| 18.10.22 | Авраменко  Давыдов  Шишкин | Initial |
| --- | --- | --- |
| 31.10.22 | Авраменко  Давыдов  Шишкин | Переписаны пункты 4, 5. Расширены 8.2, 8.3. |

**Test Plan (План тестирования)**

**1. Introduction (Введение) 1.1. Purpose**

Данный документ описывает план тестирования разрабатываемого проект - цели и мотивы тестирования, описание тестовых сценариев и критерии их прохождения, используемые программы и их окружение.

**1.2. Scope (Область применения)**

Данный проект представляет собой систему для автоматизации процесса полного цикла проведения модификации: от выбора модификации до поддержи во время восстановительного периода.

**1.3. Intended Audience (Предполагаемая аудитория)**

Этот документ рекомендован к ознакомлению Заказчику, а также команде разработки проекта.

**1.4. Document Terminology and Acronyms (Терминология документа)** 1) Glossary

**1.5. References (Ссылки)**

1) Джим Арлоу, Айла Нейштадт. UML 2 и унифицированный процесс. 2) Лукьяненко С. В. “Геном”.

3) Vision

4) Требования к системе

5) Прецеденты использования

6) Список рисков

**1.6.Document Structure (Структура документа)**

Во втором разделе документа описываются цель и мотивы тестирования вместо со справочной информацией с описанием архитектуры проекта.

В третьем разделе содержатся целевые объекты тестирования.

В четвертом разделе находится план тестов и краткое описание сценариев.

В пятом разделе можно увидеть основные подходы к тестированию. Далее в шестом разделе расположены критерии старта, окончания и паузы в тестировании.

В седьмом разделе описываются ожидаемые результаты тестирования.

Восьмой раздел описывает необходимое для тестирования окружение и используемые программы.

Девятый раздел содержит информацию об обязанностях людей.

Десятый - процесс управления тестированием.

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

**2.1. Background (Справочная информация)**

С описанием данного проекта можно ознакомиться в Vision и Use-case

Со структурой данного проекта можно ознакомиться в SAD и SDP.

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

● Проверить работоспособность основных функций системы

● Повысить соответствие системы поставленным требованиям

● Получение текущего статуса работоспособности системы, актуальной информации о ней

● Повысить отказоустойчивость системы при различных сценариях использования

● Уменьшить количество потенциальных дефектов

● Оценить работоспособность системы под нагрузкой

● Оценить максимальную нагрузку на систему

**2.3. Test Motivators (Мотивы тестирования)**

Следующие документы служат источниками информации для тестирования: ● Концепция - Vision

● Описание прецедентов - Usecase

● Список рисков - RiskList

● План разработки продукта - SDP

● Спецификация требований к продукту - SRS

● Описание архитектуры - SAD

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

* Вход в систему
* Создание нового клиента
* Создание заявки на модификацию
* Генерация коммерческого предложения
* Получение рисков по специализации
* Профили клиентов
* Подтверждение заявок
* Создание сценариев сопровождения/модификации
* Поиск модификаций/клиентов
* Обработка заявок на регистрацию
* Некорректные вводимые данные
* Производительность системы под нагрузкой

1. Классы сервисов
2. Endpoint-ы backend части
3. Frontend интерфейс

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

1. Тестирование базы данных - проверка ограничений базы данных
2. Функциональное тестирование - тестирование отдельных частей системы на соответствие функциональным требованиям
3. Тестирование бизнес-цикла - сквозное тестирование системы в соответствии с прецедентами
4. Тестирование интерфейса - проверка функциональности основных элементов интерфейса
5. Нагрузочное тестирование - тестирование, нацеленное на определение верхней границы количества запросов в единицу времени, при котором время ответа системы будет соответствовать необходимым значениям.
6. Стрессовое тестирование - определение верхнего предела количества запросов в единицу времени, при котором система становится неработоспособной
7. Тестирование безопасности и прав доступа - тестирование доступности сервисов на основании наличия определенных прав доступа пользователей
8. Тестирование производительности - проверка времени отклика системы и его соответствие требуемым значениям

**5. Test Approach (Подход к тестированию)**

**5.1. Testing Techniques and Types (Техники тестирования)**

**5.1.1. Data and Database Integrity Testing (Тестирование базы данных)**

| Technique Objective:  (Цель) | Оценить корректность обработки данных, доступность базы данных при нагрузках, обработка ошибок при запросах |
| --- | --- |
| Technique: (Описание процесса) | 1. Подготовка входных данных для негативных и позитивных сценариев, проверяющих ограничения БД, основываясь на ER-модели базы, а также на функциональных требованиях, описанных в SRS 2. Провести запросы к базе данных с использованием языка запросов SQL 3. Проверка ответа или выданной ошибки 4. Проверка соответствия состояния базы данных и сравнение его с ожидаемым состоянием после выполнения каждого теста |
| Oracles:  (Источники) | ER-модель базы данных, функциональные требования к системе (SRS) |
| Required Tools:  (Инструменты) | * Тестовое окружение * Заполненная база данных соответствующая ER-модели * Утилита PSQL |
| Success Criteria: (Критерий успеха) | Полученные ответы или ошибки соответствуют ожидаемым, как и состояние базы данных |

**5.1.2. *Function Testing (Функциональное тестирование)***

| Technique Objective:  (Цель) | Проверка реализованной функциональности для каждой предусмотренной группы пользователей. |
| --- | --- |
| Technique: (Описание процесса) | 1. Выбор целевого серверного компонента для тестирования в соответствии с функциональными требованиями описанными в SRS 2. Тестирование проводится с применением драйверов и мок классов для изоляции тестируемой функциональности. 3. Для каждого сценария обеспечиваются предусловия с помощью настройки мок классов и инициализации драйверов. 4. После выполнения теста проверяется постусловие на соответствие данных в случае успешного выполнения прецедента. 5. Проверяется корректность ошибок/статусов возвращаемых целевым компонентом при тестировании альтернативного потока |
| Oracles:  (Источники) | * Функциональные требования к системе описанные в SRS * Use-case |
| Required Tools:  (Инструменты) | * Junit * Java JDK |
| Success Criteria: (Критерий успеха) | Поведение компонентов соответствует функциональным требованиям относящимся к этому компоненту |

**5.1.3. *Business Cycle Testing (Тестирование бизнес-цикла)***

| Technique Objective:  (Цель) | Данный тип тестирования проверяет, что при выполнении определенных действий описанных в UseCase будут выведены корректные данные и система изменит свое состояние на ожидаемое с точки зрения пользователя системы |
| --- | --- |
| Technique: (Описание процесса) | 1. Тестовым сценарием является последовательность шагов описанных в TestCase 2. Создание тестовых данных для выполнения основной и альтернативной последовательности 3. Выполнение последовательности шагов описанных в сценарии в запущенной системе с тестовым набором данных для обеспечения предусловий 4. Сравнение результата с ожидаемым описанным в пункте “Постусловие” тестируемого сценария 5. Результат сравнения записывается тестировщиком в “Журнал тестирования” |
| Oracles:  (Источники) | * Функциональные требования к системе (SRS) * TestCase |
| Required Tools:  (Инструменты) | * Браузер Google Chrome * PostgreSQL |
| Success Criteria: (Критерий успеха) | Поведение системы соответствует описанному в пользовательском сценарии, а также состояние системы в конце теста соотвествует постусловию. |

**5.1.4. User Interface Testing (Тестирование интерфейса)**

| Technique Objective:  (Цель) | Проверка отображения информации, соответствующей действиям, которые выполнил пользователь, компонентов навигации, ввода и валидации данных |
| --- | --- |
| Technique: (Описание процесса) | 1. Тестовое покрытие формируется для основных UI компонентов системы, тесты должны проверять соответствие поведения компонентов описанию в TestCase   Под основными UI компонентами подразумеваются компоненты для которых приведены MockUp’ы в UseCase.   1. Тестирование должно осуществляться автоматически - с помощью системы автоматизированного тестирования Selenium. 2. Шаблоны тестов должны формироваться при помощи Selenium IDE и исполняться при помощи Selenium RC в браузерах Firefox или Chrome. |
| Oracles:  (Источники) | * Функциональные требования к системе (SRS) * TestCase * UseCase |
| Required Tools:  (Инструменты) | * Selenium * Браузеры Google Chrome * ChromeDriver |
| Success Criteria: (Критерий успеха) | Поведение UI соответствует описанному в пользовательском сценарии, а также производится валидация данных |

**5.1.5. Load Testing (Нагрузочное тестирование)**

| Technique Objective:  (Цель) | Проверка максимального количества запросов, которое способна выдержать система, обеспечивая необходимый уровень обслуживания. |
| --- | --- |
| Technique: (Описание процесса) | Приложение разворачивается в тестовой среде. Основным условием является то, что система не должна быть подвержена никакой нагрузке, кроме тестовой.   1. Сконфигурировать тесты Apache JMeter, таким образом, чтобы убедится в выполнении требований, заявленных в SRS при заданной конфигурации системы 2. Генерируется поток запросов к системе и фиксируется среднее время отклика 3. Составить график зависимости времени отклика системы от нагрузки |
| Oracles:  (Источники) | * SRS |
| Required Tools:  (Инструменты) | Apache JMeter |
| Success Criteria: (Критерий успеха) | Количество запросов равно или превышает значение, указанное в SRS, при условии соответствия характеристик сервиса заданным |

**5.1.6. Stress Testing (Стрессовое тестирование)**

| Technique Objective:  (Цель) | Найти нагрузку при которой время отклика системы превышает максимально допустимое значение и система остается работоспособной |
| --- | --- |
| Technique: (Описание процесса) | Приложение разворачивается в тестовой среде. Основным условием является то, что система не должна быть подвержена никакой нагрузке, кроме тестовой.   1. Сконфигурировать тесты Apache JMeter, таким образом, чтобы нагрузка на систему превышала заявленную пиковую нагрузку 2. Генерируется поток запросов к системе и фиксируется среднее время отклика 3. В случае если система осталась работоспособной повторить шаги 1-2 с увеличенной нагрузкой 4. Составить график зависимости времени отклика системы от нагрузки |
| Oracles:  (Источники) | * SRS |
| Required Tools:  (Инструменты) | Apache JMeter |
| Success Criteria: (Критерий успеха) | Максимальная нагрузка на систему, при которой система остается работоспособной, превышает или равна максимальной пиковой нагрузке указанной в SRS |

**5.1.7. Security and Access Control Testing (Тестирование безопасности и прав доступа)**

| Technique Objective:  (Цель) | Проверка механизмов аутентификации и авторизации пользователей, а также проверка разграничения функциональности в соответствии с ролями пользователей |
| --- | --- |
| Technique: (Описание процесса) | 1. Для каждого метода API сервера определить в соответствии с SRS необходимые права доступа. 2. Определить права доступа каждой из существующих ролей. 3. Создать тестовых пользователей для каждой роли. 4. Проверить возможность пользователей каждой роли получить доступ к методам, к которым они должны иметь доступ. |
| Oracles:  (Источники) | * Функциональные требования к системе (SRS) * Use-case диаграмма |
| Required Tools:  (Инструменты) | * Postman, PostgreSQL |
| Success Criteria: (Критерий успеха) | * Определены права доступа каждой роли ко всем публичным API системы. * Проверена доступность определенной в SRS функциональности для указанной роли и недоступность для остальных * Все реализованные тестовые сценарии выполняются успешно и без ошибок. |

**5.1.8 Performance Profiling (Тестирование производительности)**

| Technique Objective:  (Цель) | Проверка времени отклика системы и его соответствия требуемым значениям. |
| --- | --- |
| Technique:  (Описание процесса) | Приложение разворачивается в тестовой среде. Основным условием является то, что система не должна быть подвержена никакой нагрузке, кроме тестовой.   1. Сконфигурировать тесты Apache JMeter, таким образом, чтобы нагрузка на систему не превышала заявленную нагрузку 2. Генерируется поток запросов к системе и фиксируется среднее время отклика 3. После выполнения запросов тестировщик находит 95-й процентиль времени отклика системы. |
| Oracles:  (Источники) | * SRS |
| Required Tools:  (Инструменты) | * Apache JMeter |
| Success Criteria:  (Критерий успеха) | 95-й процентиль времени отклика не превышает верхнюю границу заданного в SRS времени отклика. |

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

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

* Тестовое окружение подготовлено и готово к работе, на нем развернута система или прототип системы с частичной функциональностью
* Тестирование конкретной функциональности начинается с момента появления ее реализации в системе.

***6.2. Test Plan Exit Criteria (Критерий окончания)***

● Реализованы и протестированы поставленные заказчиком требования

● Количество обнаруженных дефектов в процессе тестирования уменьшилось до незначительных значений не влияющих на основную функциональность

**6.3. Suspension and Resumption Criteria (Критерий паузы и возобновления)** Критерии приостановки тестирования:

● Тестовое окружение недоступно

● Продукт не соответствует основным требованием системы

● Найдены критические дефекты

Критерии возобновления тестирования:

● Продукт стал соответствовать требованиям

● Тестовое окружение доступно

● Критические дефекты исправлены

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

**7.1. Test Evaluation Summaries (Результаты выполнения тестов)** В случае успеха:

● Название теста

● Краткое описание теста

● Описание входных данных

● Окружение

● Используемые программы (браузер и т.д.)

● Результат

В случае обнаружения дефекта:

● Название теста

● Краткое описание теста

● Описание входных данных

● Окружение

● Используемые программы (браузер и т.д.)

● Снимки экрана при наличии

● Описание ожидаемого результата

● Описание фактического результа

● Критичность дефекта

● Предполагаемые сроки его исправления

Нагрузочное тестирование:

● Описание теста

● Входные данные

● График с полученным результатом

● Комментарии

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

Реализовано не менее 28 тестов, из которых:

1. 23 функциональных тестов (количество функциональных требований к системе);
2. Проведено не менее 6 тестов бизнес-цикла, соотносящихся с архитектурно значимыми прецеденты;
3. Тестирование интерфейса;
4. Тестирование производительности;
5. Стрессовое тестирование;
6. Тестирование безопасности и прав доступа;
7. Нагрузочное тестирование.

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

При обнаружении ошибки в системе будет создаваться прецедент с описанием ожидаемого и фактического поведения, а также описание окружения, на котором была обнаружена ошибка.

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

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

| Resource  (Ресурс) | Quantity  (Количество) | Name and Type  (Название и тип) |
| --- | --- | --- |
| Оперативная память | 1 | 8ГБ + |
| Операционная система | 1 | Windows 10 |
| Процессор | 1 | Intel Core I7-2700K or greater |

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

| Software Element Name (Название) | Version (Версия) | Type (Тип) |
| --- | --- | --- |
| Google Chrome | 90.0.4430.93 и выше | Браузер |
| Java Development Kit | 1.8.0 или новее | Комплект ПО для разработки и сборки кода на языке Java |
| ChromeDriver | версия совпадающая с мажорной версией Chrome | Драйвер для управления веб браузером Chrome для выполнения тестов с помощью Selenium |
| Git | 2.6 или новее | Система контроля версий |

**8.3. Productivity and Support Tools (Вспомогательные инструменты)**

| Tool Category or Type (Тип программы) | Tool Brand Name (Название) | Vendor (Производитель) | Version (Версия) |
| --- | --- | --- | --- |
| Инструмент для автоматизации действий веб-браузера | Selenium | Сообщество Open Source | 3 или новее |
| Тестирование API | Postman | Postman Inc | 9.18.0 |
| Инструмент для проведения нагрузочного тестирования | JMeter | Apache SW Foundation | 4 или новее |
| Фреймворк для модульного тестирования программного обеспечения на языке Java. | JUnut | - | 5.9.1 |
| IDE для Java | IntelliJ IDEA | JetBrains | 2022 или новее |

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

**9.1. People and Roles (Люди и роли)**

| Role (Роль) | Minimum Resources Recommended  (Минимально необходимое количество людей) | Specific Responsibilities (Обязанности) |
| --- | --- | --- |
| Заказчик | 1 | Предоставить требования к системе, ознакомиться с результатами тестирования |
| Разработчик | 2 | Исправлять дефекты, обнаруженные в процессе тестирования |
| Тестировщик | 2 | Написать тестовые сценарии, произвести тестирование системы |
| Пользователь | 3 | Произвести тестирование, написать отзыв/пожелания |

**10. Management Process and Procedures (Управление) 10.1. Reporting on Test Coverage (Сообщение о тестовом покрытии)**

● Тестировщик предоставляет результаты тестирования в формате, описанном в пункте 7

● Заказчик ознакамливается с полученными результатами и оставляет комментарии

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

● Руководитель проекта представляет результаты тестирования и собственные рекомендации

● Заказчик составляет документ с поправками

● Потенциальные пользователи оставляют отзыв

● Команда ознакамливается с данным документов и производит доработку продукта

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

Команда разработки презентует работоспособность проекта и результаты тестирования, далее происходит обсуждение и согласование результатов.

Утверждение данного документа осуществляется в присутствии Заказчика и Руководителя проекта.