МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

Московский государственный университет геодезии и картографии (МИИГАиК)

# ОТЧЕТ ПО РЕЗУЛЬТАТАМ ТЕСТИРОВАНИЯ

Веб-приложение «Online chat – Онлайн чат»

Дата проведения тестирования:

12.07.2025

Дата составления отчета:

12.07.2025

Выполнила: Ширяева Ю.А.

г. Москва 2025

## Содержание

## Введение

**Цель тестирования:** Комплексная проверка функциональности чат-приложения, включая ручное тестирование пользовательских сценариев и автоматизированную проверку компонентов системы.

**Методы тестирования:**

1. Ручное тестирование пользовательских сценариев
2. Автоматизированное unit-тестирование ключевых компонентов
3. Интеграционное тестирование API

## Часть 1: Ручное тестирование пользовательских сценариев

*Сценарий "Регистрация и вход в систему"*

**Предусловия:**

1. Сервер запущен на localhost:8080
2. Клиентское приложение открыто в браузере

**Шаги:**

1. Переход на страницу регистрации
2. Ввод логина "test_user" и пароля "test_pass123"
3. Подтверждение регистрации
4. Ввод тех же данных на странице входа
5. Проверка получения JWT-токена

**Ожидаемый результат:**

- Успешная регистрация с кодом 201
- Успешная аутентификация с кодом 200
- Получение валидного JWT-токена\

*Сценарий "Отправка и получение сообщений"*

**Предусловия:**

1. Пользователь авторизован
2. Имеется валидный JWT-токен

**Шаги:**

1. Отправка сообщения "Привет!" через API
2. Запрос истории сообщений
3. Проверка наличия отправленного сообщения

**Ожидаемый результат:**

- Сообщение сохраняется с кодом 200
- История содержит отправленное сообщение
- Сообщение имеет правильные атрибуты (автор, время, текст)

## Часть 2: Автоматизированное тестирование

1. *Тестирование модуля базы данных (database_test.cpp)*

**Назначение:** Проверка корректности работы с SQLite базой данных.

**Тестируемые функции:**

- Инициализация структуры БД
- Операции CRUD для пользователей
- Хранение и извлечение сообщений

**Ключевые проверки:**

```
void test_database() {
    Database db(":memory:");
    assert(db.initialize());

    // Проверка создания пользователя
    assert(db.create_user("user1", "hash1"));
    User u = db.get_user_by_login("user1");
    assert(u.id != 0); // ID должен быть присвоен
    assert(u.login == "user1"); // Корректность сохранения логина
    assert(u.password_hash == "hash1"); // Корректность сохранения хеша

    // Проверка работы с сообщениями
    assert(db.add_message(u.id, "Hello"));
    auto messages = db.get_recent_messages();
    assert(messages.size() == 1); // Сообщение должно добавиться
    assert(messages[0].text == "Hello"); // Текст должен сохраниться
}
```

2. *Тестирование сервиса аутентификации (auth_test.cpp)*

**Назначение:** Проверка безопасности и корректности работы механизмов аутентификации.

**Тестируемые функции:**

- Хеширование паролей
- Генерация и валидация JWT токенов
- Регистрация и аутентификация пользователей

**Ключевые проверки:**

```
void test_auth_service() {
    Database db(":memory:");
    assert(db.initialize());
    AuthService auth(db, "test_secret");

    // Проверка регистрации
    assert(auth.register_user("test_user", "test_pass"));

    // Проверка аутентификации
    assert(auth.authenticate("test_user", "test_pass")); // Верный пароль
    assert(!auth.authenticate("test_user", "wrong_pass")); // Неверный пароль
```

```
    // Проверка JWT токенов
    std::string token = auth.generate_token("test_user");
    assert(auth.validate_token(token)); // Валидный токен
    assert(auth.get_login_from_token(token) == "test_user"); // Извлечение данных
}
```

## Часть 3: Интеграционное тестирование API

Сценарий "Полный цикл работы"

1. Автоматическая регистрация тестового пользователя
2. Аутентификация и получение токена
3. Отправка тестового сообщения
4. Проверка сохранения сообщения
5. Запрос истории сообщений
6. Валидация данных в ответе

**Проверяемые аспекты:**

- Согласованность работы компонентов
- Корректность HTTP-кодов ответов
- Соответствие форматов данных спецификации
- Обработка ошибочных запросов

Результаты тестирования

| **Тип тестирования** | **Область проверки** | **Результат** |
|---|---|---|
| Ручное тестирование | Регистрация пользователя | Успешно |
| | Вход в систему | Успешно |
| | Отправка сообщений | Успешно |
| Unit-тесты | Работа с базой данных | Успешно |
| | Хеширование паролей | Успешно |
| | JWT токены | Успешно |
| Интеграционные тесты | Полный цикл работы | Успешно |

## Выводы

1. Основная функциональность приложения работает корректно
2. Автоматизированные тесты покрывают ключевые компоненты системы
3. Пользовательские сценарии не выявили критических проблем

Приложение готово к развертыванию в тестовой среде.