

# Пермяков Дмитрий Кириллович

iOS Developer



Проживаю в Москве, закончил 4 курс МГТУ им. Н.Э. Баумана, ИУ5 и защитил диплом с оценкой "отлично" на тему "Маркетплейс для продажи тортов на платформе iOS"

В свободное время увлекаюсь кино, спортом и прогулками на свежем воздухе.

Люблю развиваться в различных сферах программирования, изучать самые новые технологии и не останавливаться на достигнутом.

## Навыки

iOS	● ● ● ● ●
Backend	● ● ● ● ●
Soft skills	● ● ● ● ●
Hard skills	● ● ● ● ●
Frontend	● ● ● ● ●
Devops	● ● ● ● ●

## Образование

2021

**Cbeer курс C/C++ МГТУ**

Алгоритмы и структуры данных на C/C++

2023

**Цифровая кафедра МГТУ**

По программе Backend разработчик

Май 2023

**Технопарк (VK-Edu) при МГТУ по курсу iOS**

Мобильная разработка iOS

Сент 2021 - Июнь 2025

**МГТУ им. Н.Э.Баумана**

ИУ, Информатика и вычислительная техника, ИУ5

## Опыт работы

Июнь 2023- Сейчас

**iOS разработчик**

Wildberries

- Продуктовая разработка SDK для ВБ Бизнеса. Многомодульное приложение, написанное на чистом SwiftUI с использованием Tuist для ВБ Банка
- Работал в Дизайн Системе. Разработка кастомных UI компонентов как для UIKit так и для SwiftUI. Покрытие UI и Handler тестами, аналитика. Работа с низкоуровневым api.
- Внедрение акции WB Club в WB Travel. Переход на gRPC и декомпозиция проекта на модули. Встраивание отдельного приложения WB Travel в основное приложение Wildberries.
- Разработка внутреннего приложения для дизайнеров на чистом SwiftUI для тестирования кастомных UI компонентов.
- Чтение докладов и knowledge sharing-сессий на такие темы, как: Backend на Swift+Vapor, gRPC и мобильное приложение, Combine, SwiftUI

Дек 2022- 2023

**Доставка 24**

Freelancing

- Разрабатывал стартап приложение под iOS для оптовой закупки продуктов в команде с Backend и Android разработчиками.
- Клиент серверное взаимодействие по RestAPI на Combine.
- SwiftUI для разработки интерфейса.
- Архитектура MVVM.

## Контакты

+7(916) 855-99-42

dimapermyakov55@gmail.com

**GitHub:**

<https://github.com/mightyK1ngRichard>

**Telegram:**

<https://t.me/mightyK1ngRichard>

# Опыт учебной работы

## Tort&Land

Выпускная квалификационная работа, МГТУ

iOS, MacOS, Backend

Разработал полнофункциональный маркетплейс с возможностью кастомизации тортов, чатов и 3D-просмотра.

- Backend: реализован на Go в микросервисной архитектуре с использованием gRPC, MinIO (для хранения медиа), Nginx, PostgreSQL и системой рефреш-токенов для авторизации (Образы созданы в Docker).
- gRPC использовался для клиент-серверного взаимодействия между backend'ом и iOS/macOS-клиентами.
- iOS-приложение и macOS-десктоп для администратора написаны на SwiftUI. Архитектура MVVM.
- Кэширование реализовано через SwiftData и FileManager
- ARKit использовался для просмотра 3D-моделей тортов
- Встроен real-time чат на WebSocket и система уведомлений через gRPC Streaming
- Приложение декомпозированно на независимые модули.

Проект направлен на создание платформы, где пользователи могут заказывать кастомные торты от частных пекарей, настраивая внешний вид, начинку и параметры, а также взаимодействовать через встроенный чат и просматривать 3D-визуализацию торта до заказа. Для администратора было разработано десктопное приложение для отслеживания и мониторинга статусов заказа и тортов с возможностью перехода по deeplink QR в само приложение.

С кодом и презентацией проекта можно ознакомиться на моём GitHub:

- iOS | <https://github.com/mightyK1ngRichard/bmstu-2025-final-qualifying-work>
- Backend на Go | <https://github.com/mightyK1ngRichard/bmstu-2025-final-qualifying-work-backend-go>

## RealTimeMessenger

Сетевые технологии. Курсовая работа

Система обмена сообщениями с уровневой архитектурой и защитой на физическом уровне

Разработал систему обмена текстовыми сообщениями в реальном времени с выделением трёх уровней: прикладного, транспортного и канального.

Канальный уровень:

Реализовал кодирование и декодирование данных с использованием циклического [7,4] кода, обеспечивающего базовую защиту от искажений при передаче.

- Транспортный уровень: настроил взаимодействие между сервисами, включая передачу сообщений в отдельный сервис, уже интегрированный с Apache Kafka.
- Производил разбиение сообщений на байты, последующую сборку сообщений после обработки и возврат их мобильному клиенту.

Прикладной уровень:

- На стороне клиента использовал WebSocket-сервер, реализованный на Swift с использованием Vapor,
- Собранные сообщения доставлялись в прототип мобильного мессенджера (аналог Telegram), написанного на SwiftUI.

С остальным множеством моих проектов уже можно ознакомиться на моём GitHub