CoolBet

- Команда:
  - Сергей Свистунов
  - Михаил Скородумов
  - Илья Зиновкин
  - Анастасия Смолянинова
  - Андрей Николаев ☹

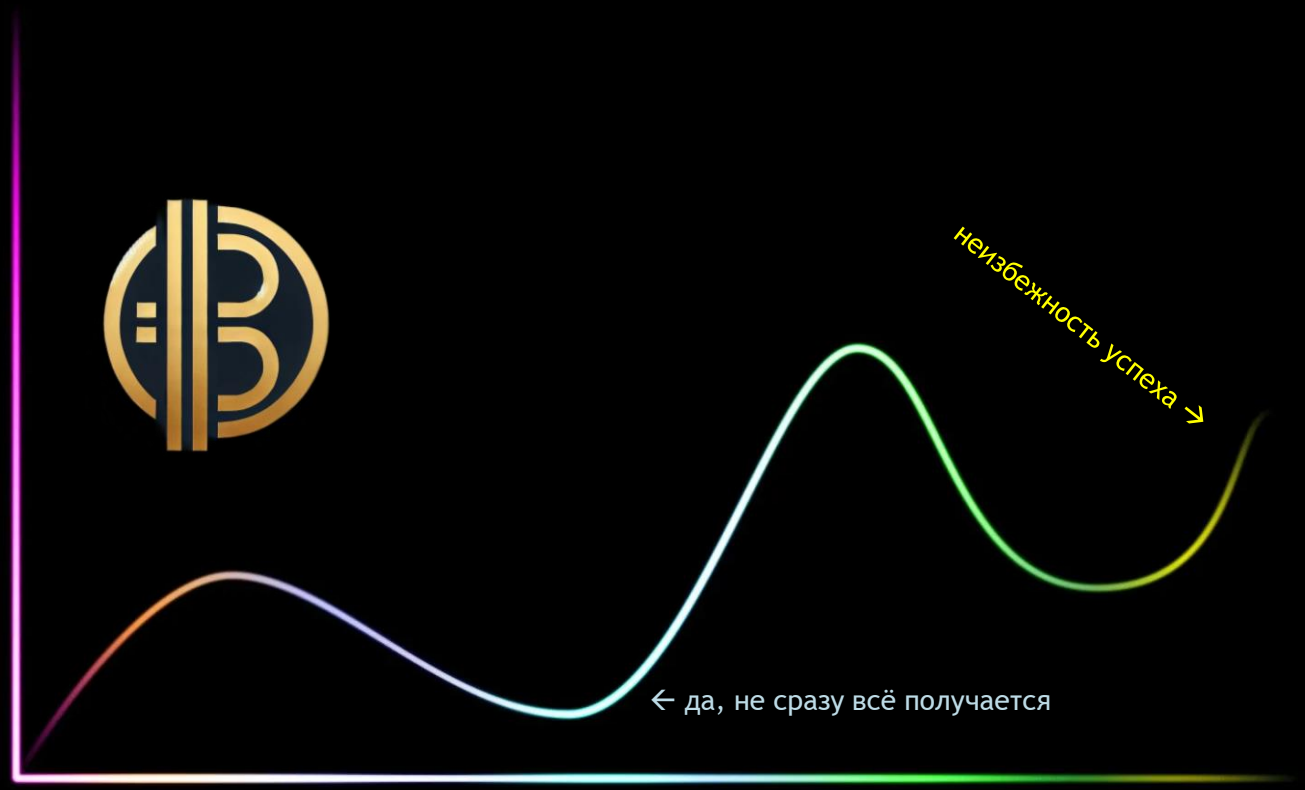
Проект CoolBet

CoolBet была основана в 2024 году группой энтузиастов, которые стремились сделать ставки на события простыми и доступными для всех. Наша платформа была создана для того, чтобы каждый пользователь мог с лёгкостью делать ставки на исходы интересующих его событий — от спортивных матчей до политических выборов и экономических прогнозов. CoolBet выделяется из стада благодаря удобному интерфейсу, прозрачным условиям ставок и честному способу расчёта выигрышей.

Мы верим, что ставки должны приносить удовольствие, и стараемся сделать их максимально понятными и безопасными для наших пользователей.

«Самая надежная платформа для ставок на различные события.
Мы предлагаем пользователям возможность делать ставки на исходы интересных и актуальных событий по всему миру!»

Цели проекта

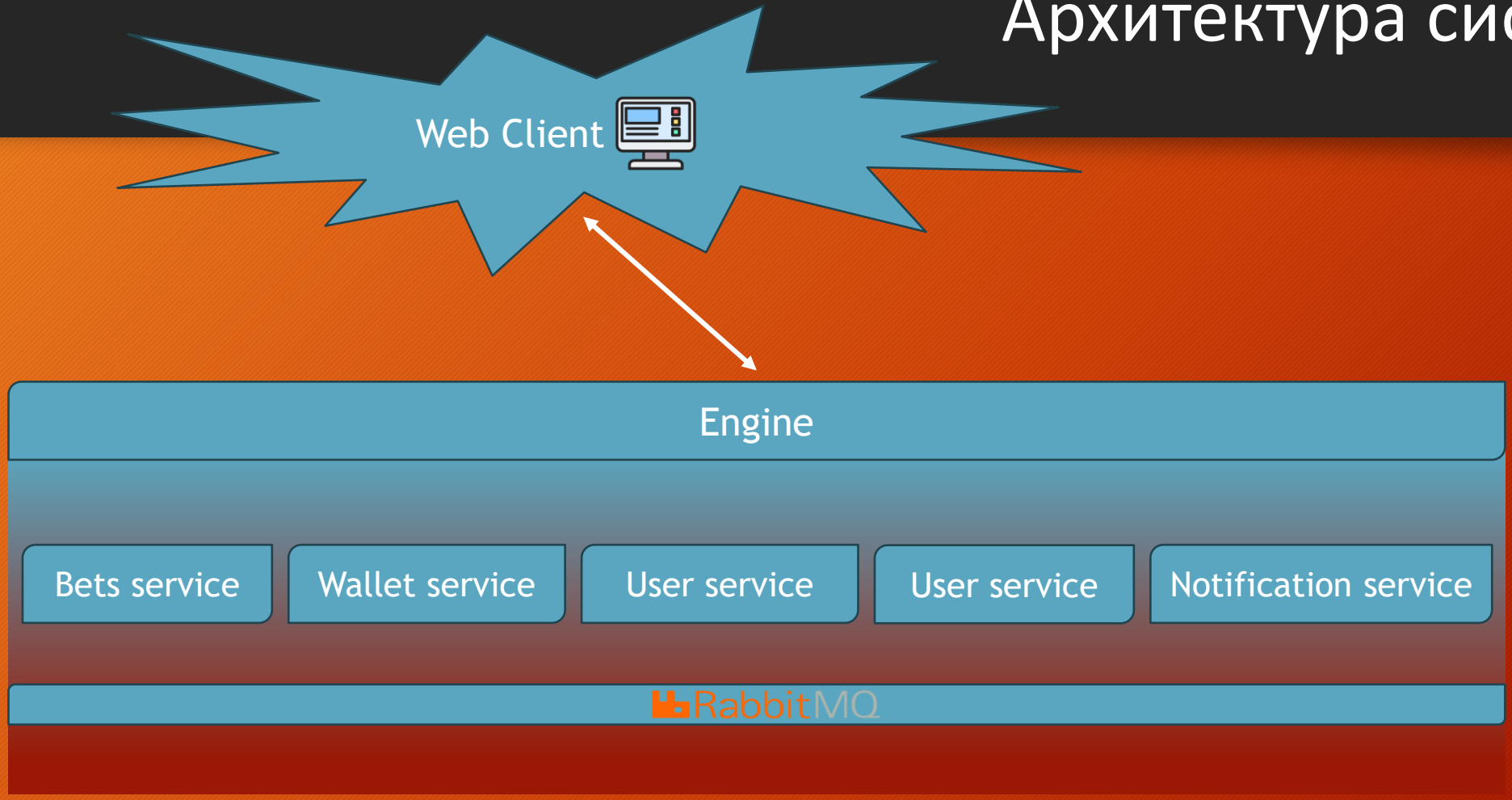


- Реализовать и продвинуть систему ставок на события с уникально понятным расчётом ставок
- Использовать передовые технологий для обеспечения превосходного пользовательского опыта
- Создать решение, обеспечивающее надёжность, масштабируемость и поддерживаемость
- Обеспечить максимальную доступность системы – ничто не должно мешать пользователям расстаться с деньгами

Прогресс по проекту

CoolBets		
Подготовка		
Сделать первоначальный план проекта	Готово, поддержка	Сергей
Сбор требований и написание спецификации на решение	Готово, поддержка	Сергей
Проработка и отрисовка архитектуры решения	Готово, поддержка	Все
Доработать и положить в репозиторий схему решения	Готово, поддержка	Андрей
Нарисовать и задеплоить схему БД для каждого сервиса	Готово, поддержка	Сергей
Engine		Сергей
Проработка требований к сервису	В работе	
Написать интерфейсы для сервисов с реализациями-фейками	Готово, поддержка	
Разработка схемы взаимодействия с сервисами	Готово, через коннекторы	
Реализация логирования Serilog	В работе	
Имплементация функционала	В работе	
Добавление кеширования		
Написание unit-тестов для разработки	План	
Тесты, которые должны запускаться при деплое		
UI		
Написание пользовательского интерфейса на MVC ASP.NET	В работе	Сергей
Добавление динамических компонентов React	План	Сергей
Bets service		Анастасия
Проработка требований к сервису	В работе	
Написание контракта сервиса	В работе	
Спецификация необходимых DTO	В работе	
Разработка нормализованной структуры БД	В работе	
Имплементация функционала	План	
Реализация логирования Serilog	План	
Добавление кеширования		
Реализация взаимодействия с RebbitMQ	План	
Написание unit-тестов для разработки	План	
Тесты, которые должны запускаться при деплое		
Wallet service		Сергей
Проработка требований к сервису	В работе	
Написание контракта сервиса	Готов драфт контракта	
Спецификация необходимых DTO	В работе	
Разработка нормализованной структуры БД	План	
Имплементация функционала	План	
Реализация логирования Serilog	План	
Добавление кеширования		
Реализация взаимодействия с RebbitMQ	План	
Написание unit-тестов для разработки	План	
Тесты, которые должны запускаться при деплое		
User service		Илья
Проработка требований к сервису	Готово	
Написание контракта сервиса	Готово, поддержка	
Спецификация необходимых DTO	Готово, поддержка	
Разработка нормализованной структуры БД	Готово	
Имплементация функционала	В работе	
Реализация логирования Serilog	План	
Добавление кеширования	План	
Реализация взаимодействия с RebbitMQ	План	
Написание unit-тестов для разработки	План	
Тесты, которые должны запускаться при деплое		
Гайдлайны по реализации функционала работы с пользователями в решении	План	
Notification service		Миша
Проработка требований к сервису	Готово	
Написание контракта сервиса	Готово, поддержка	
Спецификация необходимых DTO	Готово, поддержка	
Разработка нормализованной структуры БД	Готово	
Имплементация функционала	В работе	
Реализация логирования Serilog	План	
Добавление кеширования	План	
Реализация взаимодействия с RebbitMQ	План	
Написание unit-тестов для разработки	План	
Тесты, которые должны запускаться при деплое		
Инфраструктура		
Настройка контейнеризации	В работе	Илья
Настройка пайплайнов деплоя		Сергей
Закупка и поддержка сервера для хостинга		
Прочее		
Сделать скетч презентации проекта, задеплоить в docs для PETPO1	Готово	Настя
Сделать скетч презентации проекта, задеплоить в docs для PETPO2	Готово	Сергей
Проведение UAT		Все
Подготовка демонстрации и материалов к защите		Сергей

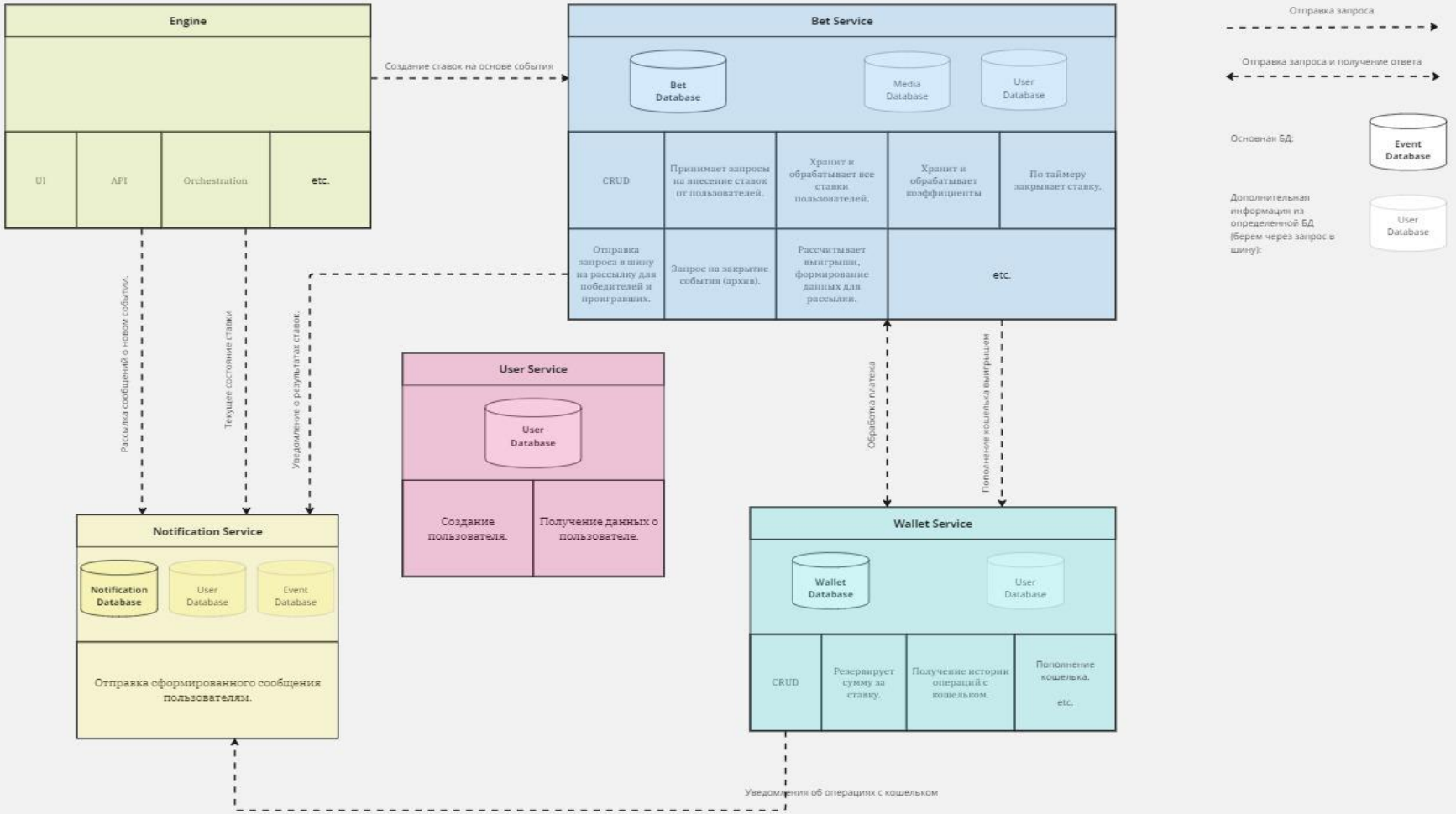
Архитектура системы



Комментыы:

- Микросервисная архитектура
- Web Client – ASP.NET + React
- Engine – движок, использует сервисы и снабжает UI
- Bets service – букмекер, управляет событиями, исходами, ставками и клирингом
- Wallet service – казначей, отвечает за ведение счетов и выплаты
- User service – вышибала, проверяет креденшлс и ведёт списки клиентов
- Notification service – почтальон, занят рассылкой писем и прочих сообщений
- RabbitMQ – канал для асинхронного общения с сервисами

Схема взаимодействия микросервисов (дополняется)



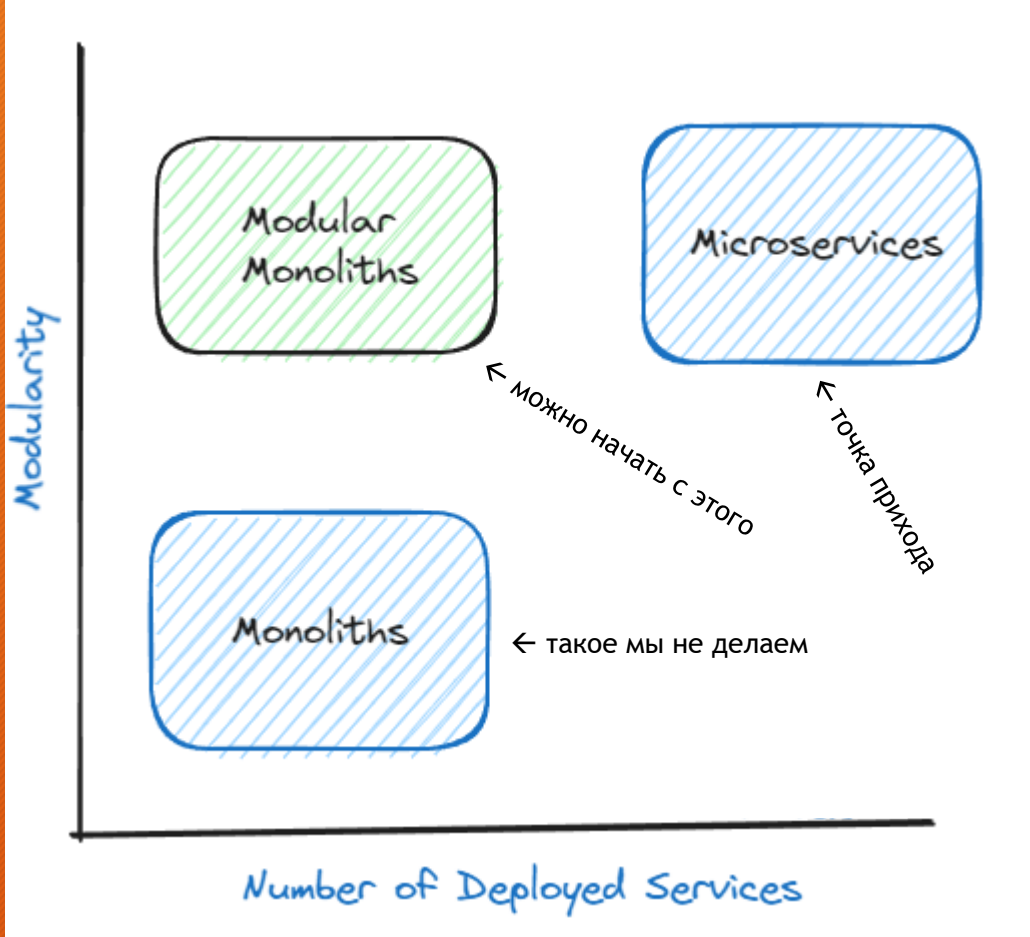
Комменты:

- Микросервисная архитектура
- Web Client – ASP.NET + React
- Engine – движок, использует сервисы и снабжает UI
- Bets service – букмекер, управляет событиями, исходами, ставками и клирингом
- Wallet service – казначей, отвечает за ведение счетов и выплаты
- User service – вышибала, проверяет креденшлс и ведёт списки клиентов
- Notification service – почтальон, занят рассылкой писем и прочих сообщений
- RabbitMQ – канал для асинхронного общения с сервисами

Ответственные:

- UI – Сергей Свистунов
- Engine – Сергей Свистунов
- Bets service – Анастасия Смолянинова
- Wallet service – Сергей Свистунов
- User service – Илья Зиновкин

Концепция коннекторов в Engine



Процесс создания решения:

- Выделяются сервисы, которые отвечают за свою изолированную бизнес-область
- Для каждого сервиса разрабатывается контракт с ДТОшками
- В Engine по этим контрактам сервисы по DI используются в работе, инжектятся в контроллеры и т.д.
- На первом этапе имплементация этих контрактов делается в виде фейков
- Дальше, фейки заменяются на коннекторы, которые общаются с внешними сервисами

Преимущества такого подхода:

- Возможность реализации движка и пользовательского интерфейса с первого дня
- Пользователи сервисов ничего не знают о их имплементации: фейк → модульный монолит → микросервисы
- Взаимодействие с внешними микросервисами осуществляется через коннектор, поэтому появляется возможность менять способ взаимодействия с ними

Недостатки:

- TBD

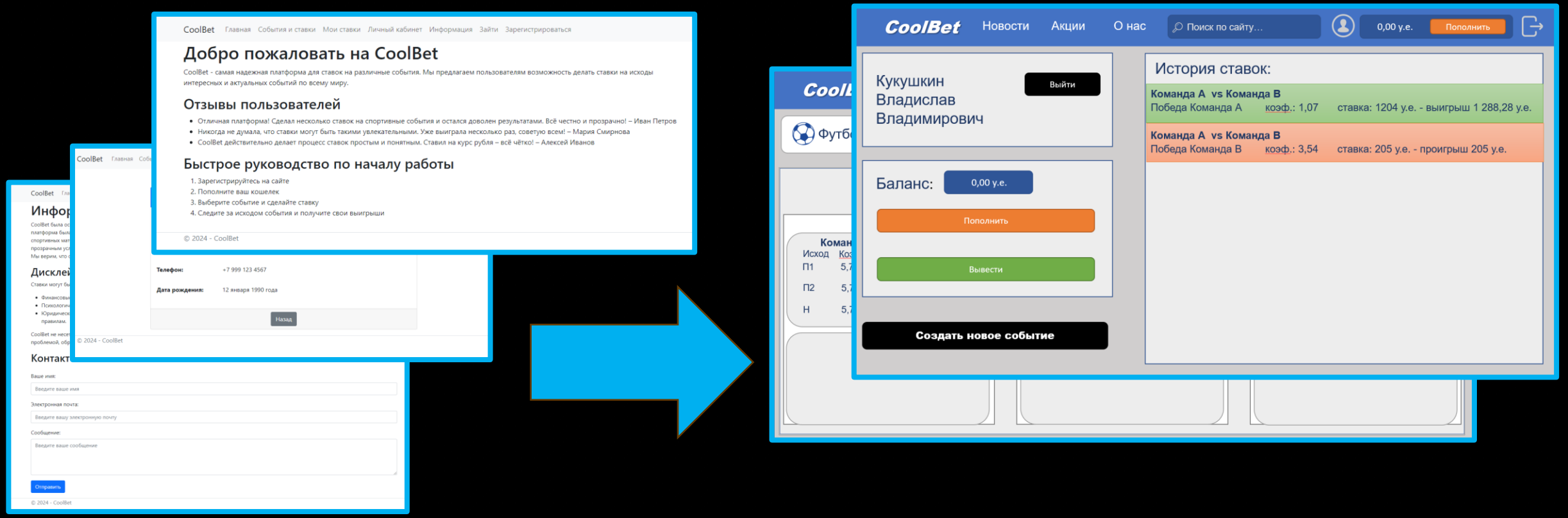
Технологии

- ASP.NET Core 8
- React
- Entity Framework
- SQL Server
- xUnit
- Docker, Docker Compose
- Swagger
- Serilog
- MediatR
- Redis
- RabbitMQ

Пользовательский интерфейс

- Пользовательский интерфейс реализуется в два этапа:

  1. ASP.NET **MVC**
  2. Добавление **React** компонентов
  … 3. **SPA**



Функциональные требования

По проекту ведётся документация:

- Функциональные требования
- Пользовательские истории
- Архитектура системы
- Спецификация UI

- Пользователи должны иметь возможность регистрироваться и входить в систему.
- Администраторы должны иметь возможность создавать, редактировать и удалять события.
- Пользователи должны иметь возможность делать ставки на доступные события.
- Пользователи должны иметь доступ к истории своих ставок и результатам.
- Внутренняя валюта (УЕ):
  - Система должна использовать внутреннюю валюту, условную единицу - "УЕ".
  - Должна быть возможность обновлять счета пользователей, когда они приобретают УЕ.
  - У пользователей должна быть возможность заказать конвертацию и вывод УЕ во вне.

Пользовательские истории

Регистрация и аутентификация:

Как пользователь, я хочу регистрироваться на сайте, чтобы получить доступ к функционалу ставок.

Как пользователь, я хочу восстанавливать пароль, чтобы получить доступ к моему аккаунту в случае его утери.

Как пользователь, я хочу входить в систему с использованием двухфакторной аутентификации для повышения безопасности моего аккаунта.

События и ставки:

Как пользователь, я хочу просматривать список доступных событий, чтобы выбирать, на что ставить.

Как пользователь, я хочу получать подробную информацию о каждом событии, чтобы делать информированные ставки.

Как пользователь, я хочу делать одиночные ставки, чтобы поставить деньги на один из исходов события.

Как пользователь, я хочу иметь возможность отменять свою ставку до начала события, если я передумал.

Профиль пользователя

Как пользователь, я хочу просматривать и редактировать информацию своего профиля, чтобы поддерживать актуальность данных.

Как пользователь, я хочу просматривать историю своих ставок, чтобы анализировать свои прошлые решения.

Тестирование

xUnit.net

Стратегия тестирования: Подход к тестированию включает юнит-тесты, интеграционные тесты и тесты интерфейса.

Тест-кейсы: Спецификации для тестирования функциональности регистрации, ставок и управления событиями и др. userStory.

CoolBet

СПАСИБО, ДРУЗЬЯ,
ЗА ВНИМАНИЕ!