

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 Telegram Mini App и бота для интерактивов на мастер-классах»
по дисциплине «Проектный практикум»

Заказчик:

Новиков М.Ю.

Куратор:

Карелина Д.Е.

Студенты команды:

Voshod

Жилин Ю.Д.

Леонова Е.С.

Третьяков А.С.

Конев И.Д.

Екатеринбург, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Область применения.....	5
2 Решение	6
3 Требования заказчика.....	8
4 Аналоги разрабатываемого продукта	9
5 Архитектура	10
5.1 Архитектура приложения.....	10
5.2 Выбор стэка для frontend-сервера.....	10
5.3 Выбор стэка для backend-сервера	10
5.4 Выбор стэка для бота в Telegram.....	11
5.5 Выбор интерфейса коммуникации между микросервисами.....	11
5.6 Выбор СУБД.....	12
6 Методология и процесс разработки.....	13
6.1 Методология	13
6.2 Процесс разработки	13
6.2.1 Этап «Аналитика».....	13
6.2.2 Этап «Проектирование».....	14
6.2.3 Этап «Разработка»	15
6.3 Процесс распределения задач внутри команды	16
ЗАКЛЮЧЕНИЕ.....	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	19
ПРИЛОЖЕНИЕ А (справочное) Примеры изображений артефактов.....	20

ВВЕДЕНИЕ

В институте ИРИТ-РТФ УРФУ часто проходят различные мероприятия для абитуриентов и их родителей. Они участвуют в интерактивах на мастер-классах. После прохождения активностей теряется дальнейшая связь с аудиторией, отсутствуют данные об участниках интерактива.

Отсутствие данных или низкий процент оставленных данных влечет за собой невозможность достижения целей заинтересованных лиц. В рамках института заинтересованных лиц являются: аналитический отдел, ведущий интерактивов, отделы принятий решений. Большое количество целей может преследовать заинтересованные лица с последующей работой с собранной информацией в процессе проведения интерактивов. Цели могут связаны с извлечением полезной информации из собранных данных.

Пример целей достижимые с помощью собранных данных:

- сохранение и поддержание обратной связи с аудиторией;
- оценка эффективности мероприятий;
- отслеживание цифрового поведения пользователя;
- оптимизация расходов на мероприятия;
- выявление удачных и неудачных подходов к проведению;
- повышение лояльности аудитории;
- принятие стратегически обоснованных решений об эффективности мастер-классов и мероприятий.

Цель проекта: найти или разработать решение, которое позволит во время интерактивов собирать эффективно данные с участников активностей.

Под эффективностью понимается увеличить процент собранных данных с аудитории, а именно больше 80%. В текущем бизнес-процессе процент, собранных данных приравнивается к 30% или 0% в конкретном интерактиве.

Список задач проекта:

- погрузиться в проблему бизнес-процесса;
- проанализировать текущий бизнес-процесс;
- предложить обоснованное решение данной проблемы;
- составить требования, ТЗ к будущему решению;
- реализовать требуемый результат в срок;
- апробировать продукт на реальных пользователей.

Требуемый результат: результатом должно быть работоспособное решение, достигшее цели проекта.

Планируемое достижение: выполнение задач, достижение цели проекта, создание работоспособного продукта, соответствующего требованиям заказчика.

1 Область применения

Область применения конечного продукта сосредоточена в рамках института ИРИТ-РТФ УРФУ и его заинтересованных сторон. Все основные проблемы, требования и задачи проекта формулировались на основе анализа текущих внутренних бизнес-процессов института, а также в результате регулярного взаимодействия с представителями заинтересованных лиц.

Продукт ориентирован на решение специфических задач, таких как организация и сопровождение интерактивных мероприятий, взаимодействие между участниками и организаторами, а также автоматизация ряда процессов.

Продукт не разрабатывался для широкого публичного использования: он содержит ограничения, связанные с доступом к ряду функций, которые требуют авторизации, либо напрямую зависят от интеграции с внутренними системами института (например, базами данных студентов, расписаниями, внутренними API и т.д.).

Таким образом, MVP реализован как внутренний корпоративный инструмент, адаптированный под конкретные сценарии использования в ИРИТ-РТФ, и требует доработки и масштабирования для возможного применения за пределами института.

2 Решение

Для достижения цели проекта команда пришла к решению, что способ достижения цели является написание собственного сервиса, который будет способен удовлетворить потребности заказчика на сегодняшний момент и в случае необходимости доработан под возникающие потребности в будущем. Возможность свободно дорабатывать функциональность под свои бизнес-потребности является одним из главных аргументов послуживший основанием при выборе способа достижения цели.

Во время аналитики были рассмотрены способы сбора данных неявным образом. По итогам рассмотренных решений было принято решение остановиться на разработке сервиса на базе мессенджера (таблица 1).

Таблица 1 – Сравнение способов сбора данных в цифровом пространстве

Критерий/Решение	Выдача уникального токена	Через fingerprint устройства	Вход через мессенджер	Участие на базе мессенджера
Идентификация пользователя на продолжительное время	-	-	+	+
Неявный способ сбора данных	+	+	-	+
Дальнейшая связь с участником	-	-	+	+
Возможность автоматизированной рассылки	-	-	-	+

С помощью проведенной аналитики команда убедилась, что реализация сервиса на базе мессенджера является лучшим решением. Мессенджер Telegram выбран в качестве платформы для нашего сервиса, благодаря своему охвату аудитории, обширной документации и опыта разработки в данном сервисе, что положительно скажется на скорости написания программного кода.

Итог: собственный сервис MiniApp с ботом на базе мессенджера Telegram, который объединит в себе следующие главные функции – возможность управлять интерактивами и проходить их, собирать информацию неявным образом с участников интерактивов.

3 Требования заказчика

К конечному продукту заказчик предъявляет следующие функциональные требования:

- функционал управление интерактивами;
- сбор данных с участников интерактива;
- создание интерактивов вида «вопрос-ответ»;
- выгрузка отчетов с данными в определенном формате;
- быстрый способ присоединения к интерактиву по QR-коду;
- реализовать весь процесс, описанный заказчиком.

К конечному продукту заказчик предъявляет следующие нефункциональные требования:

- архитектура должна быть масштабируемой и гибкой для добавления новых функций;
- должна быть ролевая модель для выдачи прав доступа к функционалу ведущего;
- требования к хранению данных в соответствии с Федеральным законом от 27.07.2006 № 152-ФЗ «О персональных данных»;
- требования к доступу данных в соответствии с Постановлением Правительства РФ от 01.11.2012 № 1119;
- требования в соответствии с Федеральным законом от 27.07.2006 № 149-ФЗ «Об информации, информационных технологиях и о защите информации»;
- требования в соответствии с Приказом ФСТЭК России от 18.02.2013 № 21.

4 Аналоги разрабатываемого продукта

Одной из приоритетных задач при разработке продукта было создание эффективного взаимодействия с основными элементами управления. С этой целью был проведён анализ существующих на рынке сервисов, представленных в данном сегменте. В качестве примеров были рассмотрены такие платформы, как Marquiz, Quizizz, Quizlet и MyQuiz. В ходе анализа были выделены наиболее полезные функциональные решения, которые были учтены при проектировании собственного сервиса.

Во время интервью с заказчиком выяснилось, что он активно использует MyQuiz для проведения интерактивов и полностью удовлетворяет его функционалом в этом аспекте. Это послужило основанием для более детального изучения данного сервиса — его функционала, дизайна, пользовательского интерфейса (UI) и пользовательского опыта (UX). В результате MyQuiz был выбран в качестве референса для разработки дизайна и прототипов веб-интерфейса нашего продукта.

Поскольку целью проекта не является конкуренция на открытом рынке, составление сравнительной таблицы с аналогами не представлялось необходимым. Изучение существующих решений использовалось исключительно для поиска дизайнерских и функциональных ориентиров, необходимых для проектирования сервиса.

Прямыми конкурентами нашего продукта являются команды дисциплины «Проектный практикум», у которых тема проекта идентичная с нашей. Сравнить наш продукт с результатом других команд не предоставляется возможным из-за конфиденциальности разработки.

Вывод: разработанный сервис удовлетворяет потребности заказчика в рамках его бизнес-процесса, предлагая уникальные возможности, которые не реализованы в существующих аналогах. В частности, это — неявный сбор данных, требуемый формат отчетности с заданной структурой, а также автоматизированная рассылка.

5 Архитектура

5.1 Архитектура приложения

Выбрана микросервисная архитектура, потому что: разделение позволит разрабатывать, масштабировать микросервисы независимо друг от друга. Независимость обеспечивает простое обновление микросервиса, без необходимости изменять весь функционал. Разделение на микросервисы позволит распределять нагрузку среди них, что увеличит производительность системы.

5.2 Выбор стэка для frontend-сервера

Стэк `nuxt.js` выбран, потому что: этот фреймворк позволяет сделать web-приложение с сервером, что необходимо для микросервисной архитектуры. Поддерживается рендеринг на сервере – `ssr`. Это снижает нагрузку с клиента. Поддерживается `websocket`, есть встроенный инструмент для `state management` – `Vuex`.

Предоставляемого функционала будет достаточно в рамках проекта. Разработчик также заинтересован разобраться в данном фреймворке так, как его использует во многих IT компаниях, что открывает широкие перспективы на дальнейшее трудоустройство и профессионального роста.

5.3 Выбор стэка для backend-сервера

Основным критерием была возможность реализовать `REST API` и `WebSocket`. Под эти критерий подходит `FastAPI` у которого есть ряд других преимуществ:

- скорость работы (по данным сайта `techempower` занимает 286, а `django` 424);
- асинхронность,
- встроенная валидация данных;

– автоматическая OpenAPI документация (То есть автоматическая генерация документаций).

Всё это стало причиной выбора FastAPI. Также у разработчиков в команде есть опыт в разработке с данной библиотекой FastAPI, что скажется на скорости создания бэкэнда.

5.4 Выбор стэка для бота в Telegram

Aiogram был выбран для того, чтобы вручную не писать запросы к телеграм, а делать это через библиотеку с удобными методами. Aiogram убирает необходимость следить за обновлениями api telegram. А также aiogram предоставляет асинхронное выполнение кода и в aiogram можно реализовать интеграцию с другими микросервисами. Также у разработчиков в команде есть опыт в разработке с данной библиотекой aiogram, что скажется на скорости создания бота.

5.5 Выбор интерфейса коммуникации между микросервисами

В качестве интерфейса коммуникации между микросервисами были выбраны REST API по протоколу HTTPS, WebSocket по протоколу WSS, потому что: websocket реализует в web-приложении двустороннюю непрерывную связь на интерактиве для обмена данными в реальном времени. Такой интерфейс поддерживается нативно в любом браузере. Его ресурсов достаточно для задач нашего продукта. REST API это общепринятый стандарт взаимодействия типа request/response web-приложений. Простота реализации и возможность масштабирования даёт возможность реализовывать новые сценарии взаимодействия. Выбраны протоколы, поддерживающие шифрование, так как приложение будет работать с конфиденциальными данными пользователей.

5.6 Выбор СУБД

В рамках MVP SQLite был выбран из-за простоты реализации с последующей возможностью перейти на более сложную архитектуру (связка PostgreSQL и Redis).

6 Методология и процесс разработки

6.1 Методология

При разработке подхода к проектной деятельности были взяты принципы Каскадной модели (Waterfall). А именно, последовательность этапов разработки, документирование, фиксирование требований на начальном этапе, ограниченное участие заказчика, ведение отчетности о выполненных задачах.

6.2 Процесс разработки

Каждый участник вне зависимости от своей роли, компетентности в команде, принимал участие на каждом этапе. Участник выполнял различные виды задач и после каждой выполненной задачи писал отчет о проделанной работе.

Вся работа была разбита на 3 основных этапа «Аналитика», «Проектирование», «Разработка». Каждый из них включал более конкретные подэтапы. Сроки реализации каждого из этапов были зафиксированы в календарном плане (в соответствии с Рисунком 1 Приложения А).

6.2.1 Этап «Аналитика»

Во время процесса разработки на этапе «Аналитика» были достигнуты следующие результаты:

- проанализировали проблему, цели, задача проекта, возможные решения для устранения проблемы проекта;
- составили ТЗ MVP, провели несколько итераций правок с заинтересованными лицами, а также согласовали с ними ТЗ.

Все наработки, аналитика и выводы, к которым пришла команда по проекту – все это структурировано зафиксировано в соответствующих документах. Трудозатраты на данному этапе (таблица 2).

Таблица 2 – Трудозатраты команды на этапе «Аналитика»

Исполнитель/ Данные	Кол. выполненных задач (шт)	Кол. затраченного времени (чел/ч)	Среднее время выполнения на одну задачу (чел/ч)
Жилин Ю.Д.	9	52	5,7
Третьяков А.С	10	35	3,5
Конев И.Д.	12	50	4,2
Леонова Е.С.	1	2	2
Командные результаты	32	139	4,35

6.2.2 Этап «Проектирование»

Во время процесса разработки на этапе «Проектирование» были достигнуты следующие результаты:

- спроектировали взаимодействие модулей в микросервисной архитектуре (в соответствии с Рисунком 3 Приложения А);
- спроектировали передачу данных между модулями сервиса (в соответствии с Рисунком 4 Приложения А);
- четко разграничили функционал, задачи микросервисов;
- спроектирован пользовательский интерфейс с учетом всего запланированного функционала (в соответствии с Рисунком 5 Приложения А);
- устранена нехватка знаний в используемых технологиях.

Все технические решения, спроектированная система зафиксированы в текстовых, схематичных артефактах. Трудозатраты на данном этапе (таблица 3).

Таблица 3 – Трудозатраты команды на этапе «Проектирование»

Исполнитель/ Данные	Кол. выполненных задач (шт)	Кол. затраченного времени (чел/ч)	Среднее время выполнения на одну задачу (чел/ч)
Жилин Ю. Д.	7	28	4
Третьяков А.С	7	47	6,7
Конев И.Д.	8	52	6,5

Продолжение таблицы 3

Исполнитель/ Данные	Кол. выполненных задач (шт)	Кол. затраченного времени (чел/ч)	Среднее время выполнения на одну задачу (чел/ч)
Леонова Е.С.	7	29	4,1
Командные результаты	29	156	5,4

6.2.3 Этап «Разработка»

Во время работы над этапом «Разработка» были достигнуты следующие результаты:

- создан Telegram Bot с запланированным функционалом (в соответствии с Рисунком 7 Приложения А);
- реализовано Api серверной части (в соответствии с Рисунком 8 Приложения А);
- сверстаны все макеты сервиса и настроена логика взаимодействия пользователя и Api (в соответствии с Рисунком 9 Приложения А);
- сервис протестирован на сценарии «happy path»;
- сервис выложен на хостинг;
- сделан дизайн всех страниц сервиса.

Трудозатраты на данному этапе (таблица 4).

Таблица 4 – Трудозатраты команды на этапе «Разработка»

Исполнитель/ Данные	Кол. выполненных задач (шт)	Кол. затраченного времени (чел/ч)	Среднее время выполнения на одну задачу (чел/ч)
Жилин Ю. Д.	2	18	18
Третьяков А.С	2	63	31,5
Конев И.Д.	2	69	34,5
Леонова Е.С.	2	17	8,5
Командные результаты	8	185	23,1

6.3 Процесс распределения задач внутри команды

Каждый участник вне зависимости от своей роли, компетентности в команде, принимал участие на каждом этапе работы над проектом, в связи с выбранным подходом к проектной деятельности. Зачастую задачи могли не соответствовать роли, компетентности участника в команде. Пример задачи на канбан доске (в соответствии с Рисунком 2 Приложения А).

Жилин Ю.Д. – тимлид. Занимался организацией рабочего процесса внутри команды, а именно распределением командных ресурсов, планированием работы, введением отчетности, поддержкой связи с заинтересованными лицами, решением конфликтов, недопониманий внутри команды, формированием единого видения деятельности команды, составлением и согласованием документов. Дополнительные задачи: разработал Telegram Bot. [1][5]

Леонова Е.С. – дизайнер. Занималась анализом дизайна, UI/UX в аналогах, определилась с референсом дизайна. По ТЗ создала необходимое количество прототипов, прошла несколько итераций правок. По согласованным прототипам разработала дизайн веб-интерфейсов ведущего и участника интерактива.

Третьяков А. С. – backend-разработчик. Занимался поиском, выбором, аргументацией итогового технического решения, проектированием архитектуры сервиса, а также взаимодействия модулей внутри системы. Участвовал в составлении ТЗ, а именно, в выборе обязательного и рационального, с точки зрения командных ресурсов, функционала. Разработал модуль, который взял на себя логику работы с данными на FastApi, а также подключил базу данных.[2][3] Протестировал сервис, сделал деплой сервиса в публичный доступ.

Конев И. Д. – frontend-разработчик. Занимался поиском, выбором, аргументацией итогового технического решения, проектированием архитектуры сервиса, а также взаимодействия модулей внутри сервиса.

Участвовал в составлении ТЗ, а именно, в выборе обязательного и рационального, с точки зрения командных ресурсов, функционала. Разработал модуль, который взял на себя логику взаимодействия пользователя и backend-сервиса на Nuxt.js, а также сверстал экраны сервиса. Протестировал сервис, сделал деплой сервиса в публичный доступ. [4]

ЗАКЛЮЧЕНИЕ

Разработанный сервис полностью соответствует требованиям заказчика. Сервис дает возможность настроить, провести интерактив, а также собрать данные неявным образом с участников интерактива. Также в сервисе заложена возможность добавления запланированного функционала, а именно, автоматизированная рассылка, ролевая модель с ограничением прав к доступу функциональности.

Первоначальное тестирование проходило во время разработки, а именно, просмотр сверстанных страниц в Telegram MiniApp и внесение правок в отображение элементов интерфейсов, проверка правильности работы эндпоинтов, WebSocket, Telegram Bot в соответствии спроектированного функционала.

Окончательное тестирование на реальных пользователях будет осуществлено в запланированный период по календарному плану. Целью финального тестирования является: убедиться в стабильной, безотказной работе сервиса вовремя проведении апробации на реальных пользователях.

В ходе работы столкнулись с проблемами: отсутствием опыта работы с сетевым протоколом WebSocket, расплывчатые требования заинтересованных лиц проекта, отсутствие опытного аналитика в команде, неочевидное поведение браузера Telegram.

Для улучшения MVP стоит активно тестировать его на реальных пользователях, чтобы выявлять ошибки и недоработки, фиксировать их и передавать команде для последующего исправления и доработки продукта. А также, добавлять функционал, связанный с рассылкой и выдачи прав, доступа к определенному функционалу.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Aiogram documentation [Электронный ресурс] / Aiogram Contributors. – URL: <https://docs.aiogram.dev/> (дата обращения: 17.05.2025).
2. FastAPI documentation [Электронный ресурс] / FastAPI Team. – URL: <https://fastapi.tiangolo.com/> (дата обращения: 17.05.2025).
3. SQLite Documentation [Электронный ресурс] / SQLite Consortium. – URL: <https://www.sqlite.org/docs.html> (дата обращения: 17.05.2025).
4. Vue.js 3 Documentation [Электронный ресурс] / Vue.js Core Team. – URL: <https://vuejs.org/guide/introduction.html> (дата обращения: 17.05.2025).
5. Григорьев, С. Методологии разработки ПО: Agile, Scrum, Waterfall и другие [Электронный ресурс] / Сергей Григорьев. – Москва, 2020. – URL: <https://habr.com/ru/companies/edison/articles/269789/> (дата обращения: 17.05.2025).

ПРИЛОЖЕНИЕ А

(справочное)

Примеры изображений артефактов

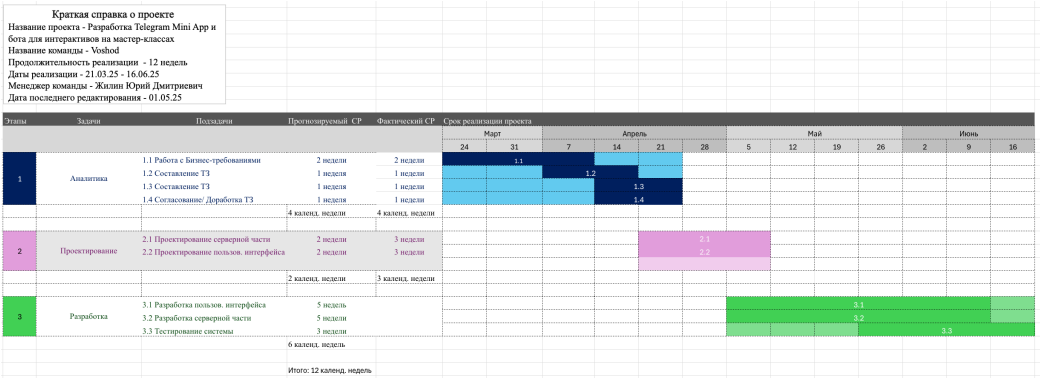


Рисунок 1 – Календарный план

☐ Спроектировать обмен данными внутри Wss

Сегмент

Для проверки

Ход выполнения

☐ Не начато

Приоритет

Средний

Дата начала

07.05.2025

Дата выполнения

08.05.2025

Повторять

Не повторяется

Заметки

Цель: окончательно разобраться и решить как будут передаваться данные внутри Wss

Описание: спроектировать работу Ws между модулями системы. Подробно со в семи техническими деталями прописать в каком виде приходят данные, как их программа парсит, какую функцию выполняют модули в данном обмене данными. Рассмотреть обмен данными подробно, со всеми шагами, которые происходят внутри процесса обмена данными на следующей ситуации. Ситуация: начинается интерактива, пользователь получает первый вопрос, отвечает на вопрос, в конце показываются результаты.

Комментарий: абстракция не нужна. Нужно подробная спроектированная схема, по которой будет видно кто, как, где хранит, обрабатывает данные. Очень ответственная задача.

Результат: полное понимание, как будет происходить обмен данными через wss между модулями, подкрепить понимание текстовым документом+ схемой.

Контрольный список

☐ Добавьте элемент

Вложения

Websocket telegram - fron
<https://docs.google.com/docu>

...

☐ Показать на карточке

Websocket telegram <-> f
<https://miro.com/app/board/u/>

...

☐ Показать на карточке

Рисунок 2 – Пример задачи участнику команды, изображенной на канбан-доске

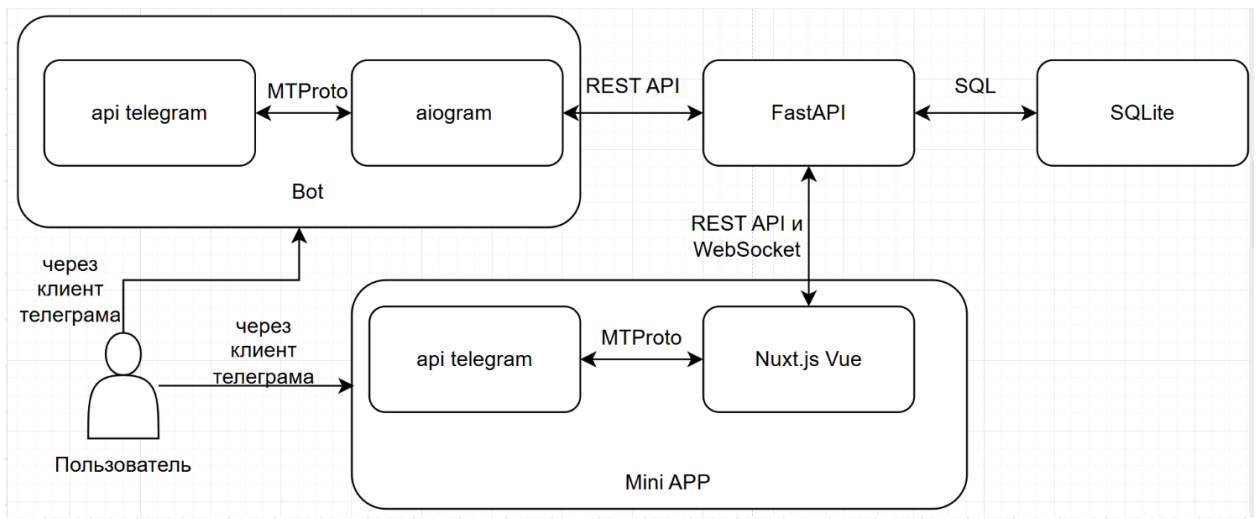


Рисунок 3 – Взаимодействие модулей в системе

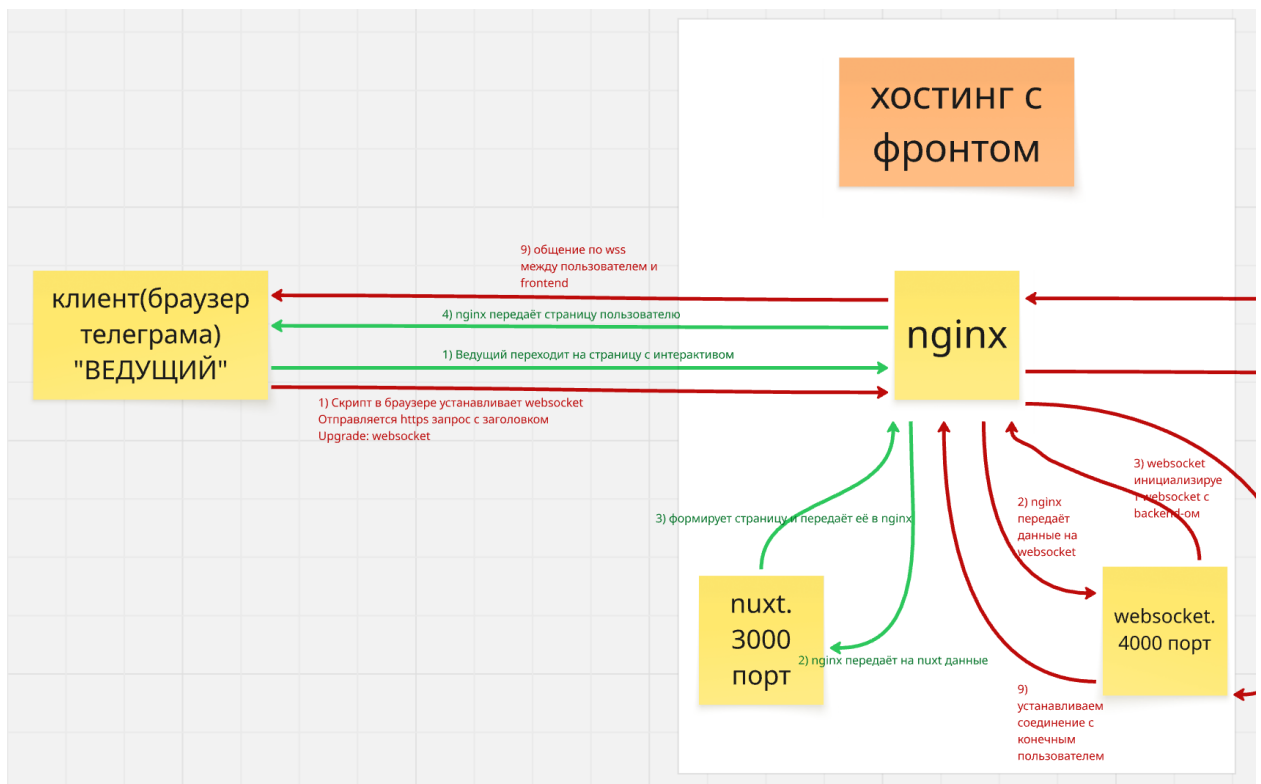


Рисунок 4 – Схема передача данных по WebSocket

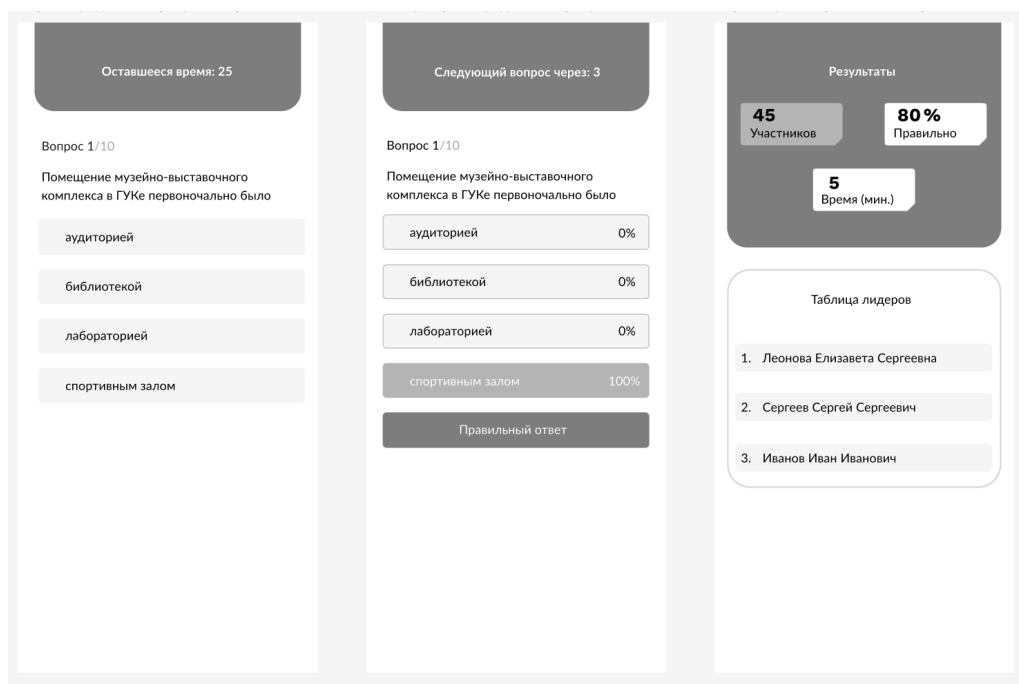


Рисунок 5 – Прототипы веб-интерфейсов участника

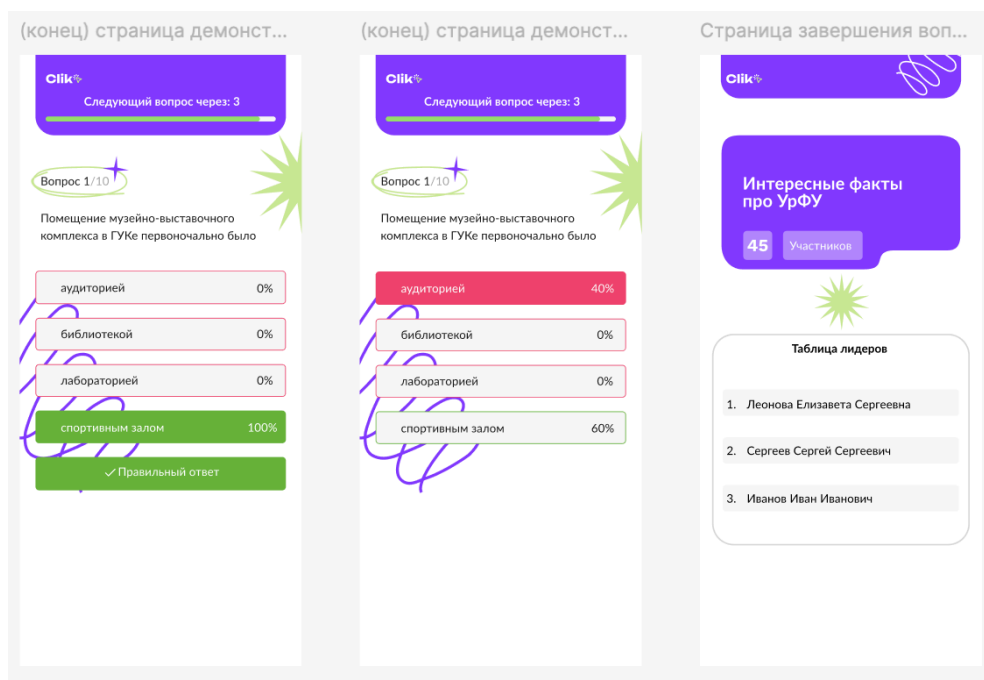


Рисунок 6 – Дизайн веб-интерфейсов участника

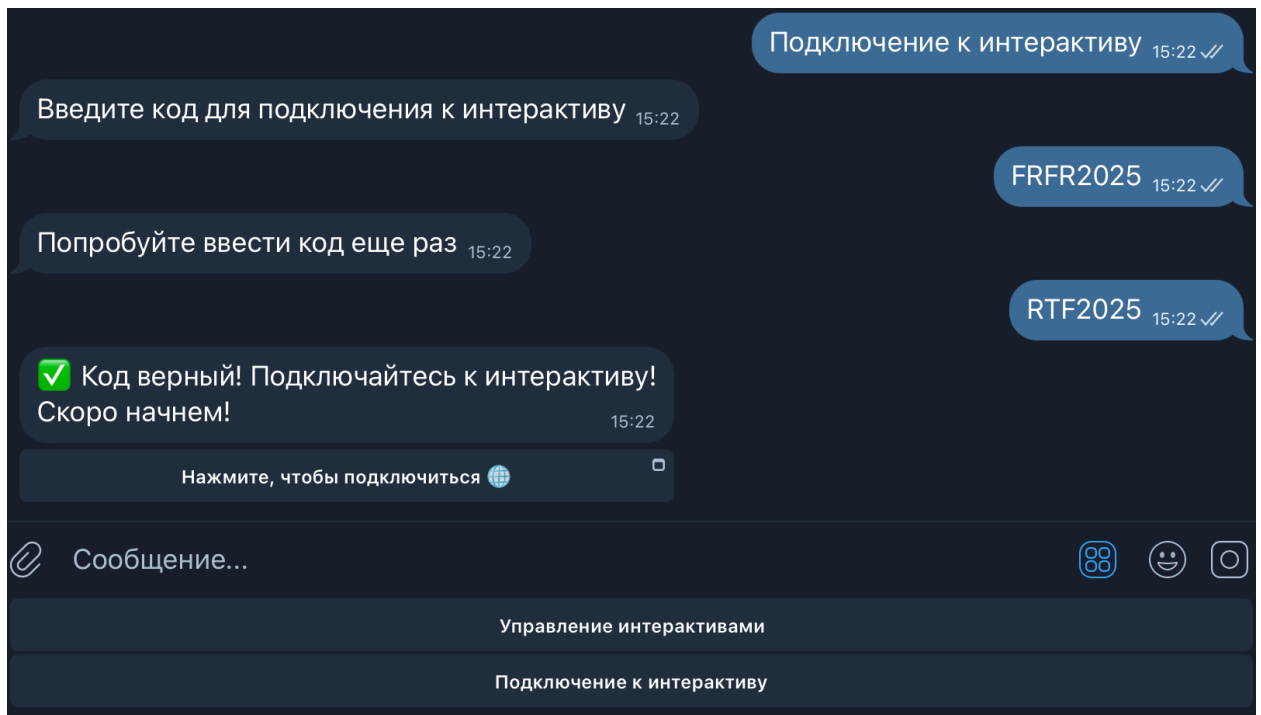


Рисунок 7 – Интерфейс бота Telegram

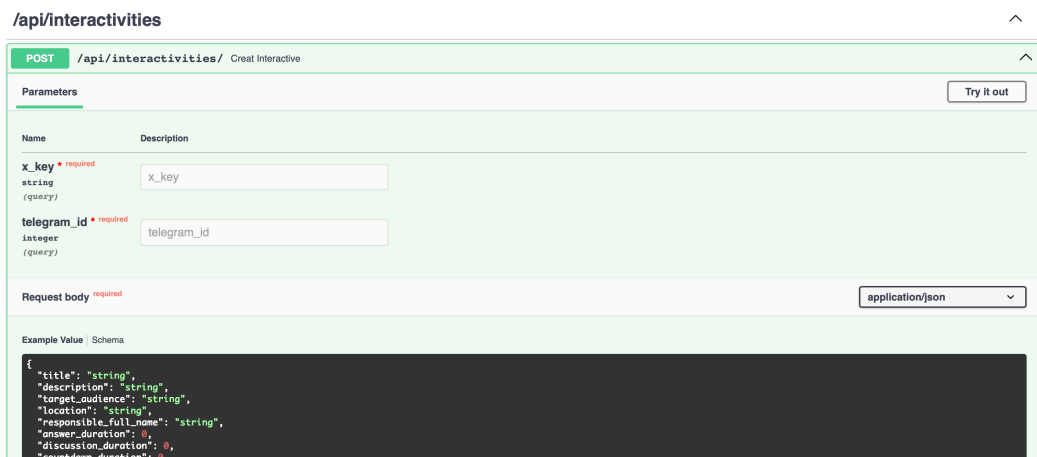


Рисунок 8 – Документация Api

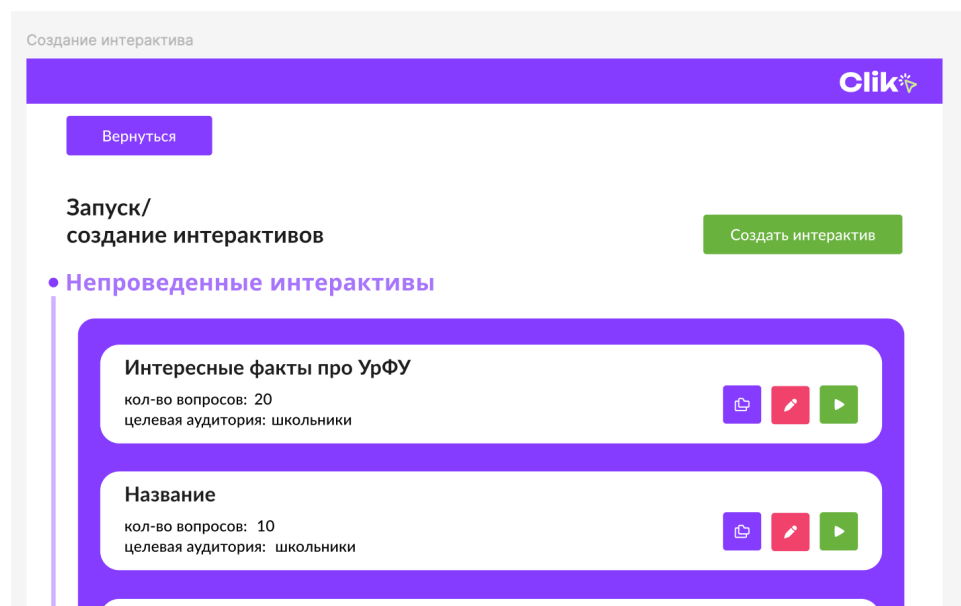


Рисунок 9 – макет «Создание интерактива»