Отчёт о Unit-тестировании проекта TechTrek Web ко второй рубежной аттестации 06.05.2025

В ходе модульного тестирования проекта TechTrek Web была проведена проверка ключевых компонентов бизнес-логики, включая аутентификацию пользователей, управление игровым процессом и постраничную загрузку сущностей. Тесты были реализованы с использованием фреймворка JUnit 5 и библиотеки Mockito для создания подмен (моков) зависимостей.

Тесты направлены на обеспечение надёжной работы следующих подсистем:

- Аутентификация и регистрация пользователей: проверена регистрация новых пользователей, обработка повторяющихся имён и e-mail, а также генерация и обновление JWT-токенов.

- Игровая логика: протестирован запуск новой игры, создание всех необходимых сущностей (ресурсы, команды, ходы), а также обработка исключительных ситуаций (отсутствие пользователя или миссии).

- Работа со справочниками (сферами): проверена корректная пагинация и обновление списка сфер на клиенте в зависимости от смещения (offset).

Тесты покрывают как позитивные сценарии (успешные операции), так и негативные (ошибки при неверных данных). Это позволило выявить и устранить ряд возможных исключений и недочетов на раннем этапе разработки.

На данный момент реализованы следующие unit-тесты:

Таблица – AuthServiceTest (тесты авторизации и регистрации пользователей)

№	Название теста	Что проверяет	Как проверяет (механизм)
1	testRegisterNewUser_Success	Успешная регистрация пользователя: сохранение, генерация токенов	Проверяет сохранение пользователя и refresh-токена, токены – через assert и verify
2	testRegisterNewUser_UsernameExists_ThrowsException	Обработка ошибки при существующем username	Проверяет выброс UserAlreadyExistsException, сохранение не вызывается (verify(..., never()))
3	testRefreshAccessToken_Success	Обновление access-токена по валидному refresh-токену	Проверяет генерацию нового access-токена, без удаления refresh-токена
4	testRefreshAccessToken_InvalidToken_Throws	Ошибка при несуществующем refresh-токене	Проверяет выброс RuntimeException, если токен не найден в репозитории
5	testAuthenticate_Success	Успешная аутентификация пользователя: генерация access/refresh токенов	Проверяет генерацию токенов, извлечение пользователя, сохранение refresh-токена

Таблица – GameServiceTest (тестирование запуска игры)

№	Название теста	Что проверяет	Как проверяет (механизм)
1	startGame_happyPath_returnsFilledDTO	Успешный запуск игры: создаются сущности, возвращается корректный DTO	Проверяет поля DTO (missionId, companyName, turnNumber, money) через assertThat() и сохранение сущностей через verify()
2	startGame_userNotFound_throws	Обработка ошибки при отсутствии пользователя по username	Проверяет выброс EntityNotFoundException, сообщение содержит "User not found"
3	startGame_missionNotFound_throws	Обработка ошибки при отсутствии миссии по ID	Проверяет выброс IllegalArgumentException, сообщение содержит "Mission not found"

Эти тесты покрывают основной сценарий и граничные случаи метода startGame(...) в GameService, гарантируя корректную инициализацию игры и корректную обработку ошибок.

Таблица – SphereServiceTest (тестирование работы со сферами)

№	Название теста	Что проверяет	Как проверяет (механизм)
1	getSpheres_shouldReturnPagedSpheres	Корректное получение первой страницы сфер заданного размера	Мокается метод findAll(pageable), проверяется размер возвращённого списка (3) и вызов репозитория
2	updateSpheres_shouldReturnNextSetOfSpheres	Получение следующей "порции" сфер с учётом offset	Вычисляется страница как offset / size, проверяется размер списка (2) и вызов findAll()

Эти тесты фокусируются на пагинации данных из репозитория сфер, обеспечивая корректное поведение методов getSpheres и updateSpheres при разных параметрах.