

Отчет тестирования функциональностей MVP

Функциональное тестирование проводилось для:

- 1) Экран авторизации: экран регистрации, экран входа и экран восстановления пароля. 7 тест кейсов из 7 прошли успешно
- 2) Главный экран: проверяется корректность отображения графика веса и активности, перенаправление на другие экраны. 6 тест кейсов из 6 прошли успешно
- 3) Блок А (экран тренировок): проверяется возможность начать тренировку, а также создание и редактирование тренировок с аккаунта админа. 9 тест кейсов из 9 прошли успешно
- 4) Личный кабинет: проверяется возможность редактирования пароля и почты, а также выход из аккаунта. 7 тест кейсов из 7 прошли успешно

Юнит тесты

Юнит-тесты проверяют отдельные компоненты приложения в изоляции, используя моки для замены внешних зависимостей. Эти тесты быстрые и не требуют подключения к базе данных или другим внешним сервисам.

В нашем проекте интеграционные тесты реализованы через моки, эмулирующие поведение внешних зависимостей. Это позволяет проверять взаимодействие компонентов без реальных внешних сервисов.

1 Unit-тесты для сервиса Auth_service

1.1 Test_auth_service

Тест	Что проверяет
test_hash_password	Проверяет корректность хеширования пароля

test_normalize_contact	Проверяет нормализацию email (удаление пробелов и приведение к нижнему регистру)
test_generate_verification_code	Проверяет генерацию кода верификации (должен возвращать строку длиной 6 символов)
test_register_user_success	Проверяет успешную регистрацию пользователя с корректными данными
test_register_user_existing_email	Проверяет обработку попытки регистрации с уже существующим email
test_register_user_invalid_email	Проверяет обработку некорректного email (содержащего пробелы)
test_verify_user_success	Проверяет успешную верификацию пользователя по коду

1.2 Test_router

Тест	Что проверяет
test_register_user_failure	Проверяет обработку ошибки при регистрации (дубликат email)
test_verify_code_success	Проверяет успешную верификацию пользователя по коду
test_verify_code_failure	Проверяет обработку неверного кода верификации
test_login_user_success	Проверяет успешный вход пользователя с верными учетными данными

test_login_user_failure	Проверяет обработку неверных учетных данных при входе
test_refresh_token_success	Проверяет успешное обновление токена доступа
test_refresh_token_blacklisted	Проверяет обработку попытки обновления отозванного токена

1.3 Test_token_service

Тест	Что проверяет
test_create_token_payload	Проверяет создание payload для токена с правильными полями и временными метками
test_create_access_token	Проверяет создание access токена с дефолтным временем жизни
test_create_access_token_with_custom_expiry	Проверяет создание access токена с кастомным временем жизни
test_create_refresh_token	Проверяет создание refresh токена с правильными параметрами
test_create_reset_password_token	Проверяет создание токена для сброса пароля
test_decode_jwt_valid	Проверяет декодирование валидного JWT токена
test_decode_jwt_expired	Проверяет обработку истекшего токена

2 Unit-тесты для сервиса Workout_service

2.1 test_exercises_service

Тест	Что проверяет
test_get_all_exercises	проверяет получение всех упражнений
test_get_exercise_by_id	проверяет получение упражнения по ID
test_create_exercise	проверяет создание нового упражнения
test_update_exercise	проверяет обновление существующего упражнения
test_delete_exercise	проверяет удаление упражнения

2.2 test_training_service

Тест	Что проверяет
test_get_trainings	проверяет получение списка всех тренировок
test_get_training_by_id	проверяет получение тренировки по ID

2.3 test_api

Тест	Что проверяет
test_read_root	проверяет ответ от корневого маршрута API
test_get_exercises_unauthorized	проверяет поведение API при неавторизованном доступе

test_get_exercises_authorized	проверяет получение списка упражнений авторизованным пользователем
test_get_exercise_by_id_authorized	проверяет получение упражнения по ID авторизованным пользователем
test_get_muscle_groups_authorized	проверяет получение списка групп мышц авторизованным пользователем

2.4 test_api_simple

Тест	Что проверяет
test_get_exercises	проверяет метод получения всех упражнений
test_get_exercise_by_id	проверяет метод получения упражнения по ID
test_get_muscle_groups	проверяет метод получения всех групп мышц

2.5 test_database

Тест	Что проверяет
test_database_connection	проверяет подключение к базе данных и выполнение простого запроса SELECT 1
test_get_exercises_table_exists	проверяет существование таблицы exercises
test_get_muscle_groups_table_exists	проверяет существование таблицы muscle_groups

test_fetch_exercises	проверяет получение всех упражнений из базы данных
test_fetch_exercise_by_id	проверяет получение упражнения по ID

Метрики

Всего на данном этапе было написано 83 теста.

Для реализованного функционала и экранов было выполнено 70 тестов.

13 тестов затрагивают нереализованный функционал.

Найденные баги

Нашли и пофиксили

Критичные баги

- 1) Не работал график из сервиса activity_service из-за ошибки типа данных
- 2) Удаление гифок не работало из-за проблем с CORS

Нашли и работаем над исправлением

Не критичные баги

- 1) Если в тренировке два одинаковых упражнения подряд идут, то кол-во повторений не считается
- 2) При заполнении упражнений возникает ошибка при задании названия более 25 символов, пока нет валидации