**Test Plan**

**(План тестирования)**

**1.** **Introduction (Введение)**

**1.1** **Purpose**

Целью данного документа является описание запланированных работ по тестированию, начиная с описания объекта, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения для разрабатываемой автоматизированной системы контроля чистоты эксперимента “Платформа”.

**1.2 Scope (Область применения)**

Этот документ относится к автоматизированной информационной системе для системы контроля эксперимента “Платформа”, которое будет разработано инициативной группой “SaDiVa”. Разрабатываемая информационная система предоставляет возможность сотрудникам “Цитадели” следить за заключенными и вмешиваться при нарушении чистоты эксперимента. Формат разрабатываемой системы будет представлять собой клиент-серверное приложение. Для работы с системой будет предусмотрено несколько сценариев использования в зависимости от прав пользователя.

Данный документ будет использоваться при разработке продукта тестировщиками, программистами и менеджерами.

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

Данный документ будет использоваться тестировщиками, разработчиками для проведения тестирования автоматизированной системы контроля чистоты эксперимента “Платформа”.

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

Glossary ([Glossary.docx](https://docs.google.com/document/d/1Ou6epUB6-zqZc1wfUPYO2CbKB8MVWuJW/edit))

**1.5** **References (Ссылки)**

нет

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

В разделах данного документа содержится информация о:

1. Целях и мотивах тестирования
2. Целевых объектах тестирования
3. Плане тестов
4. Подходах к тестированию
5. Критериях старта и конца
6. Ожидаемых результатах
7. Необходимом окружении
8. Обязанностях сотрудников
9. О мероприятиях по управлению процессом тестирования

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

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

Разрабатываемая информационная система предоставит возможность сотрудникам “Цитадели” производить контроль за соблюдением правил эксперимента, а также вмешиваться в эксперимент во избежание нарушений правил. Формат разрабатываемой системы будет представлять собой клиент-серверное приложение. Архитектура приложения реализована согласно архитектуре MVI, с разделением на фронтенд и бекенд. Общая диаграмма системы доступна в [SAD.docx](https://docs.google.com/document/d/111xonRdFK0ZgoxNtOQXzRB00LC9Ezs1f/edit) в разделе 2.

Бекенд составляющая, реализующая доступ к данным и обработку бизнес-логики реализована на языке Kotlin с использованием фреймворка Ktor. Фронтенд составляющая представляет собой тонкий клиент, реализованный с помощью compose-jetpack multiplatform.

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

Основные цели тестирования проекта:

1. Удовлетворение заявленных требований и возможностей, отраженных в разделе 4 документа [Vision.docx](https://docs.google.com/document/d/1E4bIXlkFc2ZZ4bEkHAf8uSEvKtMdK4Rw/edit?rtpof=true)
2. Валидация и верификация проекта с учетом функциональных и нефункциональных требований из раздела 3 документа [Vision.docx](https://docs.google.com/document/d/1E4bIXlkFc2ZZ4bEkHAf8uSEvKtMdK4Rw/edit?rtpof=true)
3. Обеспечение оптимальности цикла разработки через использование техник TDD и регрессионного тестирования, что реализуется через достаточное тестовое покрытие (80-90%)
4. Минимизация рисков, возникающих в процессе кодирования, отраженных в документе [RiskList.docx](https://docs.google.com/document/d/1wFBYTW8Hgl7Q1R_LKRvdZGVd30RoQ7UR/edit?rtpof=true)
5. Исполнение специфических требований к эргономике и доступности системы, так как большую часть пользователей составляют заключенные.

**2.3** **Test Motivators (“Мотиваторы” тестирования)**

В таблице перечислены основные “мотиваторы” тестирования - аспекты проекта, имеющие наибольшую важность в процессе тестирования

| Источник информации | Документ | Назначение |
| --- | --- | --- |
| Разделы 2.1, 2.2, 2.3 | [Vision.docx](https://docs.google.com/document/d/1E4bIXlkFc2ZZ4bEkHAf8uSEvKtMdK4Rw/edit?rtpof=true) | В разделах подробно описан бизнес-процесс, его текущие недостатки и особенности системы, которые должны его улучшить. Будет использовано в первую очередь для тестирования удобства использования системы |
| Описание основных потоков юзкейсов | [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) | Будет использовано для обеспечения регрессионного тестирования |
| Precondition, postcondition каждого юзкейса | [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) | Будет использовано для сохранения инвариантов системы, для обеспечения корректности моделирования бизнес-процесса |
| Риски категории ТР | [RiskList.docx](https://docs.google.com/document/d/1wFBYTW8Hgl7Q1R_LKRvdZGVd30RoQ7UR/edit?rtpof=true) | Моделирование ситуаций возникновения рисков для упрощения их смягчения (только для тех, что можно протестировать) |
| Функциональные требования | [SRS.docx](https://docs.google.com/document/d/1FPEVQYix8eWcn4zHgGHRQIlBTKPg9gIR/edit) пункт 3.1 | Обеспечение полноты тестового покрытия |
| Требования к надежности и производительности | [SRS.docx](https://docs.google.com/document/d/1FPEVQYix8eWcn4zHgGHRQIlBTKPg9gIR/edit) пункты 3.3, 3.4 | Будет использовано в качестве числовых метрик при нагрузочном и др. тестировании |

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

Исходя из общей архитектуры системы ([SAD.docx](https://docs.google.com/document/d/111xonRdFK0ZgoxNtOQXzRB00LC9Ezs1f/edit) п.2), включаемых в нее модулей ([SAD.docx](https://docs.google.com/document/d/111xonRdFK0ZgoxNtOQXzRB00LC9Ezs1f/edit) п. 5.1) и стратегии развертывания ([SAD.docx](https://docs.google.com/document/d/111xonRdFK0ZgoxNtOQXzRB00LC9Ezs1f/edit) п.7) объекты тестирования включают:

1. Модуль Service для фронтенда - обеспечение связи со сторонними сервисами
2. Модуль Service Middleware для бекенда - обеспечение связи со сторонними сервисами
3. Каждый реактивный компонент фронтенд приложения - корректность бизнес-логики и визуализации для поддержки удобства использования
4. Модуль ContentManager - корректность логики работы с ФС и кешированием, надежность
5. Модуль Domain - корректность поддержания инвариантов бизнес-логики для каждого отдельного юзкейса бизнес-процесса
6. Модуль Routing - правильность реализации заявленной REST спецификации
7. Модуль DB и его подмодули - стабильность и консистентность БД, корректность запросов и надежность

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

| №  пункта | Цель теста | Компонент системы | Вид теста |
| --- | --- | --- | --- |
| 5.1 | Корректность первичных ключей | База данных | Дымовой тест, автоматический |
| 5.1 | Проверка not-null ограничений | База данных | Дымовой тест, автоматический |
| 5.1 | Проверка стратегии удаления сущностей | База данных | Дымовой тест, автоматический |
| 5.1 | Проверка уникальности определенных полей | База данных | Дымовой тест, автоматический |
| 5.2 | Проверка инвариантов при обработке основного бизнес-цикла системы | Бекенд | Регрессионный, автоматический |
| 5.3 | Проверка основного бизнес-цикла системы с т.з. пользователя | Фронтенд/Бекенд | Интеграционный, автоматический |
| 5.4 | Проверка корректности и удобства UI | Фронтенд | Ручной |
| 5.5 | Проверка производительности системы | Бекенд | Автоматический |
| 5.6 | Проверка ожидаемой нагрузки системы | Бекенд | Автоматический |
| 5.7 | Проверка максимальной нагрузки системы | Бекенд | Автоматический |
| 5.9 | Защищенность системы в отношении распространенных уязвимостей | Фронтенд/Бекенд | Интеграционный, ручной/автоматический |

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

*5.1 Data and Database Integrity Testing (Тестирование базы данных)*

Схема БД

Для тестирования БД будет использован dump, представляющий собой скрипт для инициализации основной БД. Тестировщик использует данный dump для воссоздания развернутой у разработчиков БД. Dump предоставляется после развертывания конфигурации основной БД.

Так как тестирование всех таблиц БД нецелесообразно ввиду количества этих таблиц, было решено выделить несколько обязательных для тестирования таблиц которые будут проверены вручную. Это сделано для проверки сохранения инвариантов предметной области, которые будут аналогичным образом описаны для остальных сущностей.

Список таблиц подвергаемых тестированию:

1. platform - таблица, содержащая данные платформы перемещающейся по этажам
2. platform\_user - содержит информацию о пользователях с их идентификационными данными
3. platform\_prisoner - составная сущность, зависящая от таблиц platform и prisoner, содержит их связь, а также связь с соседом по этажу

Предполагается, что тестирование будет проводиться с помощью фреймворка PGUnit

| Technique Objective:  (Цель) | Проверка схемы БД на корректную установку ограничений с использованием первичных ключей (PK) для сохранения уникальности записей |
| --- | --- |
| Technique:  (Описание процесса) | 1. Описание приводится для таблицы platform\_user, работа с остальными таблицами производится аналогично 2. Ознакомиться с параметрами подключения развернутой базе данных системы “Платформа” (настройки подключения находятся в конфигурационном файле) 3. При помощи утилиты psql/Data Grip произвести подключение к базе данных 4. Произвести выборку списка пользователей системы (их идентификаторов)  * При отсутствии идентификаторов пользователей — создать нового пользователя с параметрами (username = “db\_test\_user”, password = “password”, activated = “true”).  1. Выбрать один из отображенных идентификаторов (либо созданный на предыдущем шаге) 2. Создать нового пользователя с id, присутствующем в системе 3. Убедиться в выводе ошибки уникальности поля 4. Произвести запрос на получение данных пользователей 5. Убедиться в отсутствии изменений в данных пользователей |
| Oracles:  (Источники) | 1. Документ: [ТЗ.docx](https://docs.google.com/document/d/1AIwE7RCRblPbgkYCq8PjF88EQ6K5zA8K/edit) |
| Required Tools:  (Инструменты) | Утилиты psql/Data Grip текстовый редактор (для чтения конфигурационного файла) |
| Success Criteria:  (Критерий успеха) | Схема БД имеет ожидаемые первичные ключи |

| Technique Objective:  (Цель) | Проверка схемы БД на корректную установку ограничений not-null полей, обеспечение null-safety при работе с БД в языке Kotlin |
| --- | --- |
| Technique:  (Описание процесса) | 1. Описание приводится для таблицы platform, работа с остальными таблицами производится аналогично 2. Ознакомиться с параметрами подключения развернутой базе данных системы “Платформа” (настройки подключения находятся в конфигурационном файле) 3. При помощи утилит psql/Data Grip произвести подключение к базе данных 4. Создать новую платформу с id, отсутствующем в системе; code указать как NULL; code = 1; задать name = ‘Unnamed’; description = ‘123’; floor\_amount; created\_at = ‘1990-01-01’; updated\_at = ‘1990-01-01’ 5. Убедиться в появлении ошибки нарушения not-null ограничений 6. Произвести запрос на получение данных пользователей 7. Убедиться в отсутствии пользователя с login = NULL |
| Oracles:  (Источники) | 1. Документ: [ТЗ.docx](https://docs.google.com/document/d/1AIwE7RCRblPbgkYCq8PjF88EQ6K5zA8K/edit) |
| Required Tools:  (Инструменты) | Утилиты psql/Data Grip, текстовый редактор (для чтения конфигурационного файла) |
| Success Criteria:  (Критерий успеха) | Схема БД имеет ожидаемые ограничения на ненулевые поля |

| Technique Objective:  (Цель) | Проверка схемы БД на корректную связь между сущностями и конфигурацию удаления таких сущностей |
| --- | --- |
| Technique:  (Описание процесса) | 1. Описание приводится для таблицы platform\_prisoner, работа с остальными таблицами производится аналогично 2. Ознакомиться с параметрами подключения развернутой базе данных системы “Платформа” (настройки подключения находятся в конфигурационном файле) 3. При помощи утилит psql/pgadmin произвести подключение к базе данных 4. Выбрать заключенного из системы с activated = 0 (при отсутствии создать в platform\_prisoner). 5. Удалить пользователя из platform\_prisoner, с которым имеется связь в сущности prisoner 6. Убедиться в наличии пользователя в системе после удаления 7. Произвести выборку заключенного из platform\_prisoner 8. Убедиться в наличии пользователя с id удаленного пользователя |
| Oracles:  (Источники) | 1. Документ: [ТЗ.docx](https://docs.google.com/document/d/1AIwE7RCRblPbgkYCq8PjF88EQ6K5zA8K/edit) |
| Required Tools:  (Инструменты) | Утилиты psql/Data Grip, текстовый редактор (для чтения конфигурационного файла) |
| Success Criteria:  (Критерий успеха) | Схема БД имеет ожидаемые связи полей таблиц |

*5.2 Function Testing (Функциональное тестирование)*

| Technique Objective:  (Цель) | Обеспечение корректности моделирования бизнес-цикла, сохранение инвариантов, полная и точная реализация требований |
| --- | --- |
| Technique:  (Описание процесса) | 1. Использовать для подробностей документы [SRS.docx](https://docs.google.com/document/d/1FPEVQYix8eWcn4zHgGHRQIlBTKPg9gIR/edit?usp=sharing&ouid=103372480510558009952&rtpof=true&sd=true) [ТЗ.docx](https://docs.google.com/document/d/1AIwE7RCRblPbgkYCq8PjF88EQ6K5zA8K/edit) [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) 2. Создать заключенного в системе и залогиниться 3. Пройти основной поток юзкейса InmateFloorAllocation 4. Убедиться, что платформа распределила заключенных согласно рейтингу |
| Oracles:  (Источники) | 1. Документы: [ТЗ.docx](https://docs.google.com/document/d/1AIwE7RCRblPbgkYCq8PjF88EQ6K5zA8K/edit) [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) 2. Юзкейсы: InmateFloorAllocation |
| Required Tools:  (Инструменты) | Junit, Mockito, TestContainers |
| Success Criteria:  (Критерий успеха) | Все инварианты, описанные в процессе, выполнены. Система допускает многократное исполнение описанного процесса |

*5.3 Business Cycle Testing (Тестирование бизнес-цикла)*

| Technique Objective:  (Цель) | Проверка основного бизнес-цикла моделируемого в системе |
| --- | --- |
| Technique:  (Описание процесса) | 1. Следующие пункты выполнить с помощью соответствующих средств тестирования Selenium WebDriver. Необходимые действия осуществлять посредством симуляции взаимодействия с соответствующими UI элементами системы 2. Войти в систему под ролью Аналитик, заполнить форму о действиях заключенного.   Возможные нарушения: Убийство/Покушение заключенного/сотрудника, Попытка сбежать.   1. Обновить балльно рейтинговую систему: за каждое нарушение у заключенного отнимаются баллы. 2. Проверить баллы заключенных, убедиться, что за Администратора балльную систему тоже можно проверить и там будут указываться те же баллы. |
| Oracles:  (Источники) | 1. Документы: [ТЗ.docx](https://docs.google.com/document/d/1AIwE7RCRblPbgkYCq8PjF88EQ6K5zA8K/edit), [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) 2. Юзкейсы: InmatePointsManagement |
| Required Tools:  (Инструменты) | Как минимум два устройства для взаимодействия с системой, Selenium WebDriver, JRE, gradle, браузер Chrome |
| Success Criteria:  (Критерий успеха) | Все действия с системой завершены с ожидаемым результатом. Система допускает многократное исполнение описанного процесса |

*5.4 User Interface Testing (Тестирование интерфейса)*

| Technique Objective:  (Цель) | Направлен на проверку корректности работы и удобства использования пользовательского интерфейса приложением пользователями |
| --- | --- |
| Technique:  (Описание процесса) | 1. Тестирование полей, проверка типов данных (страница входа, регистрации и личный кабинет) 2. Тестирование навигации по приложению 3. Тестирование отображения различного функционала за счет смены ролей 4. Тестирование возможности заключенного выбрать предпочтения при регистрации 5. Тестирование возможности Повара просматривать заказы 6. Тестирование возможности Аналитика изменять баллы заключенных 7. Тестирование возможности Администратора изменять этажи и баллы заключенных 8. Тестирование отображения подсказок и ошибок пользователю |
| Oracles:  (Источники) | usecase - источник, конкретно описание теста   1. RegisterInmates, EmployeeRegistrationModification: проверить поля в форме регистрации и логина, попробовать зарегистрировать пользователя, ввести несуществующие значения. 2. InmatePointsManagement: попробовать добавить отрицательные значения баллов 3. InmateFloorAllocation - попробовать поместить заключенного на несуществующий этаж 4. MenuCreation, DisplayMenu: войти под ролью повара и составить меню 5. ProductOrder: под ролью Повара заказать недостающие продукты 6. PlatfotmManagement: попробовать переместить платформу на несущствующий этаж 7. PrisonerFoodChoice: выбрать еду под ролью заключенного 8. EmployeeAuthentication: авторизоваться под разными ролями и просмотреть соблюдение ограничений |
| Required Tools:  (Инструменты) | Устройство с веб-браузером Jest, Gatling |
| Success Criteria:  (Критерий успеха) | Все необходимые тесты 1-8, покрывающие 90% модулей прошли успешно |

*5.5 Performance Profiling (Тестирование производительности)*

| Technique Objective:  (Цель) | Проверка доступности системы и корректности ответов при одновременной обработке множества запросов (до 100) |
| --- | --- |
| Technique:  (Описание процесса) | 1. Произвести авторизацию в системе под ролью Сотрудник 2. Получить JWT токен сотрудника 3. Создать Thread group в JMeter 4. Добавить в него HTTP Request (с адресом http://host[:port] /v2/menu при запуске на локальном устройстве), HTTP Header Manager (с Authorization равным Bearer + “ ” + JWT токен клиента) 5. Сделать дубликат данной группы 100 раз 6. Добавить “View Results Tree” 7. Перейти в созданный “View Results Tree” 8. Запустить тест 9. Убедиться в успешности выполнения HTTP запроса 10. Сравнить ответы запроса между разными запросами всех групп 11. Убедиться в том, что они одинаковы |
| Oracles:  (Источники) | 1. Документы: [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) 2. Юзкейс: EmployeeRegistrationModification |
| Required Tools:  (Инструменты) | среда выполнения Java, JMeter, DataGrid, Postgres |
| Success Criteria:  (Критерий успеха) | Все 100 запросов завершились удачно (код 200), ответы всех запросов идентичны |

*5.6 Load Testing (Нагрузочное тестирование)*

| Technique Objective:  (Цель) | Тест проводится для оценки производительности системы с нагрузкой близкой к максимально ожидаемой при регистрации заключенного |
| --- | --- |
| Technique:  (Описание процесса) | 1. Произвести авторизацию в системе под ролью Сотрудника 2. Получить JWT токен клиента 3. Создать Thread group в JMeter 4. Добавить в него HTTP Request (с адресом http://host[:port]/api/v1/prisoners при запуске на локальном устройстве), HTTP Header Manager (с Authorization равным Bearer + “ ” + JWT токен клиента). 5. Установить body:   {  "lastName": "string",  "firstName": "string",  "patronymic": "string",  "passport": "31848023743",  "weight": 180,  "birthDate": "2024-10-02",  "favoriteDish": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  }   1. Параметр clientId установить id клиента под данным JWT токеном 2. Установить количество “Number of threads” - 1000. 3. Установить “Timeouts-Response” = 5000. 4. Установить “Delay” = 1000 5. Добавить “View Results Tree” 6. Перейти в созданный “View Results Tree” 7. Запустить тест 8. Убедиться в выполнении запросов (статус каждого запроса - 200 OK) 9. Убедиться в создании записей в БД |
| Oracles:  (Источники) | 1. Документы: [Vision](https://docs.google.com/document/u/0/d/13XSo_Pde4Gs0nj5iUzKhtkMzz_gTnoI28WtpLZDGBCE/edit) [ТЗ](https://docs.google.com/document/u/0/d/1_SJQVGrcHqZKLjQt1o7OPzTX9pelvxpD5-HbUzCS2Rs/edit) 2. Юзкейс: PrisonerRegistration |
| Required Tools:  (Инструменты) | среда выполнения Java, JMeter, DataGrip, Postgres |
| Success Criteria:  (Критерий успеха) | Максимальная задержка ответа не превышает 5 секунд  Среднее время выполнения запроса = 1 секунда |

*5.7 Stress Testing (Стрессовое тестирование)*

| Technique Objective:  (Цель) | Тест проводится для оценки производительности системы с избыточной нагрузкой при регистрации заключенного |
| --- | --- |
| Technique:  (Описание процесса) | 1. Произвести авторизацию в системе под ролью Client 2. Получить JWT токен клиента 3. Создать Thread group в JMeter 4. Добавить в него HTTP Request (с адресом http://host[:port]/api/v1/prisoners при запуске на локальном устройстве), HTTP Header Manager (с Authorization равным Bearer + “ ” + JWT токен клиента). 5. Установить body: {   {  "lastName": "string",  "firstName": "string",  "patronymic": "string",  "passport": "31848023743",  "weight": 180,  "birthDate": "2024-10-02",  "favoriteDish": "3fa85f64-5717-4562-b3fc-2c963f66afa6"  }   1. Параметр clientId установить id клиента под данным JWT токеном 2. Установить количество “Number of threads” - 50000. 3. Установить “Timeouts-Response” = 5000. 4. Добавить “View Results Tree” 5. Перейти в созданный “View Results Tree” 6. Запустить тест 7. Убедиться в частичном выполнении запросов (часть запросов имеют код 200 OK, остальные отличный = 4xx) 8. Убедиться в создании записей в БД |
| Oracles:  (Источники) | 1. Документы: [Vision](https://docs.google.com/document/u/0/d/13XSo_Pde4Gs0nj5iUzKhtkMzz_gTnoI28WtpLZDGBCE/edit) [ТЗ](https://docs.google.com/document/u/0/d/1_SJQVGrcHqZKLjQt1o7OPzTX9pelvxpD5-HbUzCS2Rs/edit) 2. Юзкейс: PrisonerRegistration |
| Required Tools:  (Инструменты) | среда выполнения Java, JMeter, DataGrid, Postres |
| Success Criteria:  (Критерий успеха) | Backend функционирует  Часть запросов обработалась корректно (до 10%), остальная часть запросов не обработалась системой  Максимальная задержка ответа превышает 5 секунд |

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

| Technique Objective:  (Цель) | Проверка системы на корректное разграничение привилегий и прав доступа, тестирование системы на наличие уязвимостей |
| --- | --- |
| Technique:  (Описание процесса) | 1. Изучить доступную функциональность каждой роли в системе 2. Попытаться изменить роль пользователя в ручном режиме с помощью Burp Suite (повышение привилегий) 3. Запустить автоматические скрипты Burp Suite на поиск уязвимостей 4. Запустить автоматические скрипты dirsearch 5. Запустить автоматические скрипты sqlmap 6. Убедиться, что уязвимости не найдены 7. Запустить статический анализатор кода find-sec-bugs для приложения бекенда 8. При возникновении уязвимостей использовать стандартный порядок создания задач об исправлениях и доработках (подробнее в п. 4.3 [SDP.docx](https://docs.google.com/document/d/1_FXvPMaqVHpfFZGASbB3kRz8hdWsuA_O/edit?rtpof=true))   Примеры уязвимостей:   * Privilege Escalation – Изменение роли пользователя без проверки прав доступа. * SQL Injection – Выполнение произвольных SQL-запросов через пользовательский ввод. * Directory Listing – Доступ к конфиденциальным файлам через незащищенные директории. * CORS Misconfiguration – Разрешение кросс-доменных запросов для всех доменов. * Sensitive Data Exposure – Передача данных по незащищенному протоколу (HTTP). * Insufficient Data Validation – Валидация данных только на клиентской стороне без проверки на сервере. * Cross-Site Scripting (XSS) – Внедрение и выполнение вредоносного JavaScript-кода на странице. * Broken Authentication/Authorization – Выполнение операций без проверки прав доступа. * Vulnerabilities Detected by find-sec-bugs – Использование небезопасных методов при работе с пользовательскими данными или файлами. |
| Oracles:  (Источники) | EmployeeAuthentication |
| Required Tools:  (Инструменты) | Burp Suite (ручное и автоматическое тестирование), браузер Chrome, dirsearch, sqlmap, find-sec-bugs |
| Success Criteria:  (Критерий успеха) | Автоматические тесты не выявят ни одной уязвимости, все типы ролей будут идентифицироваться корректно. Проблемы кода выявленные статическим анализом устранены |

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

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

Тестирование может проводиться при готовности соответствующих компонентов системы, а именно:

1. Тестирование БД может производиться когда запущена и настроена СУБД, создана БД
2. Тестирование логики системы может производиться когда отражены все юзкейсы основного бизнес-процесса
3. Созданы соответствующие компоненты UI системы

Таким образом, критерии старта:

1. Корректно установлена СУБД
2. Создана схема БД
3. На бекенде реализована логика юзкейсов (юзкейсы перечислены в графе Oracles п. 5.3)
4. Реализованы соответствующие юзкейсам UI компоненты (см [Usecase.docx](https://docs.google.com/document/d/1WsK6MgHvOmjKYwUbCVXd2ZMJJeIUCtnk/edit) и [SAD.docx](https://docs.google.com/document/d/111xonRdFK0ZgoxNtOQXzRB00LC9Ezs1f/edit?rtpof=true))

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

Критерий окончания тестирования:

1. Все критерии успеха из таблиц в пункте 5 удовлетворены
2. Либо, появляются признаки рисков из пунктов 2.7, 2.9 и 2.12, описанные в [RiskList.docx](https://docs.google.com/document/d/1wFBYTW8Hgl7Q1R_LKRvdZGVd30RoQ7UR/edit?rtpof=true) при условии окончания сроков разработки. Таким образом, происходит концентрация сил на стратегию устранения риска и тестирование прекращается

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

Тестирование может быть приостановлено в рамках стратегии смягчения для рисков 2.7, 2.9 и 2.12 из [RiskList.docx](https://docs.google.com/document/d/1wFBYTW8Hgl7Q1R_LKRvdZGVd30RoQ7UR/edit?rtpof=true) при появлении соответствующих признаков. Возобновление тестирование необходимо начать как только соответствующие риски смягчены.

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

Некоторая часть тестов может быть представлена в виде исходного кода либо автоматизированной конфигурации. Таким образом, результаты тестирования включают:

1. Скрипты SQL для проверки базы данных
2. Тесты основного бизнес цикла в виде исходного кода на Java с необходимыми assert предикатами
3. Тесты производительности в виде сценариев JMeter
4. Тесты UI в виде исходного кода для Selenium, и unit тестов на JavaScript
5. Скрипты для Burp Suite

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

Так как часть тестов представлена в формате исходного кода, результаты выполнения тестов включают текстовые данные ожидаемого вывода. Результаты выполнения для каждой категории тестов:

1. Ожидаемые результаты запуска скриптов SQL в текстовом виде
2. Результаты вывода при запуске тестов бизнес цикла в текстовом виде
3. Скриншоты выполнения сценариев JMeter в виде графиков и параметров конфигурации
4. Скриншоты для подтверждения usability свойств интерфейса и ожидаемые результаты исполнения unit тестов
5. Скриншоты ожидаемых результатов тестирования безопасности
6. Покрытие тестами для функц тестирования

Покрытие тестами для функционального тестирования будет включать следующие аспекты:

| **Функциональность** | **Тип тестирования** | **Критерии покрытия** |
| --- | --- | --- |
| Авторизация и аутентификация | Функциональные тесты | Проверка успешного входа с правильными учетными данными и неуспешного с неправильными. |
| CRUD операции (Создание, Чтение, Обновление, Удаление) | Функциональные тесты | Проверка корректности выполнения всех CRUD операций для объектов (например, пользователей, записей). |
| Валидация ввода | Функциональные тесты | Проверка валидации на уровне клиентского и серверного приложений (например, обязательные поля, форматы данных). |
| Бизнес-логика | Бизнес циклы и unit тесты | Проверка корректности выполнения бизнес-логики в соответствии с требованиями и сценариями использования. |
| Производительность | Нагрузочные тесты | Оценка времени отклика и стабильности системы при увеличении нагрузки (например, количество одновременно активных пользователей). |
| Безопасность | Тесты безопасности | Проверка на наличие уязвимостей и оценка защищенности приложения (например, SQL Injection, XSS, CSRF). |
| Usability интерфейса | Usability тестирование | Оценка удобства интерфейса и удобства навигации для конечного пользователя. |

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

Отчет об оценке качества полученной системы содержит:

1. Процент тестового покрытия каждого подпункта для раздела 5 (отношение удовлетворенных пунктов раздела Technique ко всему количеству пунктов)
2. Процент разницы между ожидаемыми свойствами быстродействия системы и реальными (отношения количества запросов, одновременных пользователей и т.д.)

Отчет предоставляется в свободной форме.

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

Методы отслеживания ошибок:

1. Конфигурация логирования через Logback. В конфигурационном файле logback.xml будут настроено логирование всех сообщений уровня ERROR и выше.
2. Использования специальных log таблиц в базе данных для хранения информации об ошибках.

Ошибки, записанные в лог-файлы и таблицы базы данных, будут регулярно анализироваться командой разработчиков и тестировщиков. По каждой выявленной ошибке, обнаруженной через логи или тестирование, будет создан issue в репозитории на GitHub с обязательным указанием:

* Описания ошибки.
* Шагов для её воспроизведения.
* Приоритетности.
* Связанной роли пользователя и сценария.

**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)** |
| --- | --- | --- |
| Сервер Helios | 1 | Сервер под управлением FreeBSD, основной сервер заказчика |
| ПК | 2 | ПК с поддержкой стека технологий, описанного в [SRS.docx](https://docs.google.com/document/d/1FPEVQYix8eWcn4zHgGHRQIlBTKPg9gIR/edit)п 3.5 |

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

| **Software Element Name**  **(Название)** | **Version**  **(Версия)** | **Type**  **(Тип)** |
| --- | --- | --- |
| JRE | 17 | Среда для запуска Java приложений |
| Google Chrome | Chromium 112 | Веб-браузер |
| Postgre | 15 | База данных |
| DataGrip | 23 | Инструмент для работы с базами данных |
| Selenium | 4 | Инструмент для UI тестирования |
| JMeter | 5.6 | Инструмент для профилирования |
| Burp Suite | 2020 | Инструмент для тестирования безопасности ПО |
| Gradle | 8.0 | Система автоматизации сборки |
| Jest | 29.4. | Фреймворк для тестирования JavaScript |
| Gatling | 3.9 | Инструмент для тестирования производительности |
| dirsearch | 0.4 | Инструмент для поиска скрытых директорий |
| sqlmap | 1.6 | Инструмент для тестирования SQL-инъекций |
| find-sec-bugs | 4.4 | Статический анализатор безопасности кода |

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

| **Tool Category or Type**  **(Тип программы)** | **Tool Brand Name**  **(Название)** | **Vendor**  **(Производитель)** | **Version**  **(Версия)** |
| --- | --- | --- | --- |
| Среда разработки | Intellij Idea | JetBrains | 2023.3.2 |
| Система контроля версий | Git | Open Source Community | 2.43 |
| Веб-браузер | Google Chrome | Google | 119 |

**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 Responsbilities**  **(Обязанности)** |
| --- | --- | --- |
| DevOps | 1 | * Настроить окружение; * Управлять процессами CI/CD; * Обеспечить стабильность и производительность приложения; * Мониторинг и логирование систем. |
| Тестировщик | 1 | * Протестировать систему на уязвимости * Провести тестирование * Сформировать отчет * Анализ лог-файлов и log-таблиц |
| Проектировщик БД | 1 | * Настроить СУБД * Сформировать схему БД * Сформировать набор тестов * Описать ожидаемый вывод |
| Фронтенд/Бекенд разработчик | 1 | * Описать необходимые юзкейсы * Описать соответствующие тесты * Сформировать ожидаемый вывод * В случае выявления ошибок — внести правки в систему |
| Системный аналитик/Менеджер проекта | 1 | * Контроль проведения процесса тестирования * Отражение текущих задач * Консультации относительно корректности тестов относительно предметной области * Переработка описания теста при изменении требований к системе * Управление данным документом |

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

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

При готовности соответствующих тестов и отчетах об их выполнении в соответствии с п. 4.3.4 и 4.3.4 документа [SDP.docx](https://docs.google.com/document/d/1_FXvPMaqVHpfFZGASbB3kRz8hdWsuA_O/edit?rtpof=true) будет проводиться плановая проверка тестового покрытия. Такая проверка будет также штатно отражаться в виде задачи в Trello с перечислением готовых и проверенных подпунктов.

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

При выполнении тестирования возможны следующие общие проблемы:

1. Недоступность необходимого оборудования заказчика
2. Недоступность необходимого ПО
3. Некорректное описание теста
4. Ошибки системы

Для решения таких проблем будут проводиться следующие соответствующие мероприятия:

1. Перенос процесса тестирования на собственное оборудование
2. Поиск аналогов необходимого ПО, желательно открытых и бесплатных (с учетом стратегии лицензирования из [SRS.docx](https://docs.google.com/document/d/1FPEVQYix8eWcn4zHgGHRQIlBTKPg9gIR/edit))
3. Выявление некорректных аспектов теста и их устранение с привлечением менеджера проекта
4. Создание соответствующих задач в Trello с описанием ошибки и назначением ответственного лица (описание процесса в 7.3)

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

Данный план тестирования утверждается менеджером проекта при следующих критериях:

1. Заполнены все необходимые пункты раздела 5
2. Заполнен раздел 4, включающий обзор всего плана
3. Все тесты имеют четко описанные цели, удовлетворяющие пункт 2.2