МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
**«Волгоградский государственный университет»**

Институт экономики и управления
Кафедра экономики и управления

ОБРАЗЕЦ

# ПРАКТИЧЕСКАЯ РАБОТА

на тему:

## Разработка и тестирование модуля учёта заявок для службы технической поддержки

Выполнил(а): студент(ка) направления 38.03.01 «Экономика»
Иванова Анна Сергеевна, группа ДЕМО-01

Руководитель: кандидат экономических наук, доцент Петров Игорь Владимирович

Волгоград · 2026

Демонстрационный образец. Данные условны. Не является выполненной работой конкретного студента.

# АННОТАЦИЯ

Практическая работа посвящена разработке модуля учёта заявок для службы технической поддержки. Описаны требования, выбор инструментов, структура данных, реализация основных функций и тестирование. Приведены таблицы требований, сравнения платформ, тест-кейсов и приложение с матрицей прав доступа. Результатом является прототип, пригодный для опытной эксплуатации и дальнейшего развития.

**Ключевые слова:** учёт заявок, техническая поддержка, REST API, тестирование, прототип, требования.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4

1 Постановка задачи и требования 5

2 Выбор инструментов реализации 5

3 Проектирование структуры данных 6

4 Реализация основных функций 7

5 Тестирование модуля 7

6 Анализ результатов и ограничения 8

ЗАКЛЮЧЕНИЕ 9

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 10

ПРИЛОЖЕНИЕ А 11

# ВВЕДЕНИЕ

Практическая работа направлена на создание модуля учёта заявок для службы технической поддержки. Пользователи должны регистрировать обращения, указывать категорию и приоритет, а специалисты — получать назначенные заявки и фиксировать статус выполнения. Модуль рассматривается как часть внутренней информационной системы предприятия. Границы работы включают проектирование, реализацию и тестирование основных функций без интеграции с внешними сервисами.

Цель работы — разработать и проверить прототип модуля учёта заявок. Для достижения цели решаются задачи: сформулировать требования; выбрать инструменты; спроектировать структуру данных; реализовать основные функции; провести тестирование. Результаты позволяют оценить трудоёмкость и ограничения прототипа.

**Цель работы** — Разработать прототип модуля учёта заявок и проверить его основные функции тестированием.

**Задачи работы:**

1. Сформулировать функциональные требования к модулю.
2. Выбрать платформу и инструменты реализации.
3. Спроектировать структуру данных и методы API.
4. Реализовать и протестировать основные функции.

**Объект исследования** — Процесс обработки заявок в службе технической поддержки.

**Предмет исследования** — Методы и средства разработки модуля учёта заявок.

**Методы исследования:** Анализ требований, Проектирование, Прототипирование, Тестирование.

**Практическая значимость.** Результаты могут использоваться для оценки трудоёмкости разработки и обоснования выбора платформы.

## 1 Постановка задачи и требования

Практическая работа направлена на создание модуля учёта заявок для службы технической поддержки. Пользователи должны регистрировать обращения, указывать категорию и приоритет, а специалисты — получать назначенные заявки и фиксировать статус выполнения. Модуль рассматривается как часть внутренней информационной системы предприятия. Границы работы включают проектирование, реализацию и тестирование основных функций без интеграции с внешними сервисами.

На основе анализа типовых процессов поддержки сформулированы пять функциональных требований. Они охватывают регистрацию заявки, назначение исполнителя, историю статусов, поиск и отчётность. Приоритеты требований определены по влиянию на скорость обработки обращений. Требования зафиксированы в таблице и используются как основа для проектирования.

Таблица 1 – Функциональные требования к модулю

| Код | Требование | Приоритет |
|---|---|---|
| F-01 | Регистрация заявки с указанием категории и приоритета | высокий |
| F-02 | Назначение исполнителя и контроль сроков | высокий |
| F-03 | История изменений статусов | средний |
| F-04 | Поиск и фильтрация заявок | средний |
| F-05 | Формирование отчёта по нагрузке | низкий |

*Источник: Составлено автором.*

## 2 Выбор инструментов реализации

Для реализации рассматривались три платформы, различающиеся скоростью разработки, стоимостью лицензий и требованиями к серверу. Вариант А не требует лицензионных платежей и показывает высокую скорость разработки. Вариант Б требует затрат 45 000 руб., вариант В — 120 000 руб. и более высоких требований к инфраструктуре. Выбор сделан в пользу варианта А.

Сравнение выполнено по пяти критериям, значимым для учебного проекта. Учитывались скорость разработки, стоимость лицензий, поддержка REST API, требования к серверу и простота сопровождения. Вариант А обеспечивает достаточную функциональность при минимальных затратах.

Ограничением является меньшая готовность к масштабированию, которая для данного объёма заявок не критична.

Таблица 2 – Сравнение платформ

| Критерий | Вариант А | Вариант Б | Вариант В |
|---|---|---|---|
| Скорость разработки | высокая | средняя | низкая |
| Стоимость лицензий | 0 руб. | 45 000 руб. | 120 000 руб. |
| Поддержка REST API | да | да | да |
| Требования к серверу | низкие | средние | высокие |
| Итоговый выбор | выбран | — | — |

*Источник: Составлено автором.*

## 3 Проектирование структуры данных

Структура данных включает сущности заявки, пользователя, категории и комментария. Заявка связана с исполнителем и категорией, а комментарии хранятся в истории обработки. Такая модель позволяет фиксировать статусы и формировать отчёты без дублирования информации. Основные сущности приведены в таблице. Ключевые поля обеспечивают однозначную идентификацию записей.

При проектировании использован принцип нормализации: справочник категорий вынесен отдельно, а комментарии связаны с заявкой. Для статусов предусмотрен ограниченный набор значений: «Новая», «В работе», «Закрыта». Это упрощает контроль и отчётность. Схема данных достаточна для реализации требований и может быть расширена без переработки основы.

Таблица 3 – Основные сущности модуля

| Сущность | Ключевые поля | Связь |
|---|---|---|
| Заявка | id, тема, статус, приоритет | исполнитель, категория |
| Пользователь | id, ФИО, роль | заявки |
| Категория | id, название | заявки |
| Комментарий | id, текст, дата | заявка |

*Источник: Составлено автором.*

## 4 Реализация основных функций

Реализация выполнена в виде набора методов REST API. Создание заявки выполняется методом POST /tickets, назначение исполнителя — PATCH /tickets/{id}, история статусов доступна через GET /tickets/{id}/history. Отчёт по нагрузке формируется методом GET /reports/load. Все методы возвращают JSON и проверяют корректность входных данных.

Основные функции реализованы и проверены на тестовых данных. Создание заявки требует заполнения темы и категории, назначение исполнителя меняет статус на «В работе», закрытие требует комментария. Отчёт формируется в формате CSV. Реализованные функции сведены в таблицу. Для ошибок предусмотрены коды ответов и понятные сообщения.

Таблица 4 – Реализованные функции

| Функция | Метод | Статус |
|---|---|---|
| Создание заявки | POST /tickets | готово |
| Назначение исполнителя | PATCH /tickets/{id} | готово |
| История статусов | GET /tickets/{id}/history | готово |
| Отчёт по нагрузке | GET /reports/load | готово |

*Источник: Составлено автором.*

## 5 Тестирование модуля

Тестирование проводилось по заранее подготовленным тест-кейсам, охватывающим создание, назначение, закрытие заявки, фильтрацию и отчёт. Для каждого тест-кейса определены ожидаемый и фактический результаты. Все пять тест-кейсов пройдены успешно. Результаты приведены в таблице. Проверялись как корректные, так и ошибочные сценарии.

Отдельное внимание уделено валидации входных данных и обработке ошибок. При создании заявки без темы возвращается ошибка, при закрытии без комментария система запрашивает комментарий. Фильтр по приоритету корректно ограничивает список. Отчёт за период формируется в CSV и открывается без дополнительной обработки. По результатам тестирования критических дефектов не выявлено.

Таблица 5 – Результаты тест-кейсов

| Тест-кейс | Ожидаемый результат | Фактический результат | Статус |
|---|---|---|---|
| Создание заявки без темы | Ошибка валидации | Ошибка валидации | пройден |
| Назначение исполнителя | Статус меняется на «В работе» | Статус изменён | пройден |
| Закрытие заявки | Требуется комментарий | Комментарий запрошен | пройден |
| Фильтр по приоритету | Список отфильтрован | Список отфильтрован | пройден |
| Отчёт за период | Файл CSV | Файл CSV | пройден |

*Источник: Составлено автором.*

## 6 Анализ результатов и ограничения

Анализ результатов показывает, что модуль выполняет заявленные функции и пригоден для опытной эксплуатации. Среднее время выполнения тест-кейсов невелико, наибольшее время требуется для формирования отчёта. Ограничением является отсутствие интеграции с почтой и внешними системами. Для развития предусмотрены уведомления и расширенная аналитика.

На рисунке приведено распределение времени выполнения тест-кейсов. Наибольшее время — 15 минут — занимает проверка отчёта, наименьшее — 6 минут — фильтрация. Полученные значения достаточны для учебного прототипа. Для промышленного внедрения требуется нагрузочное тестирование и проверка безопасности. Рекомендуется добавить журналирование.

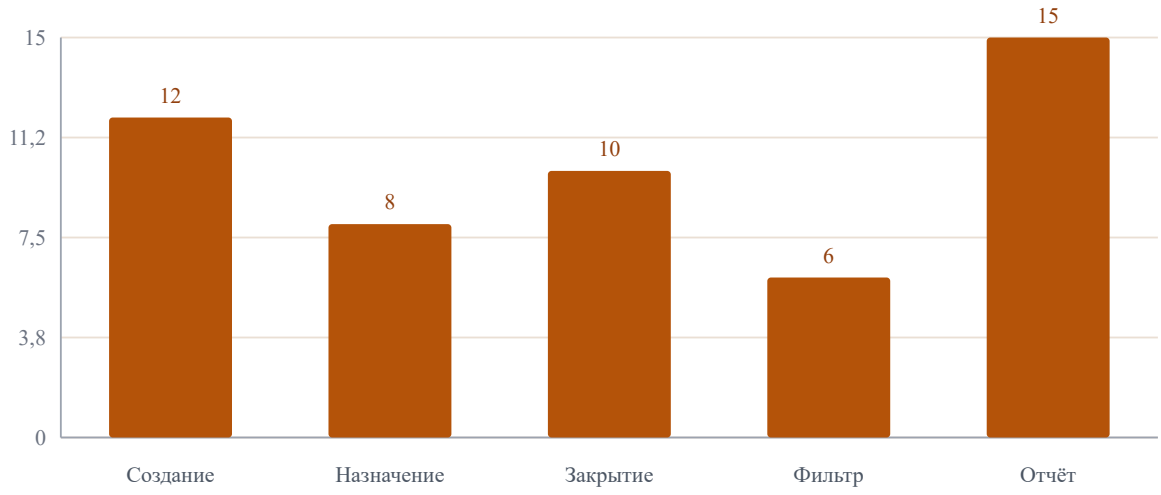



Рисунок 1 – Распределение времени выполнения тест-кейсов, мин

*Источник: Составлено автором.*

# ЗАКЛЮЧЕНИЕ

В ходе практической работы разработан прототип модуля учёта заявок. Реализованы регистрация, назначение исполнителя, история статусов, фильтрация и отчёт. Тестирование подтвердило работоспособность основных функций: все пять тест-кейсов пройдены.

Ограничениями являются отсутствие интеграции с внешними сервисами и необходимость нагрузочного тестирования. Для развития предусмотрены уведомления, расширенная аналитика и журналирование. Результаты могут использоваться для оценки трудоёмкости дальнейшей разработки.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Гома Х. UML. Проектирование систем реального времени, параллельных и распределённых приложений. — Москва: ДМК Пресс, 2020. — 704 с.
2. Фаулер М. Архитектура корпоративных программных приложений. — Москва: Вильямс, 2019. — 544 с.
3. Соммервилл И. Инженерия программного обеспечения. — Москва: Вильямс, 2020. — 624 с.
4. Куликов С. С. Тестирование программного обеспечения. — Москва: БХВ-Петербург, 2021. — 320 с.
5. Савин Р. Тестирование Дот Ком. — Москва: Дело, 2018. — 312 с.
6. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. — Москва: Стандартинформ, 2009. — 6 с.
7. ГОСТ Р 54869-2011. Проектный менеджмент. Требования к управлению проектом. — Москва: Стандартинформ, 2011. — 12 с.
8. ГОСТ Р ИСО/МЭК 25010-2015. Информационные технологии. Системы и программное обеспечение. Требования и оценка качества. — Москва: Стандартинформ, 2016. — 32 с.
9. Документация Spring Framework. — URL: https://spring.io/projects/spring-framework (дата обращения: 12.03.2025).
10. Документация PostgreSQL. — URL: https://www.postgresql.org/docs (дата обращения: 12.03.2025).
11. Документация Postman. — URL: https://learning.postman.com (дата обращения: 12.03.2025).
12. Хабр. Тестирование API. — URL: https://habr.com (дата обращения: 12.03.2025).
13. Блог Microsoft Developer. — URL: https://devblogs.microsoft.com (дата обращения: 12.03.2025).
14. Справочная система «КонсультантПлюс». — URL: https://www.consultant.ru (дата обращения: 12.03.2025).
15. Официальный сайт GitHub. — URL: https://github.com (дата обращения: 12.03.2025).

# ПРИЛОЖЕНИЕ А

Матрица ролей и прав доступа

В приложении приведена матрица прав, использованная при тестировании модуля.

Таблица А.1 – Матрица ролей и прав доступа

| Роль | Создание заявки | Назначение | Отчёт |
|---|---|---|---|
| Пользователь | да | нет | нет |
| Специалист | да | нет | нет |
| Руководитель | да | да | да |
| Администратор | да | да | да |