Rocket.Chat

Тестовая стратегия

Версия 1.0

История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 18/07/2020 | 1.0 | Первоначальная версия тестовой стратегии | Гарахмедова Е.Д. |

Оглавление

[1. Введение 4](#_Toc141709422)

[2. Типы проводимого тестирования 4](#_Toc141709423)

[2.1 Функциональное тестирование 4](#_Toc141709424)

[2.2 UI-тестирование 4](#_Toc141709425)

[2.3 Тестирование удобства использования 4](#_Toc141709426)

[2.4 Тестирование производительности 5](#_Toc141709427)

[3. Части системы, которые будут протестированы 5](#_Toc141709428)

[4. Окружение для работы 5](#_Toc141709429)

[5. Виды тестовой документации. Какие техники тест-дизайна будут использоваться при формировании тест-кейсов. 5](#_Toc141709430)

[6. Время проведения тестирования. 6](#_Toc141709431)

[Когда тестирование можно будет считать завершённым? 7](#_Toc141709432)

Тестовая стратегия

# Введение

Стратегия составлена для тестирования последней веб-версии Rocket.Chat.

**Rocket.Chat** - мессенджер для общения команд с открытым исходном кодом. Компания «Acoola» может установить Rocket.Chat на собственные сервера или использовать облачную версию. В мессенджере пользователи могут создавать каналы и группы, обмениваться файлами. Использовать Rocket.Chat можно в браузере, десктопном приложении и на мобильных устройствах.

**Особенности Rocket.Chat.**

Одна из основных особенностей мессенджера - безопасность, разработчики не хранят данные пользователей на своих серверах. В личных и групповых чатах можно подключить сквозное шифрование, а также проводить аудит безопасности чатов. Остальные особенности:

* Общение в каналах, групповых и личных чатах.
* Обеспечение защиты данных.
* Интеграция с облачными сервисами.
* Передача файлов.
* Персонализация мессенджера.
* Размещение на выделенных серверах.

##### Тестирование проекта будет проводить студент Школы21 Гарахмедова Екатерина Дмитриевна под логином «maidamar». В рамках обучения ей удалось поработать со следующими проектами:

##### TestIT

* Sbermarket
* Сберстрахование
* СберМегаМаркет
* Swagger(Fake REST API)
* Tutu.ru
* SQL

# Типы проводимого тестирования

## Функциональное тестирование

Проверка работоспособности и соответствия требованиям всех функций приложения.

## UI-тестирование

Тестирования интерфейса заключается в том, что программное обеспечение проверяется с точки зрения пользователя, чтобы итоговый продукт был удобным, привлекательным и понятным.

## Тестирование удобства использования

Проверка удобства и интуитивности интерфейса приложения, его эргономики и функциональности.

## Тестирование производительности

Проверка скорости работы приложения, потребления ресурсов устройства и его стабильности в условиях нагрузки.

# Части системы, которые будут протестированы

* Создание каналов, команд, личных переписок, обсуждений;
* Переход по ссылкам на главной (Home) странице: «Создать канал», «Join rooms», «Mobile apps», «Desktop apps», «Документация».
* Присоединение к существующим каналам(каталог);
* Работа самого чата, загрузка файлов, отправка текстовых и голосовых сообщений, создание тредов, добавление реакций, цитирование;
* Просмотр и редактирование профиля, статуса;
* Работа с участниками и уведомлениями;
* Создание звонков;
* Настройка веб-приложения: «Общие», «Присутствие пользователя», «Звуковые оповещения», «Экспорт и импорт данных», «Безопасность»;
* Панель администрирования.

# Окружение для работы

1. Операционная система macOS Monterey Версия 12.6.7
2. Браузеры:

* Safari Версия 16.5.1 (17615.2.9.11.7, 17615);
* Google Chrome Версия 114.0.5735.198 (Официальная сборка), (x86\_64).

1. TestIT
2. Microsoft Word для Mac Версия 16.75(23070901)

# Виды тестовой документации. Какие техники тест-дизайна будут использоваться при формировании тест-кейсов.

В процессе тестирования были составлены следующие виды тестовой документации:

* **Отчет о тестировании** пишется, когда функционал уж проверен и релиз либо предрелиз показывает итог проделанной работы.
* **Сценарии использования** (Use Case) - описывают в каком контексте должно проводиться каждое действие пользователя или администратора при работе с веб-версией Rocket.Chat. На основе сценариев использования составляются тест-кейсы.
* **Тест-кейсы** (Test Case) - это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или её части. Каждый тест-кейс содержит корректное название, предусловие, последовательность шагов, ожидаемый результат каждого шага, приоритетность и продолжительность.
* **Баг-репорты** - оформляется, когда баг уже локализован и его можно повторить. Если баг плавающий, нужно пытаться его повторить или занести в систему, где фиксируются баги, как плавающий баг. Ключевой момент, что баг можно повторить и воспроизвести, только тогда его заносят в систему с багами, где хранятся баг-репорты. Если создать и оформить какой-то баг, и разработчик не сможет его воспроизвести, то тут появится множество вопросов. В баг-репорте обязательно должны быть:

1. Подробное описание проблемы - что, где, когда случилось.
2. Важность дефекта, который указывает тестировщик, а уже приоритет по исправлению этой ошибки указывает менеджер либо команда из разработчиков.
3. Условия воспроизведения - версия игры, версия операционной системы и другие уникальные условия, которые могут помочь разработчику быстро найти баг, устранить и передать задачу на тестинг.
4. Алгоритм воспроизведения - пошаговые предусловия, которые необходимы для воспроизведения бага.
5. Доказательства - скрины, видео, логи с устройств.

* **Тест-план** (**Test Plan**) - это документ, описывающий весь объем работ по тестированию, начиная с описания объекта, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.

**При формировании тест-кейсом будут использоваться следующие техники тест-дизайна:**

* классы эквивалентности;
* граничные значения;
* попарное тестирование;
* прогнозирование ошибок.

# Время проведения тестирования.

|  |  |  |
| --- | --- | --- |
| **Название** | **Время начало** | **Время окончания** |
| **Функциональное тестирование** | 19.07.2023 00:20 | 31.07.2023 01:00 |
| **UI-тестирование** | 18.07.2023 14:30 | 28.07.2023 02:00 |
| **Тестирование удобства использования** | 25.07.2033 14:00 | 28.07.2023 01:00 |
| **Тестирование производительности** | 18.07.2023 14:00 | 31.07.2023 01:00 |

**Когда тестирование можно будет считать завершённым?**

* Все 100% требований учтены.
* Дефекты установлены/ожидаемое число дефектов обнаружено.
* Все дефекты, относящиеся к классу Show Stopper или Blocker, устранены, ни у одного из критических дефектов нет статуса «открытый».
* Все дефекты с высоким приоритетом идентифицированы и исправлены.
* Defect Rate (скорость дефектообразования) ниже установленного допустимого уровня.
* Очень небольшое число дефектов среднего уровня критичности «открыты», их разбор проведен.
* Число «открытых» дефектов среднего уровня, которые не влияют на пользование системой, очень небольшое.
* Все дефекты с высоким уровнем приоритета закрыты и соответствующие регрессивные сценарии успешно проведены.

Охват теста:

* Охват теста должен быть на уровне 95%.
* Pass Rate текст-кейса также должен быть 95%. Для расчета этого процентного соотношения применяется формула:

*(Общее число успешных текст-кейсов / общее число тест-кейсов) \* 100.*

* Все критически важные тест-кейсы оказались успешными
* 5% тест-кейсов могут быть провалены, но это относится к низкоприоритетным кейсам.
* Достигнуто полное покрытие функционала.
* Все крупные функциональные дефекты успешно устранены.

Сроки:

* Срок, отведенный на тестирование, истек.
* Документация по тестированию:
* Вся документация по тестированию, подлежащая сдаче (например, отчет о тестировании), подготовлена, проверена и передана.

Бюджет:

* Бюджет, выделенный на тестирование, полностью израсходован.
* Совещания в формате «Go / No Go» проведены, решение о релизе продукта принято.