Rocket.Chat

Тестовая стратегия

Версия 1.0

История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 25.07.2023 | 1.0 | Тестовая стратегия для веб версии Rocket.Chat v1.0 | Проценко Т.Е. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Оглавление

[1. Введение 4](#_Toc137505097)

[2. Типы проводимого тестирования 5](#_Toc141029286)

[2.1 Функциональное тестирование 5](#_Toc141029287)

[2.2 Тестирование производительности 5](#_Toc141029288)

[2.3 UI-тестирование 5](#_Toc141029289)

[2.4 UX-тестирование 6](#_Toc141029290)

[2.5 Тестирование совместимости 6](#_Toc141029291)

[2.6 Тестирование восстановления 6](#_Toc141029292)

[2.7 Тестирование безопасности 6](#_Toc141029293)

[3. Части системы, которые будут протестированы. 7](#_Toc137505101)

[4. Окружение для работы. 7](#_Toc137505102)

[5. Виды тестовой документации. 7](#_Toc137505103)

[6. Время проведения тестирования. 8](#_Toc137505104)

Тестовая стратегия

# Введение

**Rocket.Chat** - это открытая платформа для обмена сообщениями и совместной работы, предназначенная для бизнеса и командной работы. Он предоставляет возможность общаться в режиме реального времени через текстовые, голосовые и видео сообщения, а также поддерживает обмен файлами и интеграцию с другими сервисами.

**Основные возможности Rocket.Chat включают:**

**1.** Групповые чаты и приватные сообщения: пользователи могут создавать групповые чаты для командной работы или общения внутри отдела, а также обмениваться приватными сообщениями для индивидуального общения.

**2.** Многоуровневая структура каналов: пользователи могут организовывать каналы по темам, проектам или отделам, чтобы облегчить коммуникацию и совместную работу.

**3.** Возможность обмена файлами: Rocket.Chat позволяет пользователям обмениваться файлами различных форматов, что упрощает совместное редактирование и обмен информацией.

**4.** Уведомления и интеграции: платформа поддерживает уведомления о новых сообщениях и интеграцию с другими инструментами и сервисами, такими как GitHub, Jira, Google Drive и другие.

**5.** Голосовые и видео вызовы: Rocket.Chat предоставляет возможность совершать голосовые и видео вызовы внутри платформы, что упрощает коммуникацию и совместную работу на расстоянии.

Rocket.Chat является гибким и настраиваемым инструментом, который может быть использован в различных сферах деятельности, от бизнес-коммуникаций до образования и разработки программного обеспечения.  
  
**Цель тестирования**:

Получение информации о статусе готовности заявленной функциональности веб-приложения, а также обнаружение возможных ошибок, дефектов и недостатков продукта до его выпуска на рынок.

**Состав команды:**

* Junior QA Engineer - Проценко Татьяна Евгеньевна
* Логин: lavonnej@student.21-school.ru
* Успешно выполнила проекты Школы 21 по следующим темам:

- Введение в тестирование.

- Тестовые артефакты.

- Тест-дизайн.

- Работа с дефектами и Bug-tracking tools.

- Front-end тестирование.

- Back-end тестирование.

- Дополнительные инструменты для тестирования.

- Тестирование мобильных приложений.

- Базы данных.

В рамках проектов проводила различные тестирования приложений: TestIt, демо-версии интернет-магазина <https://www.saucedemo.com/>, СберСтрахование, Tutu.ru.

* Работала с такими приложениями, как TestIt, Swagger UI, Postman, Notion, VS code, Fiddler Everywhere, GitLab, GitBash, Android Studio.UIтестирование «СберСтрахование»

# Типы проводимого тестирования

Тестирование мессенджера является важным этапом разработки и поддерживания этой системы

Оно проводится в несколько этапов и будет включать следующие типы:

**2.1** Функциональное тестирование

**2.2** Тестирование производительности

**2.3** UI-тестирование

**2.4** UX-тестирование

**2.5** Тестирование совместимости

**2.6** Тестирование восстановления

**2.7** Тестирование безопасности

Разберём подробнее каждый тип тестирования.

2.1 Функциональное тестирование

Направлено на проверку корректности работы основных функций мессенджера. Таких, как:

* отправка и получение сообщений
* отображение входящих и исходящих сообщений
* работа с вложениями (отправка, получение, скачивание)
* работа с группами и каналами (создание, удаление, управление участниками)
* авторизация и аутентификация

2.2 Тестирование производительности

Позволяет определить, насколько хорошо веб версия справляется с нагрузкой. Также сюда будет входить нагрузочное и стресс тестирование:

* отправка большого количества сообщений за короткий промежуток времени
* работа с большим количеством вложений
* загрузка истории сообщений
* подключение большого числа пользователей

## 2.3 UI-тестирование

Тестирование пользовательского интерфейса поможет проверить функции приложения, имитируя действия пользователей.

* Приложение должно быть интуитивно понятным и простым в использовании;
* Плавная навигация;
* Интерфейс приложения должен соответствовать стандартам цвета, значков и расположению значков для ключевых функций в мессенджере
* Цвета букв и фона, размер букв и шрифт должны позволять пользователям легко читать сообщения
* Приложение должно быть доступным для людей с разным зрением, моторикой и возможностями

## 2.4 UX-тестирование

Это метод тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей приложения в контексте заданных условий.

Данный вид тестирования дает оценку уровня удобства использования приложения по следующим пунктам:

* производительность, эффективность (efficiency) - сколько времени и шагов понадобится пользователю для завершения основных задач приложения, например, размещение новости, регистрации, покупка и т.д.
* правильность (accuracy) - сколько ошибок сделал пользователь во время работы с приложением
* активизация в памяти (recall) – как много пользователь помнит о работе приложения после приостановки работы с ним на длительный период времени
* эмоциональная реакция (emotional response) – как пользователь себя чувствует после завершения задачи. Порекомендует ли пользователь систему своим друзьям

2.5. Тестирование совместимости

Тестирование на совместимость проверяет, работает ли программное обеспечение должным образом во всех браузерах, операционных системах, базах данных, устройствах, сетях и т. д.

Оно проверяет совместимость в нескольких сценариях и выявляет ошибки, которые могут сделать программное обеспечение непригодным для использования или некачественным.

2.6. Тестирование восстановления

**Это** тип [системного тестирования](https://translated.turbopages.org/proxy_u/en-ru.ru.c615abed-64ad12ba-7fc462a2-74722d776562/https/www.geeksforgeeks.org/system-testing/" \t "_self), целью которого является проверка того, может ли система восстанавливаться после сбоев или нет. Этот метод включает сбой системы, а затем проверку правильности выполнения восстановления системы.

Ожидается, что система восстановится после сбоев и возобновит свою работу в течение заранее указанного периода времени.

Приложение должно быть протестировано при восстановлении на наличие сбоев, таких как:

* Сбой источника питания
* Внешний сервер недоступен
* Потеря сигнала беспроводной сети
* Физические условия
* Внешнее устройство не отвечает
* Внешнее устройство отвечает не так, как ожидалось, и т.д.

2.7. Тестирование безопасности

Это процесс проверки программного обеспечения на наличие уязвимостей, которые могут быть использованы злоумышленниками для получения несанкционированного доступа к системе или данных. Такое тестирование включает в себя проверку системы на наличие возможности взлома, перехвата данных, атаки на сервер и других видов угроз безопасности.

Одним из важных аспектов тестирования безопасности является обеспечение конфиденциальности, целостности и доступности данных. Важным этапом тестирования безопасности является оценка уровня риска, связанного с той или иной уязвимостью. На основании этой оценки, можно предоставить рекомендации по устранению уязвимостей или рисков.

# Части системы, которые будут протестированы.

* создание каналов, команд, личных переписок, обсуждений;
* присоединение к существующим каналам (каталог);
* работа чата, загрузка файлов, отправка текстовых и голосовых сообщений;
* работа с участниками и уведомлениями;
* просмотр и редактирование профиля, статуса;
* создание звонков;
* настройка веб-приложения:
* общие, присутствие пользователя,
* звуковые оповещения,
* экспорт и импорт данных,
* безопасность;
* панель администрирования

# Окружение для работы.

**Характеристики устройства:**

Процессор: Intel(R) Pentium(R) CPU J3710 @ 1.60GHz 1.60 GHz

Оперативная память: 4,00 ГБ

Тип системы: 64-разрядная операционная система, процессор x64

**Операционная система:**

Тип системы: Windows

Версия: 22H2

**Браузер:**

Тип: Google Chrome

Версия: 115.0.5790.99 (Официальная сборка), (64 бит)

**Характеристики устройства:**

Мобильные устройства: - Apple iPhone 7

**Операционная система:** iOS 15.6.1

Приложение: Rocket.Chat Версия 4.39.0

# Виды тестовой документации.

При проведении тестирования веб-версии приложения будут использованы следующие виды тестовой документации:

* **Сценарии использования (Use Cases)** - сценарии использования помогают понять, как пользователи будут взаимодействовать с системой и как система должна реагировать на их действия. Они описывают конкретные шаги и последовательность действий, необходимые для достижения цели или выполнения функции.  
    
  Каждый сценарий использования включает в себя описание предусловий, шагов, ожидаемых результатов и возможных вариантов развития событий. Сценарии использования помогают понять требования и создать систему, которая соответствует их потребностям и ожиданиям.
* **Тест-кейсы** - документированные инструкции, которые определяют шаги для тестирования конкретного функционала или особенности программного обеспечения. Они описывают ожидаемые входные данные, ожидаемые результаты и прочие условия, необходимые для проведения тестирования. Тест-кейсы помогут систематизировать процесс тестирования, а также обеспечить полное покрытие функционала и возможные варианты использования программного продукта.
* **Тест-планы** - документ, описывающий весь объем работ по тестированию, начиная с описания тестируемых объектов, стратегии, расписания, критериев начала и окончания тестирования, до необходимого в процессе работы оборудования, специальных знаний, а также оценки рисков с вариантами их разрешения.
* **Отчет о тестировании** - документ, который описывает проведённый тест и результаты его выполнения
* **Баг-репорты** - документированные сообщения о проблеме или неожиданном поведении программного обеспечения.

**Техники тест-дизайна:**

* Эквивалентное разделение;
* Попарное тестирование;

Данные виды тестовой документации позволят качественно и в установленные сроки провести тестирование программного продукта. Описать этапы проверки, а также выявить ошибки системы, которые будут оформлены в общий отчет (отчет о тестировании), и задокументировать отдельно каждый дефект (баг-репорт)

# Время проведения тестирования.

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Тип тестирования** | **Время начала** | **Время окончания** |
| **1** | **Функциональное тестирование** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |
| **2** | **Тестирование производительности** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |
| **3** | **UI-тестирование** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |
| **4** | **UХ-тестирование** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |
| **5** | **Тестирование совместимости** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |
| **6** | **Тестирование восстановления** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |
| **7** | **Тестирование безопасности** | 25.07.2023 | 31.07.2023 |
| Формирование тестового набора | 25.07.2023 | 31.07.2023 |
| Выполнение тестирования | 25.07.2023 | 31.07.2023 |
| Фиксация результатов и подготовка отчетов о тестировании | 25.07.2023 | 31.07.2023 |

Тестирование обычно считается завершённым, когда все запланированные тесты были выполнены, и результаты этих тестов были анализированы и документированы. Завершение тестирования может также происходить, когда достигнуты конкретные критерии качества или функциональности, определённые в начале процесса тестирования.

Важно отметить, что полное завершение тестирования невозможно, поскольку всегда есть возможность обнаружить новые ошибки или улучшить качество продукта. Поэтому тестирование следует рассматривать как непрерывный процесс, который может повторяться на разных этапах разработки и поддержки программного обеспечения.