Rocket.Chat

Версия 1.0

Тестовая стратегия

История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 26/07/2023 | 1.0 | RocketChat testing strategy | Aleksandr Plotnikov |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Оглавление

[1. Введение 4](#_Toc141291141)

[2. Типы проводимого тестирования 5](#_Toc141291142)

[3. Тестируемые модули приложения. 6](#_Toc141291143)

[4. Окружение тестирования. 7](#_Toc141291144)

[5. Артефакты и тест-дизайн. 7](#_Toc141291147)

[6. Время проведения тестирования. 8](#_Toc141291149)

Тестовая стратегия

# Введение

Rocket.Chat - это мощная платформа для обмена сообщениями в реальном времени с открытым исходным кодом. Она была разработана с целью облегчить коммуникацию и сотрудничество в командах и организациях любого размера. Вот некоторые основные характеристики и функции Rocket.Chat:

1. **Коммуникации в реальном времени** - Rocket.Chat позволяет пользователям общаться друг с другом через текст, аудио или видео в реальном времени.

2. **Личные и групповые чаты** - пользователи могут обмениваться сообщениями в личных чатах или в группах. Групповые чаты могут быть открытыми или закрытыми.

3. **Файловый обмен** - Rocket.Chat поддерживает передачу файлов между пользователями, что упрощает совместную работу.

4. **Интеграция** - Rocket.Chat можно интегрировать с другими популярными сервисами, такими как Google Calendar, GitHub, Jira и другими.

5. **Безопасность** - Rocket.Chat обеспечивает высокую безопасность обмена данными, используя шифрование и различные опции аутентификации.

6. **Кастомизация** - Rocket.Chat предлагает широкие возможности для настройки и кастомизации, включая возможность создания собственных плагинов и расширений.

7. **Открытый исходный код** - Rocket.Chat является продуктом с открытым исходным кодом, что означает, что любой может улучшить его и адаптировать под свои потребности.

Имея опыт использования таких програмных продуктов как Test IT, Notion, Postman, DevTools, GitLab, используя такие инструменты как Visual Studio Code и GitHub, я, Александр Плотников ([nigellue@student.21-school.ru](mailto:nigellue@student.21-school.ru)), в данном проекте опишу и протестирую функционал этого мессенджера.

# Типы проводимого тестирования

2.1. **Тестирование функциональности:**

- регистрация и аутентификация - проверка функциональности регистрации и входа в систему, включая сброс пароля.

- управление профилем - проверка функций, связанных с профилем пользователя, таких как изменение пароля, имени пользователя, аватара и т. д.

- чаты - проверка функций чата, включая приватные чаты, групповые чаты, отправку сообщений, получение сообщений, прикрепление файлов, эмодзи и стикеры.

- уведомления - проверка работы системы уведомлений.

- поиск - проверка функциональности поиска.

2.2 **Тестирование производительности:**

- проверка скорости загрузки приложения.

- проверка времени ответа на запросы.

- проверка эффективности использования ресурсов (RAM, CPU).

- тестирование на устойчивость: тестирование системы на продолжительное время под умеренной нагрузкой.

2.3. **Тестирование безопасности:**

- проверка на наличие уязвимостей.

- проверка механизмов аутентификации и авторизации.

- проверка политики конфиденциальности и обработки данных.

2.4. **Тестирование юзабилити:**

- проверка интерфейса на интуитивность и удобство использования.

- проверка доступности для людей с ограниченными возможностями.

2.5. **UI** - **тестирование**:

- проверка дизайна и навигации.

- проверка отображения элементов на различных устройствах и разрешениях экрана.

# Тестируемые модули приложения.

Тестирование должно охватывать следующие модули и функции Rocket.Chat:

3.1. **Создание и управление каналами** - проверить функциональность создания, редактирования и удаления каналов. Проверить права доступа и возможности управления каналами для разных ролей пользователей.

3.2. **Создание и управление командами -** аналогично каналам, проверить функциональность создания, редактирования и удаления команд.

3.3. **Личные переписки -** проверить функциональность и безопасность личных переписок, включая отправку и получение сообщений, блокировку пользователей и удаление истории чатов.

3.4. **Обсуждения -** проверить функциональность создания обсуждений в рамках каналов и команд, участие в обсуждениях и управление ими.

3.5. **Присоединение к существующим каналам и командам -** проверить, что пользователи могут легко присоединиться к существующим каналам и командам, и что правила доступа соблюдаются.

3.6. **Работа чата -** проверить все аспекты функциональности чата, включая загрузку файлов, отправку текстовых и голосовых сообщений, реакции на сообщения, упоминания и так далее.

3.7. **Управление участниками и уведомлениями -** проверить управление участниками, включая добавление и удаление участников, управление ролями и правами доступа, а также систему уведомлений.

3.8. **Управление профилем пользователя -** проверить возможность просмотра и редактирования профиля пользователя, включая изменение статуса.

3.9. **Создание звонков -** проверить функциональность звонков, включая голосовые и видеозвонки, а также конференции.

3.10. **Настройки веб-приложения -** проверить аспекты настройки веб-приложения, включая общие настройки, настройки присутствия пользователя, звуковые оповещения, экспорт и импорт данных, а также настройки безопасности.

3.11. **Панель администрирования -** проверить функции панели администрирования, включая управление пользователями, каналами, командами и системными настройками.

# Окружение тестирования.

# Тестирование должно проводиться в различных окружениях, чтобы обеспечить совместимость и функциональность Rocket.Chat в различных условиях. Но в данном случае оно будет проведено в следующем окружении:

# браузер: Google Сhrome, версия 115.0.5790.1102.

**операционная система:** Windows 10

# Артефакты и тест-дизайн.

# В тестировании Rocket.Chat будут использоваться следующие виды тестовой документации (тестовые артефакты):

1. **Тест-кейсы** - описывают отдельные сценарии, которые следует проверить, чтобы убедиться, что система работает правильно. Тест-кейс включает в себя предусловия, шаги для выполнения, ожидаемые результаты и постусловия.

2. **Сценарии тестирования** - описывают последовательность тест-кейсов, которые вместе проверяют определенный функциональный аспект системы.

3. **Тест-план** - описывает подход, с помощью которого система будет тестироваться, включая тестовые цели, ресурсы, расписание и критерии успеха.

4. **Отчет о тестировании** - составляется после проведения тестирования и описывает его результаты, включая обнаруженные проблемы и общую оценку качества системы.

5. **Баг-репорты** - описывают найденные проблемы, включая шаги для их воспроизведения, ожидаемое и фактическое поведение, и, если возможно, предложения по исправлению.

Для обеспечения полного и эффективного тестирования будут использованы следующие техники тест-дизайна:

1. **Попарное тестирование** - эта техника используется, когда имеется большое количество входных параметров, и их полное сочетание невозможно или непрактично проверить. Попарное тестирование гарантирует, что каждая пара входных параметров проверяется по крайней мере один раз.

2. **Тестирование граничных значений** - эта техника используется для проверки поведения системы на границах допустимого входного пространства. Это может включать проверку максимального и минимального значения, а также значения, находящиеся непосредственно внутри и снаружи границ.

# Время проведения тестирования.

# Для более точного планирования, нужно рассматривать каждый этап тестирования Rocket.Chat более детально. Для этого мы принимаем, что у нас есть команда в одного тестировщика, работающего полный рабочий день.

1. **Создание тестовых артефактов**:

- создание тест-плана - этот процесс может занять один день.

- создание тест-кейсов и сценариев тестирования - исходя из количества функций и сложности приложения, это может занять до 1 недели.

2. **Непосредственное проведение тестирования** - прохождение всех тест-кейсов может занять до 2 недель. Это может варьироваться в зависимости от сложности тест-кейсов и количества обнаруженных проблем.

3. **Фиксация результатов** - время на документацию результатов тестирования и создание баг-репортов обычно включено в время проведения тестирования.

4. **Составление отчетности** - подготовка итогового отчета о тестировании может занять 1-2 дня.

В целом, весь процесс от создания тестовых артефактов до составления итогового отчета о тестировании может занять от 2 до 3 недель для одного тестировщика, работающего полный рабочий день. Но это приблизительные оценки. Время может варьироваться в зависимости от различных факторов основным из которых является уровень опыта тестировщика.