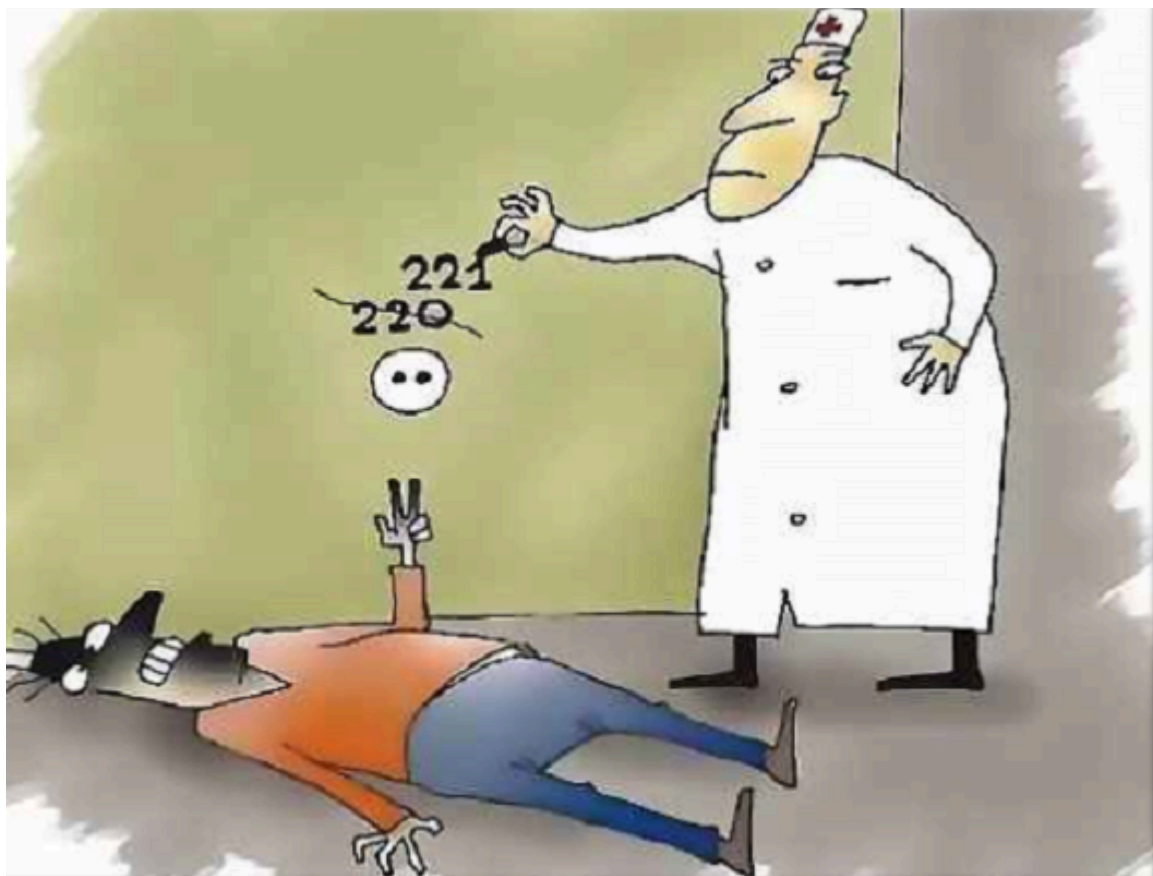


## План тестирования



# Test Plan

## (План тестирования)

### 1. Introduction (Введение)

#### 1.1 Purpose

Назначение данного документа описать план тестирования проекта САС-ИП-СА

#### 1.2 Scope (Область применения)

Данный документ относится к проекту “Производство передвижных бань в Княжестве Новгородском”, разрабатываемого командой из студентов ИТМО. Проект автоматизирует производственный процесс бань, включающий в себя создание каркаса, взаимодействие с подрядчиками, установку оборудования

Данный документ описывает объем работ по тестировании программного продукта САС-ИП-СА. Включая описание стратегии, критериев начала и окончания тестирования.

#### 1.4 Document Terminology and Acronyms (Терминология документа)

##### Glossary

#### 1.5 References (Ссылки)

Лекции Клименкова С.В.  
UML 2 and the Unified Process  
Сказка Царевна Лягушка  
Гуси Лебеди  
Фольклор с участием Бабы Яги

##### Vision

##### Software Requirements Specification

##### Use Case Specification

##### Software Development Plan

##### Glossary

##### SAD

##### Business Case

#### 1.6 Document Structure (Структура документа)

Документ содержит следующую информацию:

- Определение компонентов программного обеспечения, которые необходимо протестировать.
- Рекомендуемые требования к тестированию (высокоуровневые)
- Рекомендации и описание стратегии тестирования.
- Определение необходимых ресурсов и оценки усилий по тестированию.
- Перечисление конечных элементов тестовых мероприятий.

## **2. Evaluation Mission and Test Motivation (Цель и мотивы тестирования)**

### **2.1 Background (Справочная информация)**

Проект САС-ИП-СА – система для поддержания процесса сборки бань. Им будут пользоваться менеджеры по продажам, менеджеры сопровождения заказа, работники отдельных цехов. Архитектуру и общую информацию о проекте можно найти в документах SAD, VIsion

### **2.2 Evaluation Mission (Цели тестирования)**

- Выявить самые серьезные дефекты еще в процессе разработки, до момента передачи системы заказчику
- Убедиться в том, что система удовлетворяет функциональным требованиям
- Убедиться в том, что система удовлетворяет нефункциональным требованиям
- Выявить недостатки реализованной системы на момент передачи заказчику
- Проверить работоспособность системы в условиях близких к реальной эксплуатации

### **2.3 Test Motivators (Мотивы (?) тестирования)**

Источником требований являются документы Use Cases Specification, Business Case, Software Requirements Specification, в которых описаны как функциональные, так и нефункциональные требования, указаны юзкейсы, которые будут выполняться с помощью системы.

## **3 Target Test Items (Целевые объекты тестирования)**

- пользовательский интерфейс
- внешний АПИ сервера
- сущности базы данных
- бизнес логика

## **4. Outline of Planned Tests (План тестов)**

### **4.1. Data and Database Integrity Testing**

- Доступ к БД
- Возможность чтения и записи
- Корректность выполнения обновления записей

### **4.2. Function Testing**

- тестирование системы в сборе на соответствие функциональным требованиям

### **4.3. Business Cycle Testing**

- тестирование системы в сборе на выполнимость use cases
- проверка работы публичного api

### **4.4. User Interface Testing**

- проверка возможности реализации Use Cases с помощью пользовательского интерфейса
- проверка соответствия интерфейса требований

### **4.5. Performance Profiling**

- Замеры показателей производительности при различной нагрузке на систему

### **4.6. Load Testing**

- проверка соответствия требованиям производительности при максимальной заявленной в требованиях нагрузке

### **4.7. Stress Testing**

- определение верхних границ производительности при нагрузке, превышающей требования
- определение поведения системы при высокой нагрузке

### **4.8. Volume Testing**

- определение характеристик системы при

### **4.9. Security and Access Control Testing**

- проверка соответствия прав доступа и типов заявок \ ограничений на них наложенных

## 5. Test Approach (Подход к тестированию)

### 5.1 Testing Techniques and Types (Техники тестирования)

#### 5.2.1 Data and Database Integrity Testing (Тестирование базы данных)

Technique Objective: (Цель)	Проверка работы слоя ORM. Тестируется без публичного API.
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. Запуск БД Postgres</li><li>2. Запуск модулей приложения, работающих с БД</li><li>3. Проверка корректности схемы после выполнения миграция</li><li>4. Проверка создания OrderType с всеми типами OrderParam</li><li>5. Проверка создания OrderType с всеми повторяющимися типами OrderParam</li><li>6. Создание заявки с несуществующим в шаблоне OrderParam</li><li>7. Проверка работы репозитория для Order</li></ol>
Oracles: (Источники)	<ul style="list-style-type: none"><li>- SRS</li><li>- ER-модель</li></ul>
Required Tools: (Инструменты)	<ul style="list-style-type: none"><li>- Postgres</li><li>- Pycharm</li><li>- Python 3.9</li><li>- Pytest</li></ul>
Success Criteria: (Критерий успеха)	<ul style="list-style-type: none"><li>- Полученные ответы или ошибки соответствуют ожидаемым.</li><li>- Целостность данных не нарушается</li><li>- Итоговая созданная схема БД соответствует миграциям</li></ul>

### 5.2.2 Function Testing (Функциональное тестирование)

Technique Objective: (Цель)	Проверка корректности работы системы целиком на основе проверки соответствия функциональным требованиям.
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. С помощью автотестов системного уровня проверяем систему через публичный API</li><li>2. Проверка 3.1.5 Система должна показывать заявки пользователям с теми ролями, которым они соответствуют</li><li>3. Проверка 3.1.6 Система должна предоставлять возможность создать заявку на СаБаКо</li><li>4. Проверка 3.1.7 Система должна предоставлять возможность создать заявку на сруб</li><li>5. Проверка 3.1.13 Система должна предоставлять возможность создать заявку на брак произведенного продукта (СУКН, КУСЬ, ВТУС, СРУК, изделие ремесленников, зачарование, сруб, бревно)</li><li>6. Проверка 3.1.17 Система должна предоставлять список заказов и их статусов с фильтрацией</li><li>7. Проверка 3.1.20 Система должна предоставлять возможность указать статус любой из типов заявки</li><li>8. Проверка 3.1.23 Система должна предоставлять возможность принять заявку в работу</li><li>9. Проверка 3.1.24 Система должна предоставлять возможность согласования заявки после её рассмотрения</li><li>10. Проверка 3.1.25 Система должна предоставлять возможность принять готовность заявки</li></ol>
Oracles:	SRS

(Источники)	
Required Tools: (Инструменты)	<ul style="list-style-type: none"> <li>- Postgres</li> <li>- Keycloak</li> <li>- Pycharm</li> <li>- Pytest</li> </ul>
Success Criteria: (Критерий успеха)	Система удовлетворяет покрытым функциональным требованиям. Найденные дефекты были устранены.

### 5.2.3 Business Cycle Testing (Тестирование бизнес-цикла)

Technique Objective: (Цель)	Проверка корректности работы системы целиком на основе выполнения бизнес-сценариев, которые моделируют действия пользователей согласно юзкейсам, с использованием пользовательского интерфейса.
Technique: (Описание процесса)	<ol style="list-style-type: none"> <li>1. С помощью автотестов системного уровня проверяем систему через пользовательский интерфейс. Результаты действия валидируются через публичный API</li> <li>2. Проверка Use-Case 1 “Авторизация”. Включить альтернативные потоки.</li> <li>3. Проверка Use-Case 2 “Создать заказ на СаБаКо”</li> <li>4. Проверка Use-Case 3 “Создать заявку на компонент СаБаКо”. Включить альтернативные потоки.</li> <li>5. Проверка Use-Case 4 “Создать заявку на брак”. Включить альтернативные потоки.</li> <li>6. Проверка Use-Case 6 “Получить информацию о прогрессе по заказу”</li> </ol>
Oracles: (Источники)	Use Case Specification

<p>Required Tools:</p> <p>(Инструменты)</p>	<ul style="list-style-type: none"> <li>- Postgres</li> <li>- Keycloak</li> <li>- Python 3.9</li> <li>- Selenium</li> <li>- Pytest</li> <li>- PyCharm</li> </ul>
<p>Success Criteria:</p> <p>(Критерий успеха)</p>	<p>Юзкейсы, выбранные для покрытия успешно выполняются с помощью системы. Найденные дефекты были устранены.</p>



#### 5.2.4 User Interface Testing (Тестирование интерфейса)

Technique Objective: (Цель)	Проверка корректности работы пользовательского интерфейса системы на основе выполнения бизнес-сценариев, которые моделируют действия пользователей согласно юзкейсам, с использованием пользовательского интерфейса.
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. Запуск БД Postgres</li><li>2. Запуск Keycloak</li><li>3. Запуск серверного приложения</li><li>4. Запуск пользовательского интерфейса</li><li>5. Предоставление доступа тестовой группе пользователей. Пользователям заранее подготавливаются аккаунты с необходимыми ролями.</li><li>6. Если юз-кейс требует уже созданных заказов или заявок - такие сущности должны быть созданы.</li><li>7. Предоставление небольшого набора юзкейсов (Use Case 2 - 7)</li><li>8. Сбор обратной связи и подсчет ошибок при выполнении юзкейсов с помощью системы</li></ol>
Oracles: (Источники)	Use Case Specification
Required Tools: (Инструменты)	<ul style="list-style-type: none"><li>- Postgres</li><li>- Keycloak</li><li>- Pycharm</li><li>- Pytest</li><li>- Google Chrome</li></ul>
Success Criteria: (Критерий успеха)	Использовано более 50% юзкейсов без альтернативных потоков. Использовано более 20% юзкейсов с альтернативными потоками. Процент пользовательских ошибок не превышает 95 процентов. Количество жалоб на интерфейс в среднем не превышает 11

### 5.2.5 Performance Profiling (Тестирование производительности)

Technique Objective: (Цель)	Оценка производительности, устойчивости и способности системы справляться с заданной нагрузкой.
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. Тестовые запросы включают в себя<ol style="list-style-type: none"><li>a. Создание заказа</li><li>b. Получение заказа</li><li>c. Удаление заказа</li><li>d. Удаление заказа и его заявок</li></ol></li><li>2. Запуск набора запросов с помощью JMeter для создания нагрузки разного уровня</li><li>3. Сбор метрик времени отклика и ответа сервера</li></ol>
Oracles: (Источники)	SRS Use Case Specification
Required Tools: (Инструменты)	<ul style="list-style-type: none"><li>- Postgres</li><li>- Keycloak</li><li>- JMeter</li><li>- Python 3.9</li></ul>
Success Criteria: (Критерий успеха)	Среднее время отклика системы соответствует проектной документации и уменьшается при уменьшении нагрузки

### 5.2.6 Load Testing (Нагрузочное тестирование)

Technique Objective: (Цель)	Оценка производительности, устойчивости и способности системы справляться с нагрузкой, превышающей проектную
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. Запуск БД Postgres</li><li>2. Запуск Keycloak</li><li>3. Запуск серверного приложения</li><li>4. Нагрузить систему с помощью JMeter, который симулирует работу 20 пользователей</li><li>5. Тестовые запросы включают в себя<ol style="list-style-type: none"><li>a. Создание заказа</li><li>b. Создание нескольких заявок.</li><li>c. Изменение статусов у заявок.</li><li>d. Создание заявок на брак</li><li>e. Удаление заказа и его заявок</li></ol></li><li>6. Собрать статистику работы системы после 1 часа работы</li><li>7. Измерить среднее время отклика системы</li></ol>
Oracles: (Источники)	SRS Use Case Specification
Required Tools: (Инструменты)	Postgres Keycloak JMeter
Success Criteria: (Критерий успеха)	Система продолжает функционировать при высоких нагрузках. Не происходит отказов приложения. Среднее время ответа на запрос составляет менее 1 минуты

### 5.2.7 Stress Testing (Стрессовое тестирование)

Technique Objective: (Цель)	Оценка производительности, устойчивости и способности системы справляться с заданной нагрузкой.
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. Запуск БД Postgres</li><li>2. Запуск Keycloak</li><li>3. Запуск серверного приложения</li><li>4. Тестовые запросы включают в себя<ol style="list-style-type: none"><li>a. Создание заказа</li><li>b. Создание нескольких заявок.</li><li>c. Изменение статусов у заявок.</li><li>d. Создание заявок на брак</li><li>e. Удаление заказа и его заявок</li></ol></li><li>5. Запуск набора запросов с помощью JMeter для создания нагрузки разного уровня</li><li>6. Сбор метрик времени отклика и ответа сервера</li><li>7. Увеличение нагрузки по получения отказа сервера</li><li>8. Определение максимальных пиковых значений нагрузки</li></ol>
Oracles: (Источники)	SRS Use Case Specification
Required Tools: (Инструменты)	<ul style="list-style-type: none"><li>- Postgres</li><li>- Keycloak</li><li>- JMeter</li><li>- Python 3.9</li></ul>
Success Criteria: (Критерий успеха)	Система перестала отвечать либо невозможно увеличить нагрузку для получения отказа системы

### 5.2.8 Volume Testing (Объемное тестирование)

Technique Objective: (Цель)	Оценка производительности, устойчивости и способности системы справляться с заданной нагрузкой.
Technique: (Описание процесса)	<ol style="list-style-type: none"><li>1. Запуск БД Postgres</li><li>2. Запуск Keycloak</li><li>3. Запуск серверного приложения</li><li>4. Подготовка тестовых данных для заполнения БД и Keycloak</li><li>5. Заполнение приложения набором данных превосходящим предполагаемую нагрузку за год<ol style="list-style-type: none"><li>а. Создание заказов ~ 1000 штук</li></ol></li><li>6. Сбор метрик времени отклика и ответа сервера</li></ol>
Oracles: (Источники)	SRS Use Case Specification
Required Tools: (Инструменты)	<ul style="list-style-type: none"><li>- Postgres</li><li>- Keycloak</li><li>- JMeter</li><li>- Python 3.9</li></ul>
Success Criteria: (Критерий успеха)	Среднее время отклика системы соответствует проектной документации и уменьшается при уменьшении нагрузки

### 5.2.9 Security and Access Control Testing (Тестирование безопасности и прав доступа)

Technique Objective: (Цель)	Выявление уязвимостей программного обеспечения, которые могут повлечь за собой несанкционированный доступ к данным и функциональности системы.
Technique: (Описание процесса)	<ul style="list-style-type: none"><li>• Попытки исполнения SQL инъекций в комментарии при создании заявок на оборудование</li><li>• Проверка доступности/недоступности заказов/заявок/заявок на брак для пользователей с разными ролями</li><li>• Попытки входа с помощью некорректных данных.</li><li>• Попытки выполнить запрос с истекшим или некорректным токеном.</li></ul>
Oracles: (Источники)	RiskList
Required Tools: (Инструменты)	<ul style="list-style-type: none"><li>- Postgres</li><li>- Keycloak</li><li>- JMeter</li><li>- Python 3.9</li></ul>
Success Criteria: (Критерий успеха)	Для администраторов доступны все заявки, для роли заказчика доступны только те заказы, в которых он является автором заказа. Система не доступна по некорректному логину и паролю.

## 6. Entry and Exit Criteria (Критерии старта и окончания)

### 6.1. Test Plan Entry Criteria (Критерий старта)

Тестирование начинается после реализации первой развёрнутой версии АПИ сервера

### 6.2 Test Plan Exit Criteria (Критерий окончания)

Тестирование заканчивается после соответствия системы указанным требованиям и успешном выполнении use cases после окончания разработки требуемых функций

### 6.3 Suspension and Resumption Criteria (Критерий паузы и возобновления)

Тестирование приостанавливается когда текущий функционал уже протестирован, но новый ещё не реализован. Тестирование возобновляется при реализации новых функций \ изменении тестовых условий или развертывания.

## **7. Deliverables (Ожидаемые результаты тестирования)**

### **7.1 Test Evaluation Summaries (Результаты выполнения тестов)**

Результаты выполнения тестов будут представлены в виде html отчетов по тестовым кейсам и группам тестов.

### **7.2 Perceived Quality Reports (Оценка качества)**

Отчеты будут представлены в виде html документов и включать в себя итоговый отчет по кодовой базе со всеми группами тестов, а также отчет о покрытии кодов тестами.

### **7.3 Incident Logs and Change Requests (Журналы ошибок и изменений)**

Журнал ошибок и изменений будет представлен в виде GitHub Issues. Задачи будут создаваться на доске, у которой будут следующие статусы:

- New
- Backlog
- In Progress
- Testing
- Done

Задача будет привязываться к ветке и к репозиторию в процессе разработке и исправления ошибок. Описание задачи будет содержать следующую информацию:

- Use Case
- Требование
- Шаги для воспроизведения
- Ожидаемый результат
- Приоритет
- Тестируемый Модуль

## **8. Environmental Needs (Необходимое окружение для проведения тестирования)**

### **8.1 Base System Hardware (Базовое аппаратное обеспечение)**

<b>Resource (Ресурс)</b>	<b>Quantity (Количество)</b>	<b>Name and Type (Название и тип)</b>
Helios	1	128GB ram / 560GB space / Intel Xeon E5-2643 3.3GHz / 16 cores

## 8.2 Base Software Elements in the Test Environment (Базовые программы тестового окружения)

Software Element Name (Название)	Version (Версия)	Type (Тип)
Python interpreter	3.9	Среда выполнения кода
PostgreSQL	14	База данных
Keycloak	21.0.0	Система аутентификации

## 8.3 Productivity and Support Tools (Вспомогательные инструменты)

Tool Category or Type (Тип программы)	Tool Brand Name (Название)	Vendor (Производитель)	Version (Версия)
Инструмент тестирования API	Swagger	SmartBear Software	3
Инструмент нагрузочного тестирования	JMeter	Apache Foundation	5
Инструмент unit тестирования	Pytest	Holger Krekel	7.4
Среда разработки автотестов	Pycharm	JetBrains	2023.*
Инструмент тестирования ui	Selenium	Software Freedom Conservancy	4

## 9. Responsibilities, Staffing, and Training Needs (Обязанности сотрудников)

### 9.1 People and Roles (Люди и роли)



<b>Role</b> <b>(Роль)</b>	<b>Minimum Resources Recommended</b> <b>(Минимально необходимое количество людей)</b>	<b>Specific Responsibilities</b> <b>(Обязанности)</b>
Тестировщик	1	Ручное тестирование. Написание и исполнение интеграционных тестов.
Разработчик	1	Написание unit тестов.
Менеджер проекта	1	Проверка соответствия бизнес-требованиям

## **10. Management Process and Procedures (Управление)**

### **10.1 Reporting on Test Coverage (Сообщение о тестовом покрытии)**

Каждый тестировщик следит за процессом проверки тестовых кейсов и проверяет отчеты. После проведения тестирования, формируется отчет по каждой группе тестов, проводится анализ и формируются задачи в бэклог.

### **10.2 Problem Reporting, Escalation, and Issue Resolution (Выявление, избегание и решение проблем)**

Каждый тестировщик ведет журнал ошибок в Github Issues.

Согласно журналу тестировщик создает в системе распределения задач новые задачи. Менеджер указывает их приоритет. Разработчики берут задачи из бэклога и выполняют их в соответствии с приоритетом задач.

### **10.3 Approval and Signoff (Утверждение плана тестирования)**

После окончательной разработки плана тестирования, он утверждается Заказчиком.