МИНОБРНАУКИ РОССИИ
федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский политехнический университет Петра Великого»
Институт дополнительного образования
Высшая инженерная школа

Выпускная квалификационная работа
по программе профессиональной переподготовки
«Разработчик прикладного программного обеспечения (язык C#)»

«Веб-приложение для создания и прохождения тестов»

Выполнил: Григорьев В. А.
Руководитель: Щукин А. В.

Санкт-Петербург
2021

Предметная область

Тестирование (от слова test — испытание, проверка) применяется для определения пригодности объекта тестирования для выполнения тех или иных функций.

Традиционный тест - метод диагностики испытуемых, в котором они отвечают на одни задания, в одинаковое время, в одинаковых условиях и с одинаковой оценкой.

В заданиях открытого типа со свободным изложением испытуемый должен самостоятельно сформулировать ответ, никакие ограничения на них в задании не накладываются.

Цели и задачи

Цель ВКР:

- Подготовить веб приложение для создания и прохождения традиционных тестов с вопросами открытого типа со свободным изложением.

Задачи:

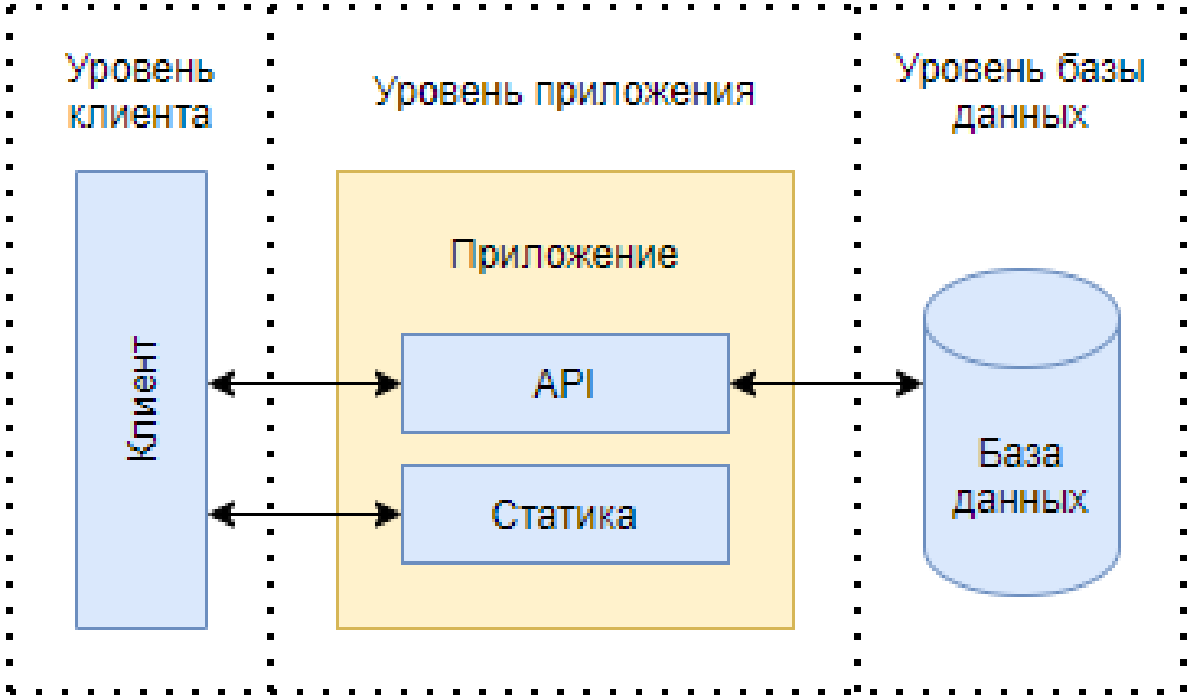
- 1. Разработать структуру базы данных для хранения данных.
- 2. Разработать backend часть приложения на ASP.NET Core WebApi.
- 3. Разработать frontend часть приложения в виде html страниц.
- 4. Провести тестирование.

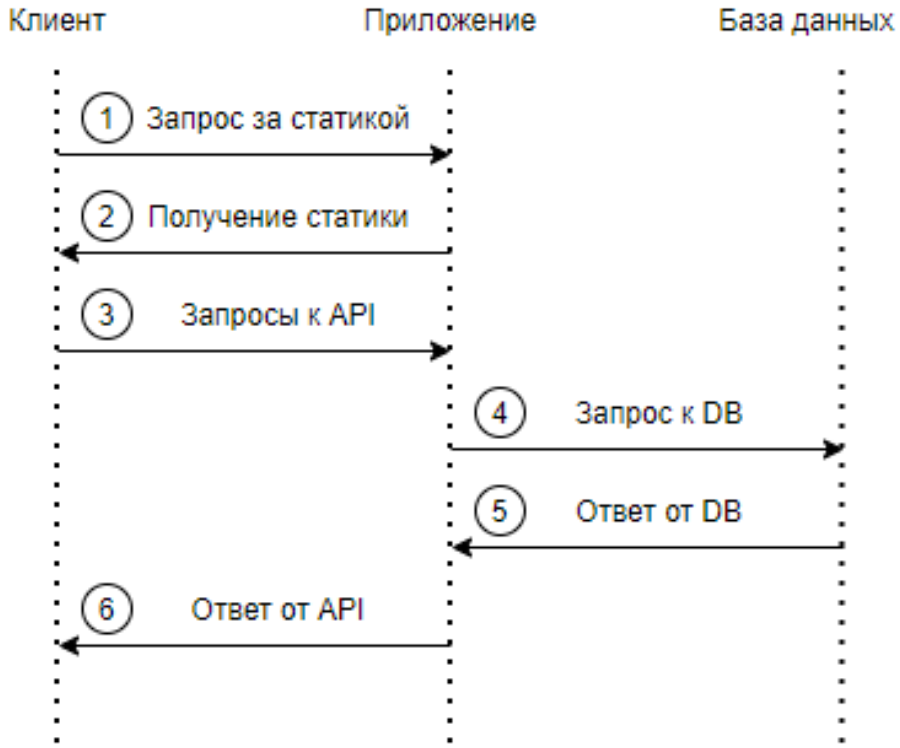
Требования

Приложение должно обеспечивать следующую функциональность:

- Авторизация и аутентификация.
- Получение списка тестов, созданных авторизованным пользователем.
- Создание и редактирование тестов.
- Прохождение существующих тестов авторизованным пользователем.
- Получение статистики по выполненным тестам.
- Вывод логов приложения в консоль и в файл.
- Сбор и отправка метрик о состоянии приложения.

Архитектура





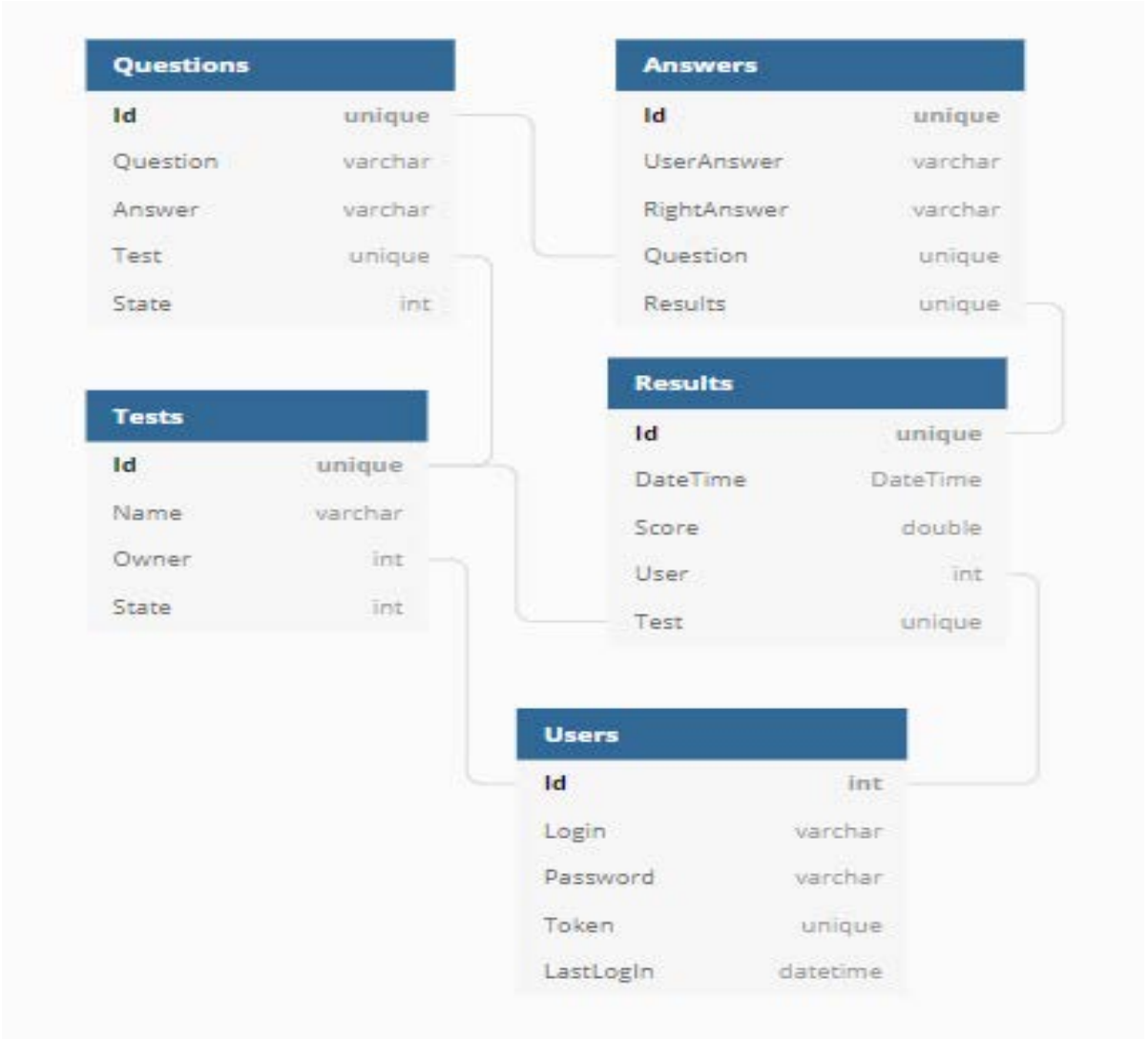
3-х уровневая клиент-серверная модель

База данных

ORM - EF Core.

СУБД - SQLite.

Таблицы: Users, Tests, Questions, Answers и Results.



Аутентификация и авторизация

Имя метода	Тип	Адрес	Параметры	Тело запроса	Тело ответа
RegisterUserAsync	POST	/register	-	UserDto	(Guid)Token
LoginAsync	POST	/login	-	UserDto	(Guid)Token
LogoutAsync	GET	/logout	-	-	-

Работа с сущностью теста

Имя метода	Тип	Адрес	Параметры	Тело запроса	Тело ответа
GetListAsync	GET	/List	-	-	List<TestDto>
AddAsync	GET	/AddEmpty	-	-	(Guid)TestId
UpdateAsync	PATCH	/	-	TestDto	-
GetAsync	GET	/	(Guid)testId	-	TestDto
RemoveAsync	DELETE	/	(Guid)testId	-	-

Работа с сущностью вопроса

Имя метода	Тип	Адрес	Параметры	Тело запроса	Тело ответа
GetListAsync	GET	/List	(Guid) testId	-	List<QuestionDto>
AddAsync	GET	/AddEmpty	(Guid) testId	-	(Guid)QuestionId
UpdateAsync	PATCH	/	-	QuestionDto	-
RemoveAsync	DELETE	/	(Guid) questionId	-	-

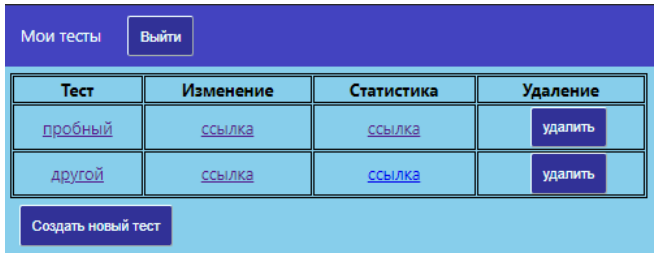
Работа с сущностью ответа

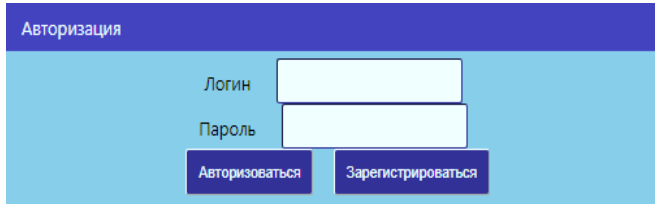
Имя метода	Тип	Адрес	Параметры	Тело запроса	Тело ответа
SendListAsync	POST	/SendList	-	List<AnswerDto>	(Guid)resultId
GetListAsync	GET	/List	(Guid) resultId	-	List<AnswerDto>

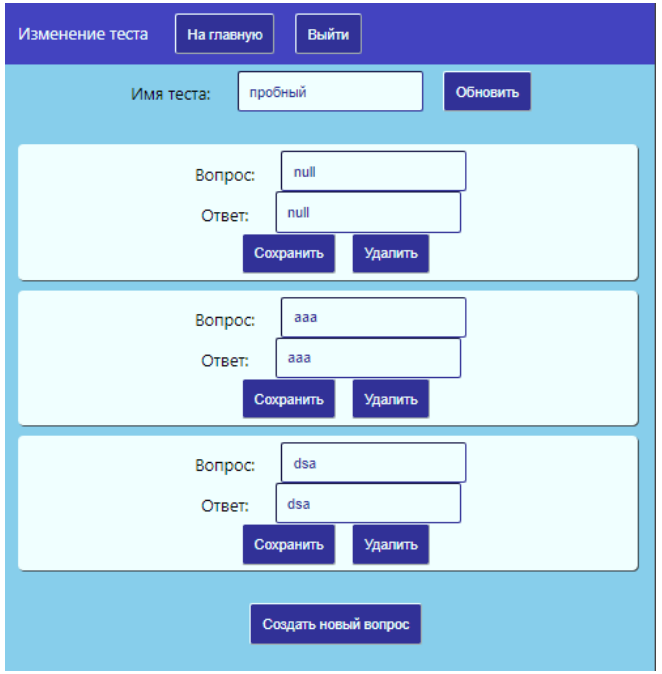
Работа по генерации статистики

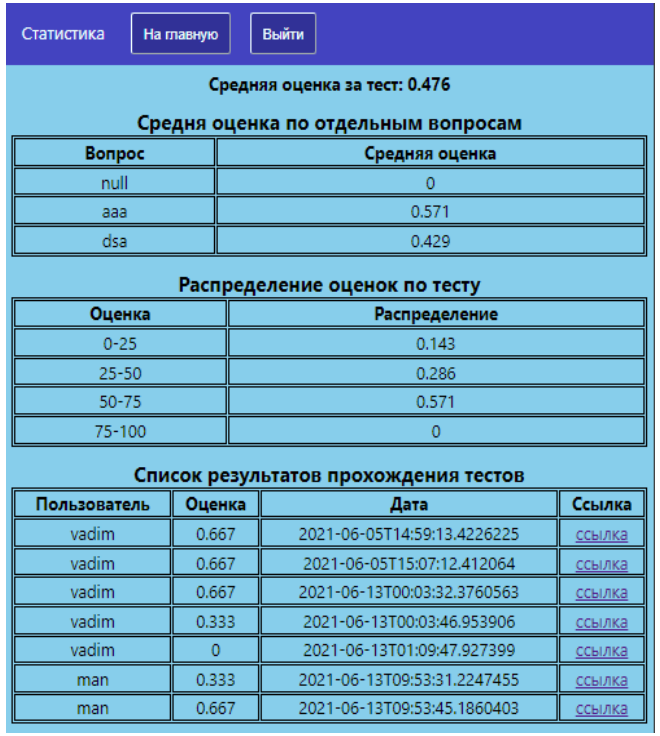
Имя метода	Тип	Адрес	Параметры	Тело запроса	Тело ответа
GetResultListAsync	GET	/GetResultList	(Guid) testId	-	List<ResultDto>
GetAverageScoreAsync	GET	/GetAverageScore	(Guid) testId	-	double
GetScorePerQuestionsAsync	GET	/GetScorePerQuestions	(Guid) testId	-	List<(string, double)>
GetScoreDistributionAsync	GET	/GetScoreDistribution	(Guid) testId	-	List<double>

Frontend







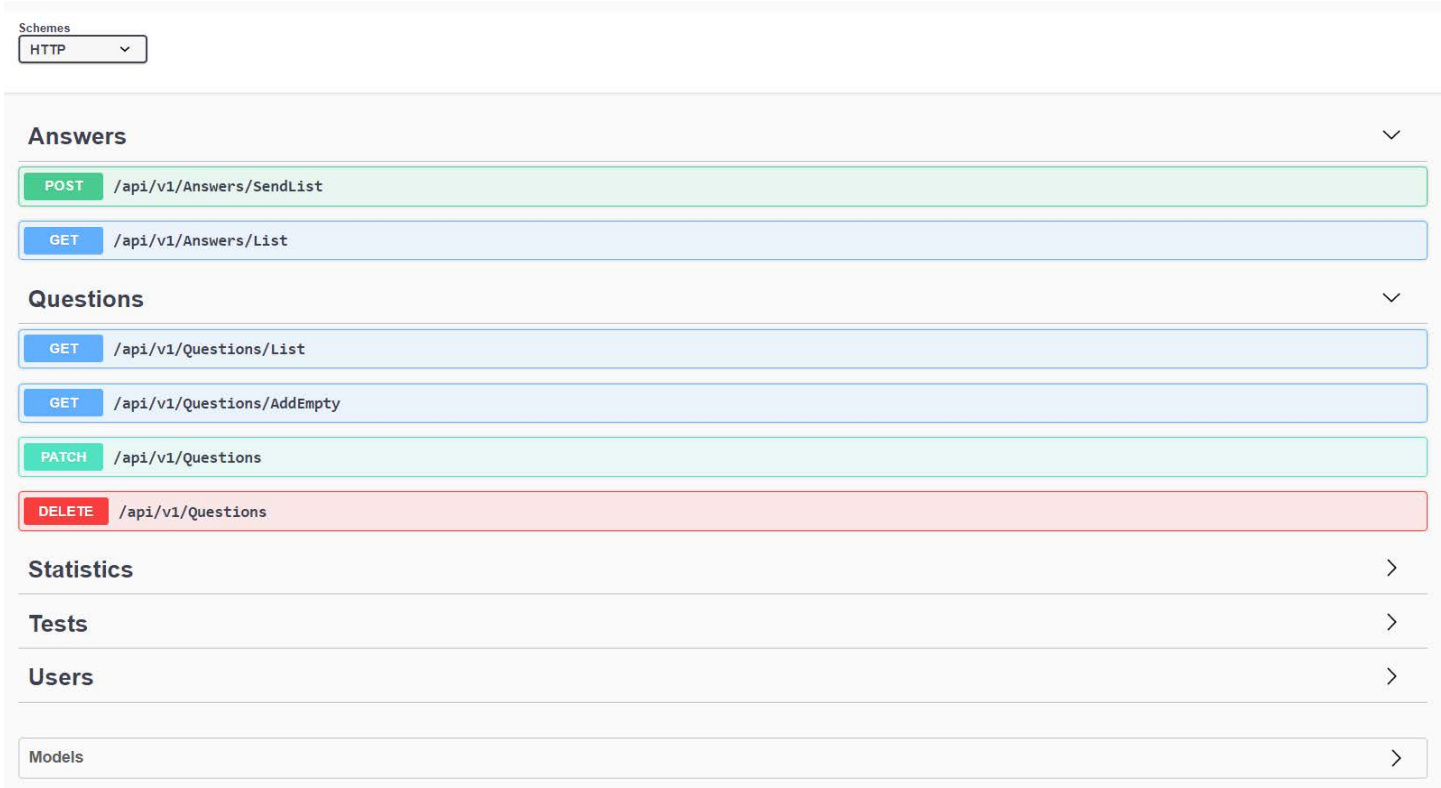


В качестве Frontend части приложения используются статические страницы (HTML+CSS+JS).

Swagger

Для документации API используется библиотека NSwag.

По пути /swagger выводится UI для Swagger.

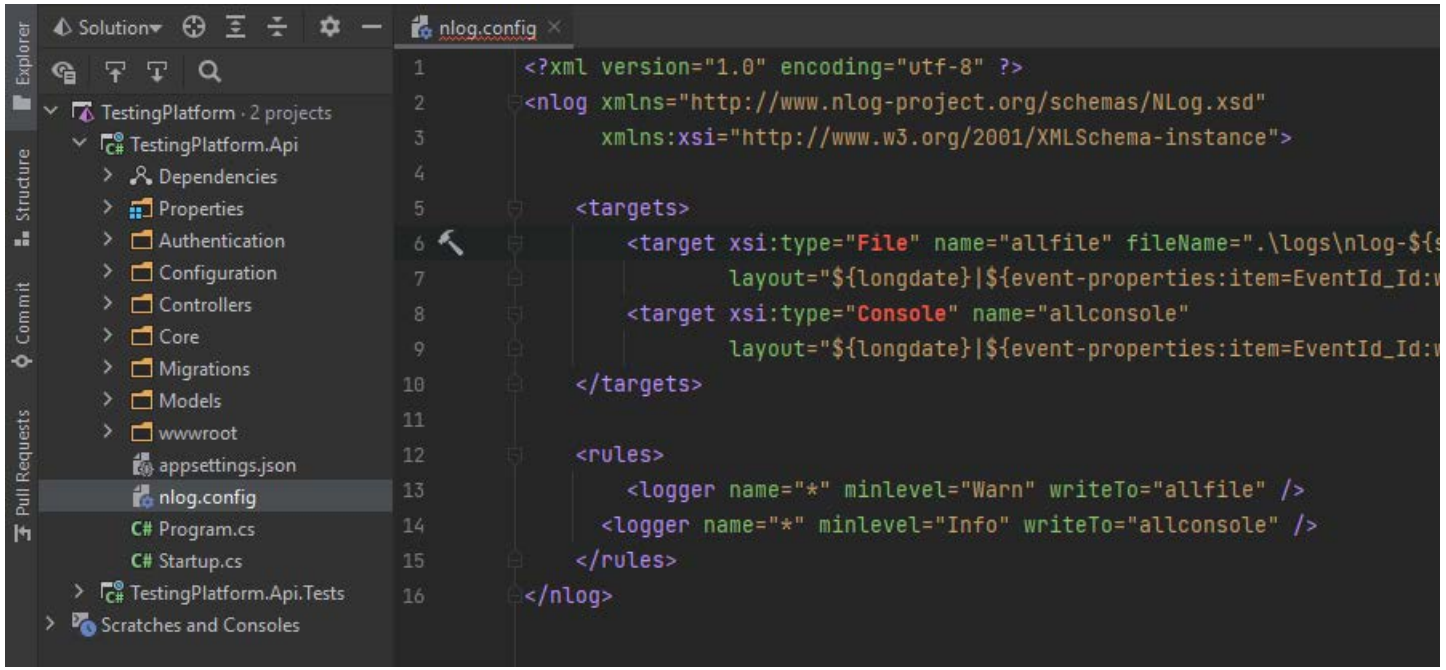


Логирование

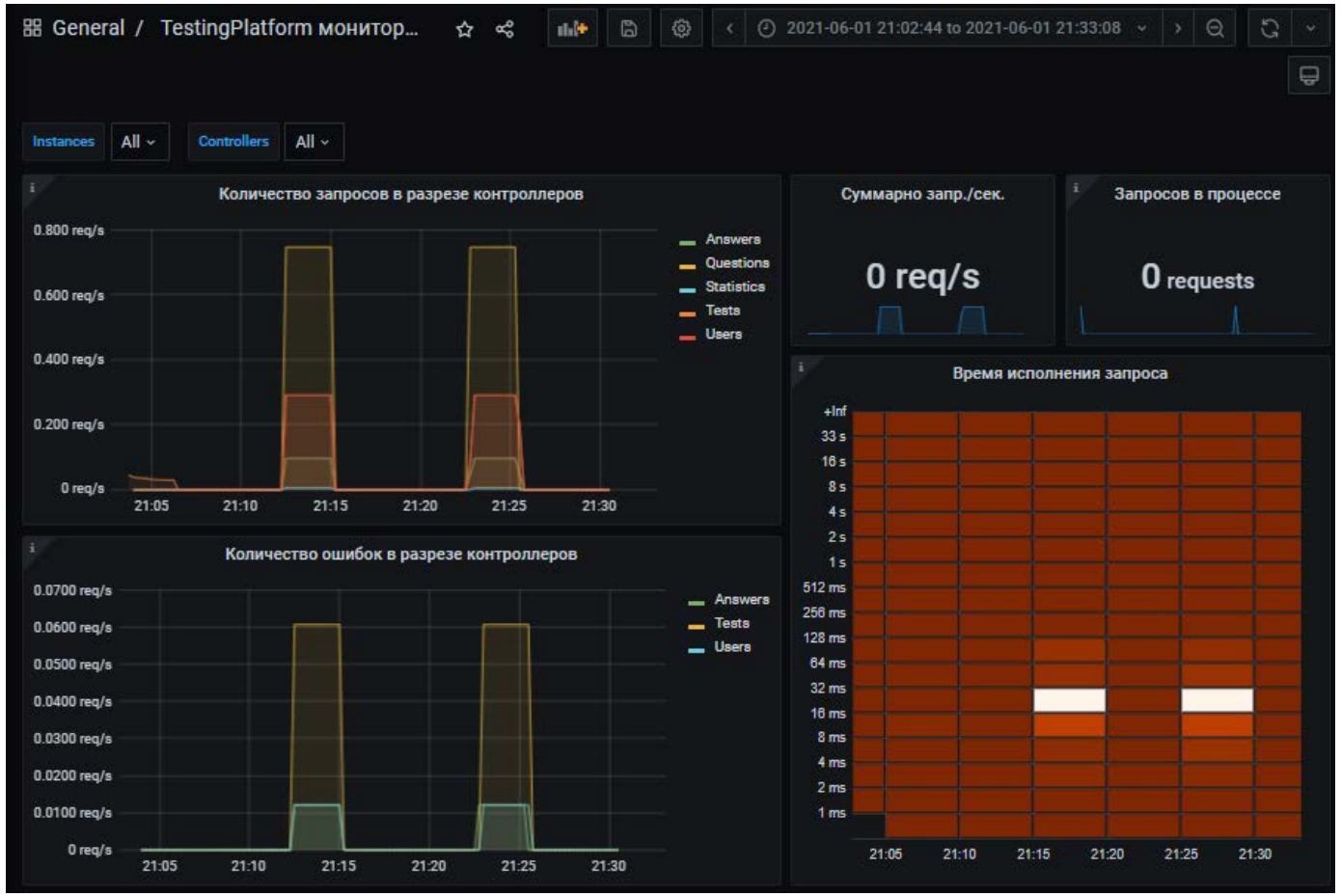
Для логирования используется библиотека NLog.

Логи выводятся в файл и в консоль.

Конфигурирование логирования находится в файле nlog.config в корне приложения.

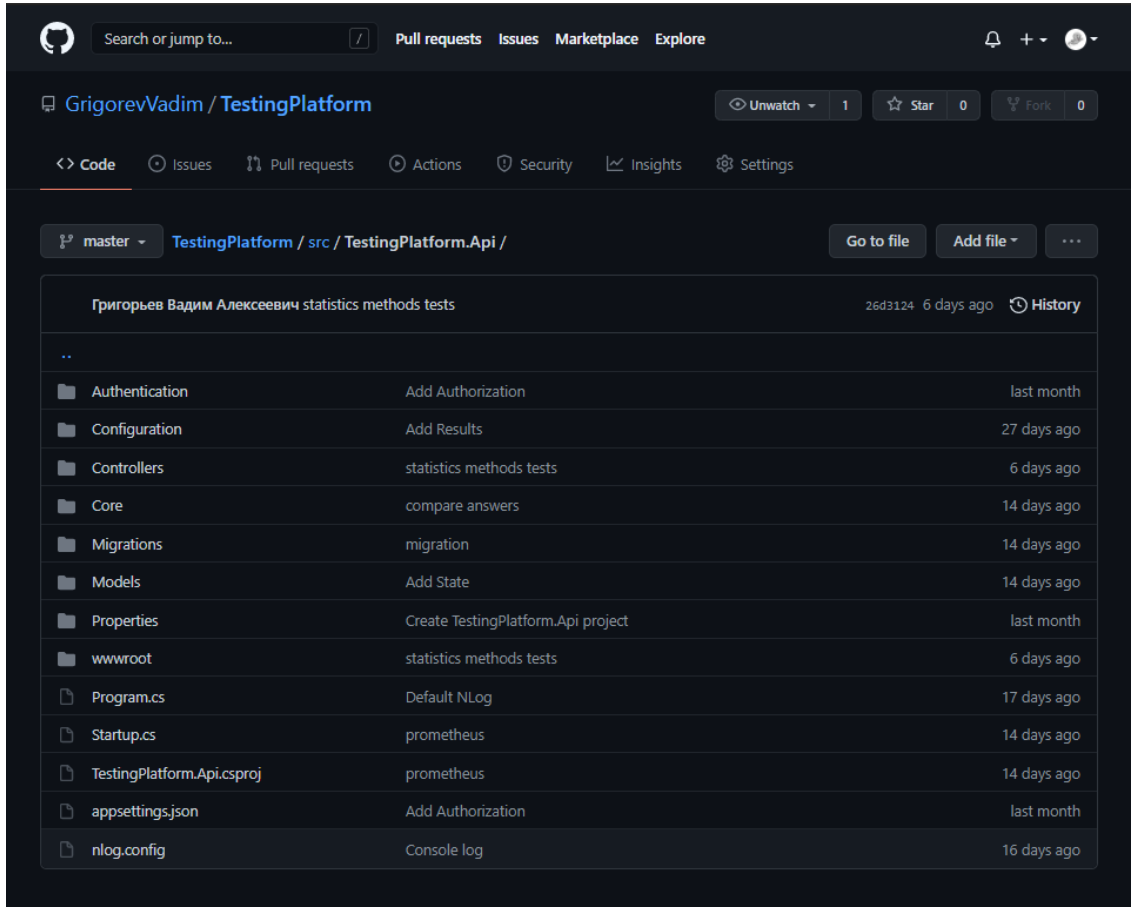


Мониторинг



Метрики собираются в Prometheus затем выводятся в Grafana

GitHub

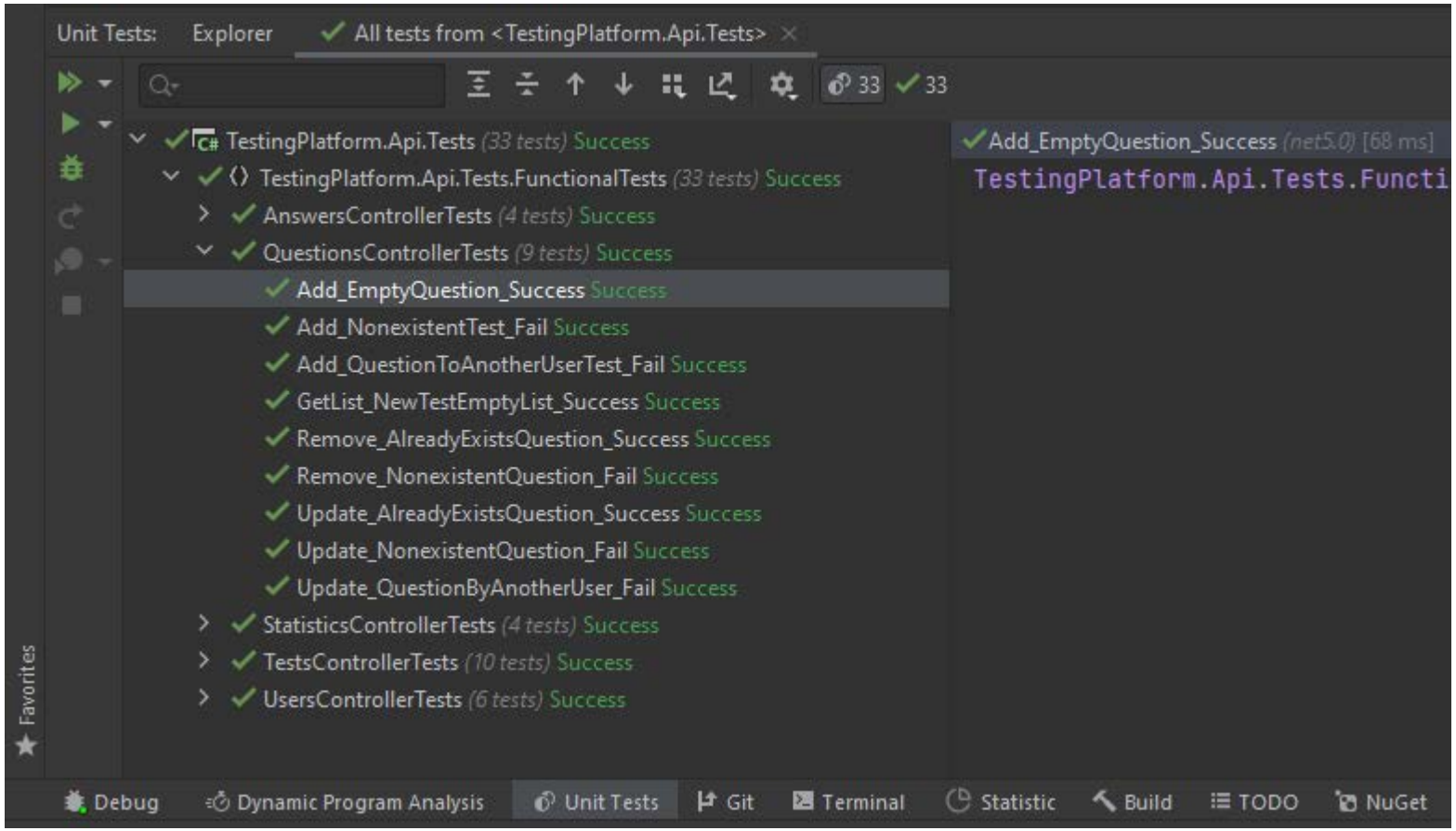


Исходный код приложения располагается по адресу https://github.com/GrigorevVadim/TestingPlatform.

Тестирование

На API приложение написано 33 функциональных теста.

Все тесты стабильно выполняются.



Заключение

Задачи:

1. Разработана структура базы данных для хранения данных.
2. Разработана backend часть приложения на ASP.NET Core WebApi.
3. Разработана frontend часть приложения в виде html страниц.
4. Проведено тестирование.

Цель ВКР:

Веб-приложение для создания и прохождения тестов подготовленно.

Спасибо за внимание!