

позволяет предоставить пользователю информативную обратную связь о его ошибках.

Тестирование в форме эссе применяется как правило для контрольных мероприятий в рамках СДО университета и т.п. В этом случае задание проверяется преподавателем. В открытых онлайн-курсах такие задания как правило не применяются, так как преподавательского ресурса недостаточно для проверки заданий всех пользователей, число которых может быть очень велико. В редких случаях, когда использование такой формы тестирования все же необходимо (например, при отсутствии технической возможности проверить задачу на программирование), прибегают к системе перекрестной проверки. В таком случае требования к ответу стараются максимально формализовать, чтобы пользователи могли более объективно оценить друг друга. При перекрестном тестировании в качестве итоговой оценки, как правило, выставляется среднее или медианное значение результатов нескольких проверок [4].

Пример интерфейса проверки задания с перекрестным оцениванием на платформе Stepik приведен на рисунке 3.

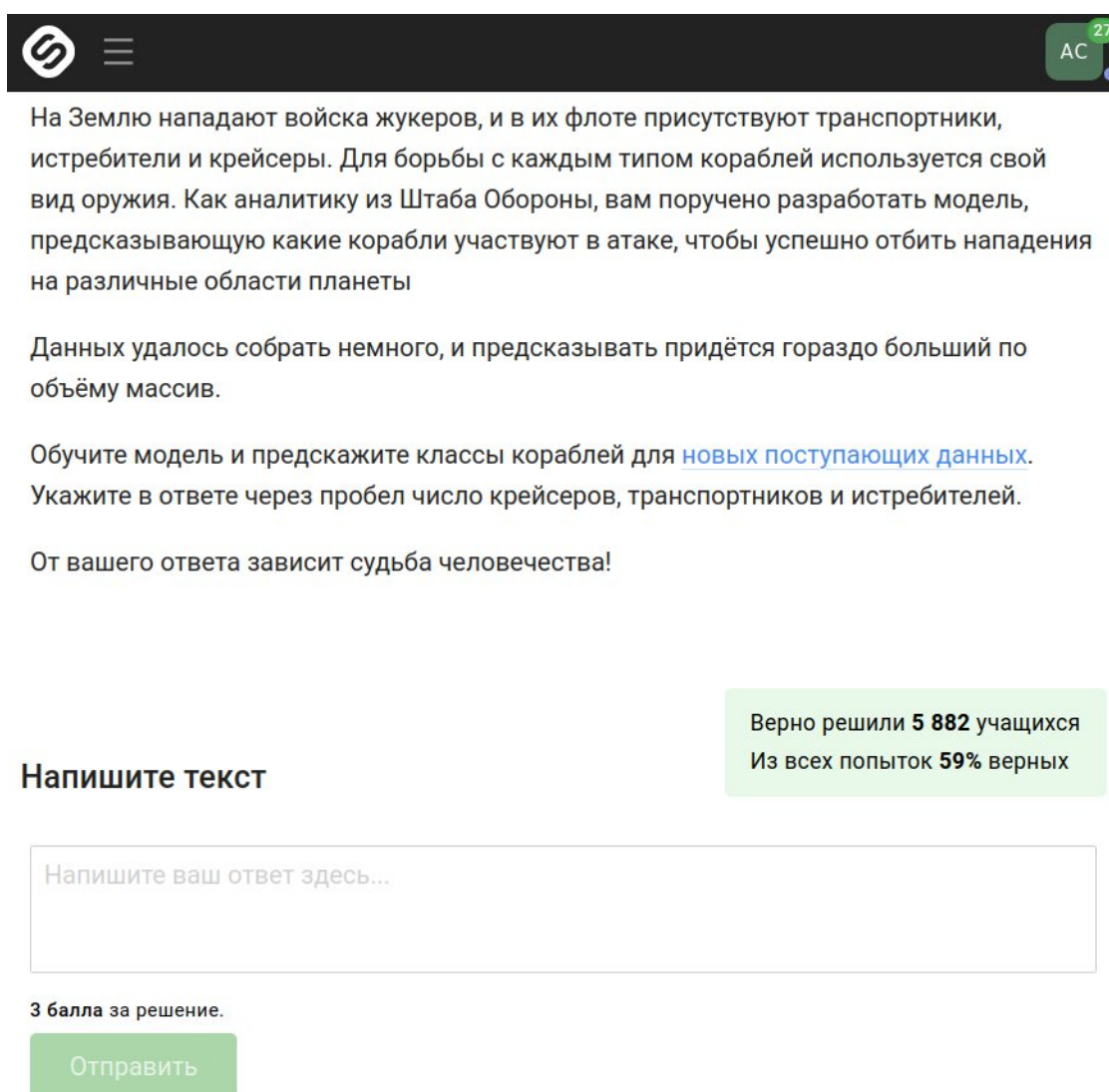
The image shows a screenshot of the Stepik review interface. On the left, there is a 'Рецензия' (Review) section with the following details: 'Шаг: PyTest — параметризация, конфигурирование, плагины Шаг 9', 'Решение: #787527998, 23 октября 2022 г., 22:04, верно', and a 'Условие:' section with a 'Развернуть' (Expand) button. Below this is the 'Решение #787527998' section, which includes a message 'Не забудьте отправить ваше решение на рецензирование.' and a text input field containing the URL 'https://github.com/ekaterinatest/stepik_auto_tests_course2.git'. On the right, there are two criteria for evaluation. 'Критерий 1' (Criterion 1) states 'Тест в репозитории можно запустить командой `pytest --language=es`, тест успешно проходит.' and has a rating of 1. 'Критерий 2' (Criterion 2) states 'Проверка работоспособности кода для разных языков. Добавьте в файл с тестом команду `time.sleep(30)` сразу после открытия ссылки. Запустите тест с параметром `--language=fr` и визуально проверьте, что фраза на кнопке добавления в корзину выглядит так: "Ajouter au panier".' and has a rating of 1. Both criteria have a 'Развернутое объяснение вашей оценки' (Expanded explanation of your rating) field and are marked as mandatory.

Рисунок 3 — Интерфейс проверки задания с перекрестным оцениванием на платформе Stepik

1.2.3 Проверка программ по референсным значениям

Зачастую, так как разработчики онлайн-портала не обладают достаточными ресурсами для создания подсистемы автоматизированного тестирования пользовательских программ, либо сама архитектура проверяемой программы не позволяет протестировать ее автоматически по техническим причинам (например, сама программа пользователя связана с тематикой автоматизированного тестирования, программа связана с машинным обучением и потребляет много вычислительных ресурсов и т.д.)

Пример таких заданий приведены на рисунках 4 и 5.



На Землю нападают войска жуков, и в их флоте присутствуют транспортники, истребители и крейсеры. Для борьбы с каждым типом кораблей используется свой вид оружия. Как аналитику из Штаба Оборона, вам поручено разработать модель, предсказывающую какие корабли участвуют в атаке, чтобы успешно отбить нападения на различные области планеты

Данных удалось собрать немного, и предсказывать придётся гораздо больший по объёму массив.

Обучите модель и предскажите классы кораблей для [новых поступающих данных](#). Укажите в ответе через пробел число крейсеров, транспортников и истребителей.

От вашего ответа зависит судьба человечества!

Верно решили **5 882** учащихся
Из всех попыток **59%** верных

Напишите текст

Напишите ваш ответ здесь...

3 балла за решение.

Отправить

Рисунок 4 — Проверка задания на машинное обучение по референсным значениям

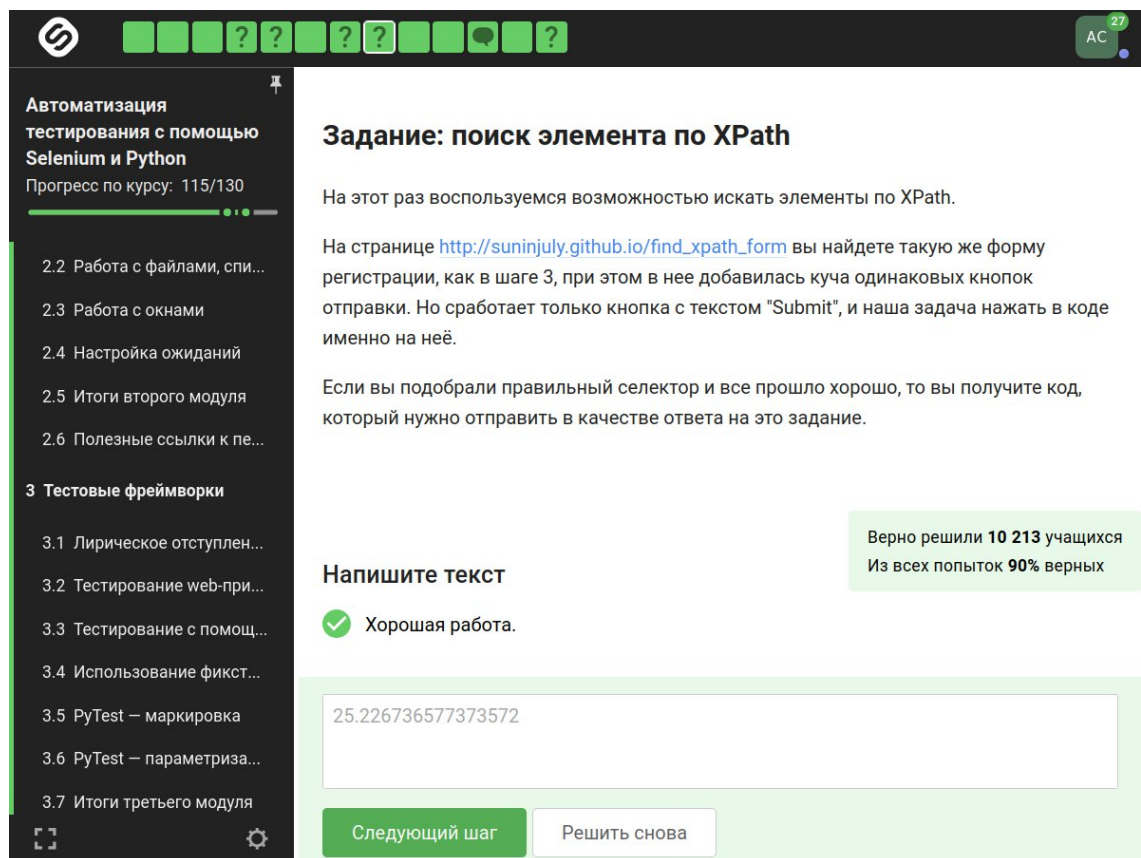


Рисунок 5 — Проверка задания на автоматизированное тестирование по референсным значениям

Функциональная модель тестирования на написание программы с проверкой по референсным значениям приведена на рисунке 6.



Рисунок 6 — Функциональная модель тестирования на написание программы с проверкой по референсным значениям

2 Проектирование программной подсистемы тестирования знаний языков описания аппаратуры

2.1 Проектирование архитектуры

Разработанная подсистема используется веб-приложением образовательного портала для управления учебными материалами, проверки пользовательских ответов на задания и работы со статистикой решения заданий.

Поскольку информация о пользователях используется как в разработанной подсистеме, так и в других компонентах программного обеспечения образовательного портала, БД используется совместно.

Обобщенная архитектура информационной системы показана с помощью контекст-диаграммы в нотации С4 на рисунке 9 [7].

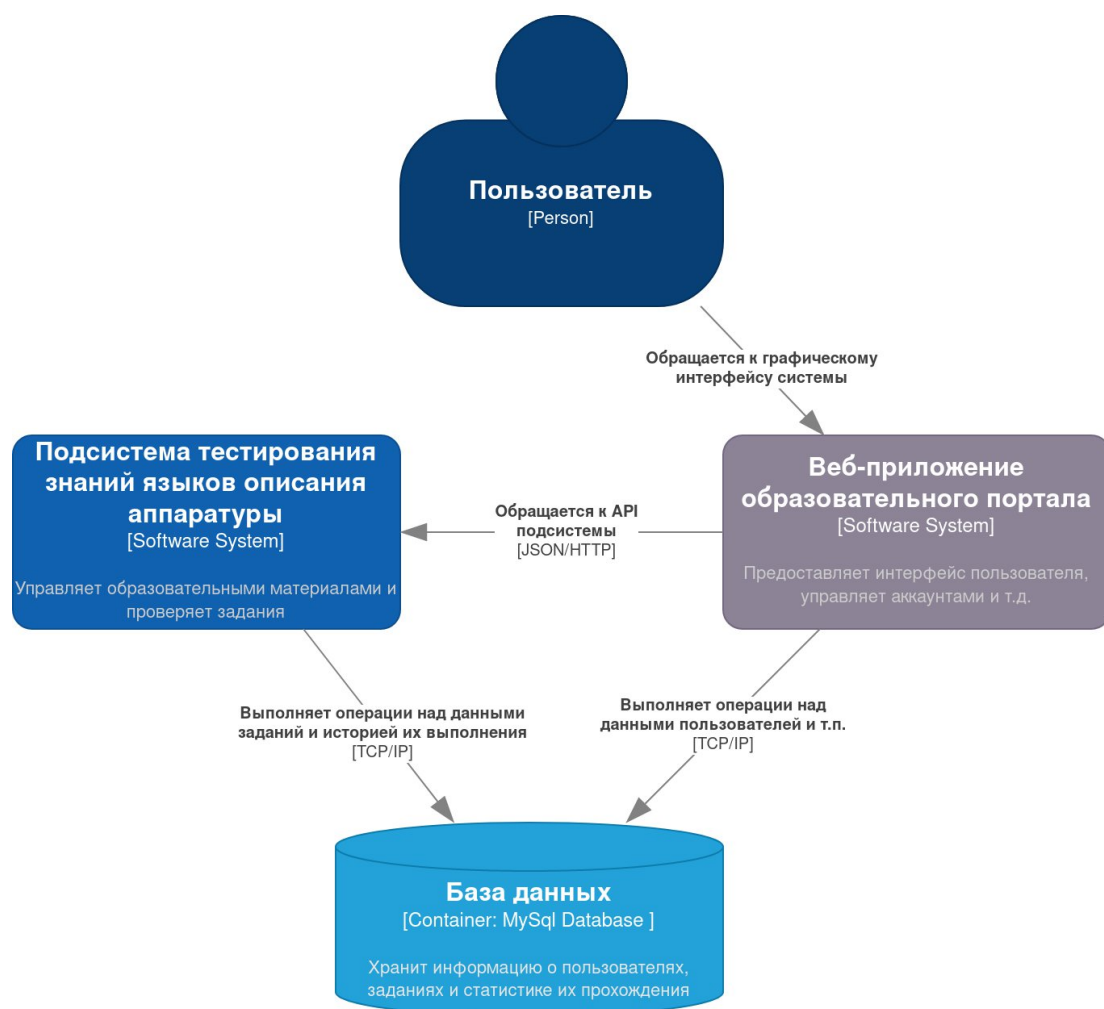


Рисунок 9 — Обобщенная архитектура информационной системы

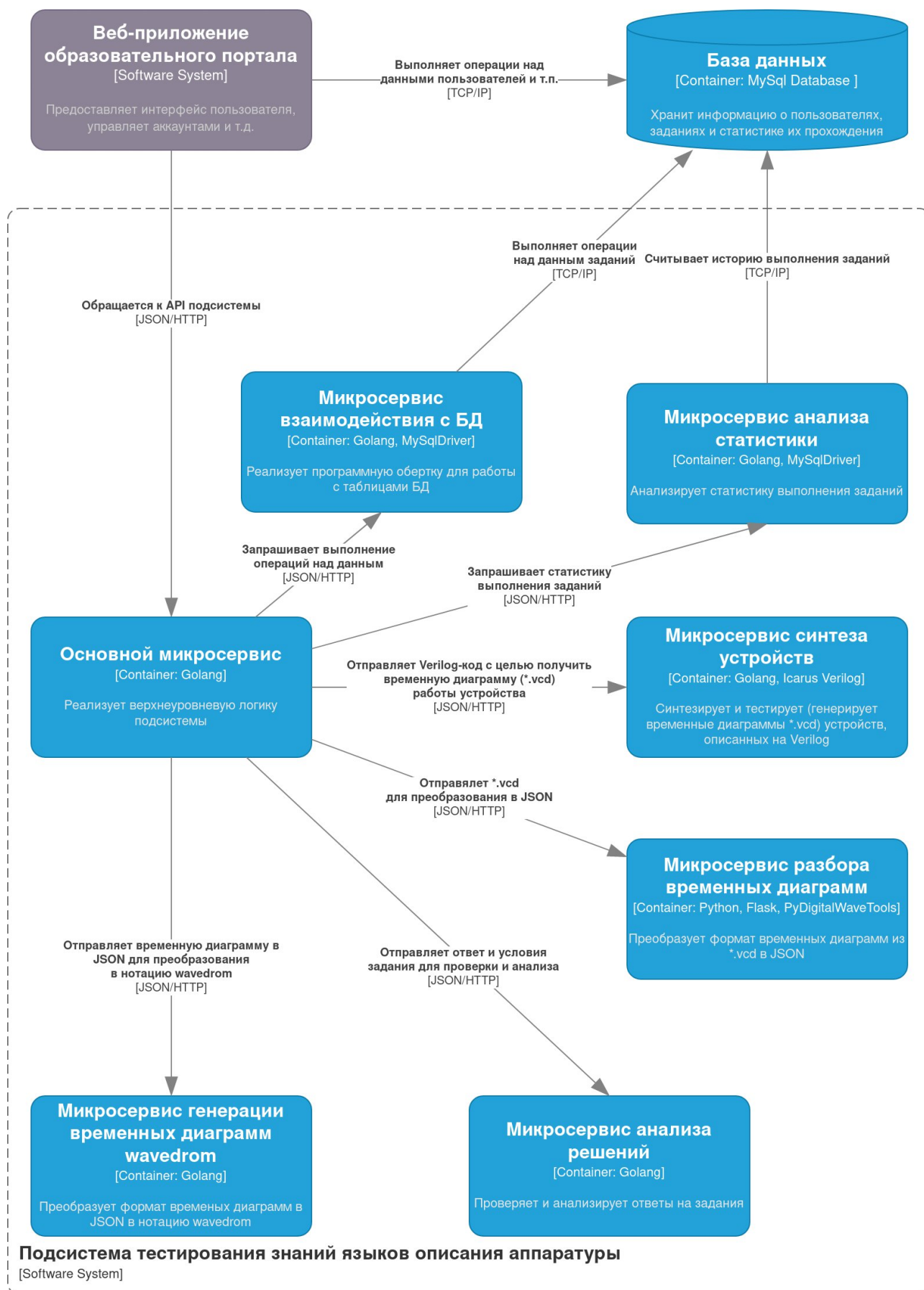


Рисунок 10 — Детализированная архитектура разработанной подсистемы

3.4 Выбор языка программирования и библиотек для функционального тестирования

Так как написание тестов на Golang требует значительного времени и такие тесты сложнее поддерживать в силу непопулярности языка среди тестировщиков, было решено тестировать разработанные микросервисы, предварительно запустив их (см. приложение Б) и обращаясь к ним по протоколу HTTP. Такой подход позволил реализовать тесты не привязываясь к языку реализации исходного ПО.

Поскольку Python обладает простым синтаксисом, большим количеством библиотек и популярен среди тестировщиков (рисунок 21), именно он был выбран для реализации тестов [12].

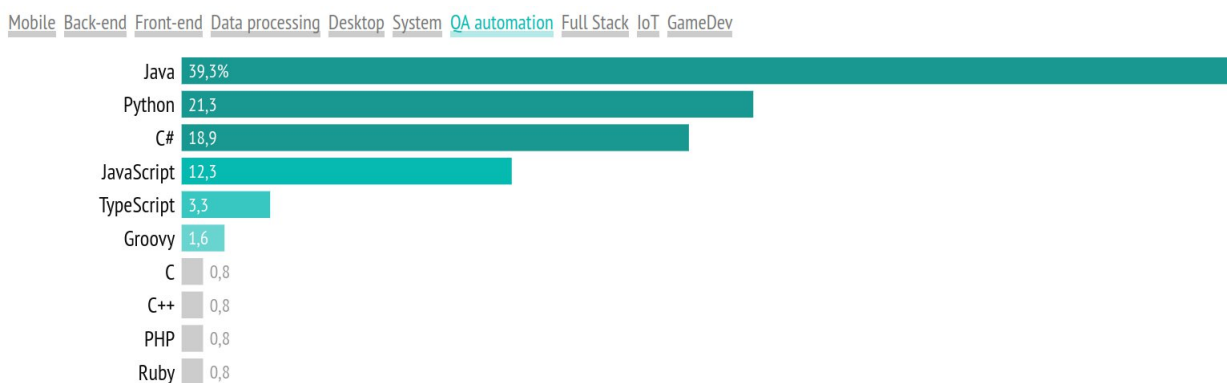


Рисунок 21 — Наиболее популярные языки в области автоматизированного тестирования

В качестве основной библиотеки для тестирования была выбрана библиотека `pytest`, являющаяся одной из наиболее популярных библиотек для автоматизированного тестирования [13].

`Pytest` обладает следующими основными преимуществами [14]:

- меньше повторяющегося кода за счет независимости от API;
- выполнение определенного набора тестов с помощью фильтрации;
- параметризация тестов — запуск одного и того же теста с разными наборами параметров;
- гибкость — архитектура библиотеки основана на плагинах, которые можно установить отдельно;

- полная обратная совместимость с unittest — возможность запуска тестов, написанных на нем;
- выполнение нескольких тестов параллельно;
- установочный код можно использовать повторно.

В дополнение к pytest была использована библиотека allure, формирующая интерактивные отчеты о прохождении тестов. Тесты в allure можно иерархически группировать и сопровождать логами и вложениями. Allure поддерживается не только для Python, но и для Java, JavaScript, Ruby, PHP, .Net и Scala.

Такой широкий набор поддерживаемых языков программирования делает allure (рисунок 22) знакомым многим разработчикам, тестировщикам и менеджерам, что упрощает поддержку тестов [15].

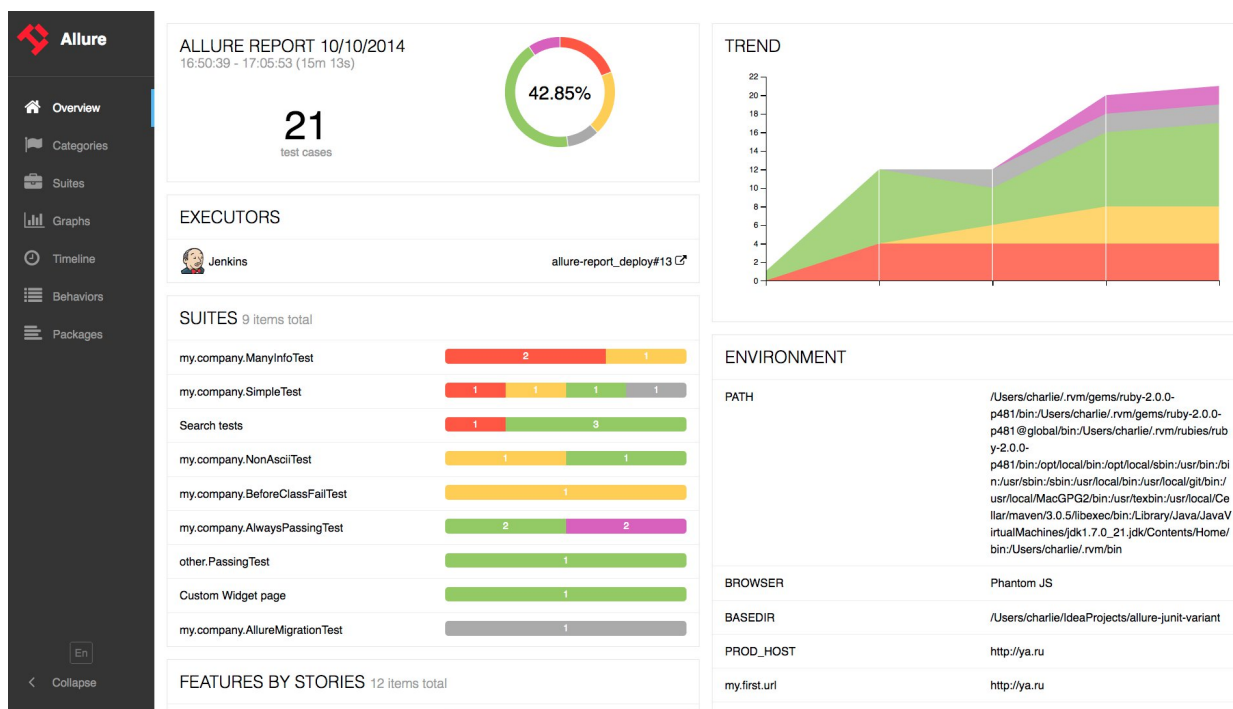


Рисунок 22 — Интерфейс allure

3.5 Реализация и проведение функциональных тестов

Для упрощения написания тестов и генерации отчетов был реализован вспомогательный модуль `utils.py`, отвечающий за отправку `http`-запросов к микросервисам, проверку `http`-ответов и их прикрепление к отчетам в allure. Программный код `utils.py` приведен в листинге 7.

Пример отчета, полученного после выполнения этого теста (и аналогичных ему) приведен на рисунках 23-24.

The screenshot shows the Allure Behaviors report. On the left is a sidebar with navigation links: Overview, Categories, Suites, Graphs, Timeline, Behaviors (selected), and Packages. The main area is titled 'Behaviors' and shows a list of test cases. The test case '#1 test_single_correct_negative' is highlighted. To the right, a detailed view of this test case is shown, including its status (Passed), severity (normal), duration (5ms), description (Test for singlechoice wrong-answered task), and execution details (Set up, Test body, Send request, Check response).

order	name	duration	status
8	test_code_correct_negative	4ms	Passed
9	test_code_correct_positive	3ms	Passed
6	test_multi_correct_false_negative	3ms	Passed
5	test_multi_correct_false_positive	3ms	Passed
4	test_multi_correct_positive	3ms	Passed
7	test_multi_error_size_mismatch	5ms	Passed
1	test_single_correct_negative	5ms	Passed
2	test_single_correct_positive	5ms	Passed
3	test_single_error_overflow	3ms	Passed

Рисунок 23 — Отчет о результате теста

The screenshot shows the Allure Behaviors report with the test case '#1 test_single_correct_negative' selected. The detailed view on the right shows the request and response payloads. The request is a GET request to 'http://127.0.0.1:8083/check' with a JSON payload. The response is a 200 status with a JSON payload indicating the test result.

```

Request URL
http://127.0.0.1:8083/check

Request payload
{
  "type": "singlechoice_test",
  "data": {
    "user_answer_id": 1,
    "task": {
      "correct_answer_id": 2,
      "answers": [
        {
          "text": "Умножение",
          "hint": "Название говорит само за себя"
        },
        {
          "text": "Вычитание",
          "hint": "Перечитай главу"
        },
        {
          "text": "Сложение",
          "hint": "Все верно"
        }
      ]
    }
  }
}

Response payload
{
  "status_str": "ok",
  "status_code": 200,
  "message": "checked",
  "is_correct": false,
  "data": {
    "hint": "Перечитай главу"
  }
}

```

Рисунок 24 — Приложения к отчету

После реализации аналогичным образом всех тестов из таблиц 4-9 они были запущены.

Благодаря параллельному запуску тестов через плагин xdist (в данном случае — в 2 потока) удалось выполнить все тесты менее, чем за 600 мс (рисунок 25).

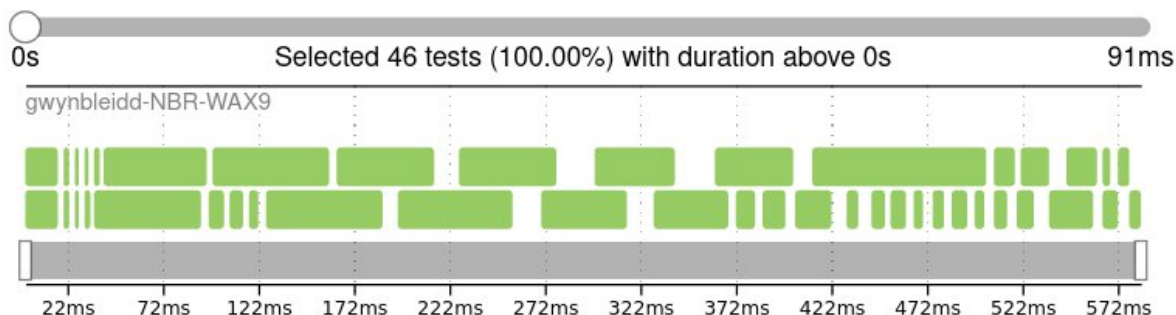


Рисунок 25 — Хронология запуска тестов

Статистика запуска тестов из отчета allure приведена на рисунке 26.

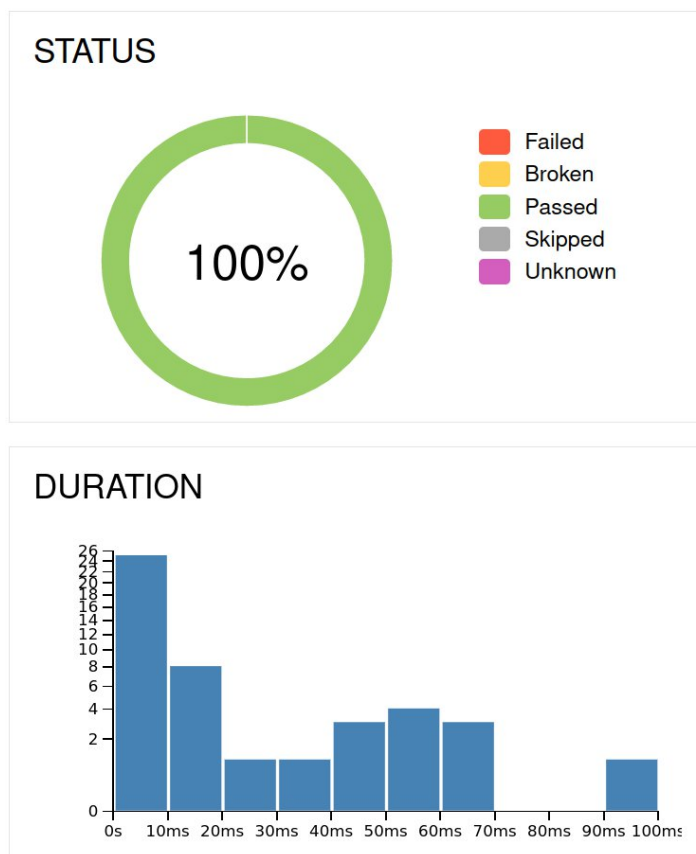


Рисунок 26 — Статистика запуска тестов

Все тесты завершились успешно, о чем свидетельствует приведенная статистика.

визуализировать статистику, допустимое среднее время ответа для запросов к БД было установлено равным 300 мс, а 95 перцентиль — 500 мс.

Результаты нагрузочного тестирования микросервиса разбора временных диаграмм приведены в таблице 10.

Таблица 10 — Результаты нагрузочного тестирования микросервиса разбора временных диаграмм

Статистика запросов							
Запросы	Ошибки	Среднее (мс)	Мин. (мс)	Макс. (мс)	Сред. размер (байт)	RPS	Ошибки / с
18458	0	261	3	706	493	279.9	0.0
Статистика ответов							
50%ile (мс)	60%ile (мс)	70%ile (мс)	80%ile (мс)	90%ile (мс)	95%ile (мс)	99%ile (мс)	100%ile (мс)
260	300	350	400	450	470	520	710

Изменения частоты запросов, времени ответа и количества пользователей показаны на рисунках 27-28.



Рисунок 27 — Результаты нагрузочного тестирования микросервиса разбора временных диаграмм



Рисунок 28 — Результаты нагрузочного тестирования микросервиса разбора временных диаграмм

Результаты нагрузочного тестирования синтезатора приведены в таблице 11.

Таблица 11 — Результаты нагрузочного тестирования синтезатора

Статистика запросов							
Запросы	Ошибки	Среднее (мс)	Мин. (мс)	Макс. (мс)	Сред. размер (байт)	RPS	Ошибки / с
12936	0	525	13	1339	3050	152.8	0.0
Статистика ответов							
50%ile (мс)	60%ile (мс)	70%ile (мс)	80%ile (мс)	90%ile (мс)	95%ile (мс)	99%ile (мс)	100%ile (мс)
580	610	640	680	770	870	950	1300

Изменения частоты запросов, времени ответа и количества пользователей показаны на рисунке 29.

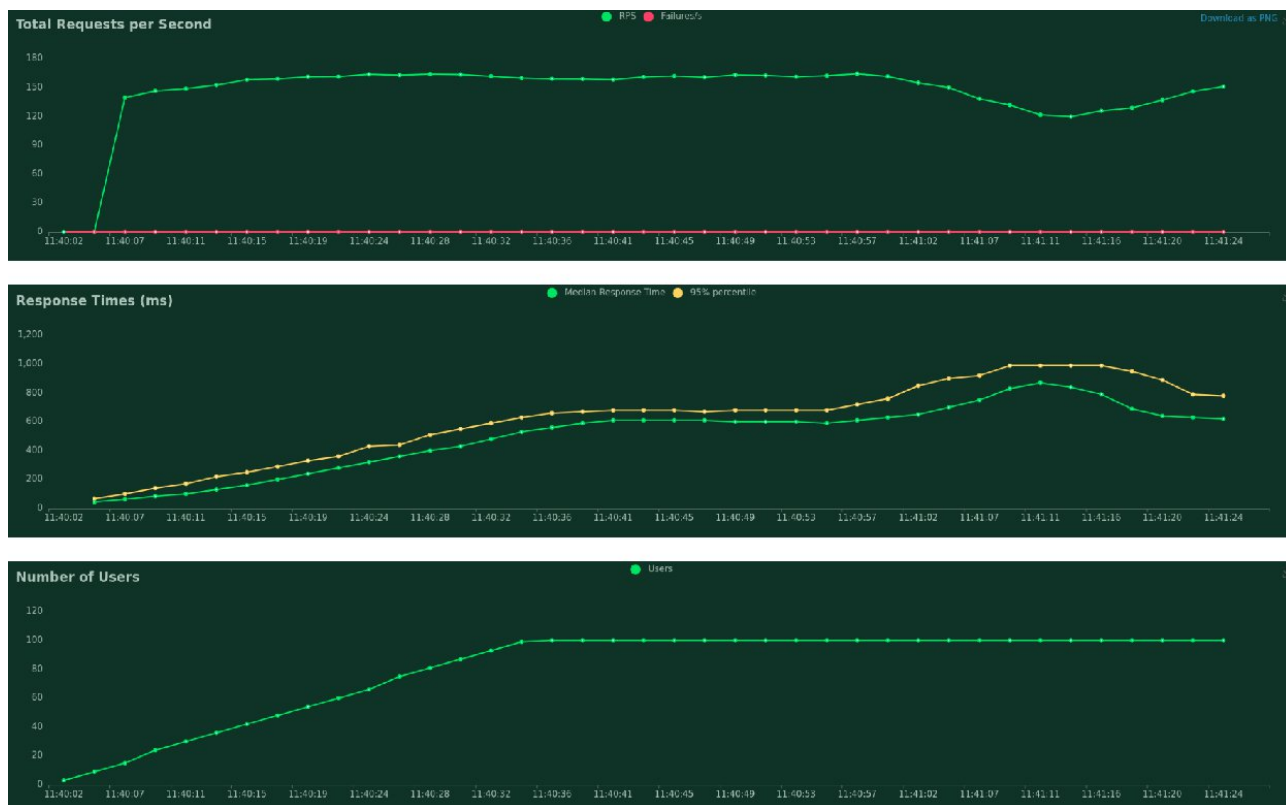


Рисунок 29 — Результаты нагрузочного тестирования синтезатора

Результаты нагрузочного тестирования обращения к БД через слой бизнес-логики приведены в таблице 12.

Таблица 12 — Результаты нагрузочного тестирования обращения к БД через слой бизнес-логики

Статистика запросов								
Маршрут	Запросы	Ошибки	Среднее (мс)	Мин. (мс)	Макс. (мс)	Сред. размер (байт)	RPS	Ошибки / с
/levels	6743	0	85	3	676	506	111.8	0.0
/stats	20486	0	161	2	1151	331	339.5	0.0
Итого	27229	0	142	2	1151	374	451.3	0.0

Продолжение таблицы 12

Статистика ответов								
Маршрут	50%ile (мс)	60%ile (мс)	70%ile (мс)	80%ile (мс)	90%ile (мс)	95%ile (мс)	99%ile (мс)	100%ile (мс)
/levels	78	94	110	130	160	180	220	680
/stats	110	140	190	270	370	460	630	1200
Итого	100	130	160	210	330	430	600	1200

Изменения частоты запросов, времени ответа и количества пользователей показаны на рисунке 30.

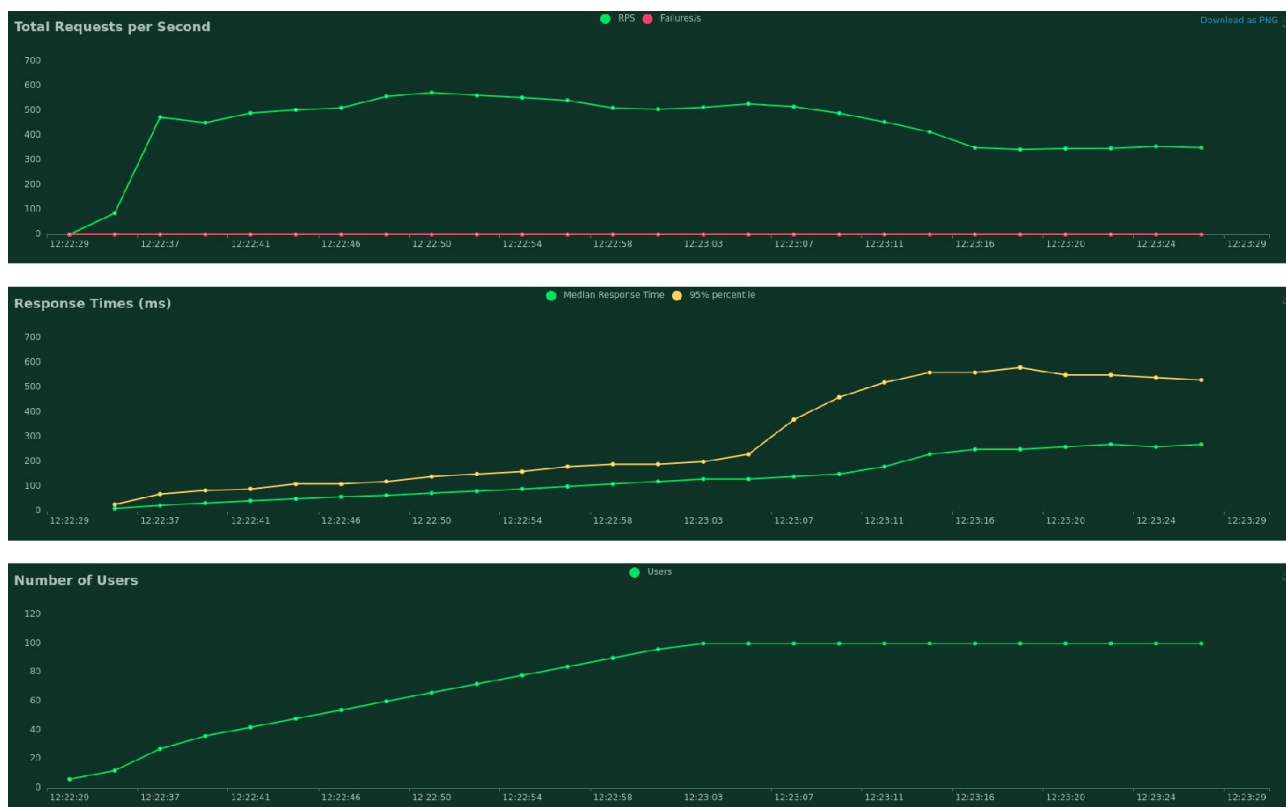


Рисунок 30 — Результаты нагрузочного тестирования обращения к БД через слой бизнес-логики

Тестируемые компоненты в ходе тестирования работали корректно, полученные задержки находятся в допустимых пределах.