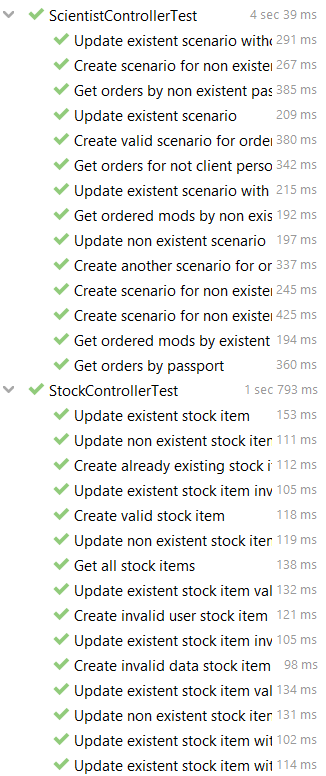
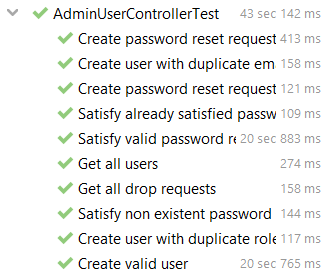
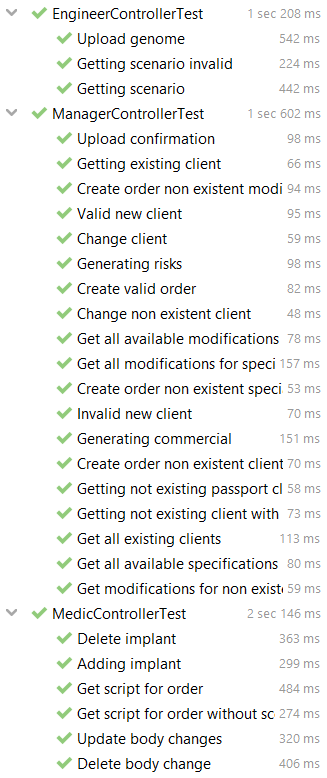
| 14.01.2022 | Авраменко  Давыдов  Шишкин | Initial |
| --- | --- | --- |

1. Функциональное тестирование

| Функциональное требование | Количество тестов | Название тестов |
| --- | --- | --- |
| FR1. Система должна предоставлять возможность создания аккаунта нового пользователя. | 3 | Create user with duplicate email  Create user with duplicate email  Create valid user |
| FR2. Система должна предоставлять возможность при создании нового пользователя выбрать ему одну из следующих ролей: администратор, медперсонал, инженер, ученый, менеджер. | 1 | Create valid user |
| FR3. Система должна предоставлять менеджеру возможность создания нового клиента. | 2 | Valid new Client  Invalid new Client |
| FR4. Система должна предоставлять менеджеру возможность редактировать данные клиента. | 2 | Change non existent client  Change client |
| FR5. Система должна предоставлять менеджеру возможность создавать новый заказ клиента | 4 | Create order non existent client  Create order for non existent specialization  Create order for non existent modification  Create valid order |
| FR6. Система должна предоставлять менеджеру возможность запросить коммерческое предложение. | 1 | Generating commercial |
| FR7. Система должна предоставлять менеджеру возможность запросить перечень рисков. | 1 | Generating risk |
| FR8. Система должна предоставлять менеджеру возможность внести подтверждение пользователя. | 1 | Upload confirmation |
| FR9. Система должна предоставлять ученому возможность просмотра модификаций ранжированных по риску в рамках заданной специализации. | 2 | Get ordered mods by non existent spec  Get ordered mods by existent spec |
| FR10. Система должна предоставлять ученому возможность внесения нового сценария модификации. | 5 | Create scenario for non existent specialization  Create scenario for no existent order  Create scenario for non existent modification  Create another scenario for order  Create valid scenario for order |
| FR11. Система должна предоставлять ученому возможность изменения существующего сценария модификации. | 4 | Update non existent scenario  Update existent scenario with extra mod  Update existent scenario scenario without all mods  Update existent scenario |
| FR12. Система должна предоставлять медицинскому персоналу возможность запроса сценария сопровождения трансформации согласно идентификатору клиента. | 2 | Get script for order without scenario  Get script for order |
| FR13. Система должна предоставлять автоматически сгенерированный сценарий сопровождения трансформации медицинскому персоналу по идентификатору клиента |
| FR14. Система должна предоставлять медицинскому персоналу возможность внесения в систему изменений клиента, произошедшие в течении трансформации | 2 | Update body changes  Delete body change |
| FR15 Система должна предоставлять медицинскому персоналу возможность внесение в систему имплантов внедренных в клиента | 2 | Adding implant  Delete implant |
| FR16. Система должна предоставлять инженеру возможность получения одобренного клиентом сценария по идентификатору заказа. | 2 | Getting scenario invalid  Getting scenario |
| FR17. Система должна предоставлять инженеру возможность внесения проведенных в геноме клиента модификаций. | 1 | Upload genome |
| FR18. Система должна предоставлять возможность списания/внесения деталей и расходных материалов с баланса склада. | 11 | Create already existing stock item  Create invalid data stock item  Create invalid user stock item  Create valid stock item  Update non existent stock item  Update existent stock item with invalid user  Update existent stock item with invalid data  Update existent stock item  Update non existent stock item amount  Update existent stock item invalid amount  Update existent stock item valid amount |
| FR19. Система должна предоставлять возможность устанавливать ограничения на количество минимально доступных расходных материалов и деталей. | 3 | Update non existent stock item min amount  Update existent stock item invalid min amount  Update existent stock item valid min amount |
| FR20. Система должна предоставлять возможность подключения сканирующих станций посредством API. |  |  |
| FR21. Система должна предоставлять администратору возможность сброса пароля по запросу пользователя. | 3 | Satisfy non existent password reset request  Satisfy already satisfied password reset request  Satisfy valid passport reset request |

Общее реализованное количество тестов - 67, все были успешно пройдены на итоговой версии системы:





2. Тестирование базы данных

| **name** | **status** | **comment** |
| --- | --- | --- |
| person\_valid | t | - |
| person\_invalid | t | null value in column "name" of relation "person" violates not-null constraint |
| passport\_valid | t | - |
| passport\_invalid | t | new row for relation "passports" violates check constraint "passports\_passport\_check" |
| passport\_fk\_invalid | t | insert or update on table "passports" violates foreign key constraint "passports\_person\_id\_fk" |
| client\_valid | t | - |
| client\_fk\_invalid | t | insert or update on table "clients" violates foreign key constraint "clients\_person\_id\_fk" |
| user\_role\_valid | t | - |
| user\_role\_unique | t | duplicate key value violates unique constraint "user\_role\_name\_uindex" |
| users\_valid | t | - |
| user\_person\_fk\_invalid | t | insert or update on table "users" violates foreign key constraint "users\_person\_id\_fk" |
| user\_role\_fk\_invalid | t | insert or update on table "users" violates foreign key constraint "users\_userrole\_id\_fk" |
| pwd\_drop\_request\_valid | t | - |
| wd\_drop\_request\_fk\_invalid | t | insert or update on table "pwd\_drop\_requests" violates foreign key constraint "pwd\_reset\_requests\_users\_id\_fk" |
| stock\_valid | t | - |
| stock\_user\_fk\_invalid | t | insert or update on table "stock" violates foreign key constraint "stock\_users\_id\_fk" |
| stock\_vendor\_invalid | t | null value in column "vendor\_code" of relation "stock" violates not-null constraint |
| implants\_valid | t | - |
| implants\_invalid | t | null value in column "name" of relation "implants" violates not-null constraint |
| client\_implant\_valid | t | - |
| client\_fk\_implant\_invalid | t | insert or update on table "client\_implant" violates foreign key constraint "client\_implant\_clients\_id\_fk" |
| lient\_implant\_fk\_invalid | t | insert or update on table "client\_implant" violates foreign key constraint "client\_implant\_implants\_id\_fk" |
| accompaniment\_valid | t | - |
| accompaniment\_invalid | t | null value in column "scenarios" of relation "accompaniment\_script" violates not-null constraint |
| specialization\_valid | t | - |
| specializations\_invalid | t | null value in column "name" of relation "specializations" violates not-null constraint |
| client\_fk\_mod\_invalid | t | insert or update on table "client\_modification" violates foreign key constraint "client\_specialization\_clients\_id\_fk" |
| scenario\_valid | t | - |
| scenario\_fk\_invalid | t | insert or update on table "scenario" violates foreign key constraint "scenario\_specializations\_id\_fk" |
| modification\_valid | t | - |
| modification\_invalid | t | null value in column "name" of relation "modifications" violates not-null constraint |
| mod\_scen\_valid | t | - |
| mod\_fk\_scen\_invalid | t | insert or update on table "modification\_scenario" violates foreign key constraint "modification\_specialization\_modifications\_id\_fk" |
| mod\_scen\_fk\_invalid | t | insert or update on table "modification\_scenario" violates foreign key constraint "modification\_specialization\_scenario\_id\_fk" |
| order\_valid | t | - |
| order\_spec\_fk\_invalid | t | insert or update on table "orders" violates foreign key constraint "orders\_specializations\_id\_fk" |
| order\_acc\_fk\_invalid | t | insert or update on table "orders" violates foreign key constraint "orders\_accompaniment\_script\_id\_fk" |
| order\_client\_fk\_invalid | t | insert or update on table "orders" violates foreign key constraint "orders\_clients\_id\_fk" |
| body\_valid | t | - |
| body\_fk\_invalid | t | insert or update on table "body\_changes" violates foreign key constraint "body\_changes\_orders\_id\_fk" |
| order\_mod\_valid | t | - |
| order\_fk\_mod\_invalid | t | insert or update on table "orders\_modifications" violates foreign key constraint "orders\_modifications\_orders\_id\_fk" |
| order\_mod\_fk\_invalid | t | insert or update on table "orders\_modifications" violates foreign key constraint "orders\_modifications\_modifications\_id\_fk" |

**Итог:**

amount | ok | not\_ok

-----------+----+--------

44 | 44 | 0

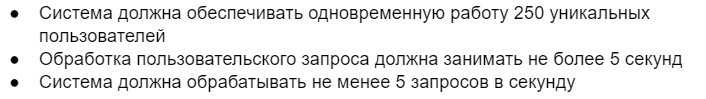
В ходе тестирования изначальной реализации были выявлены следующие недочеты:

* Отсутствовало ограничение на формат паспорта
* user\_role не уникальны
* В таблице имплантов неверный тип поля description
* Отсутствовали нескольких not null ограничений

В итоговой версии системы, выявленные недочеты были устранены.

3. Нагрузочное тестирование

В требованиях к производительности, обозначенных в SRS, содержится следующее:



То есть при наличии 250 пользователей, пропускная способность системы должна составлять не менее 5 запросов в секунду, при этом время обработки каждого из них не должно превышать 5 секунд.

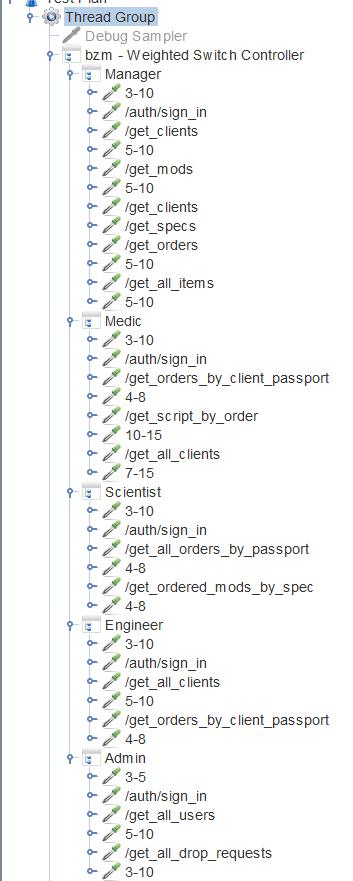
Пользователи в системе могут иметь одну из пяти доступных ролей:

* Менеджер
* Медик
* Ученый
* Инженер
* Администратор

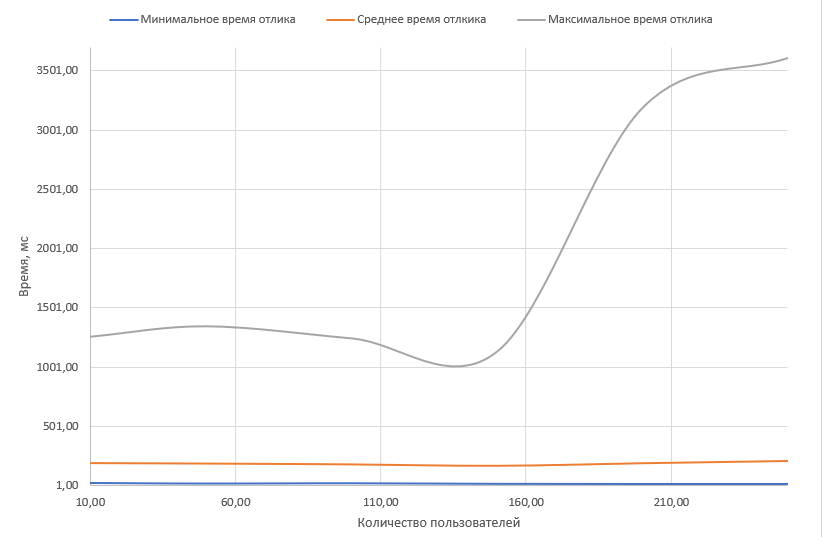
В зависимости от роли, запросы пользователя и его нагрузка на систему разнятся. В таком случае предполагаемый список профилей нагрузки по ролям с операциями и соотношение количества пользователей данного профиля представляется следующим образом:

| Профиль нагрузки | Последовательность операций | Процент пользователей с данным профилем |
| --- | --- | --- |
| Менеджер | 1. Просмотр страницы авторизации, задержка на время в промежутке 3 - 10 сек. (ввод учетных данных) 2. Авторизация в системе 3. Просмотр списка клиентов, задержка в промежутке 5 - 10 сек. 4. Просмотр данных клиента, задержка 5 - 10 сек. 5. Переход на страницу просмотра заявок, просмотр списка заявок, просмотр заявки и данных клиента, задержка в промежутке 5 - 10 сек. 6. Переход на страницу просмотра содержимого склада, задержка в промежутке 5-10 сек. | 40 |
| Медик | 1. Просмотр страницы авторизации, задержка на время в промежутке 3 - 10 сек. (ввод учетных данных) 2. Авторизация в системе 3. Ввод идентификатора клиента, просмотр заявки, задержка в промежутке 4 - 8 сек. 4. Просмотр сценария для специализации, задержка в промежутке 10-15 сек. 5. Переход на страницу отчетности, просмотр списка клиентов, задержка в промежутке 4 - 8 сек. 6. Просмотр данных клиента, задержка в промежутке 7-15 сек. | 25 |
| Ученый | 1. Просмотр страницы авторизации, задержка на время в промежутке 3 - 10 сек. (ввод учетных данных) 2. Авторизация в системе 3. Ввод идентификатора клиента, просмотр заявки, задержка в промежутке 4 - 8 сек. 4. Просмотр списка модификаций для заявки, задержка в промежутке 4 - 8 сек. | 15 |
| Инженер | 1. Просмотр страницы авторизации, задержка на время в промежутке 3 - 10 сек. (ввод учетных данных) 2. Авторизация в системе 3. Просмотр списка клиентов, задержка в промежутке 3 - 5 сек. 4. Просмотр данных клиента, задержка в промежутке 5 - 10 сек. 5. Ввод идентификатора клиента, просмотр заявки, задержка в промежутке 4 - 8 сек. | 15 |
| Администратор | 1. Просмотр страницы авторизации, задержка на время в промежутке 3 - 5 сек. (ввод учетных данных) 2. Авторизация в системе 3. Просмотр списка пользователей, задержка в промежутке 5 - 10 сек. 4. Переход на страницу с заявками на сброс пароля. просмотр списка заявок на сброс пароля, задержка в промежутке 5 - 10 сек. | 5 |

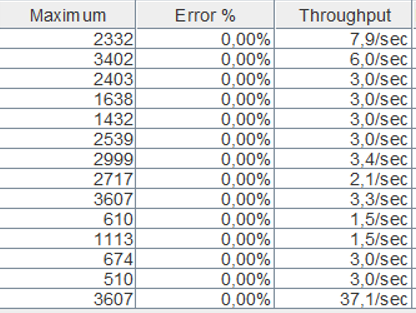
При помощи инструмента Test Script Recorder были определен порядок запросов для каждого из профилей, после чего при помощи плагина Weighted Switch Controller было воспроизведено процентное распределение пользователей. Итоговый тест план выглядит следующим образом:



По результатам тестирования был составлен график зависимости времени отклика от количества пользователей



Агрегированные данные по времени запросов для максимально требуемого количества пользователей (250)

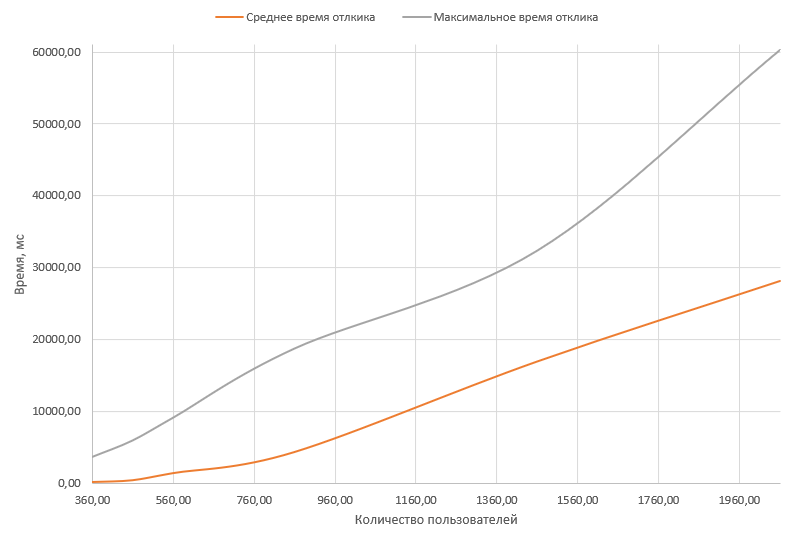


Ни один из запросов не дал ошибку, при этом максимальное время ответа составило 3.6 сек, что меньше 5 секунд, а пропускная способность - 37.1 запрос/сек, что больше требуемого 5 запрос/сек.

4. Стресс тестирование

Для стресс тестирования был использован план из нагрузочного тестирования. Количество пользователей постепенно увеличивалось, начиная с 260 до тех пор, пока система не начала отдавать ошибки или же не было превышено максимально допустимое время отклика.

График зависимости времени отклика от количества пользователей:

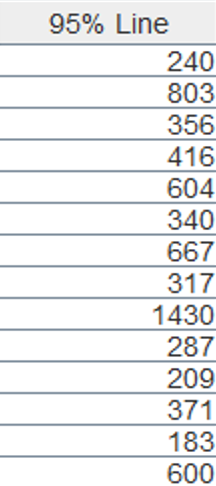


Даже при нагрузке, многократно превышающей заявленную пиковую, система не выдает ошибки и корректно обрабатывает входящие запросы. Однако с учетом того, что максимальное и среднее времена ответа так же многократно увеличиваются вплоть до значений в минуту, на данном моменте систему можно считать неработоспособной для пользователя. Указанное ограничение на максимальное время отклика в 5 секунд перестает выполняться при количестве одновременных пользователей ~440, что больше заявленного в SRS пикового значения в 250 пользователей.

5. Тестирование производительности

Для тестирования производительности был использован план из нагрузочного тестирования. Количество пользователей задавалось равным 250, пиковому возможному значению согласно SRS.

Полученные 95 перцентили времени отклика для каждого запроса и среднее:

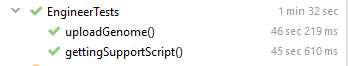


Максимальное значение 95 перцентиля составляет ~1.4 сек. что меньше чем указанное в SRS предельно допустимое значение в 5 сек.

6. Тестирование интерфейса

Для тестирование интерфейса были использованы: фреймворк JUnit и инструмент для автоматизации действий веб браузера Selenium. Были проведены тесты ключевых прецедентов использования, описанных в документе UseCase.

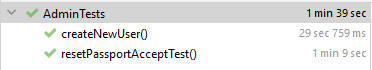
Результаты выполнения тестов:

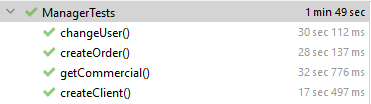












7. Тестирование бизнес-цикла

Для всех реализованных прецедентов было проведено end-to-end тестирование согласно основным и альтернативным последовательностям описанных в UseCase. В результате проведенного тестирования и исправления по его результатам, все полученные постусловия соответствовали ожидаемым.

В ходе тестирования были выявлены следующие недостатки:

* Несовпадение стилей мокапам
* Отсутствие ограничений на вводимые данные со стороны UI
* При изменении паспорта клиента вместо добавления нового паспорта изменялось значение старого

После проведения тестирования все выявленные недостатки были исправлены

8. Тестирование безопасности и прав доступа

В ходе тестирования пользовательских интерфейсов было написаны тесты позволяющие совершить вход в систему для каждой роли, а также тест который проверяющий вход с неверными учетными данными

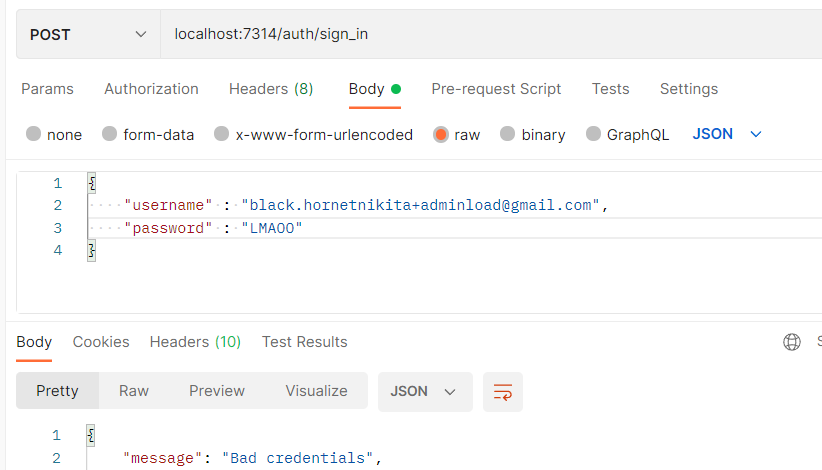




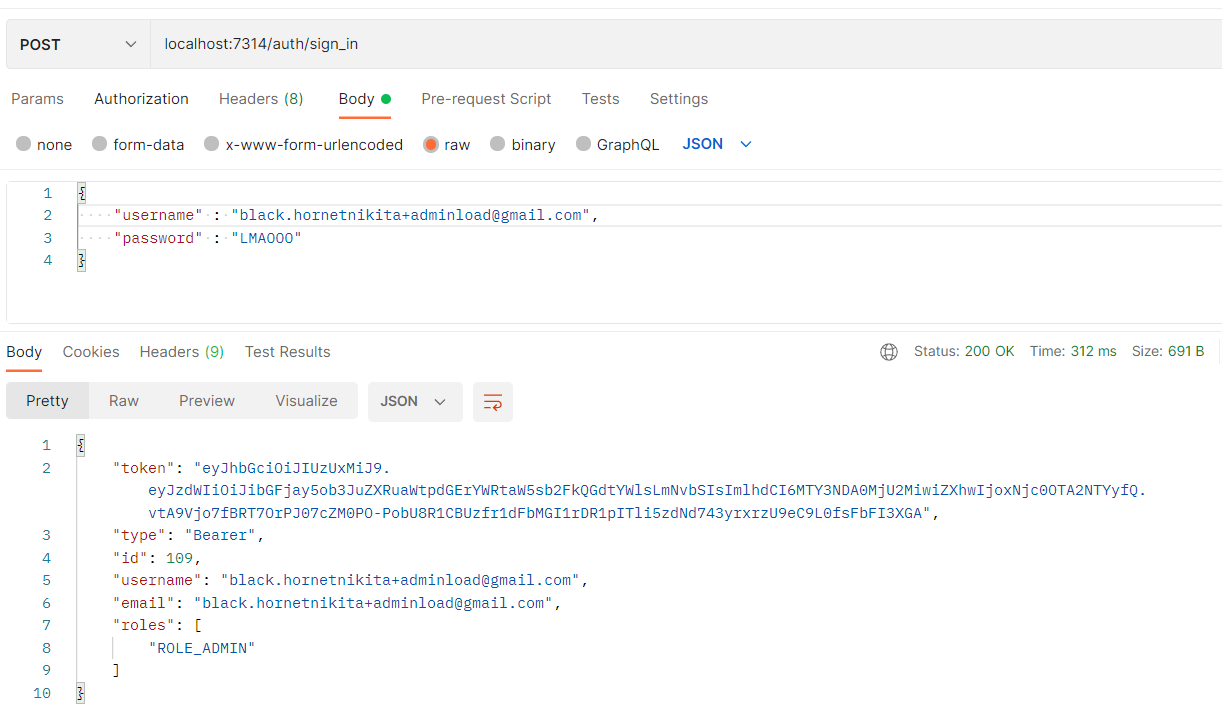


После этого было проведено тестирование эндпоинтов с использованием ПО Postman, для каждой роли была проверена доступность методов относящейся к этой роли, а также недоступность методов других ролей.

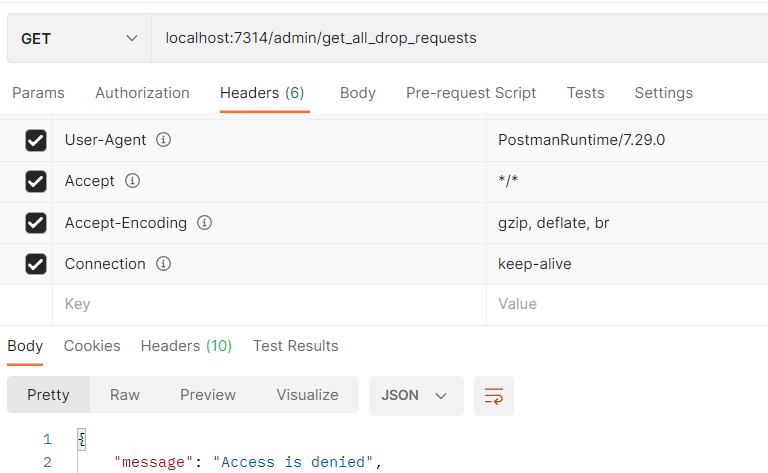
Попытка авторизации с неверными учетными данными:



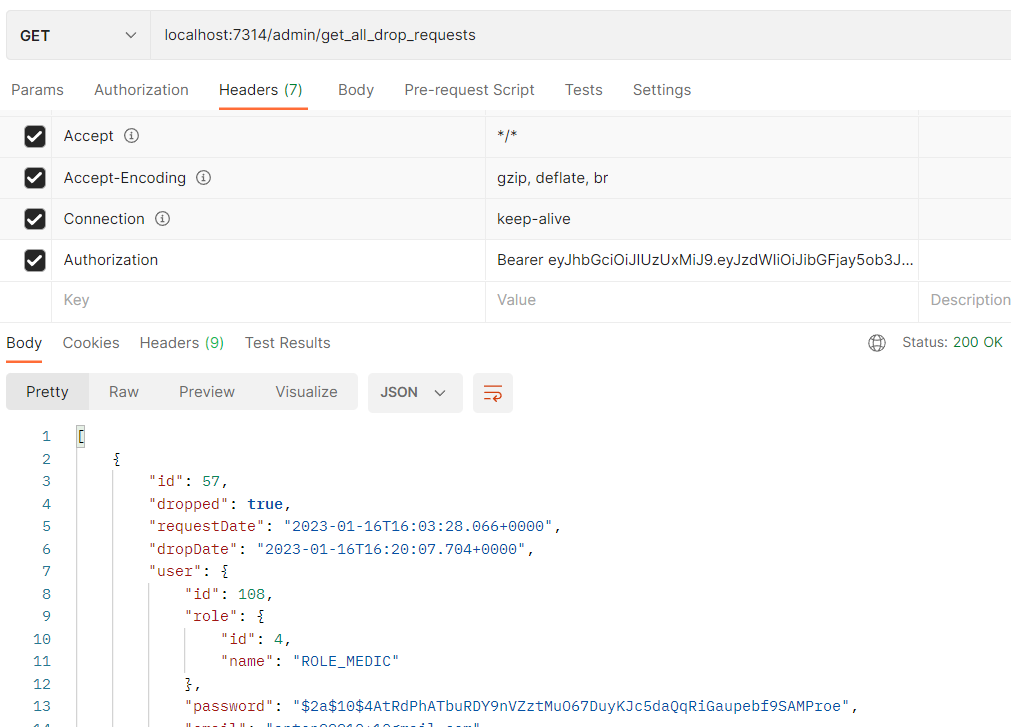
Авторизация с корректными учетными данными:



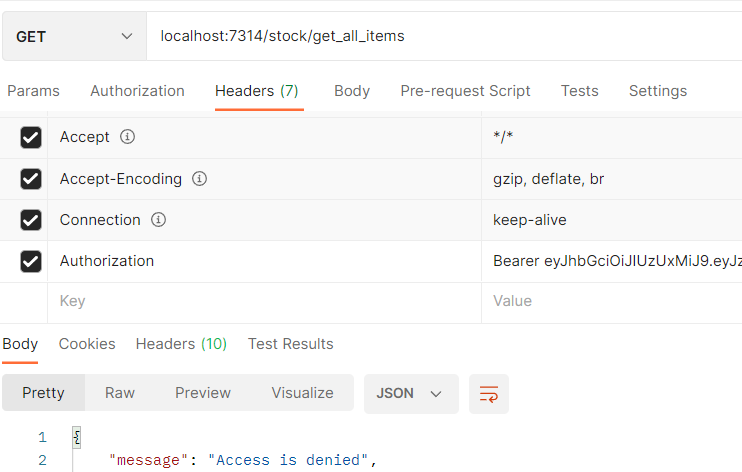
Попытка получения доступа к методу без токена-авторизации:



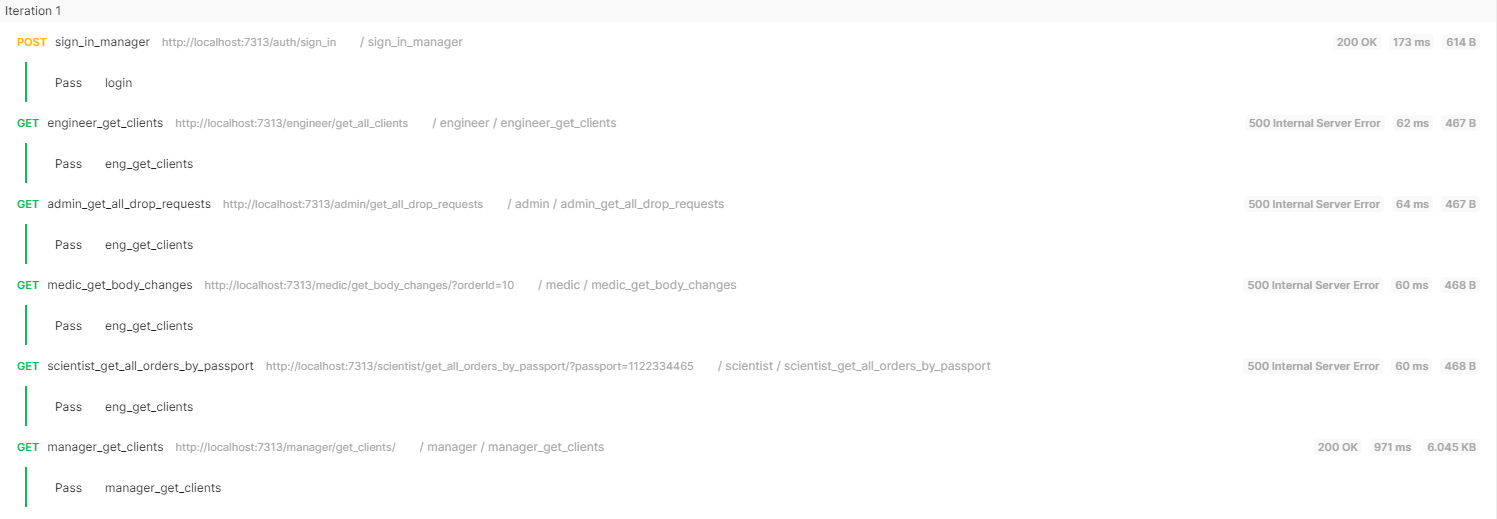
Получение доступа к метода с верными токеном авторизации:



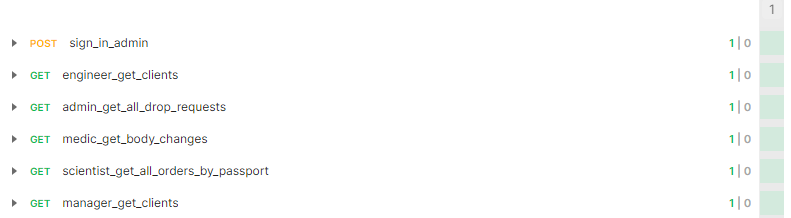
Получение доступа к методу с токеном-авторизации неверной роли:



Таким образом были написаны скрипты для проверки доступа ко всем контроллерам каждой ролью.



Администратор



Медик



Ученый



Инженер

