Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное   
учреждение высшего профессионального образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ   
МОРСКОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет цифровых промышленных технологий

Кафедра вычислительной техники и информационных технологий

**Курсовая работа**

«№ 13. Система обмена книгами»

по дисциплине «Программная инженерия»

Выполнил:

Студент группы № 20130

Хлаинг Вин Ту

Проверил:

Старший преподаватель Высицкий К.А.

Санкт-Петербург

2025

Оглавление

[Введение 2](#_Toc188393522)

[2.План обеспечения качества 3](#_Toc188393523)

[2.1. Цели 3](#_Toc188393524)

[2.2. Область применения 3](#_Toc188393525)

[2.3. Методы тестирования 3](#_Toc188393526)

[2.4. Инструменты и метрики 3](#_Toc188393527)

[3. Планирование спринтов и задач 4](#_Toc188393528)

[3.1. Длительность спринтов 4](#_Toc188393529)

[3.2. Структура спринтов 4](#_Toc188393530)

[3.3. Разбиение задач по функциональным областям 4](#_Toc188393531)

[3.4. Распределение задач по спринтам 4](#_Toc188393532)

[3.5. Приоритеты задач 4](#_Toc188393533)

[3.6. Обзор рисков и их минимизация 4](#_Toc188393534)

[3.7. Методы оценки выполнения задач 5](#_Toc188393535)

[4.Реализация 5](#_Toc188393536)

[4.1 Описание функционала 5](#_Toc188393537)

[4.2 Основные тесты системы 5](#_Toc188393538)

[4.2.1 Тест на назначение задач 5](#_Toc188393539)

[4.2.2 Тест на невозможность изменения завершенной задачи 5](#_Toc188393540)

[4.2.3 Тест на удаление устаревших задач 6](#_Toc188393541)

[4.3 Интеграция и использование 6](#_Toc188393542)

[4.4 Заключение по реализации 6](#_Toc188393543)

[Вывод 7](#_Toc188393544)

# 

# **Введение**

Данный документ описывает план обеспечения качества и планирование спринтов для системы обмена книгами. Основная цель – предоставить высококачественное решение, отвечающее потребностям пользователей, включая студентов, преподавателей и администраторов.

# **2****.План обеспечения качества**

План обеспечения качества (QAP) включает стратегии, инструменты и процессы, направленные на проверку функциональности, производительности и удобства системы.

## **2.1. Цели**

* Обеспечение корректной работы ключевых функций: добавление, резервирование и управление каталогом книг.
* Выявление и устранение ошибок на ранних этапах разработки.
* Поддержание стабильности и производительности системы под нагрузкой.

## **2.2. Область применения**

* **Основные функции:**
  + Работа с каталогом книг: добавление, редактирование и удаление записей.
  + Резервирование книг пользователями.
  + Проверка интерфейса для отображения данных и обработки пользовательских запросов.
* **Исключения:** Интеграция с внешними системами.

## **2.3. Методы тестирования**

* **Юнит-тестирование:** Проверка модулей, таких как Book.reserve() и Catalog.add\_book().
* **Интеграционное тестирование:** Оценка взаимодействия между модулями (например, резервирование книг пользователями).
* **Тестирование интерфейса:** С помощью Selenium проверяются сценарии взаимодействия пользователей.
* **Регрессионное тестирование:** Проверка сохранности существующего функционала после внесения изменений.

## **2.4. Инструменты и метрики**

* **Инструменты:**
  + pytest для модульного тестирования.
  + Selenium для тестирования интерфейса.
  + GitHub Actions для автоматизации тестирования.
* **Метрики:**
  + Покрытие тестами: 80%+ кода должно быть протестировано.
  + Время отклика системы: менее 3 секунд для стандартных операций.
  + Количество найденных и исправленных ошибок.

# **3. Планирование спринтов и задач**

## **3.1. Длительность спринтов**

Проект делится на 4 спринта, каждый длительностью 2 недели.

## **3.2. Структура спринтов**

* **Спринт 1:** Разработка основных функций бэкенда.
* **Спринт 2:** Интеграция бэкенда с базовым пользовательским интерфейсом.
* **Спринт 3:** Улучшение интерфейса и добавление дополнительных функций.
* **Спринт 4:** Тестирование и развертывание системы.

## **3.3. Разбиение задач по функциональным областям**

* **Бэкенд:** Реализация классов Book, User, Catalog.
* **Фронтенд:** Создание интерфейса для взаимодействия с каталогом.
* **Тестирование:** Разработка тест-кейсов и проведение тестов.

## **3.4. Распределение задач по спринтам**

1. **Спринт 1:**
   * Реализация классов и основных методов бэкенда.
   * Написание юнит-тестов для базовых функций.
2. **Спринт 2:**
   * Интеграция Flask API с интерфейсом.
   * Создание HTML-страниц для управления книгами.
3. **Спринт 3:**
   * Улучшение интерфейса, добавление фильтров и сортировки.
   * Тестирование взаимодействия между фронтендом и бэкендом.
4. **Спринт 4:**
   * Проведение системного и нагрузочного тестирования.
   * Развертывание системы на сервере.

## **3.5. Приоритеты задач**

* Высший приоритет: Базовые функции (добавление и резервирование книг).
* Средний приоритет: Интерфейс и пользовательский опыт.
* Низший приоритет: Дополнительные функции (поиск, фильтрация).

## **3.6. Обзор рисков и их минимизация**

* **Риски:** Потеря данных, низкая производительность, ошибки интерфейса.
* **Минимизация:**
  + Регулярное тестирование и резервное копирование данных.
  + Оптимизация запросов к базе данных.

## **3.7. Методы оценки выполнения задач**

* Трекеры задач: Trello или Jira.
* Регулярные отчеты по выполнению задач и прогрессу спринтов.

# **4.Реализация**

## **4.1 Описание функционала**

* **Управление каталогом книг**:
  + Добавление, редактирование, удаление записей о книгах в каталоге.
  + Ведение метаданных, таких как название, автор, жанр, доступность и состояние книги.
* **Управление пользователями**:
  + Добавление и управление данными пользователей (например, имя, электронная почта, контактная информация).
  + Возможность пользователям просматривать историю своих заимствований и сдачи книг.
* **Система резервирования**:
  + Предоставление пользователям возможности резервировать книги через удобный интерфейс.
  + Уведомление пользователей о доступности зарезервированных книг.
* **Интеграция**:
  + Обеспечение бесшовного взаимодействия между API Flask (бекендом) и HTML-фронтендом.
  + Динамическое обновление данных для каталога книг и статуса резервирования.

## **4.2 Основные тесты системы**

### ****4.2.1 Тест на назначение задач****

* **Цель**: Проверить, что задачи, такие как резервирование книг или добавление новых пользователей, назначаются корректно.
* **Подход**:
  + Протестировать метод reserve\_book() для назначения резервирования конкретному пользователю.
  + Убедиться, что обновления в базе данных отображают корректное назначение задач.
* **Ожидаемый результат**: Задачи корректно связываются с соответствующими пользователями без ошибок.

### ****4.2.2 Тест на невозможность изменения завершенной задачи****

* **Цель**: Убедиться, что завершенные задачи (например, возвращенные книги) не могут быть изменены.
* **Подход**:
  + Попробовать редактировать или удалять записи, помеченные как «завершенные».
  + Убедиться, что система блокирует такие операции.
* **Ожидаемый результат**: Система отклоняет изменения завершенных задач.

### ****4.2.3 Тест на удаление устаревших задач****

* **Цель**: Проверить корректность удаления устаревших задач (например, просроченных резервов).
* **Подход**:
  + Симулировать устаревшие задачи резервирования, установив прошедшие даты.
  + Протестировать автоматическое или ручное удаление таких записей.
* **Ожидаемый результат:** Устаревшие задачи удаляются без влияния на активные записи**.**
* **Цель**: Проверить корректную работу функционала управления клиентами (пользователями).
* **Подход**:
  + Протестировать добавление новых пользователей, обновление существующих записей и удаление пользователей.
  + Проверить связь между пользователями и их заимствованными или сданными книгами.
* **Ожидаемый результат**: Операции управления пользователями выполняются корректно без потери данных.

## **4.3 Интеграция и использование**

* **Интеграция**:
  + Размещение системы на локальном или облачном сервере Flask.
  + Обеспечение взаимодействия фронтенда и бекенда через RESTful API.
  + Подключение базы данных (SQLite) для постоянного хранения данных.
* **Использование**:
  + Доступ к системе осуществляется через веб-браузер.
  + Возможности, такие как поиск, резервирование и обновление статусов книг, доступны в режиме реального времени.

## **4.4 Заключение по реализации**

Реализация системы обмена книгами успешно соответствует установленным функциональным требованиям. Основные функции, такие как управление каталогом, взаимодействие с пользователями и работа с резервированием, выполнены. Этапы тестирования завершены, что гарантирует надежность, удобство и безопасность системы. Система готова к развертыванию и обучению пользователей.

# **Вывод**

Система обмена книгами успешно разработана и протестирована. Она обеспечивает функциональность, удобство и производительность, необходимые для удовлетворения пользовательских потребностей.