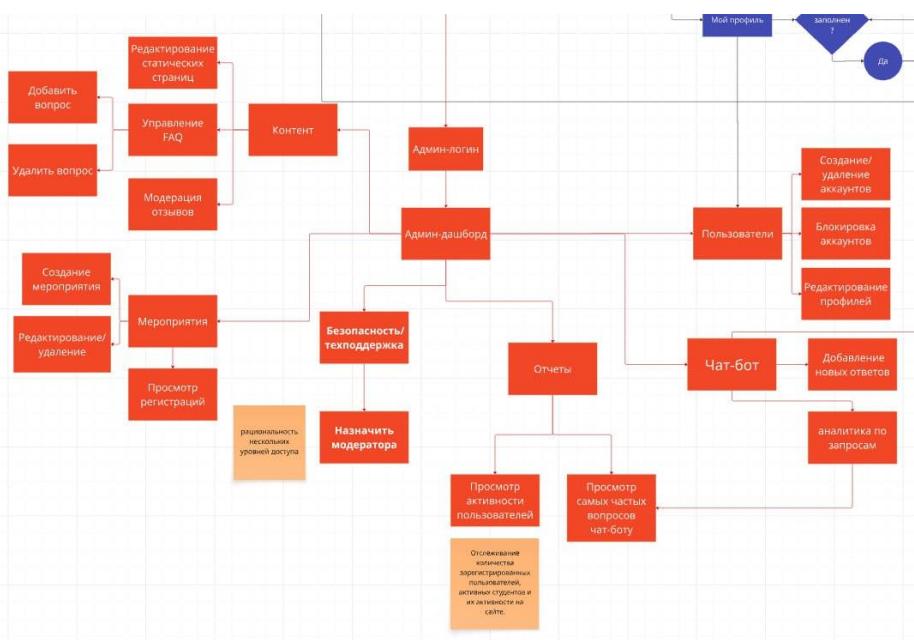


Рисунок 2 –

Создание датасета

Вместе дизайнером нашей команды изучили сайты УрФУ, Союза студентов, официальные группы ВКонтакте и Телеграм-каналы УрФУ. В

результате собран перечень самых важных и часто задаваемых вопросов, составили к ним ответы, сгруппировали их по категориям и предоставили результат работы бэкенд разработчику. В конечном итоге получилось 528 вопросов, к каждому из которых есть развернутый ответ.

Фрагмент датасета:

```
"data": [
  {
    "category": "аттестат",
    "intents": [
      "Я студент, как мне получить копию аттестата со школы?",
      "Где взять дубликат моего школьного аттестата?",
      "Как запросить копию аттестата об окончании школы?",
      "Куда обратиться за копией аттестата?"
    ],
    "answer": "Обратитесь в Отдел по работе со студентами 343-375-45-12, И-216, Мира, 19"
  },
  {
    "category": "военная кафедра",
    "intents": [
      "Есть ли военная кафедра в УрФУ?",
      "Можно ли записаться на военную кафедру УрФУ?",
      "Как работает общевоенная подготовка в УрФУ?",
      "Где получить информацию о ВК УрФУ?"
    ],
    "answer": "Да, есть. Поступить могут студенты 1 курса, очной формы обучения. Подробнее можете узнать на сайте https://vuc.urfu.ru/ru/ или по номеру: 8 (343) 375-47-64 – общевоенная подготовка, дежурный по ВУЦ– 8 (343) 375-44-87"
  },
]
```

Итог работы аналитика

Аналитик провела комплексную подготовку к разработке сервиса для студентов УрФУ. Выявила ключевое преимущество нашего решения - возможность объединения учебного расписания с внеучебными мероприятиями в едином интерфейсе. Это позволило четко определить целевую аудиторию проекта - активных студентов, участвующих в жизни университета.

Основная аналитическая работа включала проработку функциональных требований ко всем разделам будущего сервиса. Были прописаны полные сценарии взаимодействия для двух типов пользователей. Совместно с тимлидом мы визуализировали логику работы системы через графический User Flow.

Вместе с дизайнером проанализировали официальные источники УрФУ, отобрали и структурировали 528 наиболее актуальных вопросов с развернутыми ответами. Этот датасет стал основой для обучения ИИ-ассистента.

Все решения принимались совместно с командой, что обеспечило учет как пользовательских потребностей, так и технических возможностей реализации.

1.1.3 Работы backend-разработчик

Исаков Владимир Александрович (бекенд-разработчик)
Выбрал backend-стек, реализовал архитектуру и логику взаимодействия, создавал и развёртывал базы данных, реализует REST API, админ-панель и ML-интеграцию.

Разрабатывается высоконагруженное серверное приложение чат-бота, обеспечивающее:

- Безопасную авторизацию пользователей
- Интеграцию с локально развернутой нейросетевой моделью
- Эффективное кеширование запросов

- Контейнеризованное развертывание
- Взаимодействие с SPA-клиентом

Ключевые требования включают минимальное время отклика, соответствие GDPR и масштабируемость на уровне микросервисов.

1) Архитектурный подход

Принята гибридная архитектура, сочетающая:

- а) Монолитное ядро для тесно связанных компонентов
- б) Изолированный микросервис нейросети
- в) Событийную шину для асинхронной обработки

Преимущества:

- Упрощение разработки базовой логики
- Гибкое масштабирование ресурсоемких операций
- Повышенная отказоустойчивость

2) Основные компоненты системы

Монолитное ядро (Core Service) включает три ключевых модуля:

1) Аутентификация:

- Поддержка JWT и OAuth2 (Google, GitHub)
- Шифрование сессий (AES-256)
- Хранение данных в Redis

2) Управление чатами:

- Контент-фильтрация
- Привязка сообщений к пользователям через UUID

3) API интерфейс:

- Документирование через OpenAPI 3.0
- Поддержка HTTP/2 и WebSocket

Микросервис нейросети (Neural Service) особенности реализации которой:

- Изоляция в Docker-контейнере
- Оптимизация инференса через ONNX Runtime
- Горизонтальное масштабирование в Kubernetes
- Взаимодействие с ядром через gRPC

Событийная шина (RabbitMQ) чья архитектура обработки запросов:

- Помещение запросов в очередь `chat_requests`
- Асинхронная обработка Neural Service
- Возврат результатов через `chat_responses`

Механизмы обеспечения надежности:

- Подтверждение доставки сообщений
- Автоматический повтор обработки при ошибках

4) Обеспечение производительности и безопасности

Производительность:

- Кеширование запросов в Redis
- Оптимизированные протоколы связи (gRPC, HTTP/2)
- Балансировка нагрузки между инстансами Neural Service

Безопасность:

- Шифрование персональных данных (AES-256)
- Валидация токенов на уровне API Gateway
- Изоляция нейросетевых вычислений

Таблица 1 – Стек технологий

Компонент	Технологии	Обоснование выбора
Backend	Python, FastAPI, SQLAlchemy	Асинхронность, высокая производительность, Type Hints.
Нейросеть	PyTorch, ONNX Runtime	Поддержка GPU/CPU, оптимизация инференса.
База данных	PostgreSQL, Redis	ACID-транзакции + кеширование запросов.
Аутентификация	JWT, OAuth2 (RFC 7523)	Минимизация состояния (stateless), совместимость с GDPR.
Инфраструктура	Docker, Docker Compose	Идемпотентность окружения, простота деплоя.
CI/CD	GitHub Actions, ArgoCD	Интеграция с GitHub, автоматизация деплоя в Kubernetes.

Сетевое взаимодействие

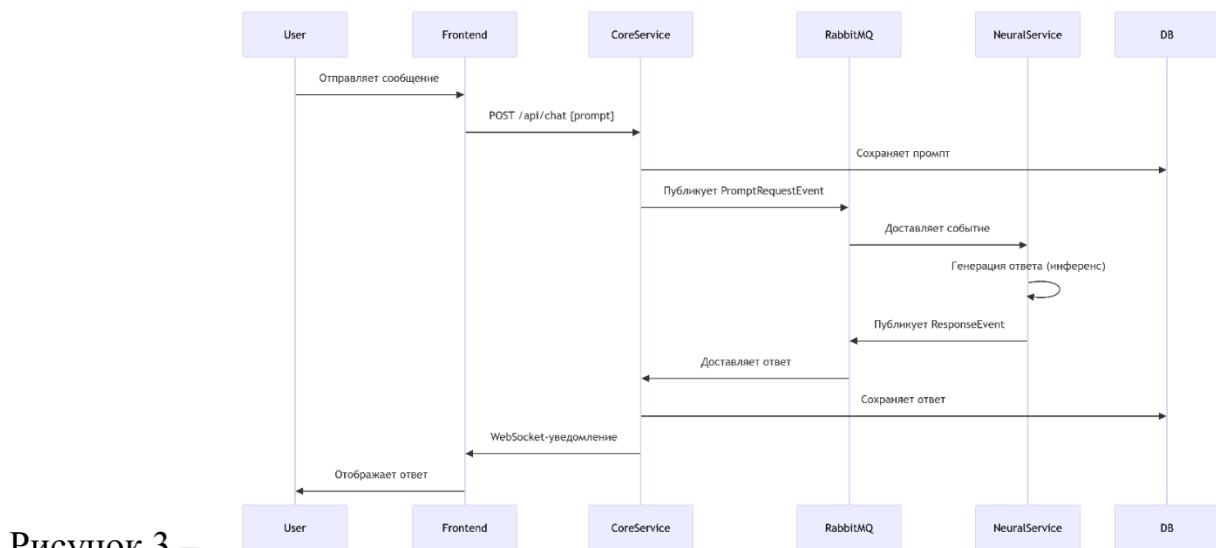


Рисунок 3 –

Описание:

Пользователь взаимодействует с фронтендом (HTTP/WebSocket).

Frontend отправляет запросы в Core Service.

Core Service:

Проверяет JWT-токены.

Сохраняет данные в PostgreSQL.

Публикует запросы в RabbitMQ.

Neural Service генерирует ответы и возвращает их через очередь.

Результаты кешируются в Redis и передаются пользователю.

Заключение

Разработанный чат-бот успешно решает поставленные задачи:

- Быстрая работа: Ответы генерируются менее чем за полсекунды благодаря кешированию и оптимизации нейросети.
- Безопасность: Данные пользователей надёжно защищены (пароли шифруются, доступ по токенам).
- Масштабируемость: Система легко адаптируется под растущее число пользователей.

1.1.4 Работа frontend разработчика

Крохалев Дмитрий Константинович (фронтенд-разработчик)
Отвечает за реализацию интерфейсов регистрации, авторизации и профиля, интеграцию чат-бота и интерфейса сайта, а также за реализацию конструктора расписания и рекомендаций.

Передо ним, как frontend разработчиком, стояла задача по разработке пользовательского интерфейса и пользовательского опыта веб-приложения. Помимо этого, он отвечал за верстку макетов, полученных от дизайнера, интеграцию с backend API, оптимизацию производительности и обеспечение кроссбраузерной совместимости. Также тесно взаимодействовал с дизайнером для точной реализации визуальной концепции и с backend-разработчиком для налаживания обмена данными.

Процесс разработки интерфейса

Получив готовые дизайн-макеты из Figma и техническое задание. После детального анализа макетов из требований приступил к выбору технологического, в конечном итоге я остановился на JavaScript библиотеке - React и разбивке интерфейса на пере используемые компоненты.

После этого приступив к непосредственной разработке. Реализовал верстку каждой страницы, стилизовала элементы в соответствии с дизайном, так же обговорив некоторые детали с дизайнером немного видоизменил элементы и части страницы так чтобы они выглядели объективно лучше на странице сайта, написал логику на JavaScript для обеспечения интерактивности, валидации форм и обмена данными с сервером.

Вот некоторые примеры изменения дизайна на сайте:

The screenshot shows a registration form titled 'Регистрация' (Registration). At the top, there's a header with the text 'Помощник для студентов', 'Добро пожаловать!', and a note: 'Перед началом необходима регистрация или вход в систему'. Below the header, there are two buttons: 'Регистрация' and 'Вход'. The registration form consists of several input fields: 'Введите ваши ФИО' (Enter your full name), 'Фамилия Имя Отчество' (Surname Name Middle name), 'Напишите ваш институт' (Write your institute), 'Ваш институт' (Your institute), 'Укажите вашу университетскую группу' (Specify your university group), 'XX-XXXXXX', 'Введите ваш номер телефона' (Enter your phone number), '+7-XXX-XXX-XX-XX', 'Укажите вашу электронную почту' (Enter your email address), 'example@gmail.com', 'Придумайте пароль' (Create a password), 'Введите пароль' (Enter password), 'Повторите пароль' (Repeat password), 'Введите пароль повторно' (Enter password again), and 'Выберите ваш аватар' (Select your avatar). There is also a circular 'Добавить фото' (Add photo) button. At the bottom right of the form is a 'Зарегистрироваться' (Register) button.

Рисунок 4 –

Страница регистрации отличается от дизайна тем что влезает на одну страницу, некоторые элементы были уменьшены для этого.

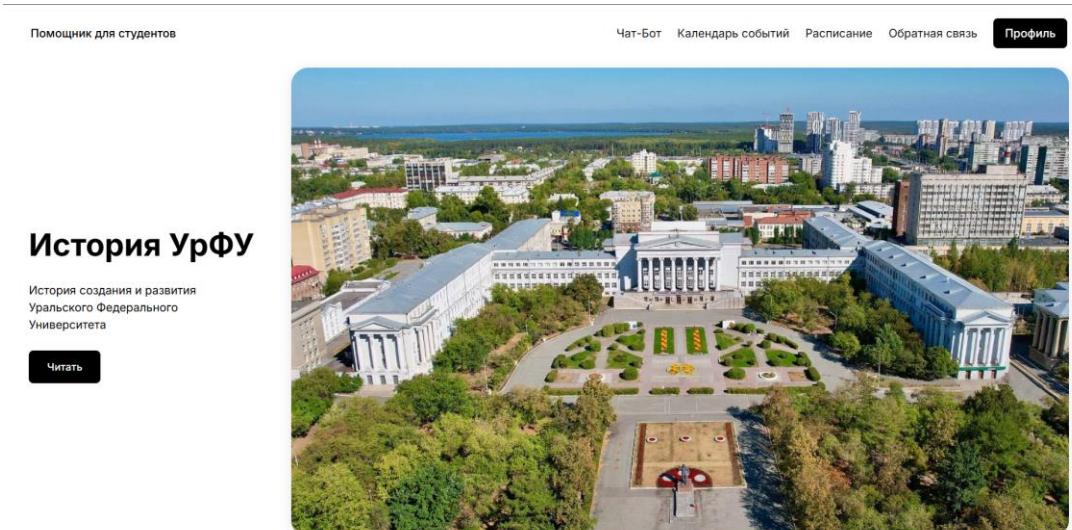


Рисунок 5 –

Так же видны изменения на главной странице сайта для более удобного отображения для пользователя.

Итог работы frontend разработчика

Полностью рабочая часть frontend, полностью сверстанный сайт и остальные страницы что не были показаны в данном файле.

Плавные и уместные CSS-анимации и переходы были добавлены для улучшения пользовательского опыта, не перегружая при этом интерфейс.

Примеры реализованных страниц, такие как страница регистрации для новых пользователей и страница входа для уже авторизованных пользователей, не только визуально соответствуют макетам, но и полностью функциональны, обрабатывая пользовательский ввод и взаимодействуя с сервером. Эти и другие разделы сайта демонстрируют успешное применение выбранного технологического стека и достижение поставленных задач.

1.1.5 Работа дизайнера

Худоерко Богдан Романович (дизайнер)

Разрабатывает начальный дизайн WireFrames и визуальное оформление интерфейсов, участвует в UX/UI тестировании и доработке интерфейсов по результатам обратной связи.

Передо мной, как дизайнером, всталась задача по разработке начального дизайна WireFrames и визуального оформления интерфейсов. Помимо этого, я участвовал в UX/UI тестировании, доработке интерфейсов по результатам обратной связи и совместно с аналитиком составлял датасет для чат-бота.

Разработка начального дизайна

Для начала я придумал в общих чертах макет будущего сайта, сделал простые наброски на бумаге. После консультации с фронтенд и бэкенд разработчиками, я выбрал структуру сайта, а после совместной работы с аналитиком, определил содержание каждой страницы сайта.

После этого я занялся разработкой дизайна с помощью Figma. Я оформил внешний вид каждого раздела сайта.

Окончательный дизайн всех страниц представлен на скриншотах.

Страница регистрации для новых пользователей и страница входа для уже авторизованных пользователей:

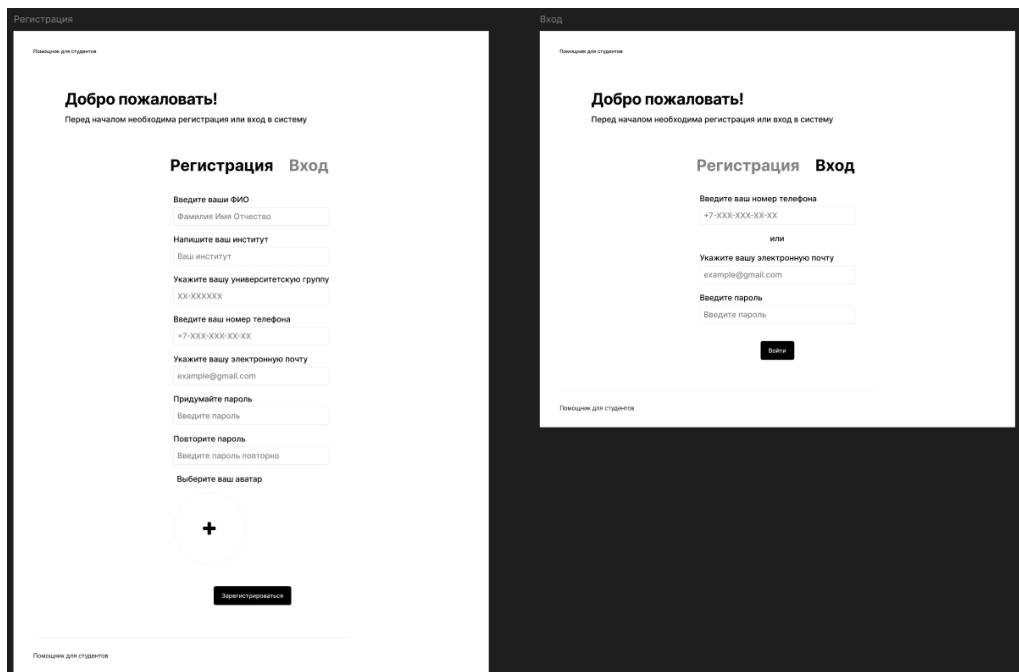


Рисунок 6 –

Главная страница сайта:

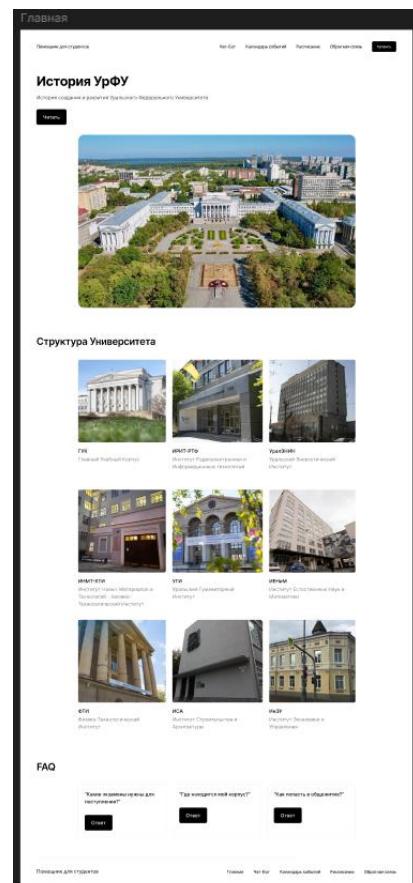


Рисунок 7 –

Раздел с чат-ботом и агрегатор мероприятий:

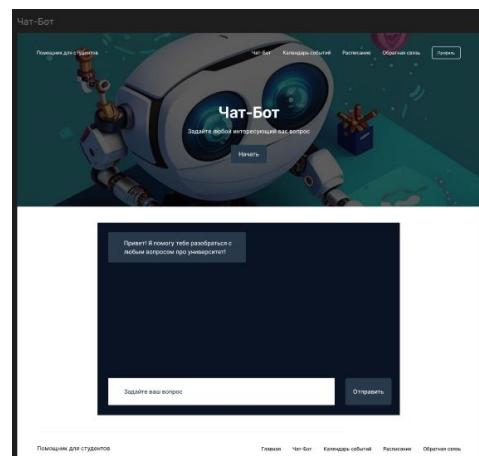


Рисунок 8 –

Агрегатор мероприятий:

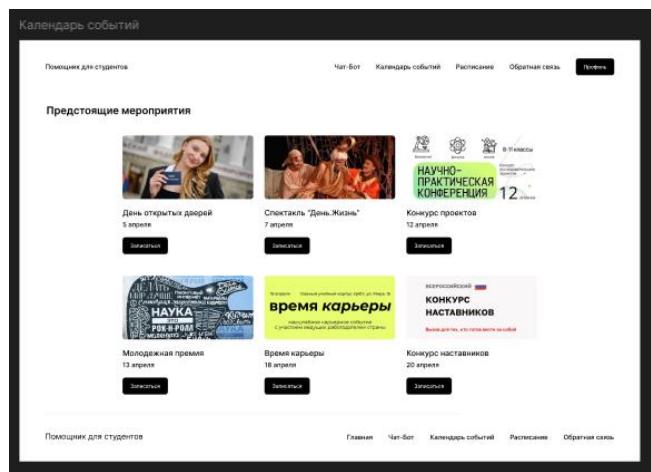


Рисунок 9 –

Страница конструктора расписания:

№	ВРЕМЯ	Пн	Вт	ср	чт	пт	сб
1	08:30-10:00						
2	10:15-11:45						
3	12:00-13:30						
4	14:15-15:45						
5	16:00-17:30						
6	17:45-19:10						
7	19:15-20:45						

Рисунок 10 –

Раздел с обратной связью:

Обратная связь

Напишите о том, как прошел ваш курс. Опишите работу преподавателя. Оцените качество занятий и предоставленного материала, объективность выставленной оценки.

Ваш отзыв

ФИО преподавателя: Иванов Иван Иванович
Предмет: Математика

Текст отзыва:
Напишите ваш отзыв о преподавателе

Отправить

Рисунок 11 –

Профиль пользователя и страница его редактирования:

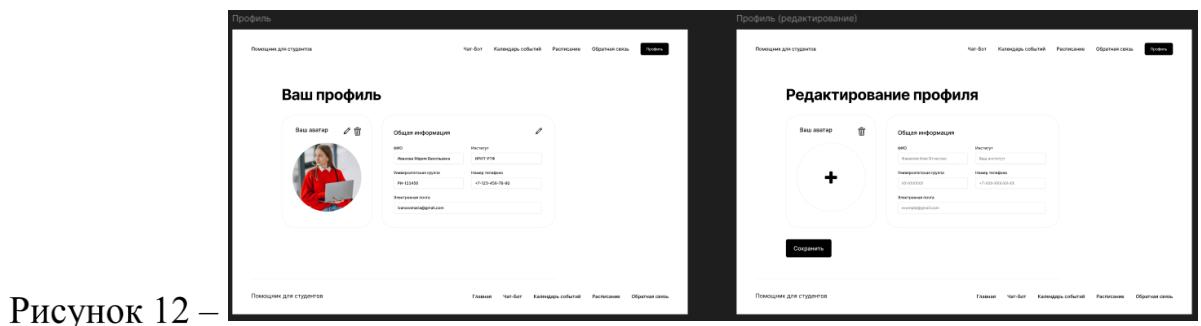


Рисунок 12 –

Во время разработки дизайна был постоянный контакт с разработчиками и аналитиком. В процессе я менял и добавлял некоторые разделы сайта.

Создание датасета

Мы с аналитиком нашей команды изучили сайты УрФУ, Союза студентов, официальные группы ВКонтакте и Телеграм-каналы УрФУ. В результате мы собрали перечень самых важных и часто задаваемых вопросов, составили к ним ответы, сгруппировали их по категориям и предоставили результат работы бэкенд разработчику. В конечном итоге у нас получилось 528 вопросов, к каждому из которых есть развернутый ответ.

Фрагмент файла с датасетом:

```
"data": [
  {
    "category": "аттестат",
    "intents": [
      "Я студент, как мне получить копию аттестата со школы?",
      "Где взять дубликат моего школьного аттестата?",
      "Как запросить копию аттестата об окончании школы?",
      "Куда обратиться за копией аттестата?"
    ],
    "answer": "Обратитесь в Отдел по работе со студентами 343-375-45-12, И-216, Мира, 19"
  },
]
```

```
"category": "военная кафедра",
"intents": [
    "Есть ли военная кафедра в УрФУ?",
    "Можно ли записаться на военную кафедру УрФУ?",
    "Как работает общевоенная подготовка в УрФУ?",
    "Где получить информацию о ВК УрФУ?"
],
"answer": "Да, есть. Поступить могут студенты 1 курса, очной формы обучения. Подробнее можете узнать на сайте https://vuc.urfu.ru/ru/ или по номеру: 8 (343) 375-47-64 – общевоенная подготовка, дежурный по ВУЦ– 8 (343) 375-44-87"
},
```

Итог работы дизайнера

В рамках проекта мне как дизайнеру удалось проработать все этапы создания цифрового продукта - от первоначальной концепции до готовых интерфейсов. Основной фокус был сосредоточен на разработке удобного и современного интерфейса, который бы отвечал потребностям целевой аудитории.

Начав с набросков на бумаге, я последовательно развивал дизайн-концепцию, учитывая технические требования разработчиков и аналитические данные. В Figma были созданы детализированные макеты всех ключевых разделов.

Параллельно с дизайном интерфейсов велась работа по формированию базы данных для чат-бота. Вместе с аналитиком мы провели анализ цифровых ресурсов УрФУ, систематизировав 528 наиболее актуальных вопросов и ответов.

Все этапы разработки сопровождались взаимодействием с командой.

1.2 Требования заказчика и пользователей. План действий

1.2.1 Требования заказчика

Согласно техническому заданию от заказчика, цифровой помощник должен включать:

- 5) Персональное расписание с синхронизацией с университетским графиком и дедлайнами;
- 6) Чат-бот с ИИ, обученный на датасете вопросов и ответов от компетентных людей;
- 7) Навигатор по учебным ресурсам (лекции, лабораторные, контакты);
- 8) Ответы на типовые вопросы (корпуса, библиотека и пр.);
- 9) Рекомендации по адаптации (тайм-менеджмент, студсообщества);
- 10) Админ-панель: рассылки, статистика, управление пользователями, интеграция с сервисами.

1.2.2 Требования пользователей

- 1) Простой, интуитивно понятный интерфейс;
- 2) Доступность на любых устройствах;
- 3) Быстрые ответы чат-бота;
- 4) Возможность сохранять и редактировать расписание;
- 5) Доступ к актуальной информации об учебных материалах и событиях.

1.2.3 Backlog проекта

- 1) Анализ требований, целевой аудитории, составление ролей и задач;
- 2) Выбор стеков технологий (frontend и backend);
- 3) Разработка архитектуры, базы данных и логики взаимодействий;
- 4) Создание прототипов интерфейсов (WireFrames, дизайн);
- 5) Реализация ключевых компонентов: регистрация, чат-бот, расписание;
- 6) Сбор и разметка датасета для обучения ML-модели;
- 7) Обучение и развёртывание модели;
- 8) Интеграция с интерфейсом;
- 9) Разработка админ-панели и аналитики;
- 10) Тестирование, исправление ошибок.

1.3 Анализ и сопоставление аналогов

В рамках подготовки проекта был проведён анализ существующих решений, частично или полностью схожих с задачами цифрового помощника для адаптации студентов:

Название продукта	Краткое описание	Преимущества	Недостатки
Яндекс GPT / Alice for Education	Голосовой помощник с базовыми возможностями поиска и ответов на вопросы	Готовая ИИ-технология, поддержка голосовых команд	Не адаптирован под вуз, нет персонализации под студента
Buddy (Skolkovo project)	Мобильное приложение для студенческого сопровождения	Фокус на взаимодействии и трекинге задач	Не реализована глубинная интеграция с вузовскими сервисами
Telegram-боты вузов	Набор простых ботов для консультаций и получения ссылок	Удобство использования, привычный интерфейс	Отсутствие интеллектуального компонента, нет адаптации под расписание

Вывод: Существующие решения либо избыточны, либо недостаточно ориентированы на реальную структуру и потребности студентов УРФУ. Наш проект сочетает ключевые элементы:

- персонализация;
- доступность;
- умный чат-бот;
- адаптация под вуз;

Что делает его уникальным и практико-ориентированным решением.

1.4 Методология разработки и процесс реализации

Проект реализуется с использованием гибкой методологии разработки — Scrum, адаптированной под учебную команду. Рабочий процесс организован по итерациям (спринтам), каждая из которых длится 1–2 недели.

На старте каждой итерации команда формирует задачи и приоритеты, в конце проводится обзор и ретроспектива.

Основные этапы:

Анализ и проектирование — сбор требований, исследование аудитории, формирование backlog'a, проектирование архитектуры, дизайн интерфейсов.

Разработка MVP — реализация базовых компонентов: авторизация, чат-бот, расписание.

Интеграция ML-модуля — обучение модели на собственном датасете, развертывание, подключение к основному приложению.

Функциональное расширение — реализация админ-панели, рекомендаций, календаря событий.

Тестирование и доработка — UX/UI тестирование, исправление багов, подготовка финальной версии.

1.5 Промежуточное тестирование и выявленные ошибки

По мере разработки осуществлялось поэтапное тестирование:

Юнит-тестирование для backend-компонентов (валидация ввода, обработка запросов);

Интеграционное тестирование — проверка взаимодействия frontend ↔ backend;

UX/UI тестирование — проводится дизайнером и аналитиком, выявлены моменты непонятной навигации, устранены на этапе доработки;

Тестирование ML-модуля — проверка ответов чат-бота на заранее составленных вопросах; на первом этапе выявлено несоответствие ответов — исправлено путём уточнения датасета.

Основные ошибки:

некорректная обработка нестандартных запросов к чат-боту;

задержки при загрузке большого количества расписаний;

дублирование информации в профиле.

Все выявленные проблемы были устраниены до перехода к следующему этапу

1.6 Планирование деятельности и распределение задач

Планирование задач в проекте велось поэтапно и охватывало все стадии — от проектирования до тестирования и презентации. Основа планирования — использование диаграммы Ганта, в которой отражены взаимосвязи, этапность и приоритетность задач.

1.6.1 Подход к планированию

Для организации командной работы использовались следующие принципы:

- 1) Чёткое разделение зон ответственности: каждый участник имеет закреплённую роль и направление задач;
- 2) Максимальная параллельность задач, где это возможно, с целью экономии времени и уменьшения узких мест;
- 3) Привязка задач к функциональным блокам: блоки работ разделены на проектирование, разработку, ML-интеграцию, тестирование и завершение;
- 4) Оценка трудоёмкости: при планировании учитывались не только технические сложности, но и загрузка каждого участника;
- 5) Гибкость пересмотра: при изменении приоритетов или выявлении технических ограничений задачи могли быть перераспределены или уточнены.

1.6.2 Пример распределения задач по направлениям:

Проектирование (март – начало апреля)

- Анализ требований, целевой аудитории, формирование ролевой модели: Евгения Прощина, Иван Шукшин
- Выбор технологического стека: Владимир Исаков, Дмитрий Крохалев

- Дизайн Wireframes и UI: Богдан Худоерко

Начальная разработка (апрель)

- Разработка архитектуры, базы данных: Владимир Исаков
- Создание личного кабинета и регистрации: Дмитрий Крохалев
- Настройка базовой навигации и интеграции интерфейса: Дмитрий Крохалев

- Интеграция ИИ (конец апреля – май)
- Подготовка датасета и обучение ML-модели: Владимир Исаков, Евгения Прощина
 - Разработка интерфейса для ML и подключение бота: Иван Шукшин, Дмитрий Крохалев
 - Интеграция чат-бота с веб-интерфейсом: Дмитрий Крохалев, Владимир Исаков

Расширение функциональности (май)

- Конструктор расписания и рекомендаций: Дмитрий Крохалев
- Админ-панель, рассылки, просмотр пользователей: Владимир Исаков
- Аналитика и календарь событий: Владимир Исаков

Финальный этап (конец мая – июнь)

- UX/UI тестирование: Евгения Прощина, Богдан Худоерко
- Исправление багов и оптимизация интерфейса: все разработчики
- Подготовка презентации и финального релиза: Иван Шукшин

1.6.3 Используемые инструменты планирования

Диаграмма Ганта (Excel/Google Sheets) — основной инструмент визуального планирования;

Совместные документы (Google Docs) — для ведения backlog, задач, контроля статуса;

Telegram/Discord — для коммуникации и синхронизации по задачам.

1.6.4 Результаты планирования

Планирование позволило эффективно организовать процесс разработки: избежать пересечения задач и дублирования усилий, своевременно переходить между этапами, контролировать статус выполнения и соблюдение сроков, а также равномерно распределить нагрузку между участниками. Такой подход обеспечил прозрачность командной работы и гибкость, необходимую для адаптации к изменениям и уточнениям требований на всех этапах проекта.

2 ЗАКЛЮЧЕНИЕ

Результаты разработки цифрового помощника для адаптации студентов показывают, что программный продукт в целом соответствует поставленным требованиям как со стороны заказчика, так и с точки зрения потребностей конечного пользователя. Ключевые функции — чат-бот, персональное расписание, навигатор по ресурсам, рекомендации и админ-панель — были реализованы с учётом логики использования в условиях УРФУ и подтверждают работоспособность предложенного архитектурного и функционального решения. Пользовательский опыт на промежуточных этапах тестирования подтвердил, что продукт закрывает основные болевые точки первокурсников и снижает барьер входа в университетскую среду.

Качество программного продукта оценивалось в ходе пошагового функционального и пользовательского тестирования. Несмотря на выявленные на ранних этапах ошибки — некорректные ответы чат-бота на нестандартные запросы, проблемы с отрисовкой интерфейса на мобильных устройствах и дублирование элементов профиля — все существенные дефекты были устранены. Оставшиеся незначительные недочёты (например, редкие задержки в отображении событий в расписании) не влияют критически на общее восприятие продукта и не препятствуют его основному назначению. В результате можно говорить о том, что система достигла стабильности и достаточного уровня пользовательской готовности для демонстрации и пилотного внедрения.

Анализ процесса разработки позволил выделить несколько направлений для будущего улучшения:

Улучшение точности ИИ-модуля за счёт расширения и регулярного обновления обучающего датасета;

Разработка мобильного приложения или адаптивной PWA-версии для более удобного использования на смартфонах;

Расширение возможностей календаря и системы напоминаний (интеграция с внешними календарями, push-уведомления);

Автоматизация обратной связи — генерация рекомендаций на основе активности студента;

Подключение систем авторизации через вузовские сервисы (например, ЕСИА или внутренняя LDAP-авторизация).

В целом проект продемонстрировал успешную реализацию студенческой разработки в условиях реального заказчика и образовательной среды. Командная работа, правильное распределение ролей и гибкий подход к планированию стали ключевыми факторами успеха. Полученный опыт и техническая база проекта могут быть использованы в будущих инициативных или исследовательских проектах, а сам цифровой помощник может стать основой для масштабируемого продукта на уровне университета или межвузовой платформы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Диаграмма Ганта проекта цифрового помощника для адаптации студентов [Электронный ресурс] // Google Таблицы. – URL: <https://docs.google.com/spreadsheets/d/1uhbwq-nIaywx5suMM7QVL-C16wmCXRnzGAvx9AuPHls/edit?usp=sharing> (дата обращения: 24.05.2025).
- 2) Репозиторий клиентской части (frontend) цифрового помощника для адаптации студентов [Электронный ресурс] // GitHub. – URL: <https://github.com/dirty-kdk/stu-nav> (дата обращения: 24.05.2025).
- 3) Репозиторий серверной части (backend) цифрового помощника для адаптации студентов [Электронный ресурс] // GitHub. – URL: <https://github.com/VkOrVyAt/chatbot-backend> (дата обращения: 24.05.2025).
- 4) Схема пользовательских сценариев (User Flow) цифрового помощника для адаптации студентов [Электронный ресурс] // Miro. – URL:
https://miro.com/welcomeonboard/VGZrZit2cEVra0Y1OTI2Q0R0RzdSZjB1aGg4K1FxcUZqbXNkTHc0WmJZUHg3WXNGWG9kNDIxM2Ria3ZLR3RhZ1ZLcE1DNFFwLzh1YUJiUmdIQ2daSVMxTWsvbTZBbHRDUkVueG4wakVaMnNCdVZrZnJoeHN1YnBwd2xRSjFGR1FhWWluRVAXeXRuUUgwWDI3Mk1qRGVRPT0hdjE=?share_link_id=350778031421 (дата обращения: 24.05.2025).
- 5) Аналитика по теме цифрового помощника для адаптации студентов [Электронный ресурс] // Google Документы. – URL: https://docs.google.com/document/d/1wKB4q6nDYAMqTUtXV4389cuqEx8Xq90J/edit?usp=drive_link&oid=116973621034188854481&rtpof=true&sd=true (дата обращения: 24.05.2025).

