

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО
Научный руководитель, доцент департамента
программной инженерии, кандидат
педагогических наук, доцент

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»

_____ С. А. Виденин
«__» _____ 2024 г.

_____ Н. А. Павлочев
«__» _____ 2024 г.

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

КофеСкаут

Программа и методика испытаний

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.05.06-01 51 02-1-ЛУ

Исполнитель
студент группы БПИ228
_____/ М. Р. Иванов /
«__» _____ 2024 г.

Москва 2024

УТВЕРЖДЕН
RU.17701729.05.06-01 51 02-1-ЛУ

КофеСкаут

Программа и методика испытаний

RU.17701729.05.06-01 51 02-1

Листов 13

Инв. № подл	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Москва 2024

СОДЕРЖАНИЕ

1. ОБЪЕКТ ИСПЫТАНИЙ	3
1.1. Наименование программы	3
1.2. Краткая характеристика области применения программы	3
2. ЦЕЛЬ ИСПЫТАНИЙ.....	4
3. ТРЕБОВАНИЯ К ПРОГРАММЕ	5
3.1. Требования к функциональным характеристикам	5
3.2. Требования к организации входных данных.....	6
3.3. Требования к организации выходных данных.....	6
3.4. Требования к надежности.....	6
3.4.1. Требования к обеспечению надежного функционирования программ	6
3.4.2. Время восстановления после отказа	6
3.4.3. Отказы из-за некорректных действий оператора	7
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ	8
4.1. Состав программной документации.....	8
4.2. Специальные требования к программной документации.....	8
5.1. Технические и программные средства используемые во время испытаний.....	9
5.2. Порядок проведения испытаний.....	9
6.1. Unit тестирование	10
6.2. Интеграционное тестирование	11
6.3. Тестирование надежности	12
6.4. Проверка программной документации	12
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ	13

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ОБЪЕКТ ИСПЫТАНИЙ**1.1. Наименование программы**

Наименование программы – «КофеСкаут».

Условное обозначение темы разработки – «CoffeeScout».

1.2. Краткая характеристика области применения программы

«КофеСкаут» - приложение для поиска интересующего напитка в кофейнях, расположенных в непосредственной близости от пользователя.

Основная область применения программы - личное использование. Также приложение предоставляет возможность кофейням продвигать свои продукты и рекламировать себя. Клиентам приложение предоставляет возможность выбора желаемого напитка из предложенных с использованием системы фильтров, а после предлагает ближайшие к пользователю кофейни, которые смогут приготовить выбранный напиток, а также предлагает сделать предзаказ. Более того, приложение указывает кратчайший путь до выбранной кофейни, что позволит пользователям сэкономить время как на выборе кофейни, так и на времени, чтобы до неё добраться.

В нынешних реалиях с большим количеством кофеен и ассортимента в них бывает трудно найти желаемый напиток поблизости, поэтому приложение «КофеСкаут» будет пользоваться спросом.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. ЦЕЛЬ ИСПЫТАНИЙ

Цель испытаний - проверить работу серверной части приложения в соответствии с представленными требованиями и выявить возможные проблемы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. ТРЕБОВАНИЯ К ПРОГРАММЕ

3.1. Требования к функциональным характеристикам

Backend приложения должен быть разработан в виде REST API поддерживающий следующие запросы:

- Авторизация пользователя в приложении с использованием OAuth.
- Отдельная регистрация для владельцев кофеен.
- Возможность поиска кофейни с учетом расстояния, ассортимента, цен.
 1. GET запрос возвращающий из базы данных список кофеен.
 2. GET запрос возвращающий из базы данных отсортированный по расстоянию список кофеен.
 3. GET запрос возвращающий из базы данных отсортированный по ценам список кофеен.
 4. GET запрос возвращающий из базы данных отфильтрованный по наличию кофейных напитков.
- Возможность поиска кофейного напитка. (GET запрос возвращающий список кофейных напитков семантически близких к искомому)
 1. GET запрос возвращающий из базы данных список кофейных напитков.
 2. GET запрос возвращающий из базы данных список любимых напитков для конкретного пользователя.
- Возможность оформления заказа и его получения кофейней. (POST запрос создающий заказ)
- Возможность отслеживания готовности заказа. (GET запрос возвращающий текущее состояние заказа)
- Возможность оставить отзыв на кофейню, напиток, напиток в конкретной кофейне. (POST запрос)
- Возможность для владельца кофейни добавить напиток в меню. (POST запрос)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

- Возможность для владельца кофейни обновить данные о напитке в меню. (PUT запрос)
- Возможность для владельца кофейни удалить напиток из меню. (DELETE запрос)
- Кофейный напиток должен иметь следующие атрибуты:
 - Название
 - Размер
 - Цена
- Кофейня должна иметь следующие атрибуты:
 - Название
 - Адрес
 - Ассортимент
 - Организация

3.2. Требования к организации входных данных

Входные данные представляют из себя либо json тела запроса, либо указанные в URL параметры запроса.

3.3. Требования к организации выходных данных

Выходные данные представляют из себя json тела и Http коды ответа на запрос.

3.4. Требования к надежности

3.4.1. Требования к обеспечению надежного функционирования программ

Пользователь с постоянным доступом к сети “Интернет” должен иметь возможность в любой момент пользоваться мобильным приложением. Бэкэнд приложения не должен завершаться аварийно ни при каких обстоятельствах.

3.4.2. Время восстановления после отказа

В случае непредвиденного прерывания работы бэкэнда приложения, он должен автоматически перезапускаться в течение 10 минут.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3.4.3. Отказы из-за некорректных действий оператора

Приложение не должно завершаться при любых некорректных действиях пользователя или администратора.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

4.1. Состав программной документации

- «КофеСкаут». Техническое задание (ГОСТ 19.201-78);
- «КофеСкаут». Программа и методика испытаний (ГОСТ 19.301-79);
- «КофеСкаут». Текст программы (ГОСТ 19.401-78);
- «КофеСкаут». Пояснительная записка (ГОСТ 19.404-79);
- «КофеСкаут». Руководство оператора (ГОСТ 19.505-79);

4.2. Специальные требования к программной документации

1. Все документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 [7] и ГОСТами к каждому виду документа (см. п. 5.1).
2. Вся документация и программа сдается в электронном виде в формате .PDF в одном архиве формата .ZIP.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

5.1. Технические и программные средства используемые во время испытаний

Технические средства:

- Apple Macbook Pro 14 на процессоре M1.

Программные средства:

- JetBrains Rider 2023.3.4
- .NET 8.0 SDK
- Safari
- Docker
- Postman
- .Env файл
- docker-compose.yml файл

5.2. Порядок проведения испытаний

- 1) Unit тестирование в соответствии с функциональными требованиями.
- 2) Интеграционное тестирование в соответствии с функциональными требованиями.
- 3) Тестирование надежности.
- 4) Проверка программной документации в соответствии с требованиями.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6. МЕТОДЫ ИСПЫТАНИЙ

6.1. Unit тестирование

На бизнес-логику приложения пишутся автоматические Unit тесты. Для этого создается отдельный проект CoffeeScoutBackend.UnitTests и выстраиваются следующие зависимости:

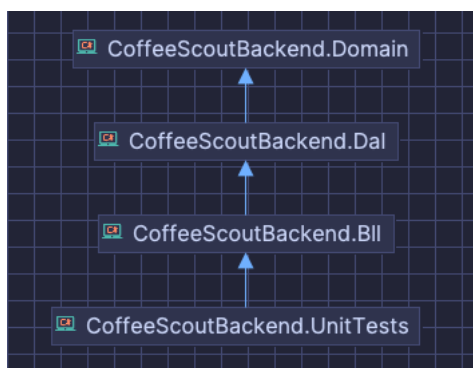


Рисунок 1 – Схема зависимостей проекта с Unit тестами.

Для написания тестов используется фреймворк XUnit и несколько сторонних библиотек: AutoBogus, FluentAssertions, Moq.

AutoBogus: Позволяет создавать экземпляры классов с автоматически сгенерированными данными.

FluentAssertions: Предоставляет набор методов расширения для проверки успешности выполнения тестов.

Moq: Предоставляет создавать моки внешних зависимостей. Моки - это объекты, которые используются вместо реальных объектов в тестах, чтобы изолировать тестируемый код от зависимостей и создать контролируемую среду исполнения для тестов.

Для написания изолированных тестов для тестов создаются моки для всех внешних зависимостей: репозиториев и внешних сервисов. Если под метод сервиса пишутся тесты, то они должны покрывать большинство позитивных и негативных сценариев. Все тесты написаны с использованием паттерна Arrange, Act, Assert, что упрощает их чтение и понимание.

Для обеспечения уверенности в надежности приложение покрытие тестами должно составлять не менее 60%. Код должен успешно проходить все написанные тесты. Текущее покрытие

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

тестами указывается на странице репозитория приложения и обновляется не реже чем раз в релиз. Код тестов также находится в репозитории: <https://github.com/Doremi203/CoffeeScout>

6.2. Интеграционное тестирование

Интеграционное тестирование производится для того, чтобы проверить взаимодействие компонентов системы между собой. В рамках проекта проверяется взаимодействие с базой данных и API. Для написания тестов создается отдельный проект CoffeeScoutBackend.IntegrationTests и выстраиваются следующие зависимости:

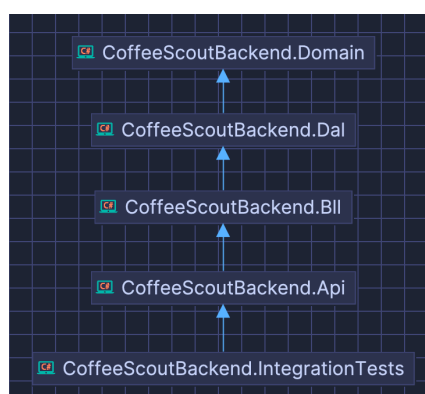


Рисунок 2 – Схема зависимостей проекта с интеграционными тестами.

Для написания интеграционных тестов также используется фреймворк XUnit. Так как требуется тестирование Api, нужно разворачивать хост приложения внутри тестовой среды, для этого реализована специальная инфраструктура: MyCustomWebApplicationFactory, HttpIntegrationTestCollection.

Для тестирования также используются некоторые сторонние библиотеки: AutoBogus, FluentAssertions, Moq, Microsoft.AspNetCore.Mvc.Testing.

Microsoft.AspNetCore.Mvc.Testing: Предоставляет инфраструктуру для тестирования контролеров.

Интеграционные тесты пишутся на каждый endpoint с использованием паттерна Arrange, Act, Assert, что упрощает их чтение и понимание. Проверяются все возможные сценарии в соответствии с возможными кодами возврата endpoint'а описанными в Swagger. Это позволяет уменьшить риск возникновения проблем при интеграции backend'а и frontend'а приложения. Все написанные тесты должны проходить успешно. Код тестов находится в репозитории приложения: <https://github.com/Doremi203/CoffeeScout>

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6.3. Тестирование надежности

Требования к обеспечению надежного функционирования программ проверяются за счет интеграционного тестирования.

Время восстановления после отказа проверяется путем вызова намеренного аварийного завершения программы во время её исполнения внутри Docker контейнера. Результат восстановления проверяется визуально путем просмотра статуса контейнера.

Отказы из-за некорректных действий оператора проверяются за счет end to end тестирования, описанного в рамках общей программы и методики испытаний.

6.4. Проверка программной документации

Программная документация проверяется визуально.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.05.06-01 51 02-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible]