[Rocket.Chat](https://www.rocket.chat/)

Тестовая стратегия

Версия 1.0

История изменений

|  |  |  |  |
| --- | --- | --- | --- |
| **Дата** | **Версия** | **Описание** | **Автор** |
| 16.07.2023 | 1.0 | Rocket.Chat — ознакомление с функционалом | Генина А.В. |
| 17.07.2023 | 1.0 | Составление тестовых сценариев | Генина А.В. |
| 19.07.2023 | 1.0 | Составление тест-кейсов | Генина А.В. |
| 15.08.2023 | 1.0 | Составление отчетности | Генина А.В. |

Оглавление

[1. Введение 4](#_Toc137505097)

[2. Типы проводимого тестирования 4](file:///C:\Users\voron\Downloads\exercise1.docx#_Toc138442781)

[2.1 Функциональное тестирование 4](file:///C:\Users\voron\Downloads\exercise1.docx#_Toc138442782)

[2.2. UI - тестирование 5](file:///C:\Users\voron\Downloads\exercise1.docx#_Toc138442783)

[2.3. Usability Testing 5](file:///C:\Users\voron\Downloads\exercise1.docx#_Toc138442788)

[2.4. Тестирование производительности 5](file:///C:\Users\voron\Downloads\exercise1.docx#_Toc138442788)

[3. Части системы, которые будут протестированы 6](#_Toc137505101)

[4. Окружение для работы 6](#_Toc137505102)

[5. Виды тестовой документации 6](#_Toc137505103)

[6. Время проведения тестирования 7](#_Toc137505104)

Тестовая стратегия

# Введение

В данном проекте рассматривается тестирование веб-версии программы для обмена сообщениями [Rocket.Chat](https://www.rocket.chat/).

**Rocket.Chat** — мессенджер для общения команд с открытым исходном кодом. Компании могут установить Rocket.Chat на собственные сервера или использовать облачную версию. В данной программе для обмена сообщений пользователи могут создавать каналы, команды, группы. Обмениваться файлами, видео и аудио сообщениями, создавать конференции и многопользовательские обсуждения. Использовать Rocket.Chat можно как в браузере, десктопном приложении, так и на мобильных устройствах.

Тестирование программы будет выполнено студентом Школы21 — Гениной Александрой Вячеславовной под логином – jaremyvi**.** В рамках пройденного обучения мной были выполнены следующие проекты по освоению различных техник тестирования и по тестированию конкретных продуктов:

1. Создание тест-кейсов в TestIT
2. Составление сценариев для тестирования продуктов с помощью инструмента Notion
3. Использование GitLab и Git при составлении тестовой документации
4. Проведение попарного тестирования – Sauce Demo
5. Частичное UI – тестирование сайта СберСтрахование.
6. Функциональное тестирование мобильного приложение tutu (нативного приложения и веб – приложения).
7. Тестирование функционала сервиса TestIT
8. Тестирование Swagger UI
9. Тестирование API в Postman
10. Запросы SQL
11. Использование эмуляторов устройств при тестировании мобильных приложений

# Типы проводимого тестирования

## Функциональное тестирование

*Функциональное тестирование* — это процесс обеспечения качества в рамках цикла разработки программного обеспечения, необходимый для проверки реализуемости функциональных требований, согласно спецификации тестируемого программного обеспечения. Функциональное тестирование проводится для оценки соответствия системы или компонента заданным функциональным требованиям. Функциональное тестирование проводится по принципу черного ящика, в связи с чем функциональность ПО можно протестировать, не зная принципа его внутренней работы. Это снижает требования к тестировщикам в части знания языков программирования или конкретных аспектов реализации программного обеспечения.

В процессе функционального тестирования будут созданы тестовые сценарии и тест-кейсы. Тестовый сценарий – это последовательность действий, которые выполняются для проверки определенной функции.

*Test case (тест-кейс, тестовый пример/случай)* — это артефакт, описывающий совокупность шагов, конкретных условий и параметров, необходимых для проверки реализации тестируемой функции или ее части.

## UI-тестирование

UI-тестирование - это деятельность, направленная на проверку качества пользовательского интерфейса, а также его соответствия всем нормам и требованиям. Такое тестирование может проводиться вручную, а может задействовать специальные инструменты.

Суть тестирования интерфейса заключается в том, что программное обеспечение проверяется с точки зрения пользователя, чтобы итоговый продукт был удобным, привлекательным и понятным для вашей аудитории. Для этого команда людей, не задействованных в разработке конкретного продукта привлекается к тестированию: они знакомятся с приложением, изучают его и взаимодействуют с ним так, как это делал бы рядовой пользователь. Задача команды разработки при этом - отследить сильные и слабые стороны, а после тестирования - усовершенствовать продукт на основе результатов тестов.

## Usability Testing (Тестирование удобства использования)

*Тестирование удобства пользования* - это метод тестирования, направленный на установление степени удобства использования, обучаемости, понятности и привлекательности для пользователей разрабатываемого продукта в контексте заданных условий. Тестирование удобства пользования 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.? (*меньше - лучше*)
* **правильность** (**accuracy**) - сколько ошибок сделал пользователь во время работы с приложением? (*меньше - лучше*)
* **активизация в памяти** (**recall**) – как много пользователь помнит о работе приложения после приостановки работы с ним на длительный период времени? (*повторное выполнение операций после перерыва должно проходить быстрее чем у нового пользователя*)
* **эмоциональная реакция** (**emotional response**) – как пользователь себя чувствует после завершения задачи - растерян, испытал стресс? Порекомендует ли пользователь систему своим друзьям? (*положительная реакция - лучше*)

Проверка удобства использования может проводиться как по отношению к готовому продукту, посредством тестирования черного ящика (black box testing), так и к интерфейсам приложения (API), используемым при разработке - тестирование белого ящика (white box testing). В этом случае проверяется удобство использования внутренних объектов, классов, методов и переменных, а также рассматривается удобство изменения, расширения системы и интеграции ее с другими модулями или системами. Использование удобных интерфейсов (API) может улучшить качество, увеличить скорость написания и поддержки разрабатываемого кода, и как следствие улучшить качество продукта в целом.

Отсюда становится очевидно, что тестирование удобства пользования может производиться на разных уровнях разработки программного обеспечения: модульном, интеграционном, системном и приемочном. При этом оно целиком и полностью будет зависит от того, кто будет использовать приложение на выделенном конкретном уровне (разработчик, бизнес пользователь системы и т.д.).

## Тестирование производительности

*Тестирование производительности* — это вид тестирования программного обеспечения, направленный на определение его способности работать с определенными рабочими нагрузками и в определенных условиях. Основная цель тестирования производительности — выявить проблемы, связанные с быстродействием, надежностью и стабильностью системы, а также определить возможные узкие места и точки отказа. Тестирование проводится с учетом негативных сценариев использования обычным пользователем системы.

# Части системы, которые будут протестированы

В данном проекте будут протестированы следующие части системы веб-приложения для обмена сообщений между пользователей Rocket.Chat:

* 1. Авторизация в системе Rocket.Chat
  2. Настройка и переход по ссылкам домашней страницы (Home)
  3. Создание, редактирование и удаление каналов, команд, личных переписок, обсуждений
  4. Добавление и удаление пользователей в различных частях мессенджера: команды, каналы, личные переписки, обсуждения
  5. Настройка уведомлений для различных чатов: команды, каналы, личные переписки, обсуждения
  6. Присоединение к существующим каналам, командам, личным перепискам, обсуждениям
  7. Работа чата: все возможности работы с чатами, звонками, видео и аудиосообщениями
  8. Создание тредов, добавление реакций, цитирование
  9. Скачивание и установка внутренних приложений
  10. Просмотр и редактирование профиля, статуса
  11. Создание звонков
  12. Настройка веб-приложения
  13. Вкладки: общие, присутствие пользователя, звуковые оповещения, экспорт и импорт данных, безопасность
  14. Панель администрирования, ее настройка и возможности

# Окружение для работы

Браузер: Яндекс Браузер; Версия 23.7.0.2526 (64-bit)

Система: Windows 10 64-bit Версия Корпоративная LTSC

Версия приложения Rocket.Chat - 6.2.8.

# Виды тестовой документации

1. *Сценарии использования* - описывают сценарии взаимодействия между пользователем и системой. На основе сценариев формируются тест-кейсы, так как они описывают, в каком контексте должно производиться каждое действие пользователя. Use Cases являются проверяемыми требованиями, так как в них всегда указана цель, которую нужно достигнуть, и шаги, которые надо для этого воспроизвести.

2. *Тест-кейсы* для проведения тестирования, созданные с помощью приложения TestIT.

3. *Тест-план* – это документ описывающей действия, которые будут происходить в процессе тестирования, описывает логику завершения задач и оценку рисков. Результатом выполнения тест-плана – является отчет о прохождении тест-кейсов (содержащихся в Тест-плане) – об успешном или провальном прохождении тестов.

4. *Баг-репорт* - технический документ, который подробно описывает ошибку в работе программы или приложения.

5. *Отчет о тестировании* – документ с выводом о результатах прохождении тестирования.

При составлении тест-кейсов для тестирования мессенджера веб-приложения Rocket.Chat будут использованы следующие *техники тест-дизайна:*

* Сценарии использования;
* Попарное тестирование;
* Интуитивное тестирование;

# Время проведения тестирования

|  |  |  |  |
| --- | --- | --- | --- |
| **Этапы тестирование** | **Время начало** | **Время окончание** | **Завершенность** |
| Написание тестовой стратегии | 16.07.2023 | 28.07.2023 | Этап завершён, когда все этапы тестовой стратегии прописаны. |
| Написание сценариев использования | 17.07.2023 | 19.07.2023 | Этап завершён, когда описаны все возможные варианты сценариев использования |
| Составление тест-кейсов | 19.07.2023 | 27.07.2023 | Этап завершён, когда описаны все возможные варианты тест-кейсов для проведение запланированных типов тестирования. |
| Составление тест - плана | 28.07.2023 | 28.07.2023 | Этап завершён, когда создан документ. |
| Проведение функционального тестирования | 28.07.2023 | 28.07.2023 | Этап завершён, когда осуществлён прогон всех написанных тест-кейсов. |
| Составление отчетности | 15.08.2023 | 15.08.2023 | Этап завершён, когда составлен документ - о результате прогона тест-кейсов. |