

# **ТЕСТ ПЛАН**

## **Система управления командировками «B-trip»**

### **1. ВВЕДЕНИЕ**

#### **1.1 Описание**

Описание: Система управления командировками предназначена для автоматизации процессов, связанных с организацией командировок внутри компании 'МТС', включая подачу заявок, утверждение, бухгалтерский учет и отчетность.

#### **1.2 Цель тестирования**

Обеспечить стабильность работы системы, ее соответствие функциональным требованиям и удобство использования для всех типов пользователей.

### **2. ПЛАН РАБОТЫ**

#### **2.1 Разработка и согласование тест-плана**

Определение областей тестирования, включая функциональное, интеграционное, и API тестирование, а также ручное и автоматизированное тестирование UI. Установление приоритетов тестирования и оценка ресурсов. Взаимодействие с разработчиками и другими заинтересованными сторонами для утверждения плана тестирования.

#### **2.2 Разработка чек-листов и сценариев тестирования**

Создание детальных чек-листов и тестовых сценариев для проверки ключевых функций и процессов системы.

#### **2.3 Выполнение тестирования**

Проведение тестирования согласно разработанным чек-листам и тестовым сценариям, включая юнит-тестирование, API тестирование и ручное/автоматизированное тестирование UI.

## **2.4 Документирование и анализ результатов**

Фиксация обнаруженных дефектов и проблем, анализ результатов тестирования.

## **2.5 Подготовка итогового отчета о тестировании**

Составление отчета с подробным описанием проведенного тестирования, обнаруженных дефектов и рекомендаций по их устранению.

# **3. ПЛАН И СТРАТЕГИЯ ТЕСТИРОВАНИЯ**

## **3.1 Процедура тестирования**

Тестирование включает проверку следующих ключевых функций и процессов:

- Переходы между статусами заявок ("not approved", "approved by master", "approved by accountant", "report-waiting", "archive") и их соответствие бизнес-логике приложения.
- Авторизацию пользователей.
- Взаимодействие с интерфейсами и функциональностью, предоставляемой модулями системы: Сотрудник, Руководитель, Бухгалтер.
- Обработку и управление заявками на командировку и соответствующими отчетами.
- Расчет и выплату авансов, обработку отчетности.
- Интеграцию микро-сервисов и передачу данных между ними.
- Отправку и получение уведомлений пользователями.

Тестирование включает проверку работы системы, соответствие функциональных требований и пользовательского опыта.

Основные типы тестирования, которые будут выполнены:

- Юнит-тестирование, проводимое разработчиками, для проверки корректности работы отдельных компонентов системы.
- API тестирование с использованием Postman для проверки интерфейсов микро-сервисов.
- API тестирование с помощью python+requests для автоматизации тестирования и интеграции с системами непрерывной интеграции.

- Ручное тестирование пользовательского интерфейса для проверки соответствия интерфейса требованиям юзабилити и дизайна.
- Автоматизированное тестирование интерфейса (если будет реализовано) для проверки поведения пользовательских сценариев в разных условиях и на различных устройствах.

### **3.2 Область тестирования**

Цель тестирования: убедиться в том, что система работает стабильно и соответствует всем требованиям, как функциональным, так и техническим.

Тестирование покрывает проверку интеграционных и функциональных аспектов работы системы, в том числе:

- Авторизацию пользователей.
- Взаимодействие с модулями системы: Сотрудник, Руководитель, Бухгалтер.
- Обработку и управление заявками на командировку.
- Расчет и выплату авансов, обработку отчетности.
- Интеграцию модулей системы и передачу данных между ними.
- Работоспособность системы на различных браузерах и операционных системах.
- Отpravку и получение уведомлений пользователями.

Входные критерии:

- Документация по требованиям завершена
- Продукт готов к тестированию
- Тестовые данные и окружение подготовлены

Критерии приемки:

- Тест-план и тестовая документация готовы
- Все чек-листы пройдены успешно
- Все smoke тесты пройдены успешно
- Все баги с высоким приоритетом исправлены
- Ручные и автоматизированные тесты пройдены

### 3.3 Сообщения об ошибках

Отчеты об ошибках создаются для того, чтобы предоставить команде разработчиков и руководителю проекта исчерпывающую информацию об обнаруженных ошибках. Они должны быть полезны при определении причин ошибок и их исправлении.

Степень серьезности ошибок можно разделить на пять категорий:

- S1 - Блокирующий (Blocker) – дефект полностью блокирует выполнение функционала, нет никакого способа его обойти
- S2 – Критический (Critical) -дефект блокирует часть функциональности, но есть альтернативный путь для его обхода
- S3 – Значительный (Major) – дефект, указывающий на некорректную работу части функциональности. Зачастую связан не с тем, что функция не работает, а с тем, что она работает неправильно
- S4 – Незначительный (Minor) – дефект, не относящийся к функциональности системы
- S5 – Тривиальный (Trivial) – дефект, не затрагивающий функциональность системы, а также оказывающий минимальное влияние на общее качество системы

Приоритет:

- High (Высокий) - ошибка, которую необходимо исправить в первую очередь. Это обычно касается ошибок, которые серьезно влияют на функциональность приложения или могут вызвать большие проблемы для пользователя.
- Medium (Средний) - ошибка, которую следует исправить, но она не так критична, как ошибки с высоким приоритетом. Обычно такие ошибки не блокируют основные функции приложения.
- Low (Низкий) - ошибки с низким приоритетом обычно не влияют на работоспособность приложения и могут быть исправлены в последнюю очередь.

Информация, указанная в каждом отчете об ошибке:

- ID
- Название баг-репорта

- Краткое описание, представляющее собой краткое описание проблемы

- Предисловия
- Шаги для воспроизведения ошибки
- Ожидаемый результат
- Фактический результат
- Дополнительная информация о дефекте в виде прикрепленных скриншотов/видеозаписей
- Окружение (ОС, наименование и версия браузера)
- Приоритет ошибки (High, Medium, Low)
- Степень серьезности ошибок

## 4. РЕСУРСЫ

### 4.1 Команда

- **Backend Разработчики:** [Шалико Салимов](#), [Давид Черных](#)
- **Системный аналитик:** [Глеб Новиков](#)
- **QA:** [Станислав Осипов](#)

### 4.2 Инструменты

Будут использоваться следующие инструменты:

Наименование	Инструмент
Чек-листы	Google docs
Баг-репорты	Google docs
Автоматизация API & UI	Postman, Selenium, pytest, request

### 4.3 Список браузеров

Наименование браузера	Версия браузера
Chrome	Версия 123.0.6312.106 (Официальная сборка), (64 бит)
MS Edge	

## Критерии качества

- Продукт должен работать в соответствии с требованиями и техническим заданием.
- Продукт не должен содержать критических и блокирующих дефектов в окончательной версии продукта.

## Риски процесса тестирования

Следующие проблемы могут повлиять на результаты тестирования:

- Изменения и модификации программного продукта, которые не были запланированы и не обсуждались заранее с командой тестирования
- Изменения в требованиях к программному обеспечению, которые не были предварительно обсуждены с командой тестирования
- Задержки в исправлении ошибок
- Обязанности участников тестовой группы

## Тестирующий:

- Сбор и изучение требований
- Проверка документации
- Планирование работ по тестированию
- Контроль за проведением испытаний, чтобы работы выполнялись в соответствии с планом
- Отчеты о ходе выполнения, количестве и серьезности обнаруженных ошибок
- Процесс контроля качества, регистрация обнаруженных ошибок в утвержденной системе отслеживания ошибок