**AccTests (v 21.02.24)**

**1.** **Introduction (Введение)**

**1.1** **Purpose**

Данный документ содержит результаты тестирования ИС “Финист”.

**1.2** **References (Ссылки)**

* [*Test Plan*](https://docs.google.com/document/d/1fQfdwJr_TJSS6kTZ6PHw5G0Rnv8ehi4UHLTIq4UINvc/edit)
* [*UC Extended*](https://docs.google.com/document/d/1fQfdwJr_TJSS6kTZ6PHw5G0Rnv8ehi4UHLTIq4UINvc/edit)

**2.** **Test Approach (Подход к тестированию)**

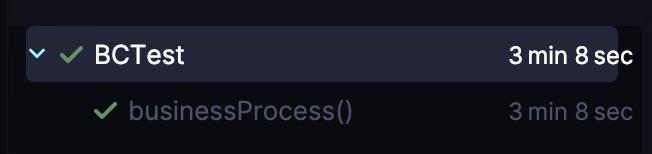
UseCase Testing

| **UC** | **Результат** |
| --- | --- |
| Показ служебной информации о пользователе |  |
| Смена роли |  |
| Комментирование состязания |  |
| Создание отчёта о свадьбе\* |  |
| Удаление аккаунта |  |
| Заполнение анкеты |  |
| Выход |  |
| Покупка стрел |  |
| Показ состязаний |  |
| Запуск стрелы\* |  |
| Отказ от состязания |  |
| Регистрация |  |
| Бан пользователя |  |
| Авторизация\* |  |
| Создание шаблона испытания\* |  |
| Назначение исполнителей |  |
| Голосование за невесту\* |  |

\*архитектурно-значимые UC

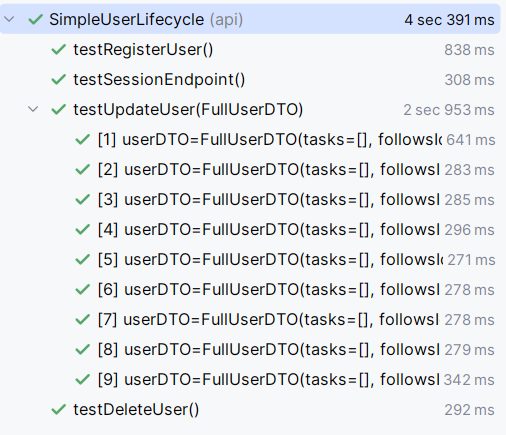
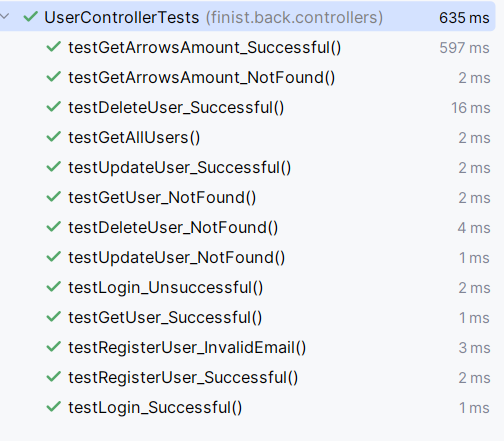
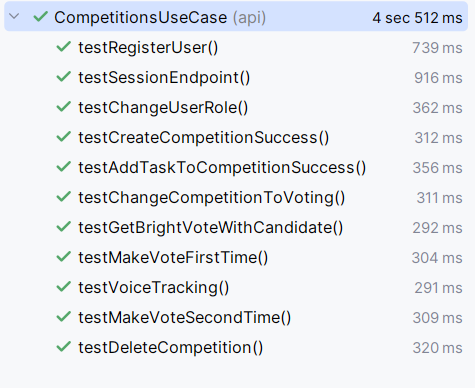
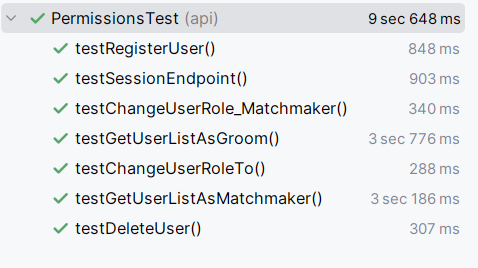
BusinessCase Testing

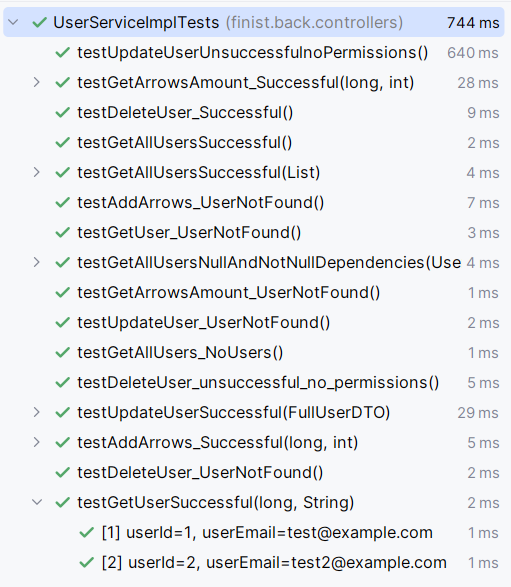
BC-тестирование представлено единым многошаговым тестом.

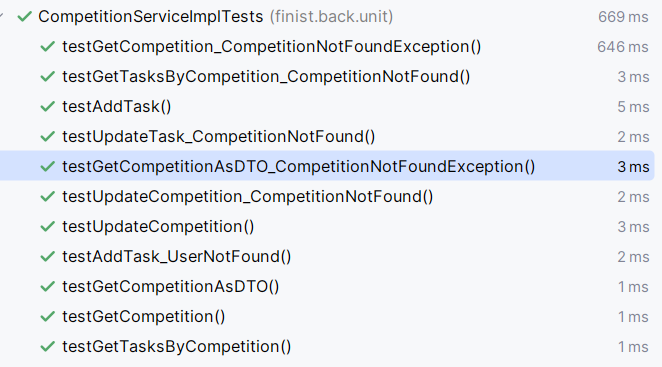


| **Шаг** | **Описание** | **role** | **Тест** |
| --- | --- | --- | --- |
| 1**✅** | Регистрация нового пользователей | BRIDE  GROOM | *var* br = User.*rand*();  *int* brId = registerNewUser(br, "bride");  UCTests.*logout*();  *var* gr = User.*rand*();  *int* grId = registerNewUser(gr, "groom"); |
| 2**✅** | Запуск стрелы | GROOM | pushArrow(gr);  UCTests.*logout*(); |
| 3**✅** | Создание шаблона испытания | MATCHMAKER | createCompTemplate(); |
| 4**✅** | Назначение участников испытания среди тех, кто свободен и испытаний им | MATCHMAKER | *int* newCompId = assignCompetitors(brId, grId); |
| 5**✅** | Заполнение отчётов о ходе испытания | GROOM BRIDE | report(br, newCompId);  report(gr, newCompId); |
| 6**✅** | Перевод состязания в статус=VOTING | MATCHMAKER | toVoting(newCompId); |
| 7**✅** | Голосование за невесту | GUEST  ENEMY  ASSISTANT | voting(newCompId); |
| 8**✅** | Перевод состязания в статус=WAITING\_AGREEMENT | MATCHMAKER | toWaitingAgreement(newCompId); |
| 9**✅** | Дача согласия на свадьбу | GROOM  1BRIDE | agreement(br, newCompId);  agreement(gr, newCompId); |
| 10**✅** | Перевод состязания в статус=MARRIAGE | MATCHMAKER | toMarriage(newCompId); |
| 11**✅** | Написание отчёта о свадьбе | MATCHMAKER | marriageReport(newCompId); |

Functional testing







| Название | Описание | Результат |
| --- | --- | --- |
| testRegisterUser | Неавторизованный пользователь может зарегистрировать аккаунт | Успех |
| testLogin | Пользователь может войти в свой аккаунт | Успех |
| testChangeUserRole | Пользователь системы может редактировать свой аккаунт: изменить роль | Успех |
| testCreateCompetitionSuccess | Пользователь с ролью “Сваха” может создать испытание | Успех |
| testAddTaskToCompetitionSuccess | Пользователь с ролью “Сваха” может добавить задания в испытание и назначить исполнителя | Успех |
| testChangeCompetitionToVoting | Пользователь с ролью “Сваха” может перевести состязание в статус “голосование” | Успех |
| testGetBrightVoteWithCandidate | Пользователи могут получить корректное голосование за лучшую невесту | Успех |
| testMakeVoteFirstTime | Авторизованный пользователь может проголосовать в рамках состязания в первый раз | Успех |
| testVoiceTracking | Проверка учета голоса | Успех |
| testMakeVoteSecondTime | Авторизованный пользователь НЕ может проголосовать в рамках состязания во второй раз | Успех |
| testDeleteCompetition | Пользователь с ролью “Сваха” может удалить состязание | Успех |
| testChangeUserRole\_Matchmaker | Авторизованный пользователь может сменить роль на “Сваха” | Успех |
| testGetUserListAsGroom | Пользователь с ролью “Жених” НЕ может получить список пользователей | Успех |
| testGetUserListAsMatchmaker | Пользователь с ролью “Сваха” может получить список пользователей | Успех |
| testDeleteUser | Пользователь может удалить аккаунт | Успех |
| testUpdateUser | Пользователь может редактировать поля своего профиля | Успех |

Load Testing (Нагрузочное тестирование)

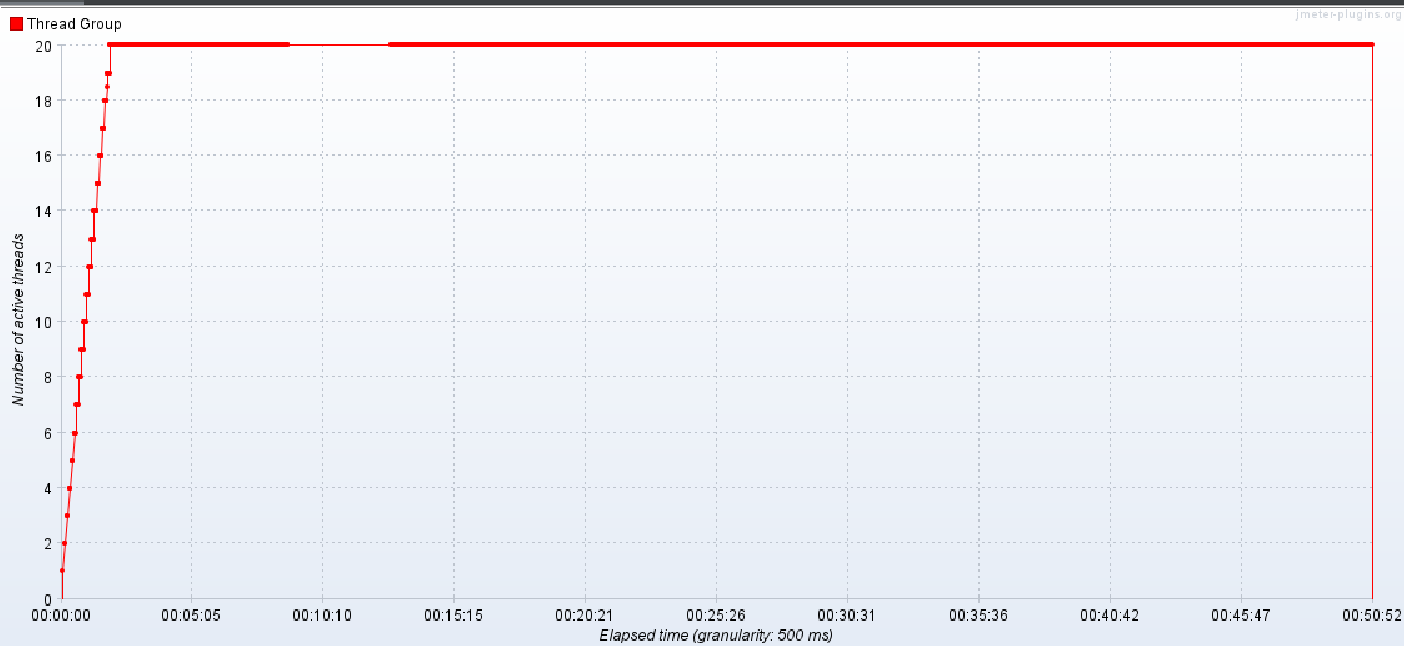
Профиль нагрузки

| Сценарий | Шаг | Нагрузка (операций в секунду) | Нагрузка (операций в час) |
| --- | --- | --- | --- |
| FullCase | registration | 0.3 ops | 1080 |
| log\_in |
| change\_role |
| launch\_arrow |
| get\_unssigned\_loverequests |
| set\_loverequest\_in\_progress |
| create\_competition |
| get\_free\_brides |
| add\_task\_to\_competition |
| update\_task |
| set\_competition\_to\_voting |
| make\_vote |

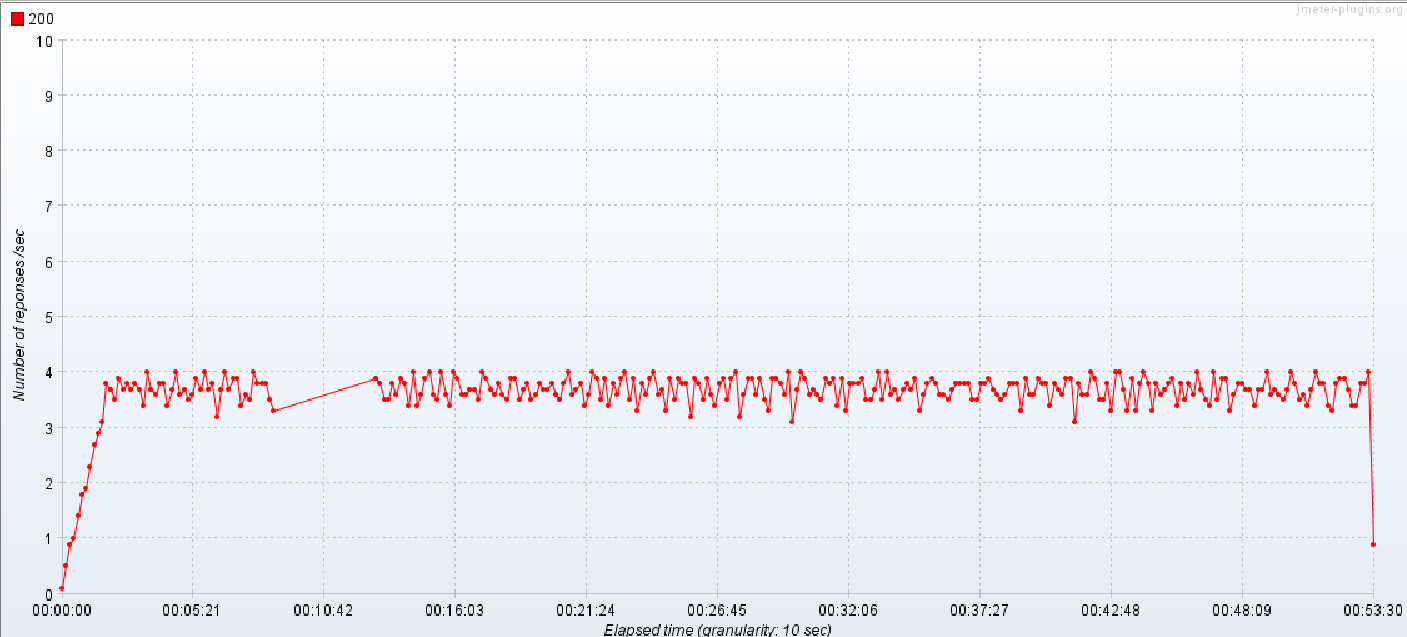
Количество запущенных виртуальных пользователей 20:

Выставленный pacing: 100 секунд

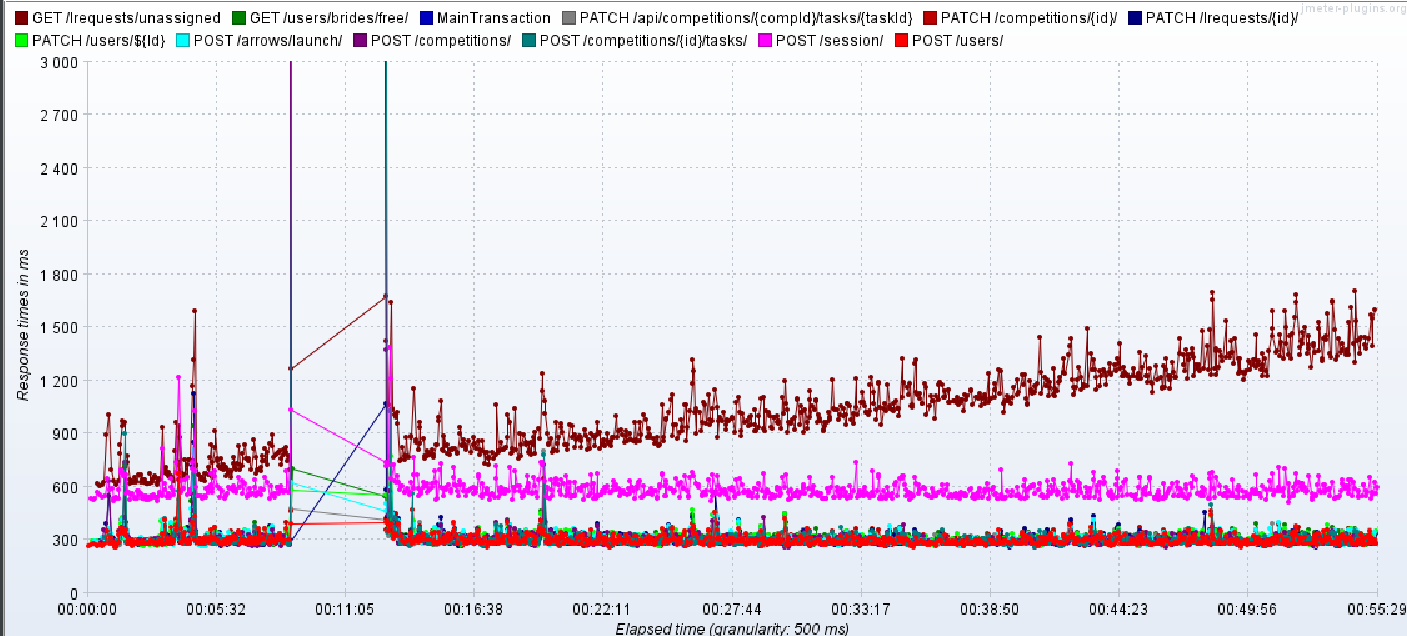
Уровень нагрузки: 0.2 ops



Утилизация сетевого трафика:



Время ответа:



* Достигнутая производительность: 0.4 ops
* Среднее время ответа не превысило 2 секунды
* Неуспешные транзакции не зафиксированы

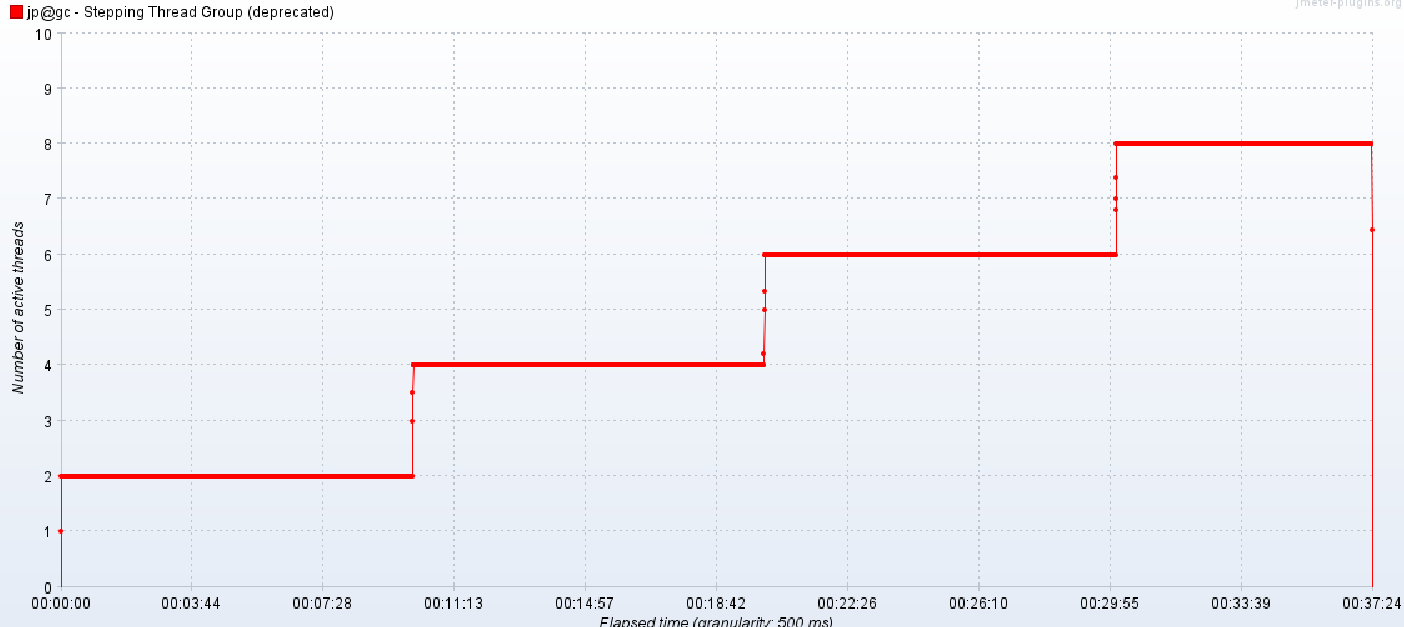
StressTesting

Профиль нагрузки

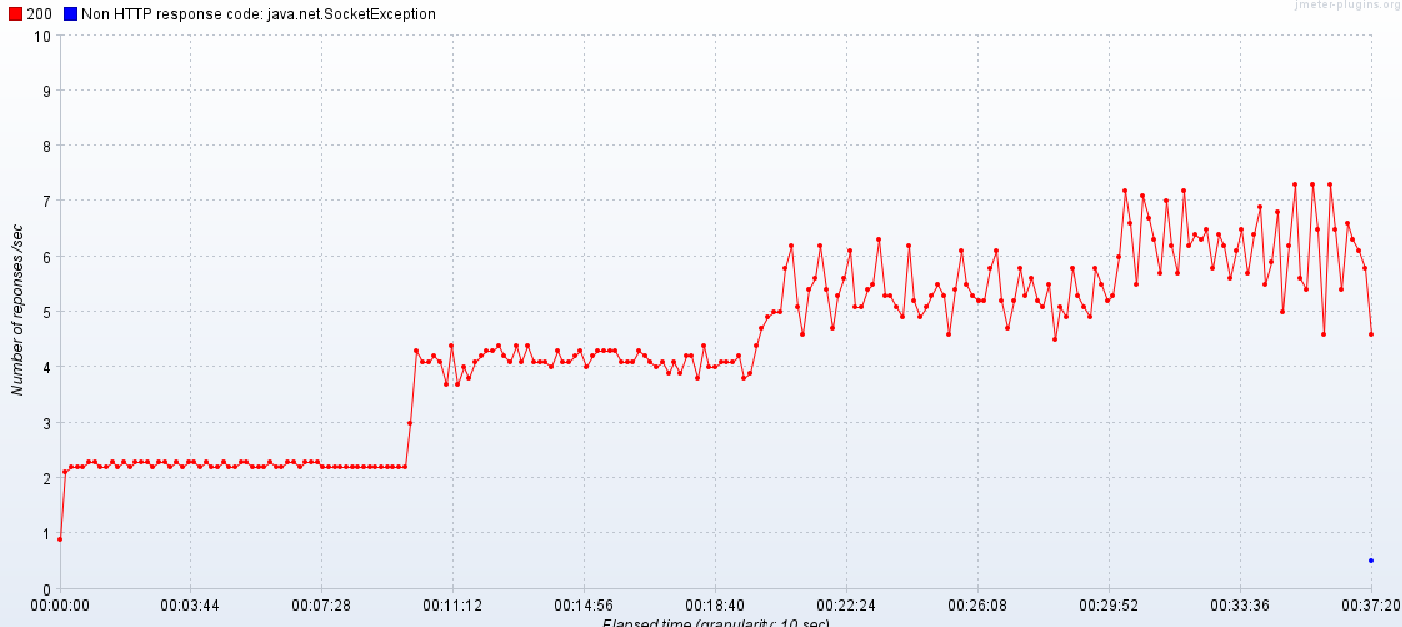
| Сценарий | Шаг |
| --- | --- |
| FullCase | registration |
| log\_in |
| change\_role |
| launch\_arrow |
| get\_unssigned\_loverequests |
| set\_loverequest\_in\_progress |
| create\_competition |
| get\_free\_brides |
| add\_task\_to\_competition |
| update\_task |
| set\_competition\_to\_voting |
| make\_vote |

Уровень нагрузки увеличивается ступенчато на 0.2 ops раз в 10 минут

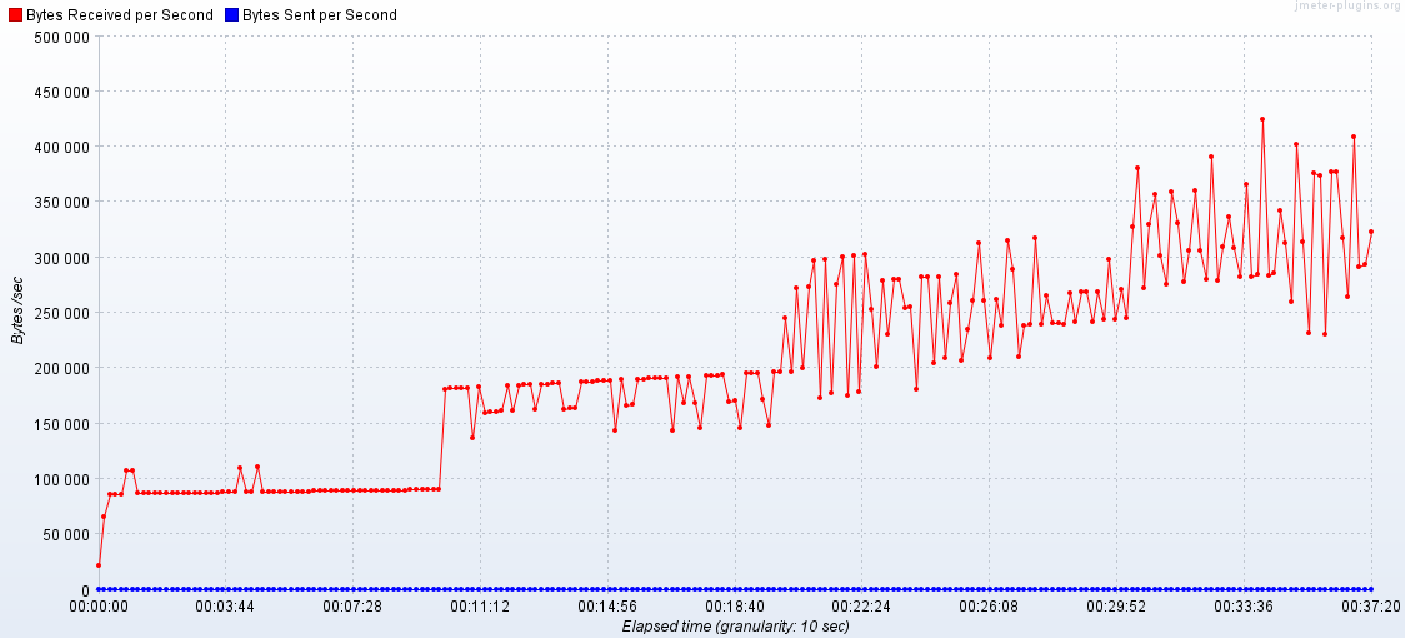
Число запущенных виртуальных пользователей (потоков):



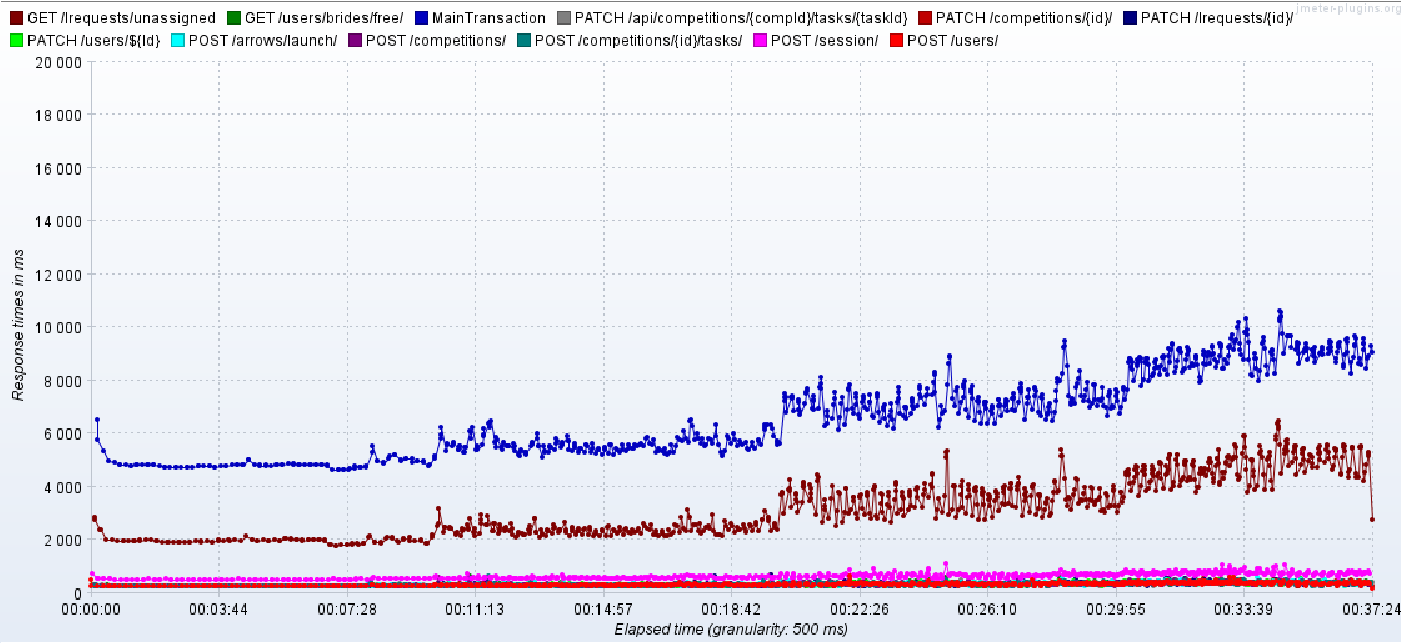
Код ответа в секунду:



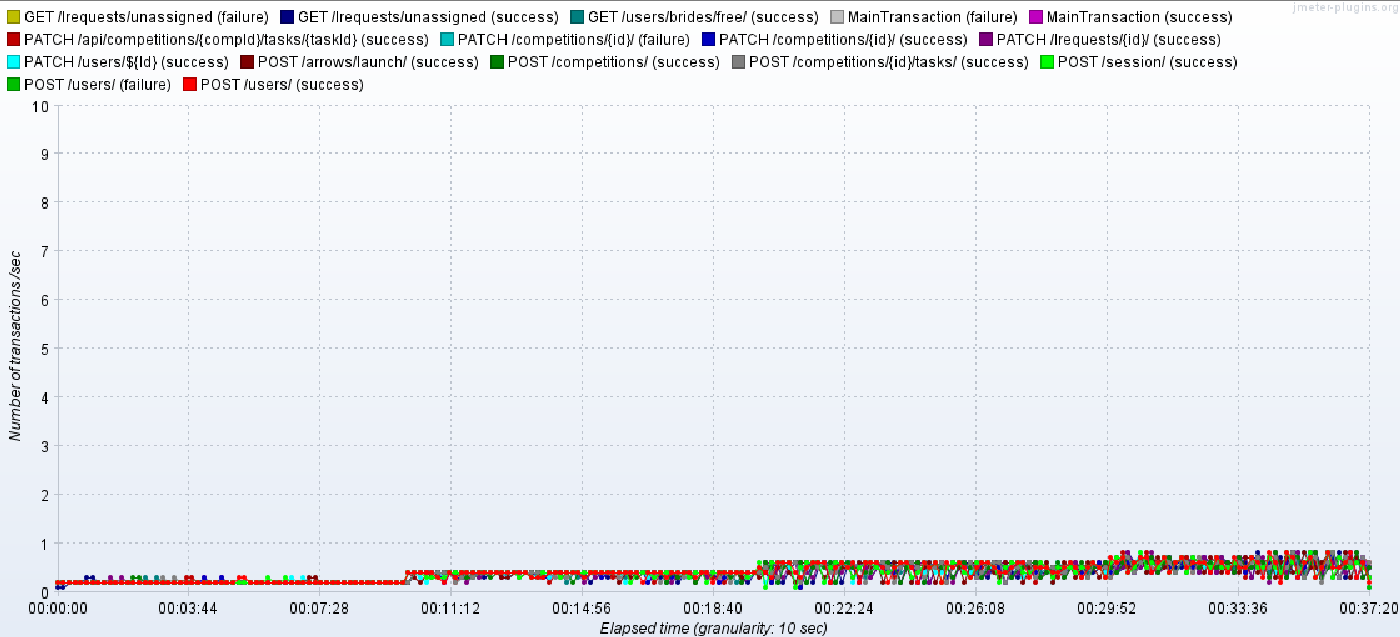
Утилизация сетевого трафика:



Время ответа:



Количество транзакций в секунду:



При увеличении нагрузки до уровня 0.6 ops время ответа превысило установленное ограничение в 3 секунды. Максимальная выдержанная нагрузка составляет 0.4 ops

Тестирование базы данных

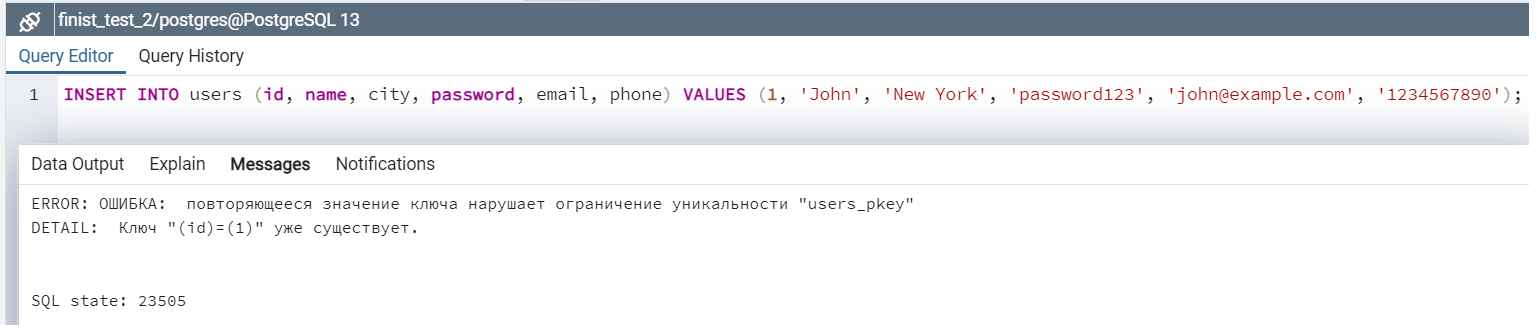
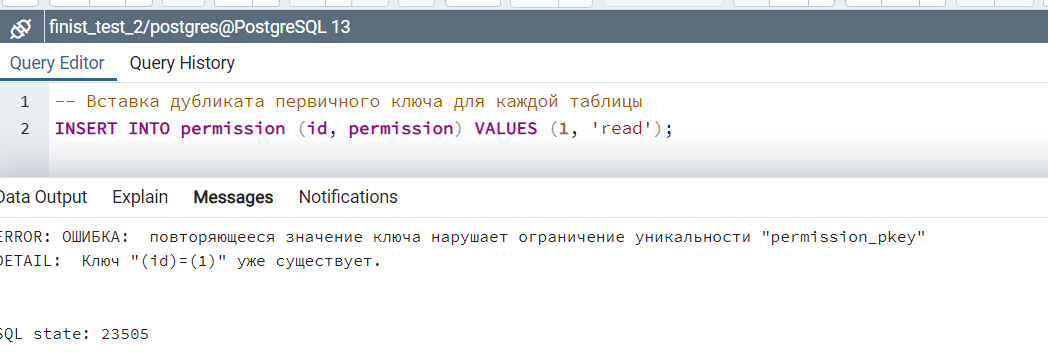
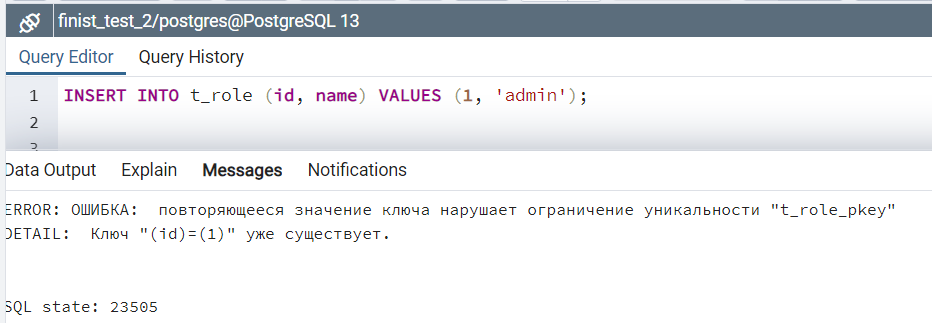
0. Подготовка тестовых данных.

База данных заполнена тестовыми данными с помощью следующих DDL команд:

| -- Заполнение таблицы permission  INSERT INTO permission (permission) VALUES ('read'), ('write'), ('execute');  -- Заполнение таблицы t\_role  INSERT INTO t\_role (name) VALUES ('admin'), ('user');  -- Заполнение таблицы users  INSERT INTO users (name, city, password, email, phone) VALUES ('John', 'New York', 'password123', 'john@example.com', '1234567890');  INSERT INTO users (name, city, password, email, phone) VALUES ('Alice', 'Los Angeles', 'password456', 'alice@example.com', '9876543210');  -- Заполнение таблицы love\_request  INSERT INTO love\_request (is\_assigned, groom\_id, matchmaker\_id) VALUES (false, 1, 2);  INSERT INTO love\_request (is\_assigned, groom\_id, matchmaker\_id) VALUES (false, 2, 1);  -- Заполнение таблицы competitions  INSERT INTO competitions (name, city, status, love\_request\_id, creator\_id) VALUES ('Competition 1', 'New York', 'IN\_PROGRESS',1, 1);  INSERT INTO competitions (name, city, status, love\_request\_id, creator\_id) VALUES ('Competition 2', 'Los Angeles', 'IN\_PROGRESS', 1, 2);  -- Заполнение таблицы tasks  INSERT INTO tasks (text, report, is\_completed, competition\_id, executor\_id) VALUES ('Task 1', 'Report 1', false, 1, 1);  INSERT INTO tasks (text, report, is\_completed, competition\_id, executor\_id) VALUES ('Task 2', 'Report 2', false, 2, 2);  -- Заполнение таблицы bride\_votes  INSERT INTO bride\_votes (end\_time, competition\_id) VALUES (CURRENT\_DATE, 1);  INSERT INTO bride\_votes (end\_time, competition\_id) VALUES (CURRENT\_DATE, 2);  -- Заполнение таблицы candidates  INSERT INTO candidates (points, bride\_id, bride\_vote\_id) VALUES (10, 1, 1);  INSERT INTO candidates (points, bride\_id, bride\_vote\_id) VALUES (5, 2, 1);  -- Заполнение таблицы marriages  INSERT INTO marriages (competition\_id, groom\_id, bride\_id, status) VALUES (1, 1, 2, 'Married');  INSERT INTO marriages (competition\_id, groom\_id, bride\_id, status) VALUES (2, 2, 1, 'Engaged');  -- Заполнение таблицы user\_votes  INSERT INTO user\_votes (candidate\_id, voter\_id, bride\_vote\_id) VALUES (1, 2, 1);  INSERT INTO user\_votes (candidate\_id, voter\_id, bride\_vote\_id) VALUES (2, 1, 2); |
| --- |

Проверка ограничений целостности:

1.1. Дубликат первичного ключа:



1.2. Несуществующий первичный ключ:



2. Проверка связей между таблицами:

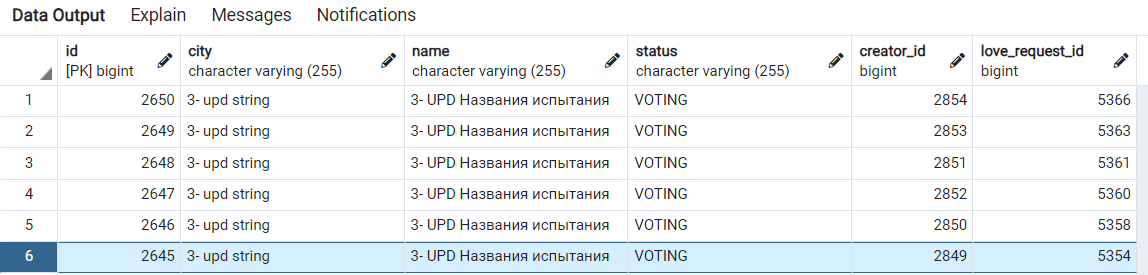
2.1. В таблице competitions имеется ограничение целостности:

*CONSTRAINT fk\_creator\_id FOREIGN KEY (creator\_id)*

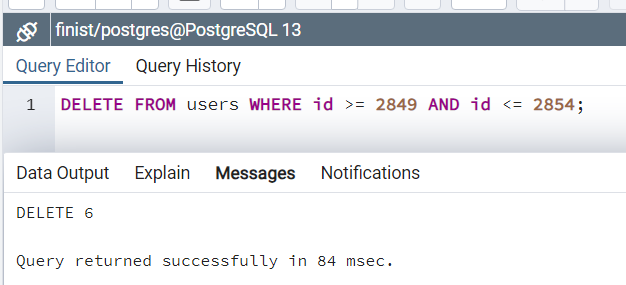
*REFERENCES public.users (id) MATCH SIMPLE*

*ON UPDATE NO ACTION*

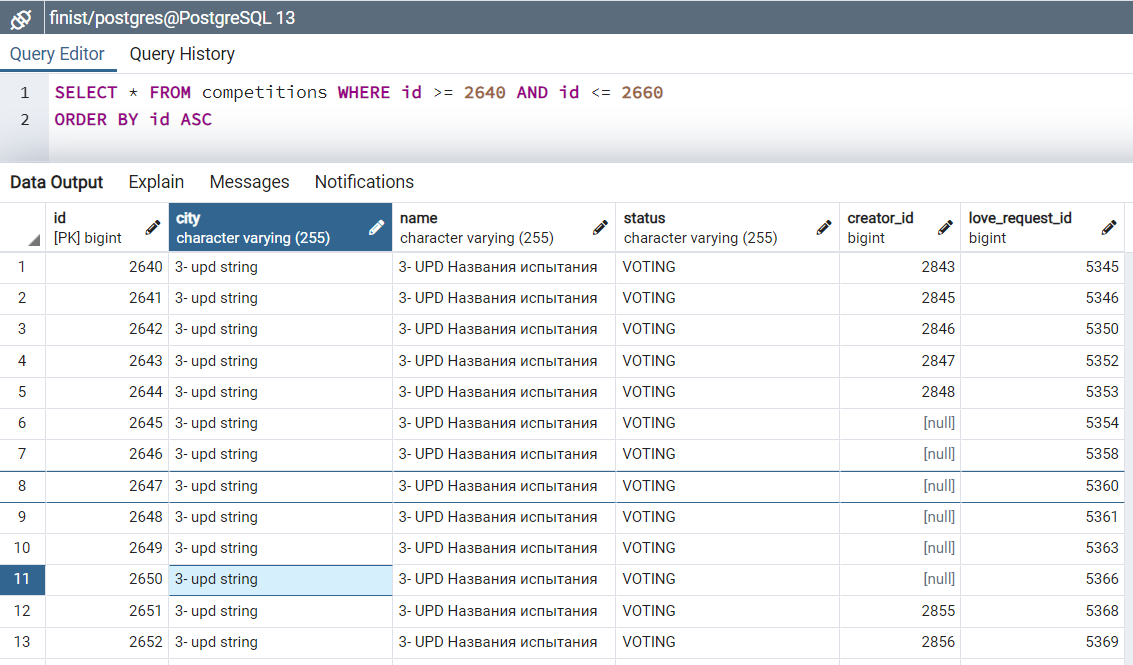
*ON DELETE SET NULL,*

Записи в competitions ДО удаления users:

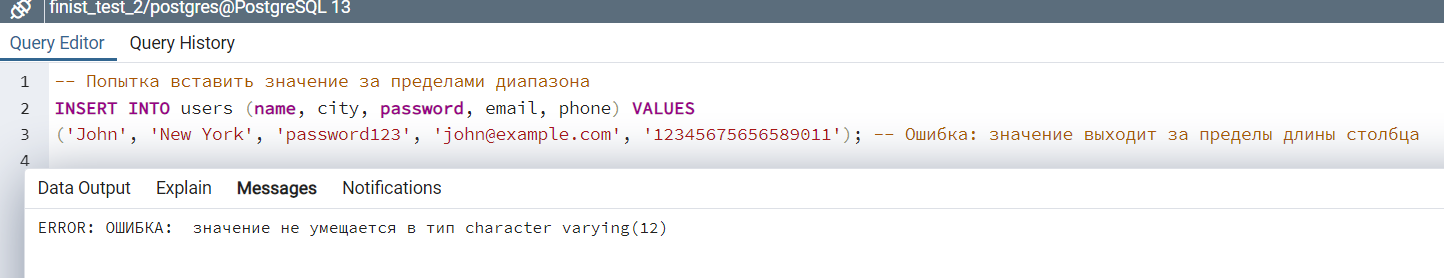
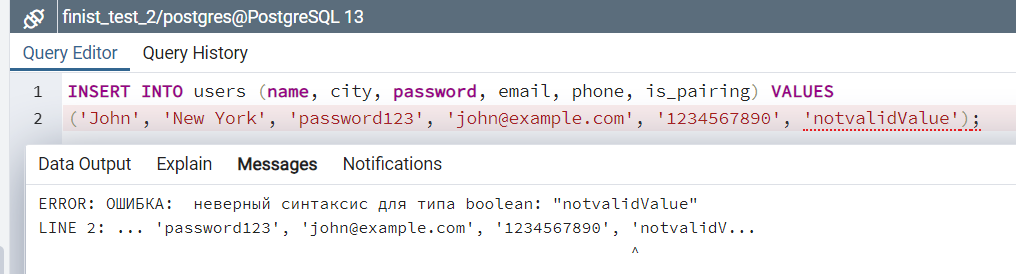
Удаление записей из таблицы users, на которые ссылаются записи в таблице competitions:



Записи в competitions ПОСЛЕ удаления users:



3. Проверка валидности типов данных:



Проведено тестирование базы данных. Все необходимые ограничения целостности соблюдены. DDL запросы выполняются корректно.

**3.** **Summary (Итоги)**

Система готова к стадии развёртывания. Все тесты выполнены успешно, отсутствуют ошибки, помеченные как критические и ошибки, мешающие выполнить UC и BC.