

Правила работы с карточками на доске:

1. Открываем любую US (колонка **US to do**) и в комментариях проверяем остался ли еще непокрытый функционал или непокрытые Acceptance Criteria, к которым можно написать TC (**МОЖНО** проверить наличие или отсутствие TC на доске, через поиск)




2. Создаем карточку TC, задаем нумерацию и название по US (смотрим примеры других TC)

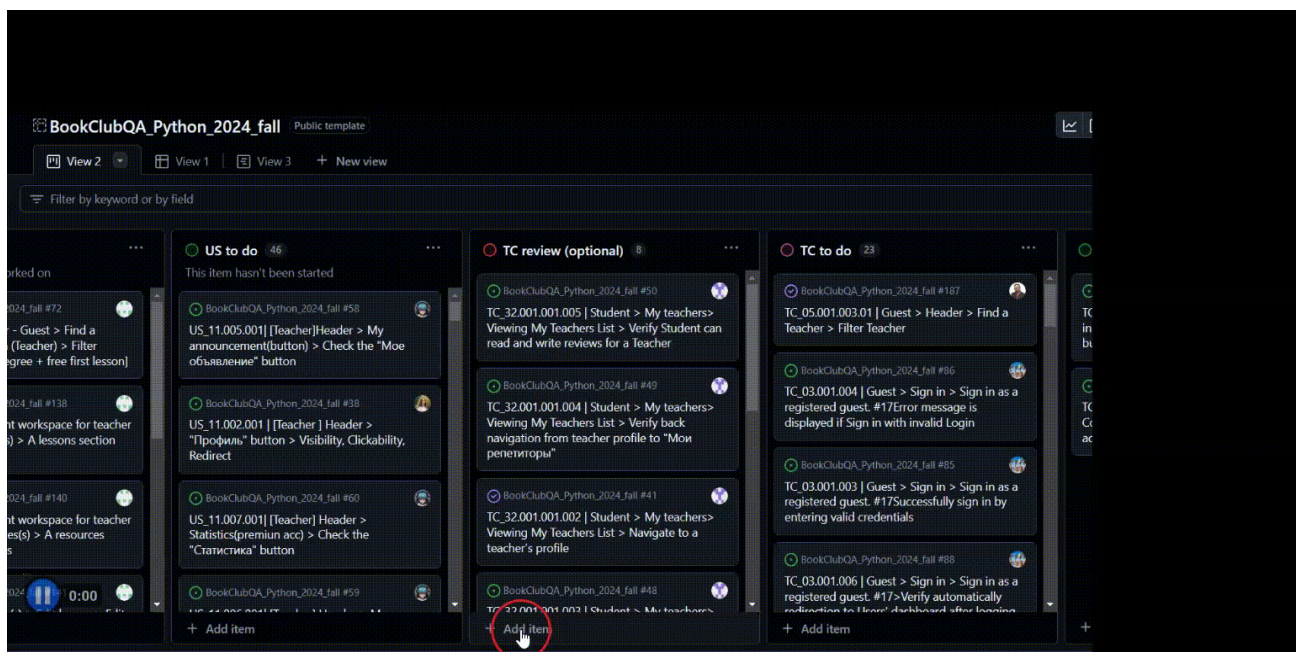
Ввести название новой карточки

TC_{номер User Story}._{порядковый номер тест кейса} | {название страницы и раздела/элемента скопировать из US} > **Verify something**

US_00.001.001 | {Epic} > {Фича, саб эпик (если есть)} > User Story Title

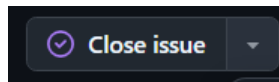
TC_00.001.001.001 | {Epic} > {Фича, саб эпик (если есть)} > User Story Title > Title
TC (Verify something)

 **Название тест кейса должно содержать краткую формулировку проводимой проверки**



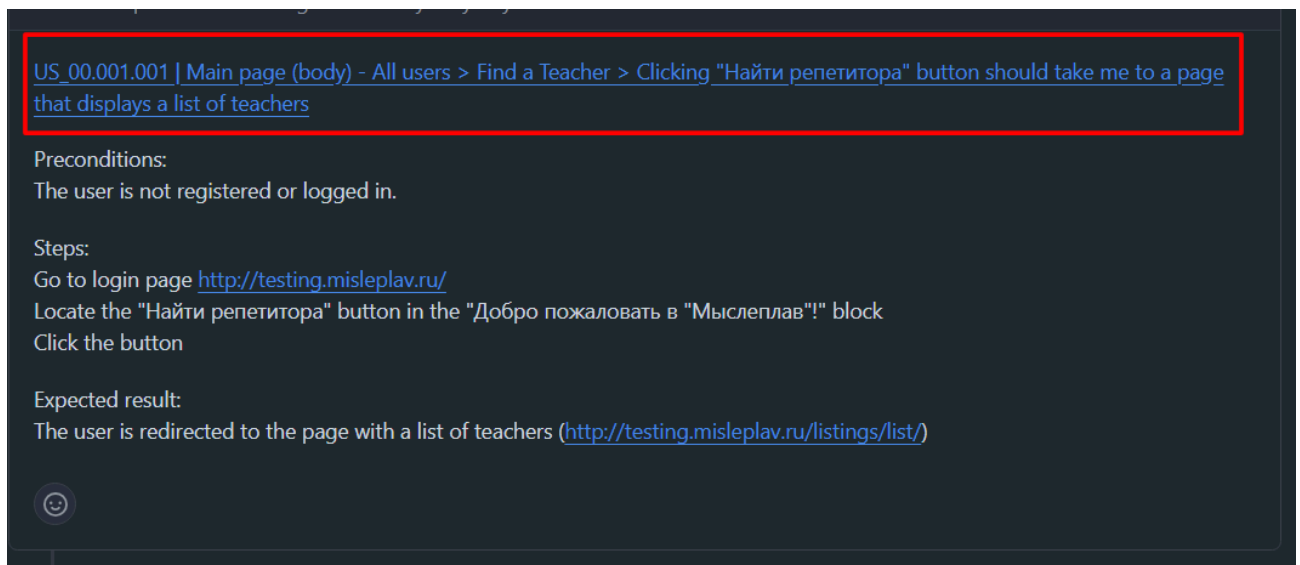
3. Новый тест-кейс помещаем в колонку **TC to do**. Если хотите, чтобы ваш тест-кейс проверили, то закиньте его в **TC review**
4. Карточку TC самостоятельно перемещаем по доске по мере своего продвижения:
 - TC to do -> новый тест-кейс
 - TC review (optional) -> проверка тест-кейса другими участниками (по желанию)
 - AT in progress -> в процессе написания автотеста(IDE)
 - AT review -> автотест находится на ревью (Pull Request)
 - AT done -> после слияния с main(After merged)

НЕ обязательно, сделать

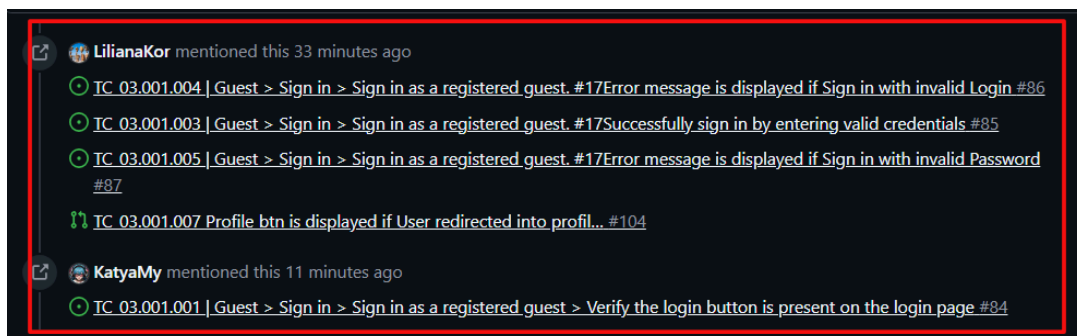


в карточке

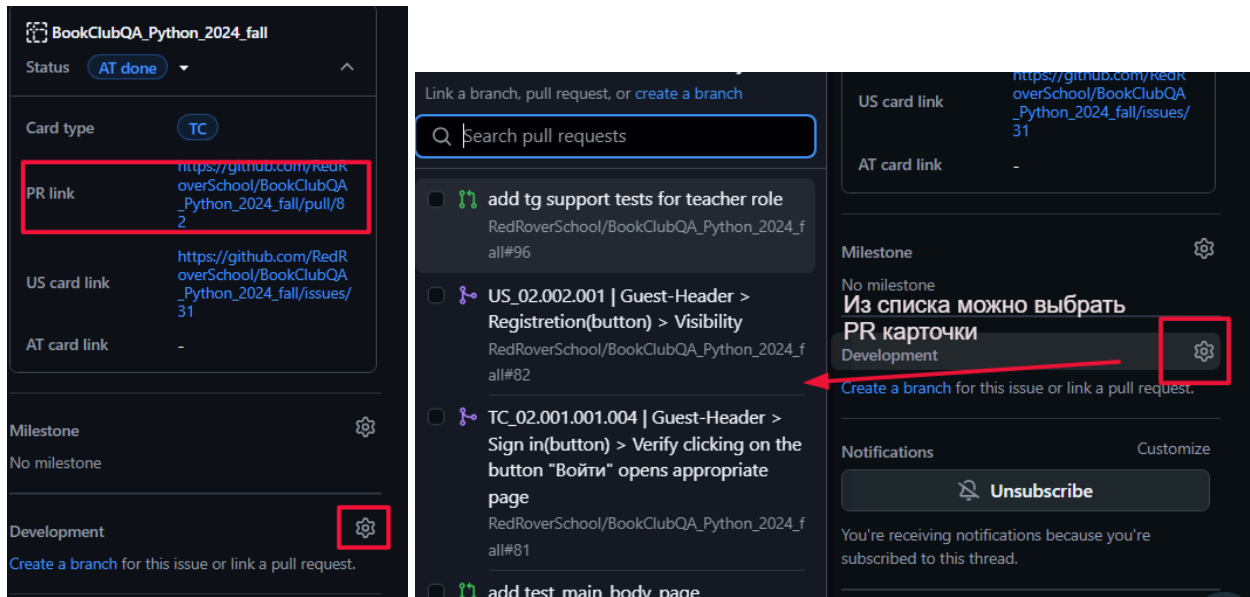
5. Название TC и ссылку на него нужно **обязательно оставляем** ссылку на US в карточке TC(вверху страницы)



В итоге под карточкой US будут вот так(легче отследить)

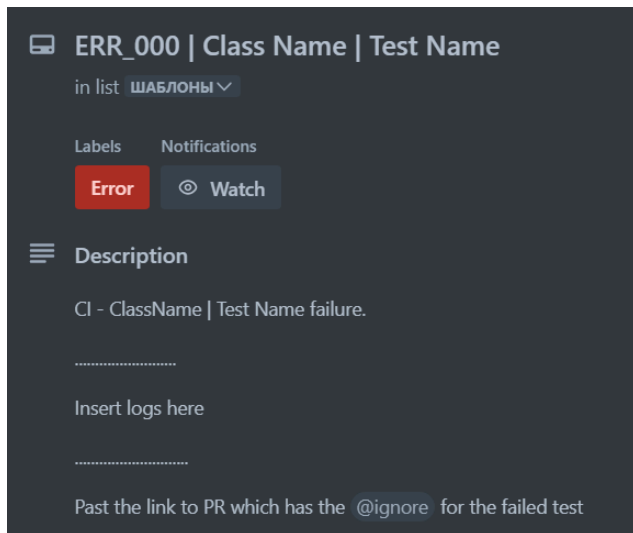


6. В карточку TC добавляем ссылку на PR

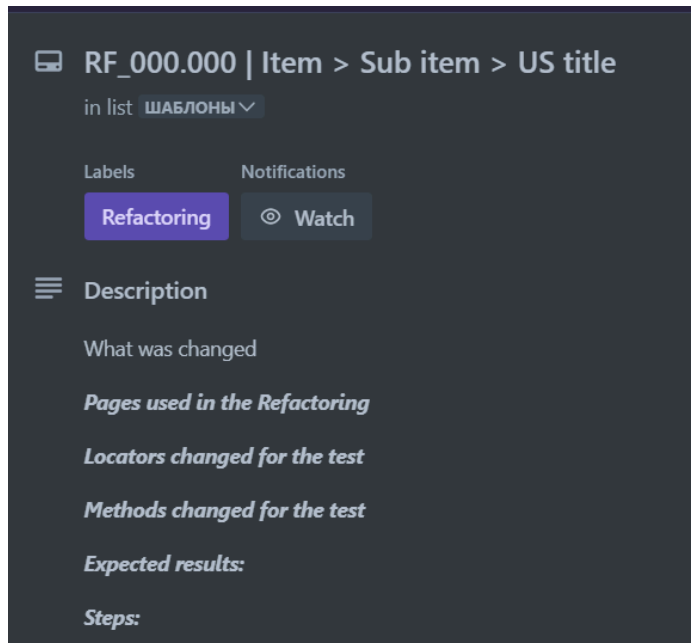


7. Писать несколько автотестов **по одному TC НЕЛЬЗЯ.**

8. **Error:** Если несколько тестов упали, заводим на каждый тест отдельную карточку Error (смотрим примеры других карточек Error)



9. **Refactoring:** Карточки создаем по примеру на доске, задаем нумерацию и название по TC и US. Несколько тестов можно делать в один RF **только** если они связаны.

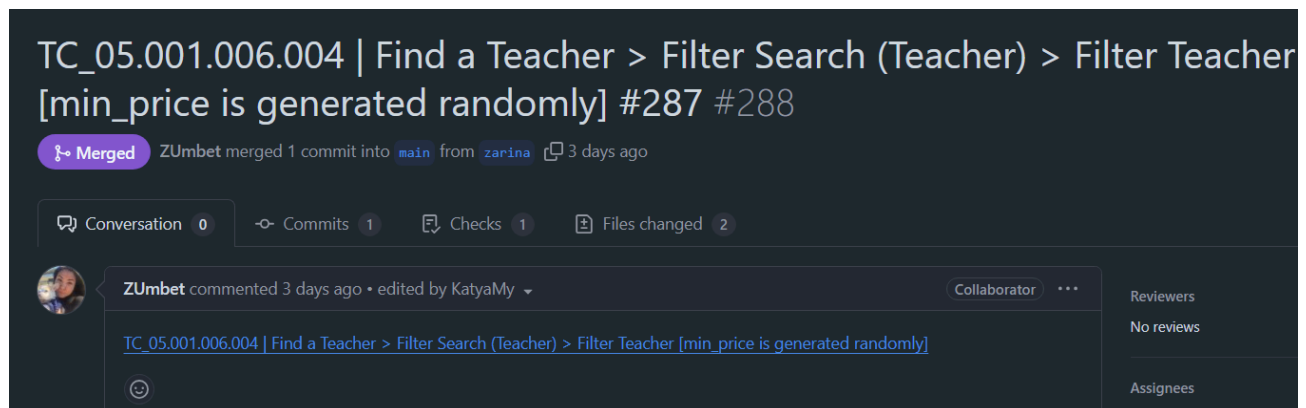


10. Для студентов, которые пишут только ТС и не будут писать АТ(мануальные тестировщики), если такие на проекте есть - ТС создавать в режиме черновика(Draft), НЕ добавляя Assignees. Такие ТС можно будет брать в работу инженерам. Ставить себя в Assignees и делать Convert to issue.

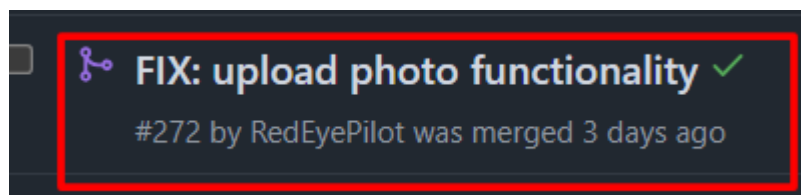
Правила работы с репозиторием:

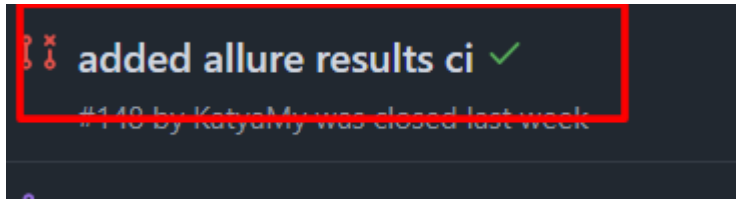
→ Работа на Python-проекте [Инструкция](#)

1. Клонировать [репозиторий](#) проекта к себе
2. В Pycharm от ветки origin/main создаем свою ветку. Название веток пишем по принципу - “инициалы/тип операции или номер ТС” (например, TG/001.003.002)
3. Используем [Ruff](#) - инструмент для анализа Python-кода(см. [Readme](#)):
4. **ПРОГОНЯЕМ ВСЕ ТЕСТЫ** перед коммитом.
5. Упавшие тесты по возможности исправляем, НО не ломая логику теста. Если не уверены в своих силах, ставим `@pytest.mark.xfail(причина)`, и при PR указываем, что метки поставлены (в идеале на доске открыть карточку ERR).
6. **В комментарии к commit добавляем номер ТС или его краткое название**
7. git push ветку в Github
8. Открываем запрос (PR) на слияние с main:
 - a. Название PR пишем по типу PR title == TC name
 - b. В комментарии вставляем ссылку на тк, не номер а полностью как на скрине



Изменения/удаление/улучшения - прописываем:



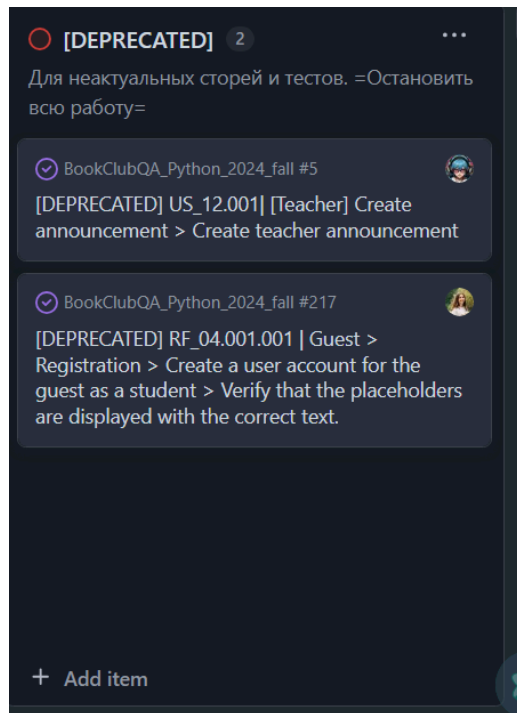


9. Отправляем ссылку на PR в свою группу и, если надо, в канал `#qa_python_project` или `team_python_project_2024_fall`
10. Ждем аппрувы, реагируем на комменты, мерджим кнопкой **"Squash and Merge"**
11. Обязательно удаляем свою ветку после слияния с main на сервере и локально

[Что делать при конфликте?](#)

[Если ваш PR не прошел проверку в CI?!](#)

Про Устаревшие USER STORY и test case - работа с ними



В каком случае User story могут попасть в колонку DEPRECATED?:

Если функционал сайта обновился и US частично или полностью теряют свою актуальность.

И что Тогда:

PM создают новые карточки на основе старых (DEPRECATED), при этом меняя нумерацию, и убирают критерии, которые не соответствуют текущей функциональности сайта.

Что же тестировщики:

US, по которой Вы пишете тесты, сменила свой статус на DEPRECATED?

1. Все работы по этой US прекращаются.
2. Если TC, над которыми Вы работаете, находится в



Отправляются туда же в колонку DEPRECATED (с соответствующей пометкой, см. скриншот > мб ярлык добавим)

P.S. Можете переписать TC под новую, актуальную US с измененной функциональностью (легко запутаться)

А как же АТ, спросите Вы?

Если Функциональность, на которую тесты ссылаются, больше не существует, тесты стоит удалить.

Оставлять такие тесты бессмысленно: они будут только создавать шум в вашем тестовом окружении и усложнять его поддержку.

Карточку ТС по тесту, который на удаление, переселяем из **AT DONE** → **DEPRECATED** (с соответствующей пометкой, см скриншот)

Что если:

Функциональность может вернуться (временные изменения).

- Можно, например: Временно пометить тесты как отключённые (*skip/ignore*) или создать папку **deprecated_tests** и зафиксировать это в документации. [Уточнить у Виктора]

Вариант Рефакторинг?

Если функциональность перенесена на другую страницу сайта или изменилась логика, старые тесты можно адаптировать для новых требований(US), вместо удаления.

Следует завести карточку **RF** со ссылкой на обновленную US и ТС , если такие имеются(а должны быть)

Подведём итог:

1. Если функциональности **точно нет** — лучше удалить тесты.
2. Создать новую US и по ней написать новый ТС проще, чем переписывать старый - не запутаешься.



ZUmbet TC_05.001.006.004 | Find a Teacher > Filter Search (Teacher) > Filter... eff4b3a · 13 hours ago 216 Commits

	.github/workflows	added registration ui, fixed test verify redirection on p...	2 days ago
	Data	added registration ui, fixed test verify redirection on p...	2 days ago
	components	TC_05.001.006.004 Find a Teacher > Filter Search (Tea...	13 hours ago
	core	fixed some tests (#263)	2 days ago
	tests	TC_05.001.006.004 Find a Teacher > Filter Search (Tea...	13 hours ago
	.gitignore	test ci	last week
	README.md	Update README.md	2 days ago
	conftest.py	fixed upload photo	yesterday
	pytest.ini	TC_03.001.005_check_invalid_username	last week
	requirements.txt	Fix allure ci (#168)	last week

README

ЧИТАТЬ НЕ ВРЕДНО, ВРЕДНО НЕ ЧИТАТЬ!



Allure-report

redroverschool.github.io/BookClubQ

Readme

Activity

Custom properties

5 stars

2 watching

13 forks

Report repository

Releases

No releases published

[Create a new release](#)

Packages

No packages published

[Publish your first package](#)

ОТЧЕТ ПО ТЕСТАМ

Возможно, Вы столкнулись с проблемой, что у Вас нет доступа к действиям в [Board Project](#).

Убедитесь что:

- Получили инвайт на почту от @victor

Отпишите пожалуйста в личку ваши данные для добавления в репозиторий хранящий тесты для приложения в формате **"Привет! Мой email привязанный к Github: email@gmail.com, мой ник в Github: nick_github"** - вас пригласят в репозиторий проекта на Github

2. Как можно быстрее примите приглашение,(приглашение активно 7 дней) потому что количество отправляемых инвайтов ограничено
3. После того, как приняли инвайт - вы уже можете делать пул реквесты и добавлять код, **НО у вас еще не будет доступа к борде проекта с карточками сторей и тестов**
4. На эту борду вас добавят **отдельно** и вручную, но только после того, как вы приняли первое приглашение.

! У кого нет прав на работу с доской , напишите свои мейлы[имена github] в Thread

! Не исключены незначительные изменения!(файл все еще в процессе редактирования)

Вернуться к → [python_project](#) документации