

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ОТЧЕТ

по производственной практике (эксплуатационной)
(вид практики)

на (в) ООО «Энергоинновационный центр»
(место прохождения практики, указать полное наименование организации)

обучающегося Шарина Романа Антоновича

Курс 3 Группа ВМК-20

Направление подготовки: 09.03.01 Информатика и вычислительная техника
(код, наименование)

Направленность ОП Вычислительные машины, комплексы, системы и сети

Руководитель практики от университета:

доцент

(должность)

Ольга Валерьевна Валова

(подпись)

(имя, отчество, фамилия)

Руководитель практики от предприятия:

Директор

(должность)

(печать организации)

Иван Флегонтович Суворов

(подпись)

(имя, отчество, фамилия)

Чита

2023

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Забайкальский государственный университет»
(ФГБОУ ВО «ЗабГУ»)

Энергетический факультет

Кафедра информатики, вычислительной техники и прикладной математики

ЗАДАНИЕ

на производственную практику (эксплуатационную)

Студенту: Шарину Роману Антоновичу

Направление подготовки: 09.03.01 Информатика и вычислительная техника

Индивидуальное задание на практику:

1. Исследование концепций проектирования клиент-сервисного приложения на базе языков программирования C# и JavaScript.
2. Анализ путей взаимодействия с внешними сервисами и из API без блокировки собственного сервиса на время выполнения запроса.
3. Прототипирование платформы администрирования Telegram-каналов и базовых необходимых её систем, а также общая оценка реализуемости и сроков реализации проекта.
4. Описание предметной области.
5. Составление технического задания.
6. Написание и защита отчета по практике.

Дата выдачи задания 19.06.2023 г.

Руководитель от университета _____ О.В. Валова
(подпись, расшифровка подписи)

Задание принял к исполнению 19.06.2023 г.

Подпись студента _____ / Р. А. Шарин /
(И.О.Ф.)

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Описание предметной области	5
2 Техническое задание	8
ЗАКЛЮЧЕНИЕ	10
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	11

ВВЕДЕНИЕ

Для построения некоторых сложных веб-сервисов особенно тех, которые должны выполняться внутри браузера, может потребоваться использовать подход деления этого приложения на Backend и Frontend части [1].

Такой подход позволяет достичь максимальной гибкости обеих частей приложения, но является максимально сложным в реализации для программиста-одиночки, ведущего проект, так как требует от него одновременно поддерживать два, по сути, разных проекта, использующих разные языки программирования и паттерны проектирования.

Так как проект делится на две кардинально разные части имеет смысл применять для их написания разные языки программирования. Для Frontend-части будет использована сборка из языков HTML+CSS+JS [3] при поддержке фреймворка React [4] исполненного на базе JavaScript надстройки под названием TypeScript [5]. Backend-часть же будет написана на языке программирования C#, основанном на .Net Core 7.0 [2].

1 Описание предметной области

Перед тем, как производить базовое деление программной части на Backend и Frontend части нужно определить их назначение. Задача Frontend-части приложения сводится к взаимодействию с пользователем, то есть она служит некоей прослойкой, которая должна передавать команды от пользователя к Backend и обратно, результат работы Backend [1] демонстрировать пользователю. Соответственно, Frontend-часть должна содержать User Interface, и средства взаимодействия с API Backend. Backend является ядром программы и должен выполнять все те задачи, которые ставит перед ним Frontend. Поэтому у Backend может не быть никаких средств взаимодействия с пользователем, так как ему нужны только средства взаимодействия с Frontend.

Язык программирования C# при использовании платформы .NET Core 7.0 [2] обеспечивает достаточно высокий уровень работы с аппаратной частью компьютера, благодаря чему не возникает потребности в реализации базового инструментария, из-за чего экономиться большее количество времени и сил разработчика. Однако же, C# находится всё ещё на достаточно низком уровне, чтобы сохранить некоторые базовые преимущества C-подобных языков программирования, к которым относятся, статическая типизация, компилируемость исходного кода, а не интерпретируемость, достаточно высокая скорость работы на больших массивах данных. Указанные мною параметры языка делают его идеальным выбором для написания Backend-части программного обеспечения.

Так как одной из поставленных целей является возможность приложения запускаться в браузере, выбор средств разработки Frontend-части достаточно ограничен и вращается лишь вокруг выбора фреймворков и паттернов, которые в своей основе каким-то образом модифицируют базовый набор языков программирования веб-разработчика HTML, CSS, JavaScript [3]. В данном случае, был выбран фреймворк React [4], который используется в подавляющем

большинстве крупных проектов, обеспечивает максимальное удобство разработки и масштабируемость веб-сервиса, однако имеет очень высокую сложность понимания для начинающего веб-разработчика, а также общий вес Frontend-части значительно увеличивается из-за размеров самого React. Так же неоднозначной частью языка JavaScript является динамическая типизация, которая не очень удобна и привычна для разработчика, пишущего свои программы на С-подобном языке программирования. Для того чтобы добавить какую-никакую статическую типизацию при написании Frontend-части, было решение добавить в проект TypeScript [5].

Общий функционал минимальной версии проекта можно оценить по следующей диаграмме:

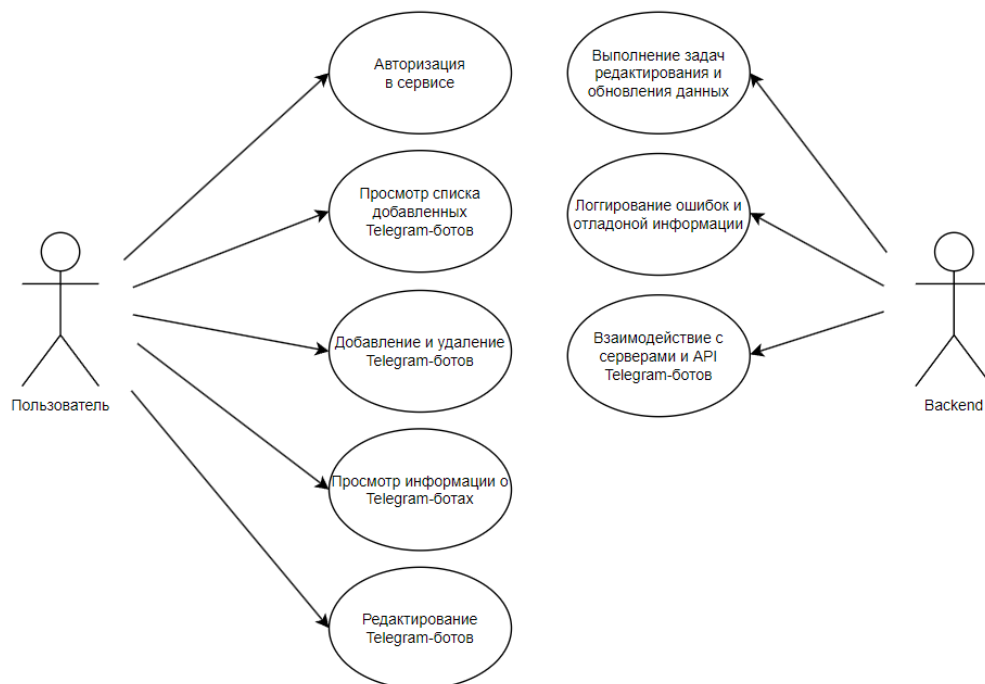


Рисунок 1 - UML диаграмма вариантов использования

Пользователь сможет:

- 1) Авторизоваться в сервисе;
- 2) Просмотреть свой список добавленных Telegram-ботов [6];
- 3) Добавить новых или удалить старых Telegram-ботов;
- 4) Просматривать информацию о Telegram-ботах;
- 5) Редактировать данные Telegram-ботов.

Задачи программного обеспечения же сводятся к задачам Backend-службы, которая должна быть непрерывно запущена и производить необходимые пользователю взаимодействия с серверами Telegram и Telegram Bot API. К основным задачам Backend-службы относятся:

- 1) Выполнение задач редактирования и обновления данных;
- 2) Логгирование ошибок и отладочной информации;
- 3) Взаимодействие с серверами и API Telegram-ботов.

2 Техническое задание

2.1 Введение

«Сервис администрирования Telegram-канал средствами Telegram-ботов» должен стать коммерчески успешным продуктом и занять свою нишу на рынке панелей и средств администрирования платформы Telegram.

2.2 Назначение разработки

Инструмент в качестве прототипа предназначен для анализа общей реализуемости и выполнимости более сложных задач взаимодействия и более глубокой интеграции с сервисами Telegram.

2.3 Требования к функциональным характеристикам

Базовый прототип этого сервиса должен обладать следующими базовыми функциональными характеристиками для пользователей:

- 1) Авторизация пользователей в сервисе;
- 2) Просмотр списка уже добавленных Telegram-ботов;
- 3) Добавление и удаление Telegram-ботов;
- 4) Просмотр информации о Telegram-ботах;
- 5) Редактирование Telegram-ботов.

В фоновых службах Backend-части должен быть реализован следующий базовый функционал:

- 1) Неблокируемое выполнение задач, связанных с обращением к API внешних сервисов. То есть интерфейс должен иметь возможность не блокироваться на несколько секунд каждый раз, как пользователь захочет что-то делать;

- 2) Надёжное хэширование паролей и шифрование и чувствительных личных данных пользователя, которые не могут быть хэшированы;
- 3) Отказоустойчивость и возможность стабильной работы под умеренными и серьёзными нагрузками на сервис.

2.4 Требования к составу и параметрам техническим средствам

Минимальные технические требования:

- 1) Процессор Intel или AMD тактовой частотой выше 2 ГГц;
- 2) Объем оперативной памяти: 4 Гб;
- 3) Свободное дисковое пространство: 2 Гб;
- 4) Стабильное беспроводное или кабельное Интернет-соединение;
- 5) Разрешение экрана не менее 1024x768.

Оптимальные технические требования:

- 1) Процессор Intel или AMD с тактовой частотой не менее 2,5 ГГц;
- 2) Объём оперативной памяти: более 4 Гб;
- 3) Свободное дисковое пространство: более 2 Гб;
- 4) Стабильное кабельное Интернет-соединение;
- 5) Разрешение экрана: более 1366x768.

2.5 Требования к информационной и программной совместимости

Сервис должен корректно отображаться в браузерах Google Chrome версии 98 и выше, запущенных на компьютерах под управлением платформы Windows 10.

2.6 Требования к программной документации

Программная документация должна содержать руководство пользователя.

ЗАКЛЮЧЕНИЕ

Программная реализация прототипа сервиса взаимодействия с внешними API позволит открыть дорогу к реализации платформы администрирования Telegram-каналов с дополнительным развитием средств и алгоритмов клиент-серверного взаимодействия.

Разработка же платформы администрирования Telegram-каналов позволит, как минимум, облегчить публикацию отложенного во времени контента, а, как максимум, даст пользователям, которые не знакомы с программированием, возможность создавать и размещать Telegram-ботов с необходимым им функционалом.

В целом, разработка платформы администрирования может стать коммерчески-успешным проектом, который восполнит недостаточные и ограниченные инструменты администрирования Telegram-каналов и займёт свою нишу на рынке платформ администрирования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Фронтенд или бэкенд: по какому пути в разработке пойти – Текст: электронный // practicum.yandex.ru: [сайт] – 2023. – URL: <https://practicum.yandex.ru/blog/chem-otlichaetsya-backend-i-frontend/> (дата обращения 03.07.2023).

2. Что такое .NET? Введение и обзор // learn.microsoft.com: [сайт] – 2023. – URL: <https://learn.microsoft.com/ru-ru/dotnet/core/introduction> (дата обращения 05.07.2023)

3. HTML, CSS и JavaScript в вебе // vc.ru: [сайт] – 2023. – URL: <https://vc.ru/flood/50683-html-css-i-javascript-v-vebe-poymut-dazhe-chayniki> (дата обращения 07.07.2023)

4. Introducing react.dev // react.dev: [сайт] – 2023. – URL: <https://react.dev/blog/2023/03/16/introducing-react-dev> (дата обращения 10.07.2023).

5. Учебник TypeScript // code.mu: [сайт] – 2023. – URL: <https://code.mu/ru/javascript/typescript/book/prime/> (дата обращения 10.07.2023).

6. Telegram Bot API // core.telegram.org: [сайт] – 2023. – URL: <https://core.telegram.org/bots/api> (дата обращения 10.07.2023).