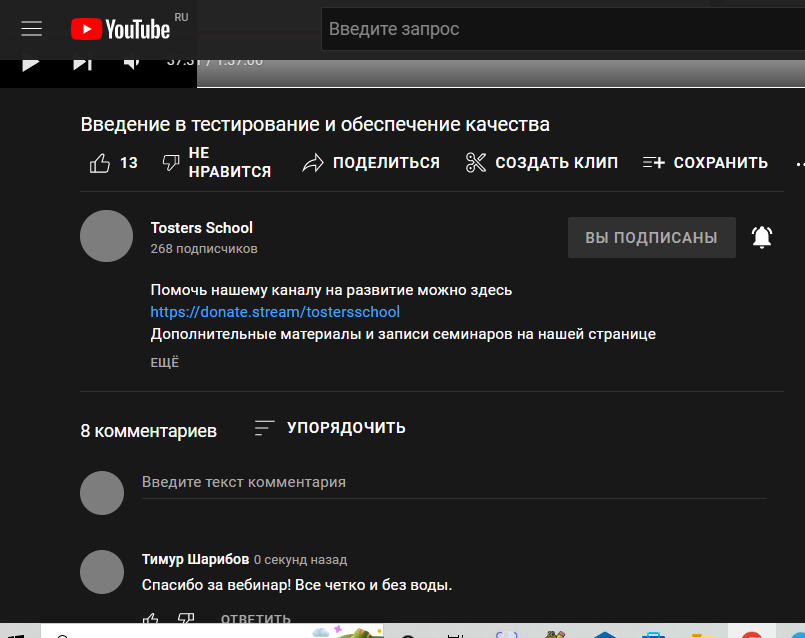
1. Оставить отзыв с комментарием на странице вебинара в YouTube
2. Опишите основные этапы разработки. Какие задачи на каждом из этапов выполняет тестировщик? В каких случаях имеет подключать тестировщика на ранних этапах разработки?
3. Опишите критерии качества ПО, раскройте каждый из параметров на примере приложения Онлайн кинотеатра (любой на ваш выбор).
4. Какой функционал онлайн кинотеатра (из п.2) вы включили бы в смоук тест и почему?
5. Опишите 2 позитивных и 2 негативных сценария тестирования для приложения из п.2
6. Выберите приложение из п.2 и опишите 2 задачи на его доработку.
   1. +1 балл указание целевой аудитории
   2. +1 балл оценка влияния на продвижение сервиса
7. В виде инструкции опишите работу с двумя функциями для приложения из п.2
8. Как вы считаете, для развития приложения из п.2 с которым вы ранее выполняли задания подойдет больше классическая методология разработки или гибкий подход и почему?
9. Оцените перспективы применения автоматизации тестирования для приложения из п.2

Дополнительные задания (12 баллов)

1. Развернуть на локальной машине Docker и поднять приложение TestIt
2. Поднять контейнер на локальной машине с приложением Petstore [swaggerapi/petstore - Docker Image | Docker Hub](https://hub.docker.com/r/swaggerapi/petstore) Дальнейшие работы провести с локальным сервисом.
3. В TestIT добавить пользователей с правами администратора, лида тестирования и тестировщика. Создать проекты Petstore (API) и Кинотеатр (приложение из предыдущих пунктов)
4. В проекте Petstore создать тест сьют PETS и написать два позитивных и два негативных тестовых сценария для метода https://petstore.swagger.io/v2/pet
5. Создать тестовый прогон для созданных тест кейсов, провести тестирование по ним и сформировать отчет о проведенном тестировании в TestIT.
6. Сформировать страницу с метриками тестирования на странице метрик проекта
7. Для проекта Кинотеатр составить 2 позитивных и два негативных тестовых сценария, сформировать тестовый прогон с их использованием и провести тестирование. Сформировать отчет по проведенному тестированию, сформировать страницу метрик тестирования для проекта.
8. Приложить скриншоты по проделанной работе

1.

2. Этапы разработки ПО:

1. Анализ требований - сбор, документирование, проверка на противоречие самих требований, т.е. формирование технического задания на разработку ПО.
2. Проектирование - проектирование основных элементов продукта: выбор подходящей БД, среды программирования, утверждение интерфейса, формирование окончательной архитектуры создаваемой системы.
3. Разработка - написание кода продукта с помощью выбранного языка программирования.
4. Тестирование - процесс проверки соответствия ПО заявленным требованиям и ожиданиям заказчика, а также выявления ошибок, т.е. неправильной работы ПО.
5. Релиз и дальнейшее его сопровождение - внедрение ПО в эксплуатацию, возможно дальнейшее его развитие.
6. Вывод из эксплуатации - закрытие проекта.

Задачи тестировщика в:

a) тестирование требований - проверка на противоречие требований, недвусмысленность.

b) выбор среды тестирования, необходимых инструментов и оборудования.

Можно уже писать чек-листы - список проверок.

с) модульное тестирование - после создания части кода можно начинать его тестирование.

d) все виды тестирования.

e) Выявление ошибок уже при реальной эксплуатации.

f) не участвует.

Цена исправления найденного бага на более ранних стадиях разработки будет стоить гораздо меньше, поэтому чем раньше подключить тестера тем лучше.

3. Функциональность - поиск фильма по названию, возможность добавлять в избранное, продолжать просмотр, фильтровать по жанрам (популярности, дате выхода)

Надежность - бесперебойная работа с выбранным качеством даже при одновременном пользовании ресурсом всех пользователей.

Юзабилити - удобство поиска, фильтра, возможно предложение фильмов (или сериалов) наиболее часто просматриваемых жанров.

Эффективность - количество просмотров, подписок.

Защищенность - невозможность просмотра без подписки платного контента.

Мобильность - возможность просматривать на разных мобильных устройствах, ПК с разными ОС.

Модифицируемость - возможность внесения новых изменений, например, нового фильтра.

4. Регистрация, вход, поиск фильма, возможность просмотра.

5. Вход при использовании валидных данных, поиск фильма из видеотеки.

Попытка входа с невалидными данными, поиск несуществующего фильма.

6. Фильтр только доступных фильмов(тв-каналов) по оплаченной подписке.

Целевая аудитория - все кому надоела реклама.

Удобство пользования, сохранение пользователей и возможно привлечение новых.

(Это одна из причин из-за чего я поменял оператора).

Возможность добавления(покупки) телеканала,которого нет в пакете тарифа.

Целевая аудитория - иностранные граждане и будущие эмигранты.

Увеличение количества пользователей.

7. Фильтр по годам.

a) Открыть страницу kion.ru

b) в правом верхнем углу страницы нажать кнопку “Вход - регистрация”.

c) Ввести валидный номер телефона, на который придет код.

d) Ввести код. ( Автоматический откроется онлайн кинотеатр kion.ru)

e) Нажать кнопку “Фильмы” в верхней строчке.(откроется окно с подборками фильмов)

f) Нажать кнопку “Все фильмы”. (Откроется окно со списком фильмов).

1. Выбрать в фильтре по годам “2022”. (Откроется список фильмов вышедших в 2022г)

Пополнение списка “Продолжить просмотр”.

a) Открыть страницу kion.ru

b) в правом верхнем углу страницы нажать кнопку “Вход - регистрация”.

c) Ввести валидный номер телефона, на который придет код.

d) Ввести код. ( Автоматический откроется онлайн кинотеатр kion.ru)

e) Нажать на любой фильм из любого списка.

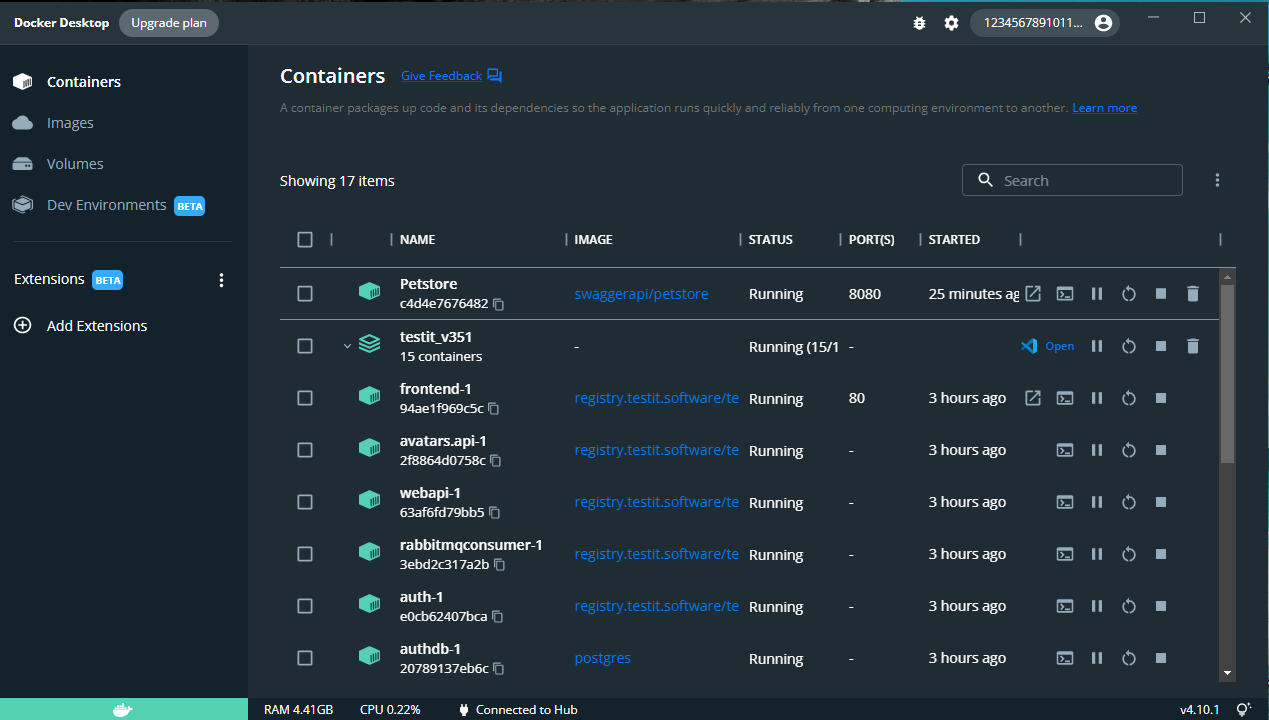
f) Нажать кнопку “Смотреть”

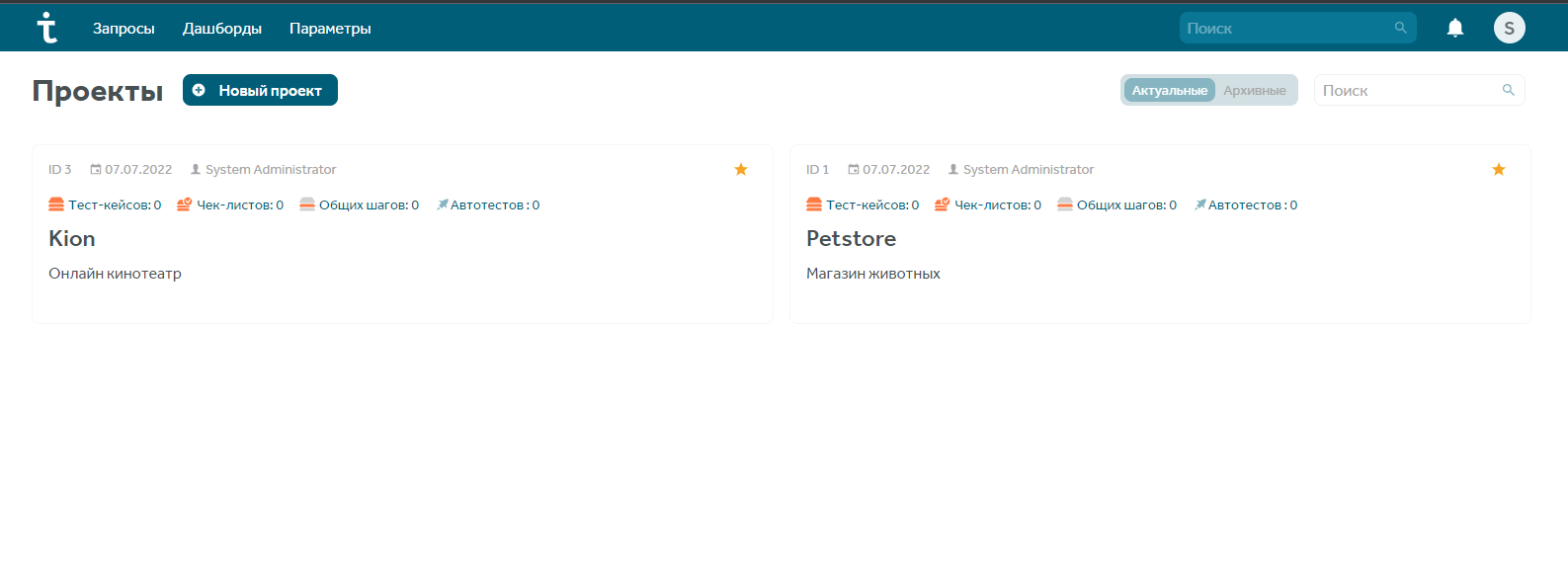
g) Передвинуть шкалу времени на 25:25

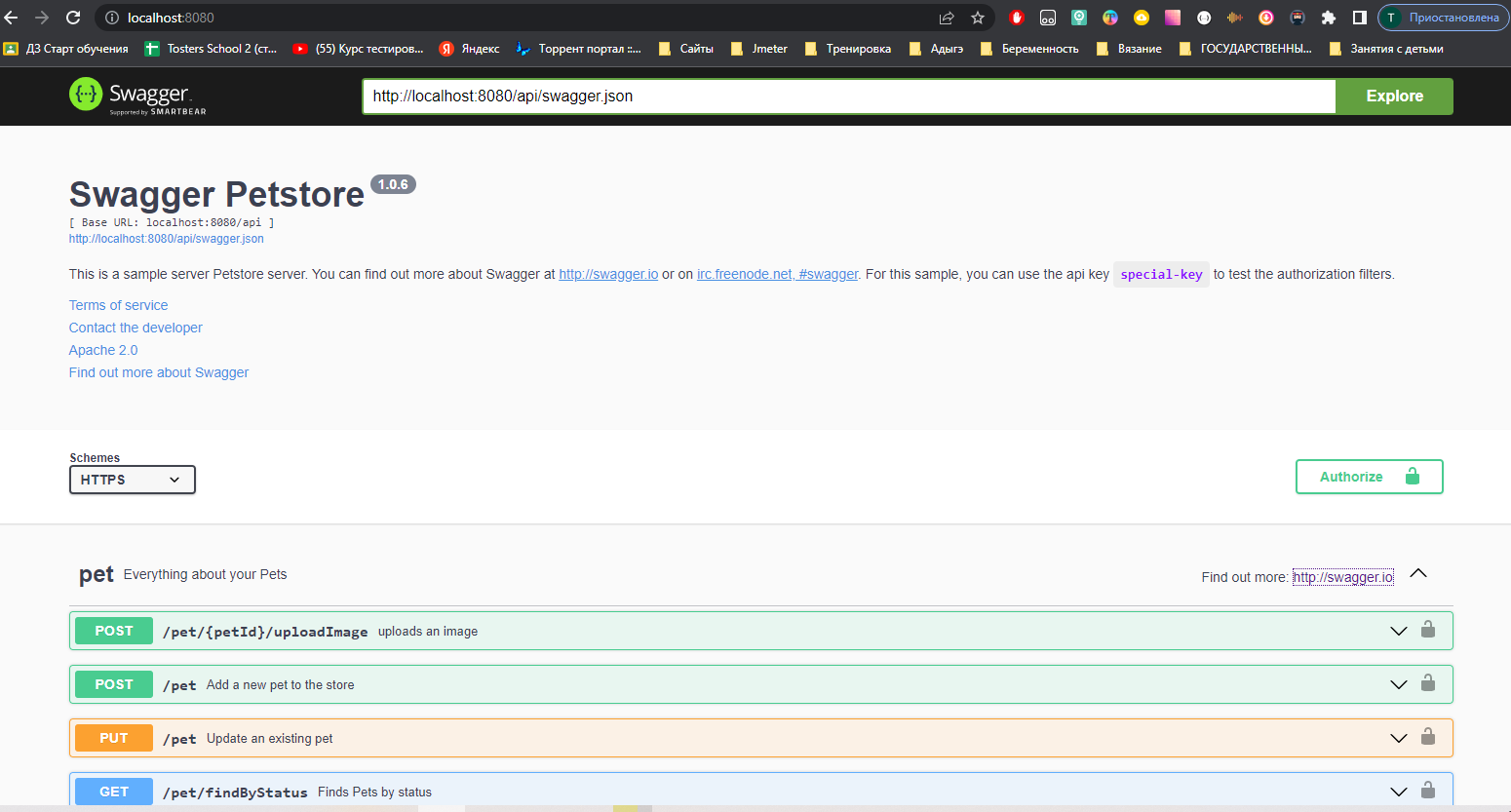
1. Нажать кнопку “Главная” в верхней строке.(На странице первым в списке “Продолжить просмотр” появится последний не до конца просмотренный фильм.

8. Гибкий, т.к. страница непрерывно меняется , появляются новые функции, фильтры, и многое другое. Возможно гибкая методология это необходимость для такого быстро развивающегося приложения.

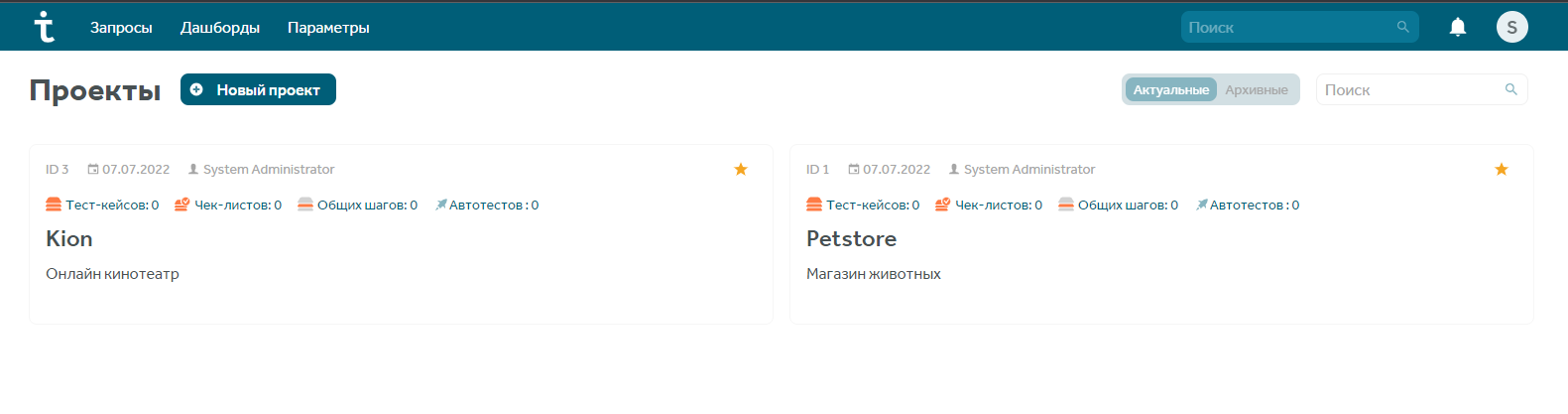
9. Целесообразно автоматизировать тестирование фильтров, т.к. их много, а их комбинаций еще больше.

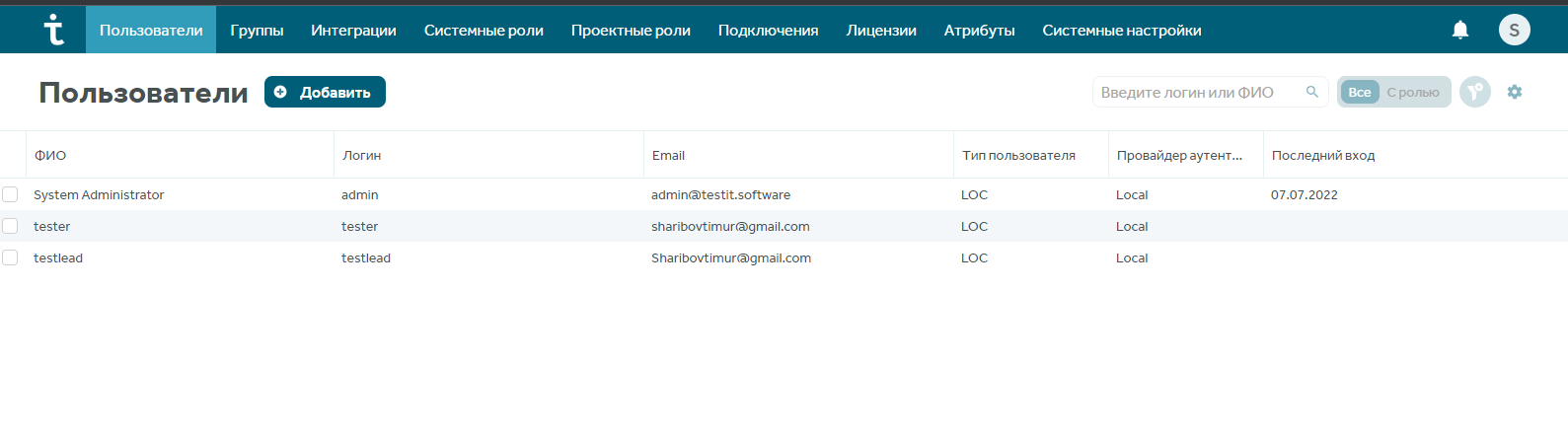




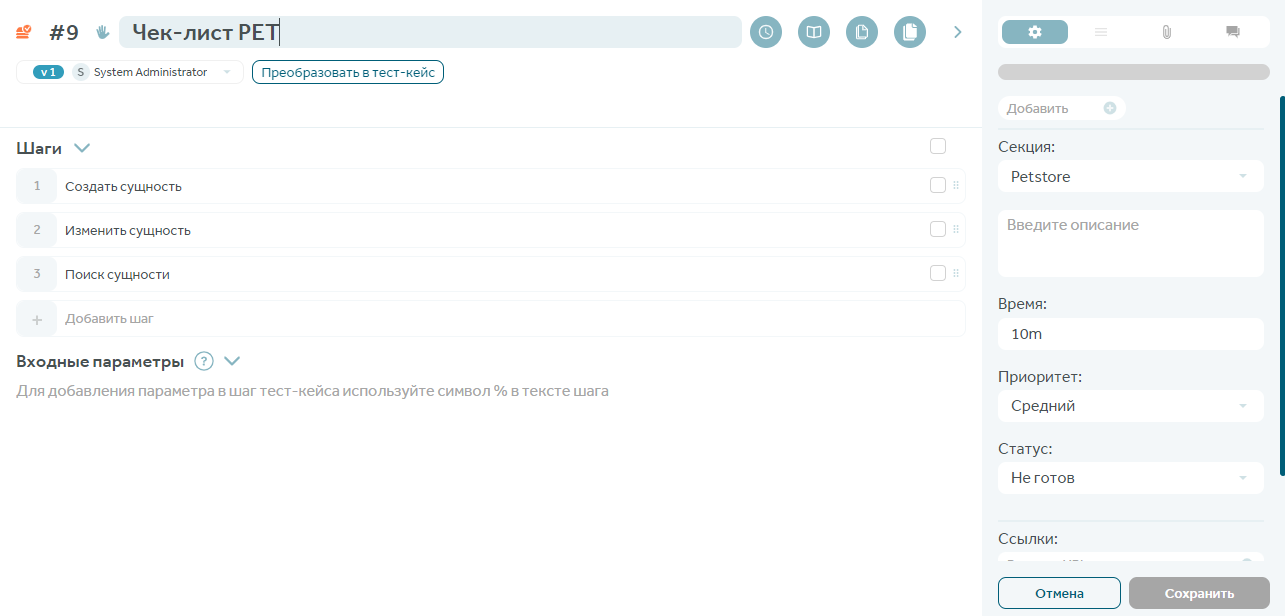
2,

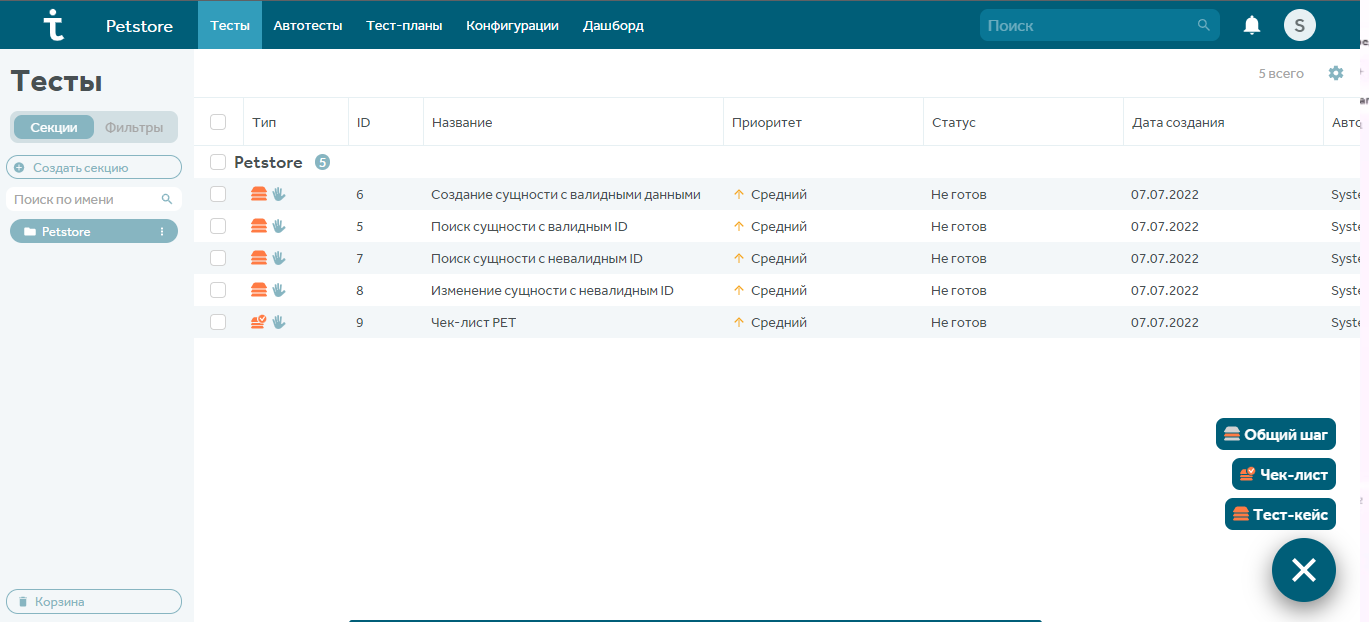
3,

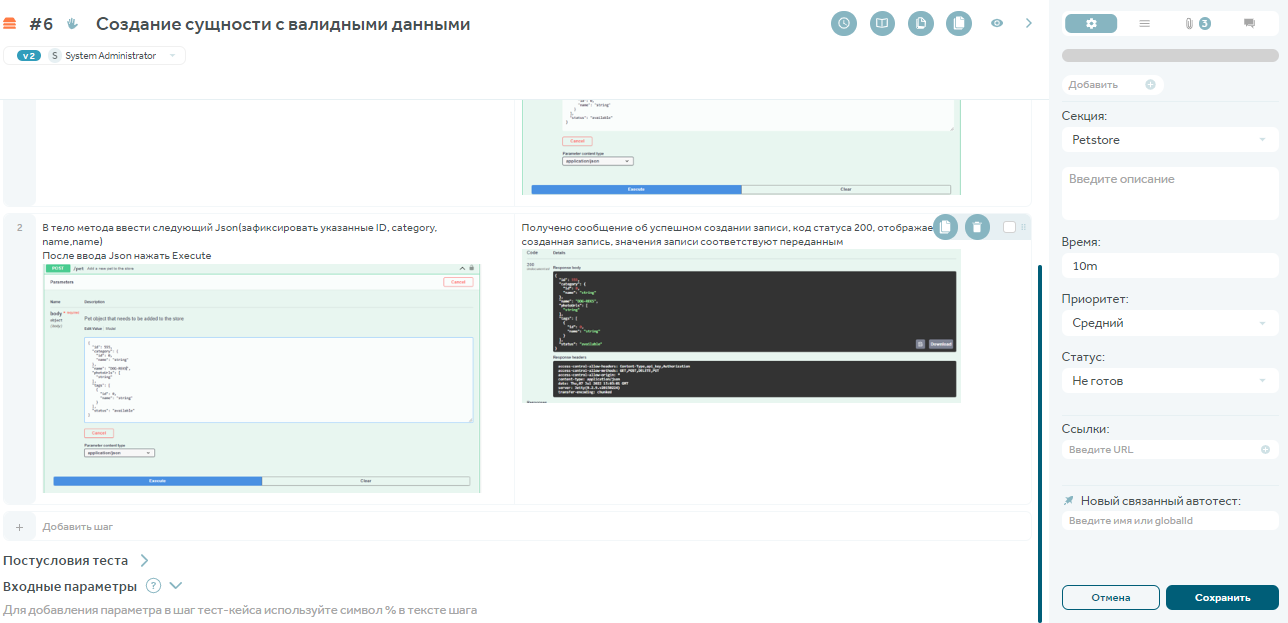


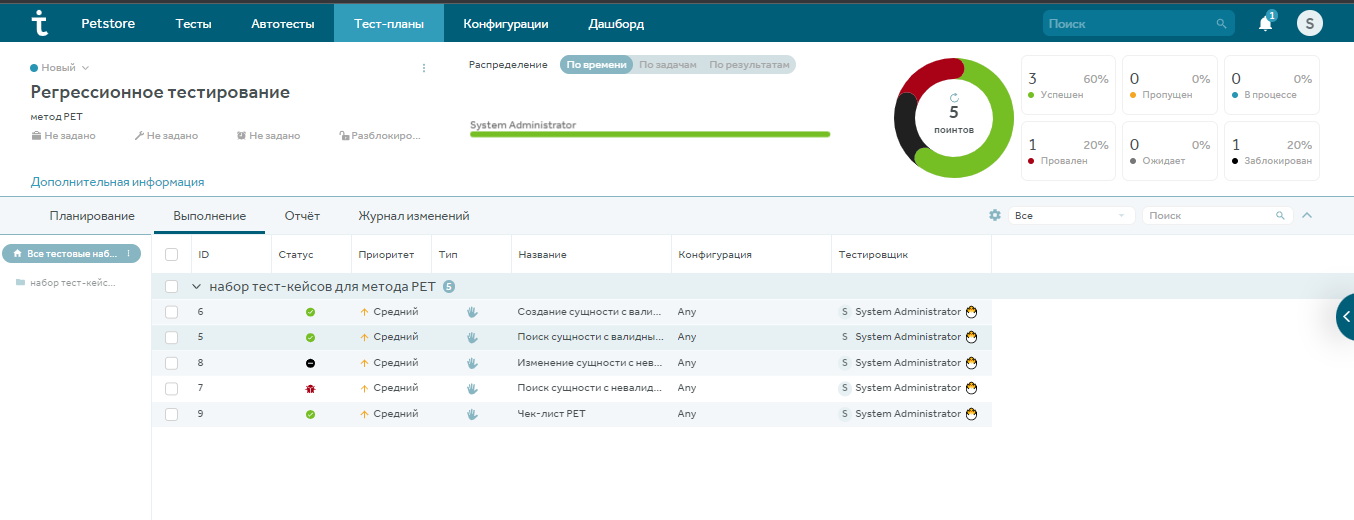


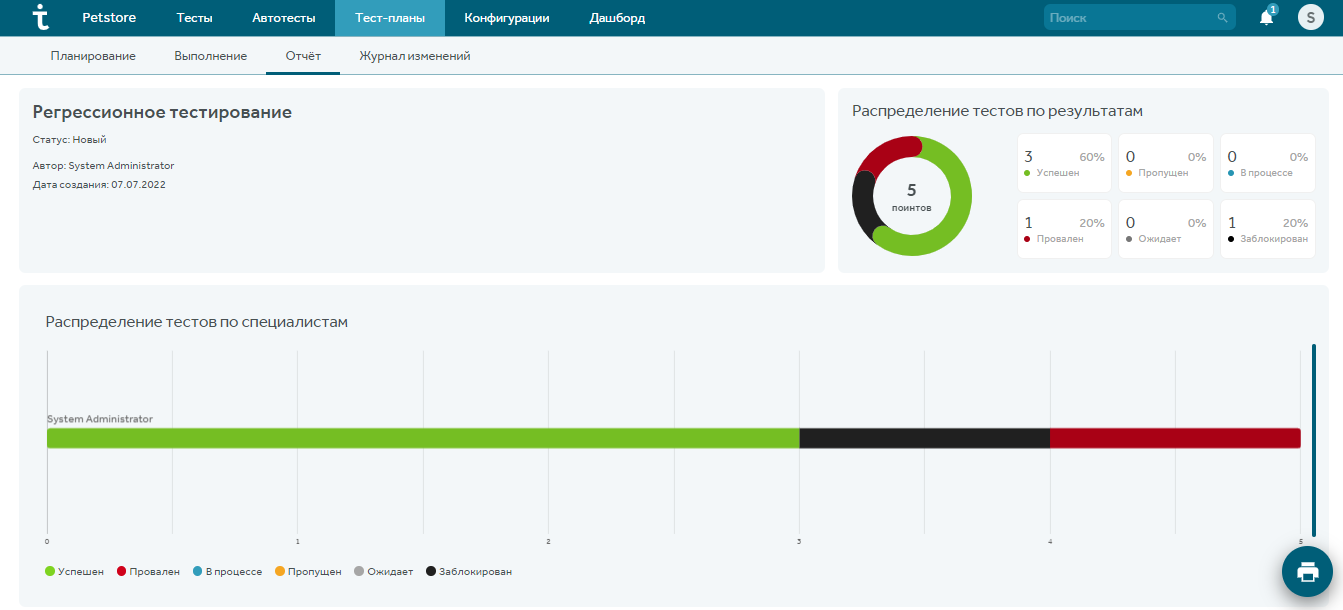
4.

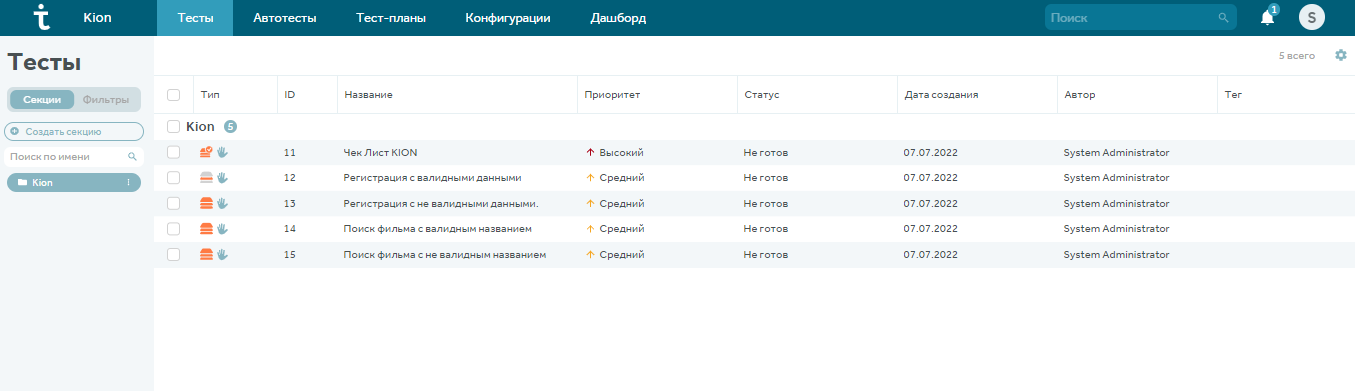


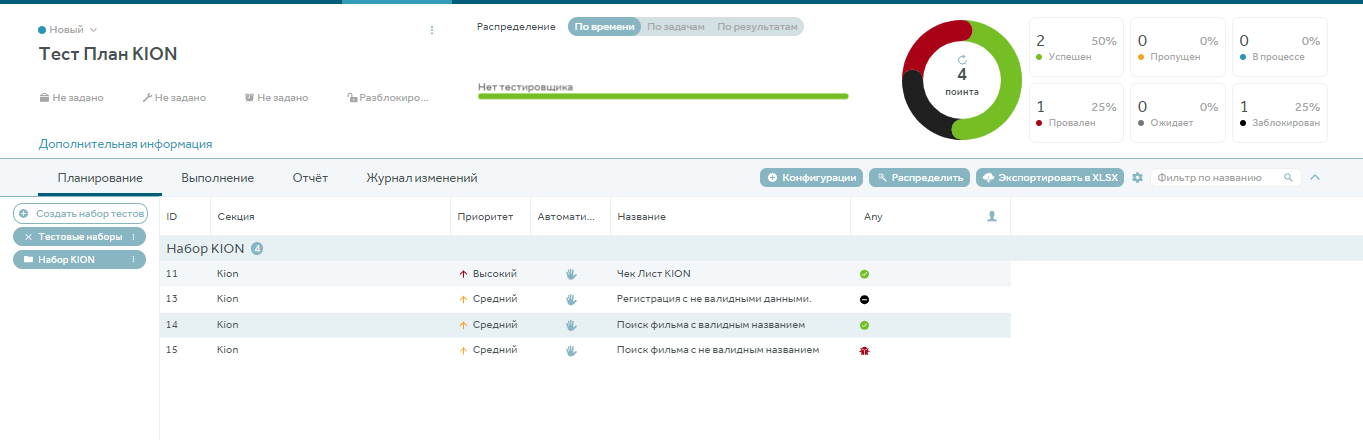


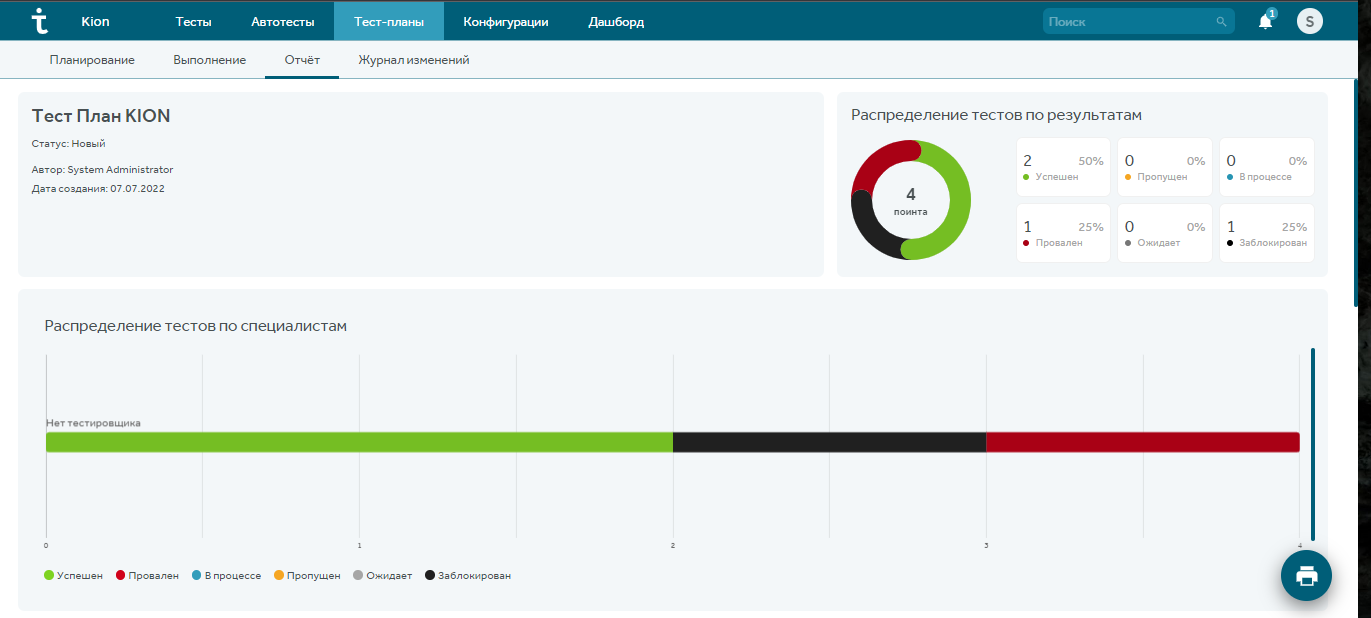


5.

6.

7. 





8. Скриншоты приложены по пунктам.