

3 Тестирование

3.1 Введение

Цель документа: Подтвердить, что программный продукт «MedBook — Система записи к врачу» реализован корректно и соответствует требованиям ТЗ.

Объект тестирования: Веб-приложение для онлайн-записи к врачам, включая backend (Node.js/Express), frontend (HTML/CSS/JS) и базу данных SQLite.

3.2 Область тестирования

Функциональные возможности:

- регистрация и авторизация пользователей;
- просмотр и поиск врачей по специализациям;
- создание и управление записями на приём;
- админ-панель: управление пользователями, врачами и расписанием;
- безопасность: JWT-токены, хэширование паролей.

Нефункциональные требования:

- время отклика API ≤ 200 мс при 50 одновременных запросах;
- покрытие кода тестами $\geq 70\%$;
- стабильность работы при стандартной нагрузке.

3.3 Окружение тестирования

- сервер: Node.js / Express.js, локально (<http://localhost:3000/>);
- база данных: SQLite (database.db);
- клиент: Chrome, Firefox;
- язык и фреймворки: JavaScript (ES6+), Node.js, Express.js;

- инструменты тестирования: Jest (unit/functional), Supertest (API integration), npm scripts;
- CI/CD (опционально): GitHub Actions для запуска тестов при каждом PR.

3.4 Методы и критерии тестирования

Уровни тестирования:

- Unit (модульное) — проверка отдельных функций и модулей (Jest);
- Integration (интеграционное) — взаимодействие API и базы данных (Supertest);
- System / Acceptance (системное/приёмочное) — тестирование всей системы через UI/API;
- Regression (регрессионное) — автоматические тесты при CI.

Методы тест-дизайна:

- эквивалентное разбиение (Equivalence Partitioning);
- анализ граничных значений (Boundary Value Analysis);
- табличное тестирование (Decision Tables);
- тестирование состояний (State Transition).

Критерии успешности:

- все критические тесты пройдены;
- покрытие кода $\geq 70\%$;
- отсутствие критических дефектов (высокий приоритет).

3.5 План тестов и тест-кейсы

Примеры тест-кейсов

Таблица 1

ID	Название	Тип	Предусловия	Шаги
----	----------	-----	-------------	------

ТС-001	Регистрация нового пользователя	функциональный / black-box	Нет аккаунта	1) Открыть /register 2) Заполнить форму 3) Отправить
ТС-002	Авторизация	функциональный / black-box	Пользователь зарегистрирован	1) Открыть /login 2) Ввести логин/пароль 3) Войти
ТС-003	Просмотр списка врачей	функциональный / black-box	Пользователь вошёл	1) Открыть /doctors
ТС-004	Создание записи на приём	функциональный / black-box	Пользователь вошёл	1) Выбрать врача и время 2) Подтвердить запись
ТС-005	Время отклика API	нефункциональный / performance	API запущено	Отправить 50 параллельных запросов

Автоматические тесты: находятся в папке tests/, запускаются командой:
npm test

