

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 телеграм-бота для дистанционного банковского обслуживания в  
NeoBank»

по дисциплине «Проектный практикум»

Заказчик: Фамилия И.О.

Бушмелев П

Куратор: Фамилия И.О.

Бушмелев П

ученая степень, ученое звание, должность

Студенты команды arbyzyata

Зиновьев М.В.

Фамилия И.О.

Трофимов Н.А.

Фамилия И.О.

Фамилия И.О.

Ежов Н.О.

Фамилия И.О.

Сырков П.Г.

Екатеринбург, 2025

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Основная часть .....	5
2 Вклад участников команды .....	6
ЗАКЛЮЧЕНИЕ .....	8
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	10
ПРИЛОЖЕНИЕ А .....	11
ПРИЛОЖЕНИЕ Б.....	12
ПРИЛОЖЕНИЕ В .....	13
ПРИЛОЖЕНИЕ Г .....	15
ПРИЛОЖЕНИЕ Д .....	16

## ВВЕДЕНИЕ

В условиях стремительного развития цифровых технологий и растущих ожиданий пользователей в отношении удобства и скорости финансовых операций, особую актуальность приобретает создание эффективных и доступных инструментов взаимодействия с банковскими сервисами. Одним из наиболее востребованных решений сегодня являются чат-боты, интегрированные в привычные пользователю платформы — такие как Telegram.

Цель данного проекта заключается в разработке Telegram-бота для клиентов NeoBank — современного цифрового банка, ориентированного на удобство, прозрачность и оперативность финансового обслуживания. Бот должен позволить пользователям осуществлять базовые операции со своими банковскими счетами напрямую из мессенджера: просматривать баланс, отслеживать последние транзакции, переводить средства между своими счетами и другим получателям, а также получать уведомления о состоянии счёта и движении средств.

Необходимость такого инструмента обусловлена как потребностью в повышении качества клиентского сервиса, так и стремлением к снижению нагрузки на традиционные каналы поддержки. Telegram-бот обеспечит круглосуточный, быстрый и интуитивно понятный доступ к функциям интернет-банкинга без необходимости установки дополнительных приложений.

Проект ориентирован на широкую аудиторию пользователей NeoBank, включая как частных лиц, так и представителей малого бизнеса, которым важна мобильность и простота управления своими финансами. Разработка подобного программного продукта также открывает возможности для масштабируемости и внедрения дополнительных функций в будущем — от консультаций по продуктам до проведения инвестиционных операций.

Ожидается, что по завершении проекта будет создан стабильный, безопасный и удобный в использовании Telegram-бот, полностью интегрированный с системой NeoBank. Он станет дополнительным каналом обслуживания клиентов, повысит удовлетворенность пользователей, улучшит доступность банковских услуг и укрепит имидж банка как технологичного и клиентоориентированного учреждения.

## 1 Основная часть

В рамках выполнения проекта по созданию Telegram-бота для дистанционного банковского обслуживания клиентов NeoBank команда провела полный цикл разработки программного продукта — от анализа требований до тестирования и документирования. На начальном этапе был проведён детальный разбор ожиданий заказчика (компании NeoFlex: требовался безопасный, доступный и устойчивый к нагрузке бот, способный выполнять ключевые операции (авторизация, просмотр баланса и списка операций, взаимодействие с счетами). На основе этих требований был сформирован план работ в формате backlog, включающий следующие ключевые задачи:

1. составлена схема работы бота (рисунок 3),
2. создана схема диалогового интерфейса (рисунок 1)
3. настройка окружения и подключение к Telegram Bot API,
4. интеграция с API NeoBank (процесс описан рисунком 4),
5. реализация авторизации, аутентификации пользователей (рисунок 2),
6. вывод информации о балансе и транзакциях,
7. функции перевода средств и взаимодействия со счетами,
8. проведение тестирования,
9. создание документации и подготовка релиза.

## **2 Вклад участников команды**

Распределение обязанностей строилось с учётом компетенций и интересов членов команды. Трофимов Никита Алексеевич и Ежов Никита Олегович отвечали за техническую реализацию: интеграцию с API NeoBank, логику взаимодействия через Telegram, настройку микросервисной архитектуры и подключение OpenAPI. Зиновьев Михаил Владимирович проводил системное тестирование, включая модульные и интеграционные тесты, и проверял корректность работы бота на каждом этапе. Сырков Павел Григорьевич выполнял функции координатора — следил за соблюдением контрольных точек, готовил промежуточные отчёты, участвовал в тестировании и оказывал помощь в реализации отдельных участков кода. Разработка велась по инкрементной методологии со спринтами: на каждом этапе проводилось планирование, ревью кода и тестирование. Модель «авторизация – работа с дебетовыми счетами – вклады – кредиты – перевод средств – просмотр истории переводов» позволила масштабировать систему и разделить зоны ответственности. По завершению проекта команда закончила разработку Telegram-бота с реализованными основными функциями, протестированной архитектурой и сопроводительной документацией.

## ПРАКТИЧЕСКАЯ ЧАСТЬ РАБОТЫ

Практическая реализация Telegram-бота для NeoBank была построена по следующему алгоритму: сначала развернуто тестовое окружение бота, затем настроен приём и обработка входящих обновлений от Telegram. На основе схемы авторизации (Приложение Б) реализована последовательность проверки пользователя: получение кода из мессенджера – передача в API NeoBank – верификация и сохранение токена сеанса.

После успешной авторизации бот переходит к основным операциям, описанным в Приложении В:

1. при получении команды бот формирует запрос к REST-интерфейсу NeoBank,
2. результаты (баланс, список счетов, выписка операций) преобразуются в удобный для чтения формат,
3. по событию «перевод средств» запускается диалог уточнения реквизитов и суммы.

Логика взаимодействия бота с банковским API оформлена в виде микросервисов, каждый из которых отвечает за отдельную группу операций (авторизация, счета, вклады, кредиты, переводы). Межсервисное взаимодействие осуществлялось по схеме из Приложения Г, а конечный интерфейс, показанный пользователю, согласован с макетом в Приложении Д.

Функциональность проверялась во время каждого спринта. Ручное тестирование гарантировало удобство диалога и корректность обработки ошибок. В результате сложился компактный и последовательный алгоритм работы бота, обеспечивающий быстрый ответ на команды пользователя и надёжную интеграцию с NeoBank.

## ЗАКЛЮЧЕНИЕ

В результате проделанной работы по созданию Telegram-бота для клиентов NeoBank можно с уверенностью сказать, что программный продукт в целом соответствует ключевым требованиям, предъявленным заказчиком. Бот обеспечивает необходимую функциональность для удалённого управления банковскими счетами — от авторизации и просмотра баланса до выполнения переводов внутри банка. Такой набор функций позволяет удобно и оперативно взаимодействовать с банком через привычный мессенджер.

Анализ результатов тестирования позволил выявить высокий уровень стабильности и корректности работы основных функций, что свидетельствует о продуманной архитектуре и тщательном подходе к разработке. Несмотря на это, в ходе тестирования были зафиксированы отдельные дефекты, в основном связанные с обработкой нестандартных сценариев. Эти недостатки не нарушают общую работоспособность системы, однако могут повлиять на восприятие пользовательского опыта, особенно при интенсивных нагрузках. Таким образом, качество продукта можно оценить как удовлетворительное с потенциалом для дальнейшего улучшения.

Для развития проекта видится перспективным расширение интерфейса бота за счёт интеграции полноценного фронтенда в виде Telegram Mini App - небольшого веб-приложения внутри мессенджера, которое позволит реализовать более сложные и визуально насыщенные сценарии взаимодействия. Это улучшит удобство использования и создаст почву для добавления новых функций боту. Также целесообразно продолжать совершенствовать систему обработки ошибок и оптимизации производительности, чтобы повысить устойчивость к нагрузкам и повысить качество обслуживания.

Работая над проектом, команде удалось успешно поработать с API банка, применить свои знания на практике. В целом, выполненный проект является удачной демонстрацией возможности создания востребованного



банковского инструмента с использованием современных технологий и методологий разработки, а предложения по дальнейшему развитию заложат прочную основу для трансформации Telegram-бота в многофункциональный и удобный сервис, способный отвечать меняющимся требованиям рынка и клиентов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Что такое Swagger и как он облегчает работу системного аналитика с API / яндекс практикум <https://practicum.yandex.ru/blog/chto-takoe-swagger/>
2. Использование диаграммы вариантов использования UML при проектировании программного обеспечения / habr <https://habr.com/ru/articles/566218/>
3. Как тестировать API, или Postman / Geek brains <https://gb.ru/blog/kak-testirovat-api-postman/>

## ПРИЛОЖЕНИЕ А

### тест-бот NeoBank

Схема диалогового интерфейса бота:

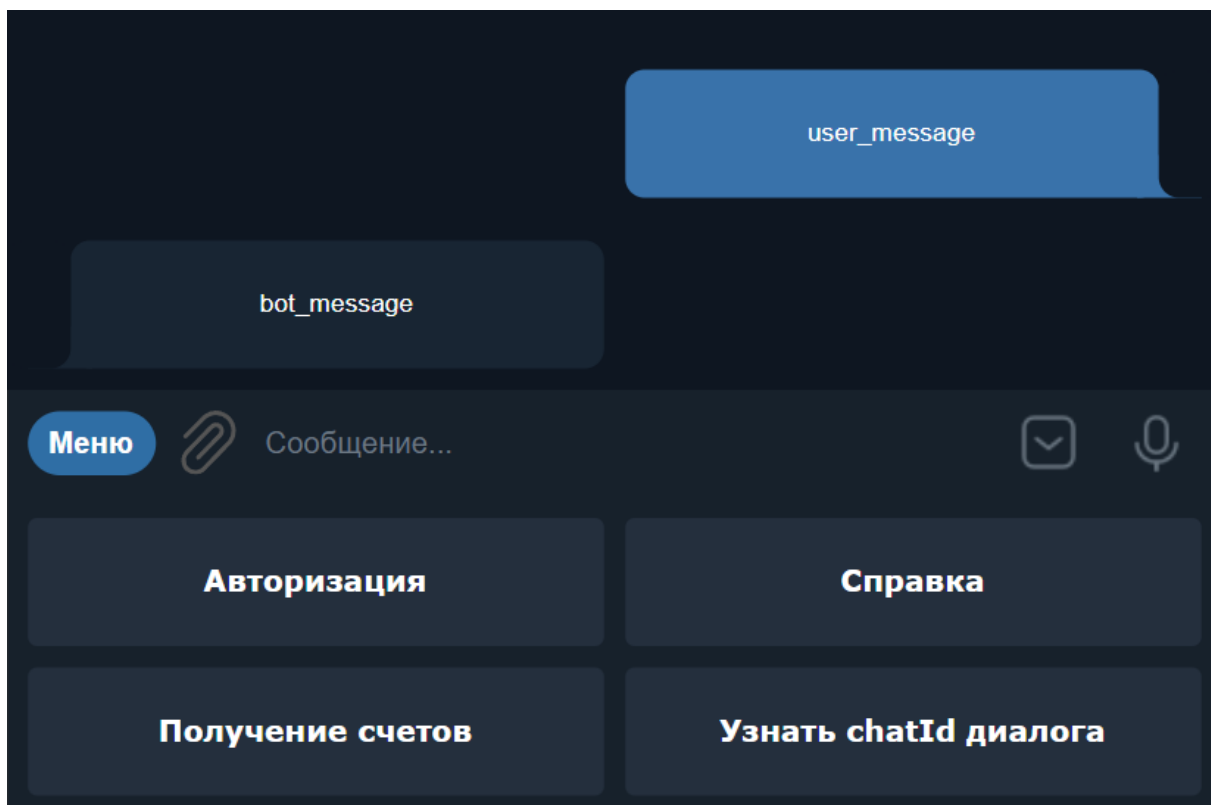


Рисунок 1 – Схема диалогового интерфейса



## **ПРИЛОЖЕНИЕ В**

### **Команды для отправки и ответы бота**

Доступные команды для отправки боту посредством нажатия кнопок интерфейса:

1. Start – присылает пользователю кнопку для авторизации в NeoBank ,
2. Счета – с помощью этой команды происходит взаимодействие с счетами, появляются новые кнопки «Мои счета», «Новый счет», «Закрыть счет», «Назад», «Отмена операции»,
  - 2.1. Мои счета – бот присылает список счетов, страницы которого можно переключать. Каждый счёт имеет в себе номер счёта, ID, баланс, доступные средства, указана валюта баланса, статус счёта и дата его создания,
  - 2.2. Новый счет – бот будет ожидать ввода суммы для нового счёта, затем предложит по кнопке выбрать валюту и будет подтверждение создания,
  - 2.3. Закрыть счет – закрыть можно только пустой счет, поэтому сначала нужно будет перевести с него деньги
  - 2.4. Назад – перейти на главный экран,
3. Вклады – позволяет пользователю работать с вкладами. После нажатия появляются кнопки «Мои вклады», «Новый вклад», «Закрыть вклад», «Назад», «Отмена операции»,
  - 3.1. Мои вклады – позволяет ознакомиться с текущими вкладами пользователя,
  - 3.2. Новый вклад – с помощью ID счета полученного при создании счёта или из списка всех счетов, можно сделать новый вклад. Будет предложено 4 опции – «Идеальный старт», «Турецкий старт», «Блестящий запуск», «Отличное начало». Затем будет выбор процентной ставки (3 месяца – 7%, 6 месяцев – 7.5%, 9 месяцев – 7.7%, 12 месяцев – 8%) и предложение об автопродлении,
  - 3.3. Назад – перейти на главный экран,

4. Кредиты – позволяет пользователю управлять своими кредитами. После вызова появляются кнопки «Мои кредиты», «Новый кредит», «Погасить кредит», «График платежей», «Назад», «Отмена операции»,
  - 4.1. Мои кредиты – позволяют увидеть текущие кредиты и информацию о них,
  - 4.2. Новый кредит – можно взять кредит, выбрав срок с процентной ставкой,
  - 4.3. Погасить кредит – с помощью бота можно погасить кредит согласно графику платежей или сделать это досрочно,
  - 4.4. График платежей – позволяет ознакомиться с графиком платежей по кредиту и увидеть ожидаемые суммы этих платежей,
  - 4.5. Назад – перейти на главный экран,
5. Валюты – бот показывает пользователю доступные виды валют, а также информацию о них, такую как код, номер и символ валюты,
6. Продукты – при нажатии кнопки, бот спросит информацию о каком типе банковского продукта пользователь хочет получить. Из вариантов для выбора есть «Счет», «Вклад», «Кредит». Например, выбор «Кредит» покажет информацию об ипотеке, автокредите и потребительском кредите,
7. Переводы – перевод с счета на счет происходит с помощью ID счетов. Для начала бот спросит ID счета с которого идет отправка, затем ID счета куда поступят средства и сумму перевода.

# ПРИЛОЖЕНИЕ Г

## Процесс работы Telegram-бота

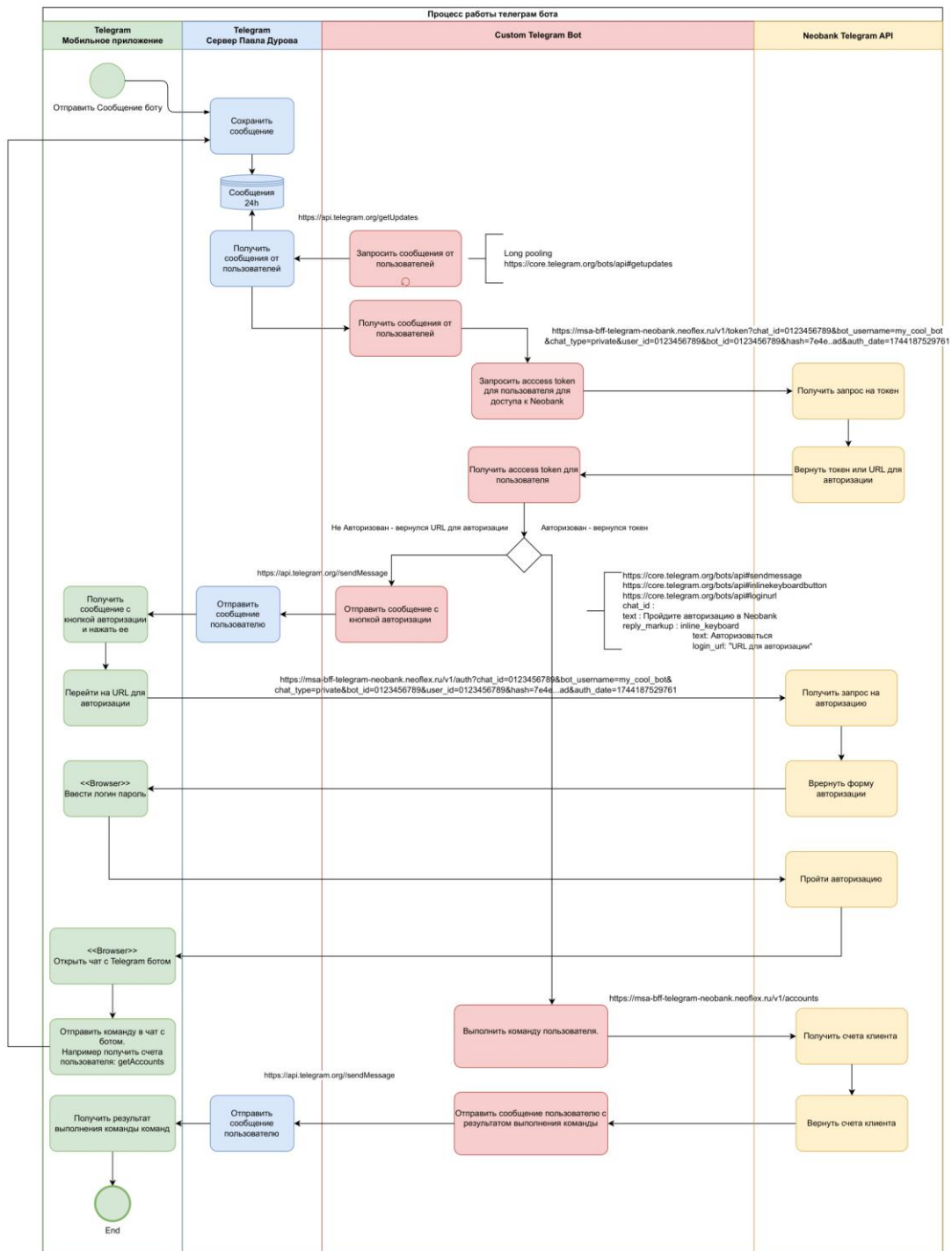


Рисунок 3 – Схема процесса работы бота

## ПРИЛОЖЕНИЕ Д

### Диалоговый интерфейс в конечном варианте

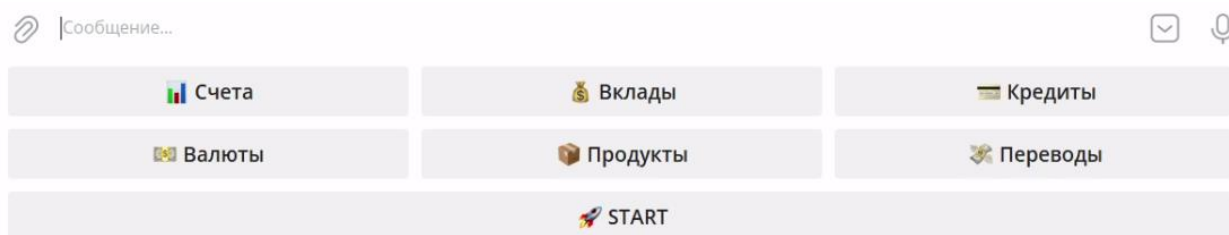


Рисунок 4 – Диалоговый интерфейс в конечном варианте